



FUNDAMENTOS DE REDES Y DE LAS COMUNICACIONES

RUBÉN MARÍN LUCAS
PEDRO MIGUEL CARMONA

INDICE

Introducción	3
Manual del usuario	4
Tecla F1	4
Tecla F2	4
Tecla F3	6
Tecla F5	7
Tecla F6	8
Tecla F7	10
Manual del programador	11
Jerarquía de clases (si se tienen clases) o jerarquía de librerías	11
Algoritmos de sondeo y selección	14
ALGORITMO SELECCIÓN	14
ALGORITMO SONDEO	16
Batería de pruebas	18
Prueba 1 - OK	18
Prueba 2 - OK	18
Prueba 3 - OK	20
Caracteres de control utilizados	23
Opciones adicionales	23
FUNCIONALIDAD DE TECLA ESC	25

Introducción

A lo largo de este documento se expone una documentación externa sobre el código empleado para construir la aplicación sobre el puerto RS-232C que se ha ido exponiendo a lo largo de las clases prácticas en un total de 6 sesiones. Se trata de un resumen de la funcionalidad en general de toda la práctica, así como de los documentos solicitados de presentar. Para saber cualquier funcionalidad de forma más precisa diríjase a la documentación interna del código aportado.

Manual del usuario

Explicación breve de todas las funcionalidades de la práctica con capturas de pantalla y con ejemplos simples de la ejecución de cada una de ellas

Tecla F1

Cuando se pulsa esta tecla se envía el mensaje escrito anteriormente en la consola. Si no se ha escrito nada, no se envía nada.

```
----- PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES -----  
----- CURSO 2019/20 -----  
----- SESION1.CPP -----  
-----  
Seleccionar el puerto a utilizar:(1-4)  
1. COM1  
2. COM2  
3. COM3  
4. COM4  
Puerto COM1 seleccionado  
Seleccionar la velocidad de transmision:(1-5)  
1. 1200  
2. 2400  
3. 4800  
4. 9600  
5. 19200  
Velocidad seleccionada: 9600  
Puerto COM1 abierto correctamente  
Hola Juan  
Hola María, que tal?  
Bien, que tal tu?  
Tambien bien
```

Como vemos hay dos colores, el color verde se corresponde a los mensajes que envía el emisor al receptor y el color amarillo se corresponde a los mensajes que se reciben, es decir los que se envían desde el receptor al emisor

Tecla F2

Cuando se pulsa esta tecla aparecerá el siguiente menú:

```
Seleccionar la trama de control a enviar:(1-4)  
1. Trama EOT  
2. Trama ENQ  
3. Trama ACK  
4. Trama NACK
```

En este menú se nos da 4 opciones para enviar tramas diferentes (Trama EOT, Trama ENQ, Trama ACK y Trama NACK). Si pulsamos en el número

correspondiente a cada una de ellas se enviará esa trama que hemos seleccionado.

```
Seleccionar la trama de control a enviar:(1-4)
  1. Trama EOT
  2. Trama ENQ
  3. Trama ACK
  4. Trama NACK
Trama EOT enviada
Seleccionar la trama de control a enviar:(1-4)
  1. Trama EOT
  2. Trama ENQ
  3. Trama ACK
  4. Trama NACK
Trama ENQ enviada
Seleccionar la trama de control a enviar:(1-4)
  1. Trama EOT
  2. Trama ENQ
  3. Trama ACK
  4. Trama NACK
Trama ACK enviada
Seleccionar la trama de control a enviar:(1-4)
  1. Trama EOT
  2. Trama ENQ
  3. Trama ACK
  4. Trama NACK
Trama NACK enviada
```

Aquí vemos como nos indica una trama diferente, diferenciadas por colores, según el número que seleccionemos.

```
Se ha recibido una trama EOT
Se ha recibido una trama ENQ
Se ha recibido una trama ACK
Se ha recibido una trama NACK
```

Aquí podemos observar como el receptor recibe las tramas, indicando que se ha recibido con el mismo color que estaban diferenciadas en la imagen anterior para identificarlas mejor

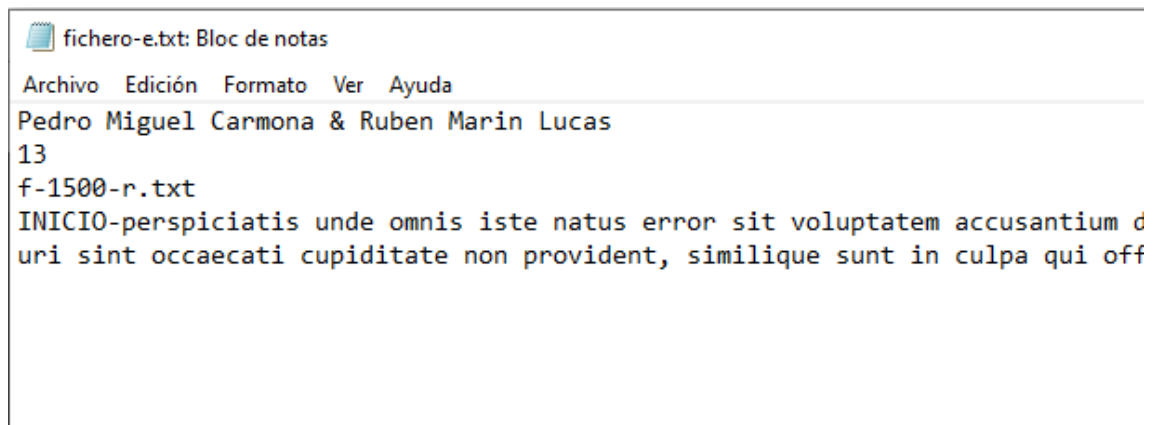
Tecla F3

Cuando se pulsa esta tecla aparecerá lo siguiente en la consola que se ha pulsado:

```
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas  
Fichero enviado
```

Como podemos leer nos informa de que el fichero se está enviando por los autores escritos en el fichero y aparecerá el mensaje "Fichero enviado" cuando se envíe en su totalidad.

Aparecerá en ese color azul, y el archivo que envía debe llamarse obligatoriamente "fichero-e.txt" y debe tener la siguiente estructura:



```
fichero-e.txt: Bloc de notas  
Archivo Edición Formato Ver Ayuda  
Pedro Miguel Carmona & Ruben Marin Lucas  
13  
f-1500-r.txt  
INICIO-perspiciatis unde omnis iste natus error sit voluptatem accusantium d  
uri sint occaecati cupiditate non provident, similique sunt in culpa qui off
```

- La primera línea del fichero debe contener el nombre de los autores de la practica
- En la segunda línea contendrá el color con el que debe salir el mensaje de recepción del fichero en la consola receptora
- En la tercera línea debe contener el nombre del fichero con el que se va a identificar la recepción del fichero
- Por último, estará el cuerpo del fichero, que será tan grande como se desee

En la consola receptora aparecerá lo siguiente:

```
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marín Lucas  
Fichero recibido  
El fichero recibido tiene un tamaño de 1500 bytes
```

Que nos informa de los autores del fichero que estamos recibiendo.

Cuando acabe de recibir el fichero aparecerá el mensaje "Fichero recibido" informando que ha acabado de recibir el fichero, además de otro mensaje que nos indica el tamaño del mismo en bytes.

Otra cosa que vemos es que el mensaje es del color que se indicó en el fichero "fichero-e.txt", que en este ejemplo era 13, que corresponde a ese fucsia que podemos ver en la imagen.

En el caso de no existir el fichero "fichero-e.txt" aparecerá el siguiente mensaje en el emisor y no se enviará nada al receptor:

```
ERROR: El fichero fichero-e.txt no existe
```

Tecla F5

Cuando se pulsa esta tecla se volcará todo lo que se envié a un fichero llamado "log.txt". Al usuario se le notificará con el siguiente mensaje:

```
Todo lo que esciba se escribira en el log, tenga cuidado con lo que escribe :)
```

Si el usuario lo pulsa de nuevo le aparecerá el siguiente mensaje:

```
La funcion de F5 ya esta activa  
La funcion de F5 ya esta activa  
La funcion de F5 ya esta activa  
La funcion de F5 ya esta activa  
La funcion de F5 ya esta activa  
La funcion de F5 ya esta activa
```

Este fichero parará de escribir en él solo cuando se finalice el programa, o cuando el usuario inicie el protocolo MAESTRO-ESCLAVO mediante la tecla F6.

Tecla F6

Cuando se pulsa esta tecla se iniciará el protocolo MAESTRO-ESCLAVO en la consola donde se ha pulsado dándole a elegir si quiere ser maestro o esclavo:

```
PROTOCOLO MAESTRO-ESCLAVO

Seleccione maestro o esclavo:(1-2)
  1. Maestro
  2. Esclavo
```

Si se selecciona maestro, en la misma consola, aparecerá un segundo menú para elegir la operación a realizar:

```
PROTOCOLO MAESTRO-ESCLAVO

Seleccione maestro o esclavo:(1-2)
  1. Maestro
  2. Esclavo

Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
  1. Seleccion
  2. Sondeo
```

En la otra consola se iniciará el protocolo, no dejando elegir, y se le comunicará que se ha elegido la consola como ESCLAVO

```
PROTOCOLO MAESTRO-ESCLAVO

Ha seleccionado ESCLAVO
```

Si se selecciona la consola como ESCLAVO en el primer menú, el segundo menú aparecerá en la otra consola y esta se seleccionará como ESCLAVO.

Si seleccionamos la opción de *selección* se ejecutará el protocolo mostrando todas las tramas de este, así como diferenciando por colores las distintas fases:

PROTOCOLO MAESTRO-ESCLAVO	PROTOCOLO MAESTRO-ESCLAVO
<pre> # seleccionando ESCLAVO E R ENQ 0 E R ACK 0 E R STX 0 87 87 E R ACK 0 E R STX 1 55 55 E R ACK 1 E R STX 0 14 14 E R ACK 0 Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas E R STX 1 37 37 E R ACK 1 E R STX 0 51 51 E R ACK 0 E R STX 1 187 187 E R ACK 1 E R STX 0 46 46 E R ACK 0 E R STX 1 6 6 E R ACK 1 E R STX 0 120 120 E R ACK 0 E R STX 1 87 87 E R ACK 1 E R STX 0 33 33 E R ACK 0 E R STX 1 94 94 E R ACK 1 E R STX 0 20 20 E R ACK 0 E R STX 1 3 3 E R ACK 1 Fichero recibido El fichero recibido tiene un tamaño de 2540 bytes E R EOT 0 E R ACK 0 FIN PROTOCOLO </pre>	<pre> Selección maestro o esclavo:(1-2) 1. Maestro 2. Esclavo # seleccionando MAESTRO, seleccione la operacion a realizar:(1-2) 1. Selección 2. Sondeo E R ENQ 0 E R ACK 0 E R STX 0 87 87 E R ACK 0 E R STX 1 55 55 E R ACK 1 E R STX 0 14 14 E R ACK 0 Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas E R STX 1 37 37 E R ACK 1 E R STX 0 51 51 E R ACK 0 E R STX 1 187 187 E R ACK 1 E R STX 0 46 46 E R ACK 0 E R STX 1 6 6 E R ACK 1 E R STX 0 120 120 E R ACK 0 E R STX 1 87 87 E R ACK 1 E R STX 0 33 33 E R ACK 0 E R STX 1 94 94 E R ACK 1 E R STX 0 20 20 E R ACK 0 E R STX 1 3 3 E R ACK 1 Fichero enviado E R EOT 0 E R ACK 0 FIN PROTOCOLO </pre>

Si seleccionamos la opción de *sondeo* se ejecutará el protocolo mostrando todas las tramas de este, así como diferenciando por colores las distintas fases, y en la última fase, el maestro elige si se finaliza la conexión, apareciendo un menú para elegir si se finaliza.

PROTOCOLO MAESTRO-ESCLAVO	PROTOCOLO MAESTRO-ESCLAVO
<pre> Selección maestro o esclavo:(1-2) 1. Maestro 2. Esclavo # seleccionando ESCLAVO E T ENQ 0 E T ACK 0 E T STX 0 87 87 E T ACK 0 E T STX 1 55 55 E T ACK 1 E T STX 0 14 14 E T ACK 0 Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas E T STX 1 37 37 E T ACK 1 E T STX 0 51 51 E T ACK 0 E T STX 1 187 187 E T ACK 1 E T STX 0 46 46 E T ACK 0 E T STX 1 6 6 E T ACK 1 E T STX 0 120 120 E T ACK 0 E T STX 1 87 87 E T ACK 1 E T STX 0 33 33 E T ACK 0 E T STX 1 94 94 E T ACK 1 E T STX 0 20 20 E T ACK 0 E T STX 1 3 3 E T ACK 1 Fichero enviado E T EOT 0 E T ACK 0 FIN PROTOCOLO </pre>	<pre> # seleccionando MAESTRO, seleccione la operacion a realizar:(1-2) 1. Selección 2. Sondeo E T ENQ 0 E T ACK 0 E T STX 0 87 87 E T ACK 0 E T STX 1 55 55 E T ACK 1 E T STX 0 14 14 E T ACK 0 Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas E T STX 1 37 37 E T ACK 1 E T STX 0 51 51 E T ACK 0 E T STX 1 187 187 E T ACK 1 E T STX 0 46 46 E T ACK 0 E T STX 1 6 6 E T ACK 1 E T STX 0 120 120 E T ACK 0 E T STX 1 87 87 E T ACK 1 E T STX 0 33 33 E T ACK 0 E T STX 1 94 94 E T ACK 1 E T STX 0 20 20 E T ACK 0 E T STX 1 3 3 E T ACK 1 Fichero recibido El fichero recibido tiene un tamaño de 2540 bytes E T EOT 0 E T ACK 0 Después el cierre de la comunicación:(1-2) 1. Si 2. No E T ACK 0 FIN PROTOCOLO </pre>

En el caso del maestro elegir no, el esclavo no dejará de intentar finalizar la conexión hasta que el maestro le diga que finalice:

```

2. Esclavo
Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Selección
2. Envío

R T STX 0 87 87
R T ACK 0
R T STX 1 55 55
R T ACK 1
R T STX 0 14 14
R T ACK 0

Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas

R T STX 1 37 37
R T ACK 1
R T STX 0 51 51
R T ACK 0
R T STX 1 187 187
R T ACK 1
R T STX 0 46 46
R T ACK 0
R T STX 1 6 6
R T ACK 1
R T STX 0 120 120
R T ACK 0
R T STX 1 87 87
R T ACK 1
R T STX 0 33 33
R T ACK 0
R T STX 1 94 94
R T ACK 1
R T STX 0 25 25
R T ACK 0
R T STX 1 3 3
R T ACK 1

Fichero recibido
El fichero recibido tiene un tamaño de 2540 bytes

R T EOT 0
Después el cierre de la comunicación?(1-2)
1. Si
2. No

R T NACK 0
R T EOT 1
Después el cierre de la comunicación?(1-2)
1. Si
2. No

R T ACK 1
FIN PROTOCOLO

PROTOCOLO MAESTRO-ESCLAVO
Ha seleccionado ESCLAVO

R T STX 0
R T ACK 0
R T STX 0 87 87
R T ACK 0
R T STX 1 55 55
R T ACK 1
R T STX 0 14 14
R T ACK 0

Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas

R T STX 1 37 37
R T ACK 1
R T STX 0 51 51
R T ACK 0
R T STX 1 187 187
R T ACK 1
R T STX 0 46 46
R T ACK 0
R T STX 1 6 6
R T ACK 1
R T STX 0 120 120
R T ACK 0
R T STX 1 87 87
R T ACK 1
R T STX 0 33 33
R T ACK 0
R T STX 1 94 94
R T ACK 1
R T STX 0 25 25
R T ACK 0
R T STX 1 3 3
R T ACK 1

Fichero enviado

R T EOT 0
R T NACK 0
R T EOT 1
R T ACK 1
FIN PROTOCOLO

```

Tecla F7

Si pulsamos esta tecla, solo tendrá su efecto en el protocolo MAESTRO-ESCLAVO durante la fase de trasferencia y durante el envío del cuerpo del fichero.

Cuando se pulse esta tecla en el cuerpo del fichero, se introducirá un error en los datos a enviar, más precisamente en Datos [0] = 'ç'

El fichero debe llamarse obligatoriamente "EProtoc.txt" y debe tener la siguiente estructura:

```

EProtoc.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
Pedro Miguel Carmona & Ruben Marin Lucas
7
Fich-P10.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula
l, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt
n ut eros et nisl sagittis vestibulum. Nullam nulla eros, ultricies sit amet,

```

- La primera línea del fichero debe contener el nombre de los autores de la practica
- En la segunda línea contendrá el color con el que debe salir el mensaje de recepción del fichero en la consola receptora

- En la tercera línea debe contener el nombre del fichero con el que se va a identificar la recepción del fichero
- Por último, estará el cuerpo del fichero, que será tan grande como se desee

En la siguiente foto se muestra una prueba que se han metido tres errores, donde se puede apreciar que la consola esclava los reconoce:

```

PROTOCOLO MAESTRO-ESCLAVO
seleccione maestro o esclavo:(1-2)
1. Maestro
2. Esclavo
Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Seleccion
2. Sonda
R R STX 0
R R ACK 0
R R STX 0 87
R R ACK 0
R R STX 1 55
R R ACK 1
R R STX 0 14
R R ACK 0
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
R R STX 1 37
R R ACK 1
R R STX 0 51
R R ACK 0
R R STX 1 187
R R ACK 1
R R STX 0 46
R R ACK 0
R R STX 1 6
R R ACK 1
R R STX 0 120
R R ACK 0
R R STX 0 120
R R NACK 1
R R STX 1 87
R R ACK 0
R R STX 1 87
R R ACK 1
R R STX 0 33
R R NACK 1
R R STX 0 33
R R ACK 0
R R STX 1 94
R R ACK 1
R R STX 0 28
R R ACK 0
R R STX 1 3
R R ACK 1
Fichero enviado
R R EOT 0
R R ACK 0
FIN PROTOCOLO

Velocidad seleccionada: 1200
Puerto COM1 abierto correctamente
PROTOCOLO MAESTRO-ESCLAVO
Ha seleccionado ESCLAVO
R R STX 0
R R ACK 0
R R STX 0 87 87
R R ACK 0
R R STX 1 55 55
R R ACK 1
R R STX 0 14 14
R R ACK 0
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
R R STX 1 37 37
R R ACK 1
R R STX 0 51 51
R R ACK 0
R R STX 1 187 187
R R ACK 1
R R STX 0 46 46
R R ACK 0
R R STX 1 6 6
R R ACK 1
R R STX 0 120 120
R R NACK 1
R R STX 1 87 87
R R ACK 0
R R STX 1 87 87
R R ACK 1
R R STX 0 33 33
R R NACK 1
R R STX 0 33 33
R R ACK 0
R R STX 1 94 94
R R ACK 1
R R STX 0 28 28
R R ACK 0
R R STX 1 3 3
R R ACK 1
Fichero recibido
El fichero recibido tiene un tamaño de 2548 bytes
R R EOT 0
R R ACK 0
FIN PROTOCOLO

```

Manual del programador

Jerarquía de clases (si se tienen clases) o jerarquía de librerías

Para empezar, se seguirá una estructura comentando desde los **niveles** más **bajos** de la aplicación **hacia** los más **altos**. Por ello, tenemos que comentar en el **primer nivel** a la librería **PuertoSerie** proporcionada por los profesores de la asignatura que básicamente sirve para **controlar** el **puerto** por el cual va a fluir la comunicación.

En un **segundo nivel** nos encontramos con una **librería** realizada por nosotros, **Gestor** que incluye a la librería **PuertoSerie**, y contiene 3 métodos, unos para mostrar el encabezado de la práctica(**void encabezado()**); otro para mostrar los mensajes e interactuar con el usuario en cuanto a las propiedades del puerto, se encarga en otras palabras de abrir el puerto(**void IniciarPuerto()**); y

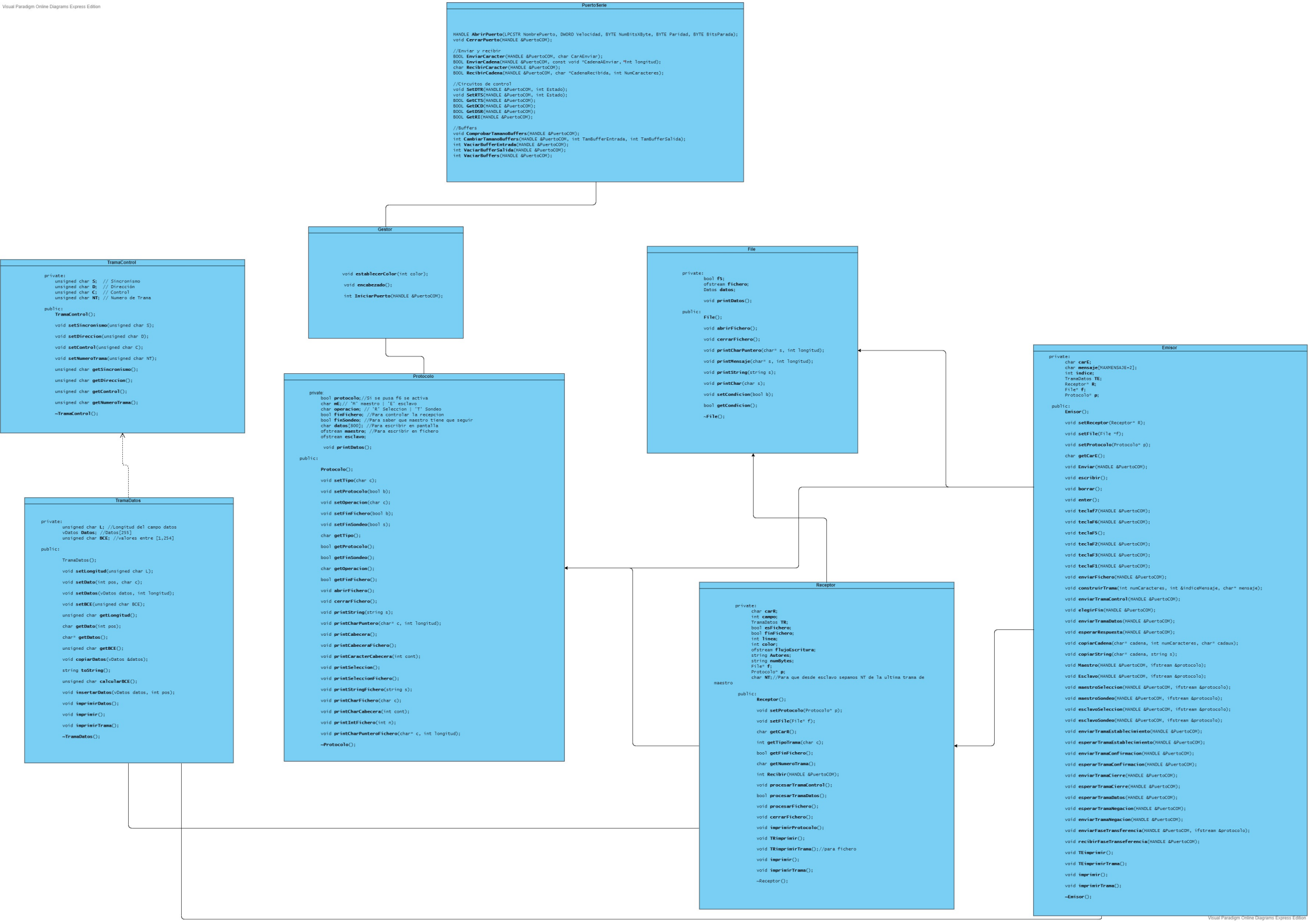
por último un método que es usado por muchas clases que consiste en establecer el color de los mensajes imprimidos por pantalla (**void establecerColor()**).

Posteriormente podemos incluir en un **tercer nivel clases auxiliares** creadas por nosotros. Primero nos encontramos con la **clase TramaControl** que sirve para formar tramas de control para el envío de tramas. Como **atributos** contiene los campos de sincronismo(**SYN**), dirección(**D**), control(**C**) y número de tramas(**NT**) y como métodos tiene el constructor y destructor por defecto, así como sus propios métodos setter y getter. Además, tenemos también la **clase TramaDatos** que hereda de ésta anterior todos sus métodos y atributos por lo que además de esos campos le añadimos longitud(**L**), vector de char(**datos**) y bce(**BCE**), éstos nos ayudarán en la transmisión de mensajes que contienen texto; además la clase cuenta con sus métodos setter y getter, añadiéndoles otros de variada utilidad. Como segunda clase a comentar en este nivel nos encontramos con la **clase File** que cuya función es **ayudarnos** en la implementación de la **función F5**, que sirve para guardar en un fichero(log.txt) toda la información escrita en pantalla; así tenemos de **atributo** un bool (**f5**) para saber cuándo está activo este modo, un flujo(**fichero**) para escribir en el fichero y un vector de char (**datos**) para imprimir por pantalla. En este nivel también tenemos a la **clase Protocolo** que tiene una función similar a File, pero para la **función F6** que se trata del modo protocolo, para ello tenemos dos flujos (**maestro** y **esclavo**) para escribir en dos ficheros diferentes (prolog-m y prolog-e) y un campo datos para imprimir en pantalla, además de una serie de tributos bool y char que nos ayudan a saber desde otras clases(emisor y receptor) qué rol hemos escogido dentro del modo protocolo(**mE**), si el modo protocolo está activo(**protocolo**), si se ha terminado de enviar el fichero(**finFichero**)...

Por último en el **cuarto nivel** nos encontramos con la **clase Receptor y Emisor**, que sirven para recibir y enviar información (tramas de datos y de control) respectivamente. Iniciamos comentando la **clase Receptor**, ésta es una clase que sirve para **recibir y procesar** las **tramas** principalmente. Para ello dispone de una serie de métodos y atributos de los que conviene mencionar los más importantes. En cuanto a los atributos destacamos los dos punteros que contiene, uno a la clase **Protocolo(p)**, y otro a la clase **File(f)**, su utilidad es la de "compartir" estos recursos con la clase **Emisor**; y de atributos destacamos su método principal que es el que se encarga de recibir carácter a carácter cada trama y procesarla (**void Recibir()**). Ahora comentaremos la **clase Emisor** que como veníamos diciendo principalmente se encarga de enviar las tramas, y al igual que la clase **Receptor**, (y con el mismo objetivo) contiene dos punteros a la clase **Protocolo(p)**, y a la clase **File(f)**, además contiene otro puntero a la clase **Receptor**, ya que en ocasiones es necesario conocer información que guarda esta clase (ej: uno de sus usos es la llamada del método recibir mientras se está enviando un fichero para conseguir que no exista exclusión entre envío y recepción). Para los métodos podemos comentar muchos, el propio método para enviar caracteres y tramas (**void Enviar()**), que detecta las funciones que se activan(las teclas f que se usan),

así como muchos otros, como los diseñados para el modo protocolo (**Maestro()**, **Esclavo()**...).

>>Se trata de un resumen para entender la disposición de las distintas clases y librerías, para más información mirar la documentación interna.



Algoritmos de sondeo y selección.

En este punto comentaremos como funciona cada una de las funciones del modo protocolo, en las que participan principalmente entre las clases ya comentadas, las clases Emisor, Receptor y Protocolo.

ALGORITMO SELECCIÓN

La base de la función de selección es la de enviar (el maestro) un fichero al esclavo, por lo tanto, en la fase de transferencia consistirá en el envío de tramas de datos(STX) por parte del maestro que recibirá el esclavo, y si llegan correctamente entonces éste enviará tramas de confirmación (ACK) hasta que se acabe el fichero y el maestro termine la conversación.

Definimos más formalmente con un pseudocódigo comentado:

MAESTRO

```
enviarTramaEstablecimiento(); → Se construye trama de control con  
                                control = 04  
esperarTramaConfirmación(); → Se espera una trama de control con  
                                control = 06  
  
enviarFaseTransferencia() → Envío del fichero  
  
enviarTramaCierre() → Se construye trama de control con control = 05  
  
esperarTramaConfirmación() → Se espera una trama de control con  
                                control = 06  
print("FIN PROTOCOLO")  
  
cerrarFichero() → Cerrar fichero de texto abierto a la hora de elegir  
                  maestro  
  
setFinProtocolo() → El bool que controla la activación del protocolo se  
                    pone a false
```

Los métodos de espera de tramas tienen todos una estructura similar. Están compuestos por un bucle que espera un número, ese número será el devuelto por el método Recibir de la clase Receptor cuando detecta una trama de control o una trama de datos. Cada tipo de trama tiene asociado un número. Ejemplo:

```
esperarTramaConfirmación() → while(R → Recibir() != 4); //El 4 se asocia a  
                                una trama ACK
```

La clase receptor siempre devuelve 0 en el caso de que se reciba un carácter que todavía no sabemos si se trata de una trama de control o de datos, en el punto en el que sabe que es una trama de control lo devuelve en ese punto; sin

embargo, cuando es una trama de datos lo devuelve una vez procesada la trama.

La fase de transferencia consiste en el envío de un fichero de texto, y por cada trama de datos que se envía habrá una llamada al método que espera una trama de confirmación, porque necesita que el esclavo le diga que le ha llegado correctamente la información para seguir.

ESCLAVO

esperaTramaEstablecimiento() → Se espera una trama de control con control = 04

enviarTramaConfirmación() → Se construye trama de control con control = 06

esperaFaseTransferencia() → Recepción del fichero

esperarTramaCierre() → Se espera una trama de control con control = 05

enviarTramaConfirmacion() → Se construye trama de control con control = 06

print("FIN PROTOCOLO")

cerrarFichero() → Cerrar fichero de texto abierto a la hora de elegir maestro

setFinProtocolo() → El bool que controla la activación del protocolo se pone a false

La fase de transferencia consiste en la espera de un fichero en forma de tramas de datos, por lo que de forma general el algoritmo consistirá en llamar en bucle mientras no se halla enviado el fichero entero al método que espera una trama de datos. En principio el fichero termina con la última trama de datos y se envía el carácter '}' para saberlo, pero tenemos que tener en cuenta que después tenemos que recibir otra trama de datos con el número de bytes del fichero. Por ello se usa un bool en la clase Protocolo que será true cuando ya haya llegado esa última trama. El diseño se decidió hacer así al final para mostrar de forma cómoda los mensajes de "Se ha recibido fichero..." cuando ya se haya recibido todas las tramas de datos, para que quede mejor de forma visual, porque si no se mezclaban esos mensajes con la información de las tramas.

En pseudocódigo sería algo como:

```
while(!p→ getFinFichero()){  
    esperarTramaDatos() → Se espera una trama de control con  
                           control = 02  
    enviarTramaConfirmacion() → Se construye trama de control con  
                               control = 06  
}
```

ALGORITMO SONDEO

Este algoritmo prácticamente igual que el explicado anteriormente, solo que ahora es el esclavo el que envía el fichero de texto al maestro. Debido a este "cambio de papeles" el esclavo cuando acabe de mandar todo el fichero tiene que solicitar el fin de la comunicación al maestro (él directamente no puede acabar, porque el que manda es el maestro). Por ello cambia un poco la fase de liberación, pero por lo demás se sigue la misma estructura.

MAESTRO

```
esperaTramaEstablecimiento() → Se espera una trama de control con  
                               control = 04  
enviarTramaConfirmación() → Se construye trama de control con  
                             control = 06  
  
esperaFaseTransferencia() → Recepción del fichero  
  
while(!p→ getFinSondeo()){  
    elegirFin() → Menú que elige enviar trama de confirmación o  
                  negación  
}  
  
printf("FIN PROTOCOLO")  
  
cerrarFichero() → Cerrar fichero de texto abierto a la hora de elegir  
                  maestro  
  
setFinProtocolo() → El bool que controla la activación del protocolo se  
                    pone a false
```

Una vez que termina la fase de transferencia entremos en un bucle en el que hasta que no enviemos una trama de confirmación no salimos de él, para ello se ha usado el bool de finSondeo. Cuando se envíe una trama de confirmación se activa el bool y termina el modo protocolo para el maestro.

ESCLAVO

```
enviarTramaEstablecimiento(); → Se construye trama de control con  
                                control = 04  
esperarTramaConfirmación(); → Se espera una trama de control con  
                                control = 06  
  
enviarFaseTransferencia() → Envío del fichero  
  
enviarTramaCierre() → Se construye trama de control con control = 05  
  
while(!p→ getFinSondeo()){  
    esperarRespuesta() → Espera una trama de confirmación o de  
                        negación  
  
    if(!p→ getFinFichero()){//Puede haber cambiado mientras esperaba  
        enviarTramaCierre() → Se construye trama de control con  
                                control = 05  
    }  
}  
  
print("FIN PROTOCOLO")  
  
cerrarFichero() → Cerrar fichero de texto abierto a la hora de elegir  
                maestro  
setFinProtocolo() → El bool que controla la activación del protocolo se  
                    pone a false
```

Cuando termine la fase de transferencia se solicita al maestro terminar la comunicación enviando una trama de cierre, y en función de lo que conteste el maestro, se seguirá mandando esa trama de cierre o terminará la comunicación. Igualmente se usa el bool finSondeo.

El esperar respuesta es un poco distinto a los otros esperar, así que se comenta un poco más detalladamente:

```
while( (n=R→ Recibir()) < 3) // < 3 tiene sentido porque confirmación y  
                                negación tienen los valores 3 y 4 en el  
                                recibir  
  
if(n==4) p→ setFinsondeo() → en el caso de que reciba una trama de  
                                confirmación, se activa el bool para que  
                                termine la fase de liberación
```

Batería de pruebas

Prueba 1 - OK

Envío de un fichero y recepción de una cadena de 800 caracteres y una trama de control durante el envío. Mostrar la captura (emisor y receptor) que evidencie que se ha probado este caso

```
=====
----- PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES -----
----- CURSO 2019/20 -----
----- SESION1.CPP -----
=====

Seleccionar el puerto a utilizar:(1-4)
1. COM1
2. COM2
3. COM3
4. COM4
Puerto COM1 seleccionado
Seleccionar la velocidad de transmision:(1-5)
1. 1200
2. 2400
3. 4800
4. 9600
5. 19200
Velocidad seleccionada: 2400
Puerto COM1 abierto correctamente
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa
. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis
, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo
, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vita
e, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum
semper nisl. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleife
nd ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut
metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies
Seleccionar la trama de control a enviar:(1-4)
1. Trama EOT
2. Trama ENQ
3. Trama ACK
4. Trama NACK
Trama EOT enviada
Fichero recibido
El fichero recibido tiene un tamaño de 100000 bytes
```

```
=====
----- PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES -----
----- CURSO 2019/20 -----
----- SESION1.CPP -----
=====

Seleccionar el puerto a utilizar:(1-4)
1. COM1
2. COM2
3. COM3
4. COM4
Puerto COM1 seleccionado
Seleccionar la velocidad de transmision:(1-5)
1. 1200
2. 2400
3. 4800
4. 9600
5. 19200
Velocidad seleccionada: 2400
Puerto COM1 abierto correctamente
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa
. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis
, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo
, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vita
e, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum
semper nisl. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleife
nd ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut
metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies
Se ha recibido una trama EOT
Fichero enviado
```

Se puede observar cómo mientras la estación de la izquierda estaba recibiendo un fichero ha enviado un mensaje de 800 caracteres y una trama de control, en este caso una trama EOT.

Prueba 2 - OK

Realizar sondeo con el fichero que contiene como cuerpo del fichero 10 tramas exactas (está en el campus virtual). Mostrar la captura (emisor y receptor) de la salida en pantalla de esta funcionalidad. Igualmente mostrar una captura de los ficheros log de maestro y esclavo.

```
PROTOCOLO MAESTRO-ESCLAVO

Ha seleccionado ESCLAVO

R T ENQ 0
E T ACK 0

E T STX 0 87
R T ACK 0
E T STX 1 55
R T ACK 1
E T STX 0 14
R T ACK 0

Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas

E T STX 1 37
R T ACK 1
E T STX 0 51
R T ACK 0
E T STX 1 107
R T ACK 1
E T STX 0 46
R T ACK 0
E T STX 1 6
R T ACK 1
E T STX 0 120
R T ACK 0
E T STX 1 87
R T ACK 1
E T STX 0 33
R T ACK 0
E T STX 1 94
R T ACK 1
E T STX 0 26
R T ACK 0
E T STX 1 3
R T ACK 1

Fichero enviado

E T EOT 0
R T NACK 0

E T EOT 1
R T NACK 1

FIN PROTOCOLO

PROTOCOLO MAESTRO-ESCLAVO

Seleccione maestro o esclavo:(1-2)
1. Maestro
2. Esclavo

Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Selection
2. Sondeo

E T ENQ 0
R T ACK 0

R T STX 0 87 87
E T ACK 0
R T STX 1 55 55
E T ACK 1
R T STX 0 14 14
E T ACK 0

Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas

R T STX 1 37 37
E T ACK 1
R T STX 0 51 51
E T ACK 0
R T STX 1 107 107
E T ACK 1
R T STX 0 46 46
E T ACK 0
R T STX 1 6 6
E T ACK 1
R T STX 0 120 120
E T ACK 0
R T STX 1 87 87
E T ACK 1
R T STX 0 33 33
E T ACK 0
R T STX 1 94 94
E T ACK 1
R T STX 0 26 26
E T ACK 0
R T STX 1 3 3
E T ACK 1

Fichero recibido
```

```
R T ACK 1
E T STX 0 51
R T ACK 0
E T STX 1 107
R T ACK 1
E T STX 0 46
R T ACK 0
E T STX 1 6
R T ACK 1
E T STX 0 120
R T ACK 0
E T STX 1 87
R T ACK 1
E T STX 0 33
R T ACK 0
E T STX 1 94
R T ACK 1
E T STX 0 26
R T ACK 0
E T STX 1 3
R T ACK 1

Fichero enviado

E T EOT 0
R T NACK 0

E T EOT 1
R T NACK 1

FIN PROTOCOLO

E T ACK 1
R T STX 0 33 33
E T ACK 0
R T STX 1 94 94
E T ACK 1
R T STX 0 26 26
E T ACK 0
R T STX 1 3 3
E T ACK 1

Fichero recibido
El fichero recibido tiene un tamaño de 2540 bytes

R T EOT 0
Desea el cierre de la comunicacion?(1-2)
1. Si
2. No

E T NACK 0

R T EOT 1
Desea el cierre de la comunicacion?(1-2)
1. Si
2. No

E T NACK 1

R T EOT 0
Desea el cierre de la comunicacion?(1-2)
1. Si
2. No

E T ACK 0

FIN PROTOCOLO
```

En estas 2 capturas de pantalla se puede observar una ejecución del algoritmo de sondeo, en el que se rechazó el fin de la comunicación por parte del maestro 2 veces. Ahora se muestra la información que han guardado los ficheros de *prolog-M.txt* y *prolog-E.txt*:

```
Prolog-e: Bloc de notas
Archivo Edición Formato Ver Ayuda
***** PROTOCOLO MAESTRO-ESCLAVO *****
*****

Ha seleccionado ESCLAVO

R T ENQ 0
E T ACK 0

E T STX 0 87
R T ACK 0
E T STX 1 55
R T ACK 1
E T STX 0 14
R T ACK 0

Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas

E T STX 1 37
R T ACK 1
E T STX 0 51
R T ACK 0
E T STX 1 107
R T ACK 1
E T STX 0 46
R T ACK 0
E T STX 1 6
R T ACK 1
E T STX 0 120
R T ACK 0
E T STX 1 87
R T ACK 1
E T STX 0 33
R T ACK 0
E T STX 1 94
R T ACK 1
E T STX 0 26
R T ACK 0
E T STX 1 3
R T ACK 1
. . .

Prolog-m: Bloc de notas
Archivo Edición Formato Ver Ayuda
***** PROTOCOLO MAESTRO-ESCLAVO *****
*****

Seleccione maestro o esclavo:(1-2)
1. Maestro
2. Esclavo

Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Selecccion
2. Sondeo

E T ENQ 0
R T ACK 0

R T STX 0 87 87
E T ACK 0
R T STX 1 55 55
E T ACK 1
R T STX 0 14 14
E T ACK 0

Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas

R T STX 1 37 37
E T ACK 1
R T STX 0 51 51
E T ACK 0
R T STX 1 107 107
E T ACK 1
R T STX 0 46 46
E T ACK 0
R T STX 1 6 6
E T ACK 1
R T STX 0 120 120
E T ACK 0
R T STX 1 87 87
E T ACK 1
R T STX 0 33 33
E T ACK 0
```

```
Prolog-e: Bloc de notas
Archivo Edición Formato Ver Ayuda
E T STX 0 14
R T ACK 0

Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas

E T STX 1 37
R T ACK 1
E T STX 0 51
R T ACK 0
E T STX 1 107
R T ACK 1
E T STX 0 46
R T ACK 0
E T STX 1 6
R T ACK 1
E T STX 0 120
R T ACK 0
E T STX 1 87
R T ACK 1
E T STX 0 33
R T ACK 0
E T STX 1 94
R T ACK 1
E T STX 0 26
R T ACK 0
E T STX 1 3
R T ACK 1

Fichero enviado

E T EOT 0
R T NACK 0

E T EOT 1
R T NACK 1

E T EOT 0
R T ACK 0

FIN PROTOCOLO

Prolog-m: Bloc de notas
Archivo Edición Formato Ver Ayuda
E T ACK 0
R T STX 1 6 6
E T ACK 1
R T STX 0 120 120
E T ACK 0
R T STX 1 87 87
E T ACK 1
R T STX 0 33 33
E T ACK 0
R T STX 1 94 94
E T ACK 1
R T STX 0 26 26
E T ACK 0
R T STX 1 3 3
E T ACK 1

Fichero recibido
El fichero recibido tiene un tamaño de 2540 bytes

R T EOT 0
Desea el cierre de la comunicacion?(1-2)
1. Si
2. No

E T NACK 0

R T EOT 1
Desea el cierre de la comunicacion?(1-2)
1. Si
2. No

E T NACK 1

R T EOT 0
Desea el cierre de la comunicacion?(1-2)
1. Si
2. No

E T ACK 0

FIN PROTOCOLO
```

Prueba 3 - OK

Realizar selección con el fichero que contiene como cuerpo del fichero 10 tramas exactas (está en el campus virtual), introduciendo 3 errores consecutivos. Mostrar la captura (emisor y receptor) de la salida en pantalla de

esta funcionalidad. Igualmente mostrar una captura de los ficheros log de maestro y esclavo

PROTOCOLO MAESTRO-ESCLAVO

seleccione maestro o esclavo:(1-2)
1. Maestro
2. Esclavo
Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Seleccion
2. Sondeo
E R ENQ 0
E R ACK 0
E R STX 0 87
E R ACK 0
E R STX 1 55 55
E R ACK 1
E R STX 0 14 14
E R ACK 0
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
E R STX 1 37 37
E R ACK 1
E R STX 0 51
E R ACK 0
E R STX 1 187
E R ACK 1
E R STX 0 46
E R ACK 0
E R STX 1 6 6
E R ACK 1
E R STX 0 120
E R NACK 1
E R STX 0 120
E R ACK 0
E R STX 1 87
E R NACK 0
E R STX 1 87
E R ACK 1
E R STX 0 33
E R NACK 1
E R STX 0 33
E R ACK 0
E R STX 1 94
E R ACK 1
E R STX 0 26
E R ACK 0
E R STX 1 3
E R ACK 1
Fichero enviado
E R EOT 0
E R ACK 0
FIN PROTOCOLO

Velocidad seleccionada: 1200
Puerto COM1 abierto correctamente

PROTOCOLO MAESTRO-ESCLAVO

Ha seleccionado ESCLAVO
E R ENQ 0
E R ACK 0
E R STX 0 87 87
E R ACK 0
E R STX 1 55 55
E R ACK 1
E R STX 0 14 14
E R ACK 0
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
E R STX 1 37 37
E R ACK 1
E R STX 0 51 51
E R ACK 0
E R STX 1 187 187
E R ACK 1
E R STX 0 46 46
E R ACK 0
E R STX 1 6 6
E R ACK 1
E R STX 0 120 177
E R NACK 1
E R STX 0 120 120
E R ACK 0
E R STX 1 87 221
E R NACK 0
E R STX 1 87 87
E R ACK 1
E R STX 0 33 175
E R NACK 1
E R STX 0 33 33
E R ACK 0
E R STX 1 94 94
E R ACK 1
E R STX 0 26 26
E R ACK 0
E R STX 1 3 3
E R ACK 1
Fichero recibido
El fichero recibido tiene un tamano de 2540 bytes
E R EOT 0
E R ACK 0
FIN PROTOCOLO

En esta captura se puede observar las tres pulsaciones seguidas, es decir, los tres errores consecutivos metidos, y como en la consola esclava los reconoce.

Prólogo-m.txt Bloc de notas

Archivo Edición Formato Ver Ayuda

***** PROTOCOLO MAESTRO-ESCLAVO *****
***** PROTOCOLO MAESTRO-ESCLAVO *****
Seleccione maestro o esclavo:(1-2)
1. Maestro
2. Esclavo
Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Seleccion
2. Sondeo
E R ENQ 0
E R ACK 0
E R STX 0 87
E R ACK 0
E R STX 1 55
E R ACK 1
E R STX 0 14
E R ACK 0
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
E R STX 1 37
E R ACK 1
E R STX 0 51
E R ACK 0
E R STX 1 187
E R ACK 1
E R STX 0 46
E R ACK 0
E R STX 1 6
E R ACK 1
E R STX 0 120
E R NACK 1
E R STX 0 120
E R ACK 0
E R STX 1 87
E R NACK 0
E R STX 1 87
E R ACK 1
E R STX 0 33
E R NACK 1
E R STX 0 33
E R ACK 0
E R STX 1 94
E R ACK 1
E R STX 0 26
E R ACK 0
E R STX 1 3
E R ACK 1

Prólogo-a.txt Bloc de notas

Archivo Edición Formato Ver Ayuda

***** PROTOCOLO MAESTRO-ESCLAVO *****
***** PROTOCOLO MAESTRO-ESCLAVO *****
Ha seleccionado ESCLAVO
E R ENQ 0
E R ACK 0
E R STX 0 87 87
E R ACK 0
E R STX 1 55 55
E R ACK 1
E R STX 0 14 14
E R ACK 0
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
E R STX 1 37 37
E R ACK 1
E R STX 0 51 51
E R ACK 0
E R STX 1 187 187
E R ACK 1
E R STX 0 46 46
E R ACK 0
E R STX 1 6 6
E R ACK 1
E R STX 0 120 177
E R NACK 1
E R STX 0 120 120
E R ACK 0
E R STX 1 87 221
E R NACK 0
E R STX 1 87 87
E R ACK 1
E R STX 0 33 175
E R NACK 1
E R STX 0 33 33
E R ACK 0
E R STX 1 94 94
E R ACK 1
E R STX 0 26 26
E R ACK 0
E R STX 1 3 3
E R ACK 1
Fichero recibido
El fichero recibido tiene un tamano de 2540 bytes
E R EOT 0
E R ACK 0

```
Prolog-e.txt: Bloc de notes
Archivo Edición Formato Ver Ayuda
Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Selecccion
2. Sondeo

E R ENQ 0
R R ACK 0

E R STX 0 87
R R ACK 0
E R STX 1 55
R R ACK 1
E R STX 0 14
R R ACK 0

Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas

E R STX 1 37
R R ACK 1
E R STX 0 51
R R ACK 0
E R STX 1 107
R R ACK 1
E R STX 0 46
R R ACK 0
E R STX 1 6
R R ACK 1
E R STX 0 120
R R NACK 1
E R STX 0 120
R R ACK 0
E R STX 1 87
R R NACK 0
E R STX 1 87
R R ACK 1
E R STX 0 33
R R NACK 1
E R STX 0 33
R R ACK 0
E R STX 1 94
R R ACK 1
E R STX 0 26
R R ACK 0
E R STX 1 3
R R ACK 1

Fichero enviado

E R EOT 0
R R ACK 0

FIN PROTOCOLO

Prolog-m.txt: Bloc de notes
Archivo Edición Formato Ver Ayuda
Ha seleccionado ESCLAVO

R R ENQ 0
E R ACK 0

R R STX 0 87 87
E R ACK 0
R R STX 1 55 55
E R ACK 1
R R STX 0 14 14
E R ACK 0

Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas

R R STX 1 37 37
E R ACK 1
R R STX 0 51 51
E R ACK 0
R R STX 1 107 107
E R ACK 1
R R STX 0 46 46
E R ACK 0
R R STX 1 6 6
E R ACK 1
R R STX 0 120 177
E R NACK 1
R R STX 0 120 120
E R ACK 0
R R STX 1 87 221
E R NACK 0
R R STX 1 87 87
E R ACK 1
R R STX 0 33 175
E R NACK 1
R R STX 0 33 33
E R ACK 0
R R STX 1 94 94
E R ACK 1
R R STX 0 26 26
E R ACK 0
R R STX 1 3 3
E R ACK 1

Fichero recibido
El fichero recibido tiene un tamaño de 2540 bytes

E R EOT 0
R R ACK 0

FIN PROTOCOLO
```

En estas dos capturas se puede ver los ficheros *Prolog-e.txt* y *Prolog-m.txt* de la ejecución de la captura anterior.

En una de las ejecuciones se ha podido hasta meter 6 fallos seguidos:

```
Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2)
1. Selecccion
2. Sondeo

E R ENQ 0
R R ACK 0

E R STX 0 80
R R ACK 0
E R STX 1 55
R R ACK 1
E R STX 0 14
R R ACK 0

Enviando fichero por Belinda Soto & Mario Borrego

E R STX 1 37
R R NACK 0
E R STX 1 37
R R ACK 1
E R STX 0 51
R R ACK 0
E R STX 1 107
R R ACK 1
E R STX 0 46
R R NACK 0
E R STX 1 6
R R ACK 1
E R STX 0 120
R R NACK 1
E R STX 0 120
R R ACK 0
E R STX 1 87
R R NACK 0
E R STX 1 87
R R ACK 1
E R STX 0 33
R R NACK 1
E R STX 0 33
R R ACK 0
E R STX 1 94
R R NACK 0
E R STX 1 94
R R ACK 1
E R STX 0 26
R R NACK 1
E R STX 0 26
R R ACK 0
E R STX 1 3
R R ACK 1

Fichero enviado

E R EOT 0
R R ACK 0

Ha seleccionado ESCLAVO

R R ENQ 0
E R ACK 0

R R STX 0 80 80
E R ACK 0
R R STX 1 55 55
E R ACK 1
R R STX 0 14 14
E R ACK 0
R R STX 1 37 142
E R NACK 0 1

Recibiendo fichero por Belinda Soto & Mario Borrego

R R STX 1 37 37
E R ACK 1
R R STX 0 51 51
E R ACK 0
R R STX 1 107 107
E R ACK 1
R R STX 0 46 46
E R ACK 0
R R STX 1 6 120
E R NACK 0 1
R R STX 1 6 6
E R ACK 1
R R STX 0 120 177
E R NACK 1 1
R R STX 0 120 120
E R ACK 0 1
R R STX 1 87 221
E R NACK 0 1
R R STX 1 87 87
E R ACK 1 1
R R STX 0 33 175
E R NACK 1 1
R R STX 0 33 33
E R ACK 0 1
R R STX 0 94 216
E R NACK 0 1
R R STX 1 94 94
E R ACK 1 1
R R STX 0 26 211
E R NACK 1 1
R R STX 0 26 26
E R ACK 0 1
R R STX 1 3 3
E R ACK 1 1

Fichero recibido
El fichero recibido tiene un tamaño de 2540 bytes

E R EOT 0
R R ACK 0
```

Caracteres de control utilizados

Carácter 'E'	Utilizado para enviar a la otra estación el tipo de esta, esclava
Carácter 'M'	Utilizado para enviar a la otra estación el tipo de esta, maestra
Carácter '{'	Utilizado para enviar a la otra estación e indicar el comienzo de envío de fichero
Carácter '}'	Utilizado para enviar a la otra estación e indicar el final de envío de fichero
Carácter 'ç'	Carácter utilizado para meter errores en la pulsación de la tecla f7
Carácter 27	Utilizado para enviar a la otra estación e indicar la pulsación de la tecla ESC para finalizar la acción

Opciones adicionales

Como en la entrega 6 nos sobró tiempo hemos podido implementar la opción individual de la entrega 4 que consistía en el envío y recepción de un fichero simultáneamente mediante la pulsación del F3.

```

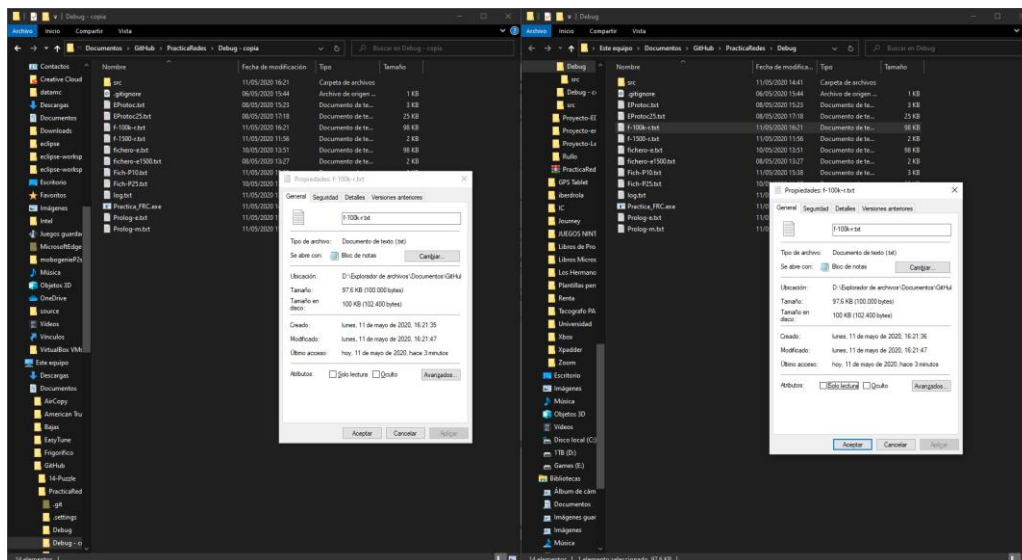
D:\Explorador de archivos\Documentos\GitHub\Practicas\Debug\Practica_FRC.exe
PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES
CURSO 2019/20
SESION1.CPP

Seleccionar el puerto a utilizar:(1-4)
1. COM1
2. COM2
3. COM3
4. COM4
Puerto COM1 seleccionado
Seleccionar la velocidad de transmision:(1-5)
1. 1200
2. 2400
3. 4800
4. 9600
5. 19200
Velocidad seleccionada: 9600
Puerto COM1 abierto correctamente
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Fichero enviado
Fichero recibido
El fichero recibido tiene un tamaño de 100000 bytes

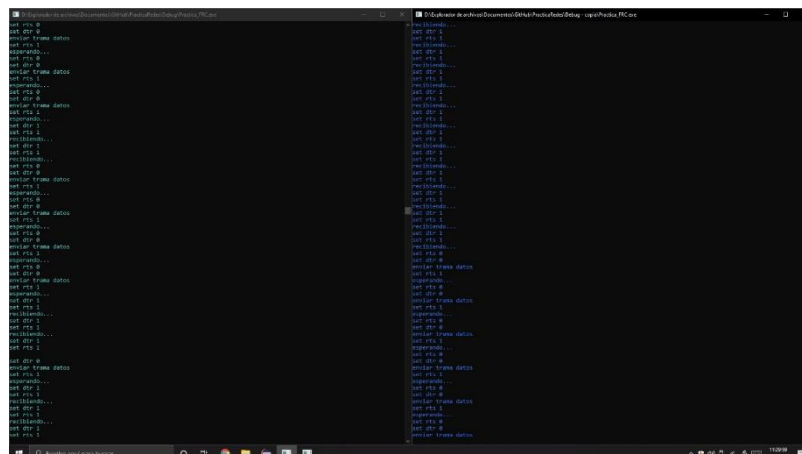
D:\Explorador de archivos\Documentos\GitHub\Practicas\Debug - copia\Practica_FRC.exe
PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES
CURSO 2019/20
SESION1.CPP

Seleccionar el puerto a utilizar:(1-4)
1. COM1
2. COM2
3. COM3
4. COM4
Puerto COM1 seleccionado
Seleccionar la velocidad de transmision:(1-5)
1. 1200
2. 2400
3. 4800
4. 9600
5. 19200
Velocidad seleccionada: 9600
Puerto COM1 abierto correctamente
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Fichero enviado
Fichero recibido
El fichero recibido tiene un tamaño de 100000 bytes
  
```


En cada estación tendremos el fichero con el nombre correspondiente al que se le puso en el fichero de envío, en la segunda línea de cabecera, como podemos apreciar en la siguiente imagen.



Si ponemos mensajes en cada acción que realiza al enviar y recibir los ficheros podemos observar que efectivamente, recibe y envía intermitentemente y que no envía todo el fichero y después lo recibe.



FUNCIONALIDAD DE TECLA ESC

Para comenzar desde el inicio de la práctica la tecla ESC ha estado implementada en el menú principal, es decir, una vez abierto el puerto y sin estar en alguna otra función (en modo protocolo, enviando un fichero...) si pulsas la tecla susodicha podrás salir de la aplicación. Ejemplo:

```
=====
----- PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES -----
----- CURSO 2019/20 -----
----- SESION1.CPP -----
=====

Seleccionar el puerto a utilizar:(1-4)
    1. COM1
    2. COM2
    3. COM3
    4. COM4
Puerto COM1 seleccionado
Seleccionar la velocidad de transmision:(1-5)
    1. 1200
    2. 2400
    3. 4800
    4. 9600
    5. 19200
Velocidad seleccionada: 19200
Puerto COM1 abierto correctamente
```

```
=====
----- PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES -----
----- CURSO 2019/20 -----
----- SESION1.CPP -----
=====

Seleccionar el puerto a utilizar:(1-4)
    1. COM1
    2. COM2
    3. COM3
    4. COM4
Puerto COM1 seleccionado
Seleccionar la velocidad de transmision:(1-5)
    1. 1200
    2. 2400
    3. 4800
    4. 9600
    5. 19200
Velocidad seleccionada: 19200
Puerto COM1 abierto correctamente
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis na
oque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec qu
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Fichero enviado
```

En cualquiera de estas dos situaciones cuando se pulse escape la aplicación se cerrará puesto que no nos encontramos realizando ninguna función, estaríamos en el menú principal.

Ahora bien, el resto de la funcionalidad de la tecla se ha ido añadiendo a lo largo de la práctica, por lo general hemos seguido un mismo diseño en toda ella de manera que siempre que haya un menú se podrá pulsar esta tecla para salir de este menú, o incluso también se podrá pulsar mientras se está enviado un mensaje, un fichero... Lo que pasará es que se cancelará el envío y volveremos al “menú principal” . Explicaremos los lugares donde se puede pulsar ESC con capturas:

- Antes de abrir el puerto hay dos menús de selección en los que se podrá pulsar la tecla escape y lo que ocurrirá será que aparecerá un mensaje “Error al abrir el puerto” y si se pulsa otra vez se podrá salir de la aplicación.

```
=====
----- PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES -----
----- CURSO 2019/20 -----
----- SESION1.CPP -----
=====

Seleccionar el puerto a utilizar:(1-4)
    1. COM1
    2. COM2
    3. COM3
    4. COM4
Error al abrir el puerto
```

- Antes de abrir el puerto hay dos menús de selección en los que se podrá pulsar la tecla escape y lo que ocurrirá será que aparecerá un mensaje “Error al abrir el puerto COM 1” y si se pulsa otra vez se podrá salir de la aplicación.

```
=====
----- PRACTICAS DE FUNDAMENTOS DE REDES DE COMUNICACIONES -----
----- CURSO 2019/20 -----
----- SESION1.CPP -----
=====

Seleccionar el puerto a utilizar:(1-4)
  1. COM1
  2. COM2
  3. COM3
  4. COM4
Puerto COM1 seleccionado
Seleccionar la velocidad de transmision:(1-5)
  1. 1200
  2. 2400
  3. 4800
  4. 9600
  5. 19200
Error al abrir el puerto COM1
```

- Al enviar un mensaje (f1), una trama de control (f2) o un fichero de texto (f3), si se pulsa la tecla escape se cancelará el envío (se avisa con un mensaje) y se volverá al “menú principal” .

```
Seleccionar la trama de control a enviar:(1-4)
  1. Trama EOT
  2. Trama ENQ
  3. Trama ACK
  4. Trama NACK

Se cancela el envio de la trama de control
```

```
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Fichero recibido
El fichero recibido tiene un tamaño de 100000 bytes
Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Fichero recibido
El fichero recibido tiene un tamaño de 28448 bytes
```

```
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Fichero enviado
Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
Se cancela la acción
Fichero enviado
```

Ahí se puede observar cómo se ha cancelado el envío y solo se han recibido 28448 bytes de 100000.

***Nosotros al enviar con f1 no hemos sido capaces de probar que funciona ya que el envío es muy rápido, pero está implementado igual que en estas situaciones.**

- La funcionalidad de esta tecla también llega al modo protocolo, se puede pulsar en el menú cuando comienza o al elegir la función a realizar, lo que pasará es que terminará el estado protocolo y volverá al “menú principal”. También se podrá pulsar en el momento en el que se envíe el fichero en la fase de transferencia, en la estación que esté enviando el fichero (en selección → maestro, en sondeo → esclavo), de

esta manera se cancelará el envío, terminando la fase de transferencia y procediendo a la fase de liberación.

```
PROTOCOLO MAESTRO-ESCLAVO

Seleccione maestro o esclavo:(1-2)
    1. Maestro
    2. Esclavo

Se cancela la accion
```

PROTOCOLO MAESTRO-ESCLAVO	PROTOCOLO MAESTRO-ESCLAVO
Seleccione maestro o esclavo:(1-2) 1. Maestro 2. Esclavo Ha seleccionado MAESTRO, seleccione la operacion a realizar:(1-2) 1. Seleccion 2. Sondeo Se cancela la accion	Ha seleccionado ESCLAVO Se cancela la accion

Recibiendo fichero por Pedro Miguel Carmona & Ruben Marin Lucas	Enviando fichero por Pedro Miguel Carmona & Ruben Marin Lucas
E R STX 1 37 37 E R ACK 1 E R STX 0 51 51 E R ACK 0 E R STX 1 107 107 E R ACK 1 E R STX 0 46 46 E R ACK 0 E R STX 1 6 6 E R ACK 1 E R STX 0 120 120 E R ACK 0 E R STX 1 87 87 E R ACK 1 E R STX 0 33 33 E R ACK 0 E R STX 1 94 94 E R ACK 1 E R STX 0 42 42 E R ACK 0 E R STX 1 39 39 E R ACK 1 E R STX 0 82 82 E R ACK 0 E R STX 1 30 30 E R ACK 1 E R STX 0 62 62 E R ACK 0 E R STX 1 123 123 E R ACK 1 E R STX 0 37 37 E R ACK 0 E R STX 1 60 60 E R ACK 1 E R STX 0 74 74 E R ACK 0 E R STX 1 57 57 E R ACK 1 Se cancela la accion E R STX 0 13 13 E R ACK 0 Fichero recibido El fichero recibido tiene un tamaño de 5080 bytes	E R STX 1 37 E R ACK 1 E R STX 0 51 E R ACK 0 E R STX 1 107 E R ACK 1 E R STX 0 46 E R ACK 0 E R STX 1 6 E R ACK 1 E R STX 0 120 E R ACK 0 E R STX 1 87 E R ACK 1 E R STX 0 33 E R ACK 0 E R STX 1 94 E R ACK 1 E R STX 0 42 E R ACK 0 E R STX 1 39 E R ACK 1 E R STX 0 82 E R ACK 0 E R STX 1 30 E R ACK 1 E R STX 0 62 E R ACK 0 E R STX 1 123 E R ACK 1 E R STX 0 37 E R ACK 0 E R STX 1 60 E R ACK 1 E R STX 0 74 E R ACK 0 E R STX 1 57 E R ACK 1 Se cancela la accion E R STX 0 13 E R ACK 0 Fichero enviado

Se observa que funciona al igual que en el envío de fichero normal, se han enviado 5080 bytes de 25000