

---

ESCOM-IPN

# Práctica 1

## Analizador de Trama LLC

REDES DE COMPUTADORAS

Laura Andrea Morales López

MSc. NIDIA ASUNCIÓN CORTEZ DUARTE

Marzo 2018

### Resumen

*In this report we will see an analysis of LLC frames and a program to do it quickly.*

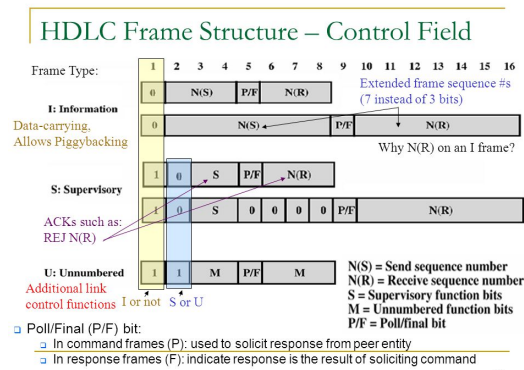
# Índice

1. Introducción: Control de enlace lógico	2
2. Problema	2
3. Hipotesis:	2
4. Software	2
5. Procedimiento	3
6. Resultados	4
7. Conclusión	6
8. Código	6
9. Mapa de memoria	8

## 1. Introducción: Control de enlace lógico

El protocolo LLC (control lógico de enlace) es un protocolo de capa de enlace de datos derivado de HDLC, del cual hereda su campo de control, y fue estandarizado por la IEEE bajo la denominación 802.2. Igual que en HDLC se tienen tramas de información, supervisión y no numeradas distinguiéndose entre ellas por los bits menos significativos de su campo de control.

Tenemos la siguiente cabecera:



## 2. Problema

Realizar este tipo de análisis es muy específico y fácil de realizar, sin embargo tiende a los errores humanos.

Como vimos en la cabecera tenemos los bloques de la información y sus respectivos significados con lo cual podemos leer de alguna manera lo que nos están diciendo las tramas.

## 3. Hipotesis:

Una cabecera LLC puede ser leída de manera secuencial, por lo tanto es posible programar la lectura de las mismas.

## 4. Software

- Librería `stdlib.h`

- Librería stdio.h
- Librería stdbool.h
- Librería string.h

## 5. Procedimiento

- Se analiza cada trama por separado
- Procesamos el campo de control que se encuentra en el byte 17.
- Checamos el tipo de trama que tendremos, para esto realizamos un & con 3.
  - Si es un 0 o un 2 entonces es de Información. En este tipo de trama la cabecera de 2 bytes es de la siguiente manera con lo que realizando únicamente un corrimiento de  $\gg 1$  podemos obtener N(s) del byte 17 o N(R) del byte 18. Y además obtenemos el P/F con un &1.



- Si es un 1 tenemos una trama de Supervisión.

Con esta trama podemos obtener el valor del comando realizando un corrimiento  $\gg 2$  y luego un &3.

Después chequeando el P/F podemos determinar si es necesario char el SAPd para saber si es comando o respuesta. Si P/F=1 chequeamos el Sap (byte 16) y si es 0 es Comando y si es 1 es Respuesta.

El P/F lo podemos obtener realizando un & 1 en el byte 18.

Y el Sap con un & 1 con el byte 16.

Con esto tenemos la posición del comando o respuesta a usar dependiendo de la siguiente tabla.



SS	Bits	Comando	Significado
0	00	RR	Receptor Listo
1	01	RNR	Receptor no listo
2	10	REJ	Rechazo
3	11	SREJ	Rechazo selectivo

- Si es un 3 tenemos una trama Sin numerar.

Como en la de supervisión checamos el P/F y el SAP y con corrimientos y and's:  $((byte17 \gg 2) \& 3) | ((byte17 \gg 3) \& 28)$  podemos obtener los números MM y con la siguiente tabla podemos obtener su significado.

**M M M P/F M M 1 1**

M	Comando	Respuesta	Significado C	Significado R
0	UI	UI	Unnumbered Information	Unnumbered Information
1	SIM	RIM	Set Inicialization Mode	Request Inicialization Mo
3	SARM	DM	Modo Desconectado	
4	UP		Unnumbered Pull	
7	SABM			
8	DISC	RD	Disconnect	Request Disconnect
11	SARME			
12		UA		Unnumbered Acknowledgm
15	SABME			
16	SNRM		Set Normal Response Mode	
17		FRMR		
19	RSET			
23	XID	XID	Exchange Station Identification	Exchange Station Identifica
27	SNMRE			

## 6. Resultados

Tenemos los siguientes resultados:

Trama 1 ToT 3 T-Unnumbered Comando SABME Trama 2 ToT 3 T-Unnumbered Respuesta UA Trama 3 ToT 4 T-Supervisión N(R):0 RR Trama 4 ToT 4 T-Supervisión N(R):0 RR Trama 5 ToT 18 T-Informacion N(s):0 N(r):0 Trama 6 ToT 18 T-Informacion N(s):0 N(r):1 Trama 7 ToT 4 T-Supervisión N(R):1 RR Trama 8 ToT 4 T-Supervisión N(R):1 RR Trama 9 ToT 172 T-Información N(s):1 N(r):1 Trama 10 ToT 4 T-Supervisión N(R):2 RR Trama 11 ToT 95 T-Información N(s):1 N(r):2 Trama 12 ToT 4 T-Supervisión N(R):2 RR Trama 13 ToT 145 T-Informacion N(s):2 N(r):2 Trama 14 ToT 4 T-Supervisión N(R):3 RR Trama 15 ToT 70 T-Informacion N(s):2 N(r):3 Trama 16 ToT 4 T-Supervisión N(R):3 RR Trama 17 ToT 126 T-Información N(s):3 N(r):3 Trama 18 ToT 4 T-Supervisión N(R):4 RR Trama 19 ToT 4 T-Supervisión N(R):4 RR Trama 20 ToT 126 T-Informacion N(s):4 N(r):4 Trama 21 ToT 4 T-Supervisión N(R):5 RR Trama 22 ToT 4 T-Supervisión N(R):5 RR Trama 23 ToT 18 T-Información N(s):5 N(r):5 Trama 24 ToT 4 T-Supervisión N(R):6 RR Trama 25 ToT 139 T-Unnumbered

Trama 1	ToT 3	T-Unnumerated Comando		SABME	
Trama 2	ToT 3	T-Unnumerated Respuesta		UA	
Trama 3	ToT 4	T-Supervisión	N(R):0		RR
Trama 4	ToT 4	T-Supervisión	N(R):0		RR
Trama 5	ToT 18	T-Infomacion N(s):0	N(r):0		
Trama 6	ToT 18	T-Infomacion N(s):0	N(r):1		
Trama 7	ToT 4	T-Supervisión	N(R):1		RR
Trama 8	ToT 4	T-Supervisión	N(R):1		RR
Trama 9	ToT 172	T-Infomación N(s):1	N(r):1		
Trama 10	ToT 4	T-Supervisión	N(R):2		RR
Trama 11	ToT 95	T-Infomación N(s):1	N(r):2		
Trama 12	ToT 4	T-Supervisión	N(R):2		RR
Trama 13	ToT 145	T-Infomación N(s):2	N(r):2		
Trama 14	ToT 4	T-Supervisión	N(R):3		RR
Trama 15	ToT 70	T-Infomacion N(s):2	N(r):3		
Trama 16	ToT 4	T-Supervisión	N(R):3		RR
Trama 17	ToT 126	T-Infomación N(s):3	N(r):3		
Trama 18	ToT 4	T-Supervisión	N(R):4		RR
Trama 19	ToT 4	T-Supervisión	N(R):4		RR
Trama 20	ToT 126	T-Infomacion N(s):4	N(r):4		
Trama 21	ToT 4	T-Supervisión	N(R):5		RR
Trama 22	ToT 4	T-Supervisión	N(R):5		RR
Trama 23	ToT 18	T-Infomación N(s):5	N(r):5		
Trama 24	ToT 4	T-Supervisión	N(R):6		RR
Trama 25	ToT 139	T-Unnumerated Sin PF			
Trama 26	ToT 53	T-Infomacion N(s):6	N(r):5		
Trama 27	ToT 53	T-Infomacion N(s):6	N(r):7		
Trama 28	ToT 18	T-Infomación N(s):7	N(r):6		
Trama 29	ToT 4	T-Supervisión	N(R):8		RR
Trama 30	ToT 18	T-Infomacion N(s):8	N(r):6		
Trama 31	ToT 4	T-Supervisión	N(R):9		RR
Trama 32	ToT 3	T-Unnumerated Comando		DISC	
Trama 33	ToT 3	T-Unnumerated Respuesta		UA	

Sin PF Trama 26 ToT 53 T-Infomacion N(s):6 N(r):5 Trama 27 ToT 53 T-Infomacion N(s):6 N(r):7 Trama 28 ToT 18 T-Infomación N(s):7 N(r):6 Trama 29 ToT 4 T-Supervisión N(R):8 RR Trama 30 ToT 18 T-Infomacion N(s):8 N(r):6 Trama 31 ToT 4 T-Supervisión N(R):9 RR Trama 32 ToT 3 T-Unnumerated Comando DISC Trama 33 ToT 3 T-Unnumerated Respuesta UA

## 7. Conclusión

Realizar este tipo de programa evita tener que realizar operaciones que tienden a error humano, tales como olvidar un bit o colocar alguno demás, ademas de así mejorar el tiempo en que se realizaría.

Como se vió mecanizarlo es

## 8. Código

```
1  /##### TRAMA ANALIZER #####
2  * author Laura Andrea Morales* version 0.1
3  * team CompilandoConocimiento* date 4/03/2018
4  * compile "gcc TramaLLC.c -o TramaLLC" * run "./TramaLLC "
5  */
6  #include <stdlib.h>
7  #include <stdio.h>
8  #include <stdbool.h>
9  #include <string.h>
10 #define C1 16
11 #define C2 17
12 #define NoTramas 33
13 typedef unsigned int uint;
14 typedef unsigned char byte;
15 typedef unsigned int sint;
16
17 char Supervision [4][5]={ "RR", "RNR", "REJ", "SREJ" };
18 char
19   UnumerC[33][7]={ "UI", "SIM", "-", "SARM", "UP", "-", "-", "SABM", "DISC", "-", "-", "SARME", "-", "-", "-", "SABME", "SNRM", "-----"
20   UA, "-", "-", "-", "FRMR", "-", "-----", "XID" };
21
22 char resultado[100000]={};
23
24 bool Analizer(byte T[], int i){
25   sint ToT=0;
26   ToT=(ToT<<8)|T[12];
27   ToT=T[13];
28   printf(resultado+strlen(resultado), "\n Trama %-3d ", i+1);
29   printf(resultado+strlen(resultado), " ToT %-4d ", ToT);
30
31   if(ToT<1500){
32     //printf(resultado+strlen(resultado), "Tipo %d ", T[C1]&0b11);
33     switch(T[C1]&0b11){
34       case 0:
35         // printf("Entre a caso 0\n");
36         printf(resultado+strlen(resultado), "T-Informacion N(s):%-5dN(r):%-5d", T[C1]>>1, T[C2]>>1);
37         break;
38
39       case 1:
40         // printf("Entre a caso Trama de Supervisión\n");
41         //printf("%d\n", (T[C1]>>2)&3);
42
43         printf(resultado+strlen(resultado), "T-Supervisión N(R):%-5d %s", T[C2]>>1, Supervision[(T[C1]>>2)&3]);
44         break;
45
46       case 2:
47         // printf("Entre a caso Trama de Información\n");
48         printf(resultado+strlen(resultado), "T-Información N(s):%-5dN(r) %-5d", T[C1]>>1, T[C2]>>1);
49         break;
50
51       case 3:
52         //printf("Entre caso 3 Unnumbered\n");
53         if (((T[C1]>>4)&1))
54         {
55           // printf("Checa Sap\n");
56           if (T[15]&1)
57           {
58             //printf("Respuesta %d", (T[C1]>>2)&3)|((T[C1]>>3)&28);
59             printf(resultado+strlen(resultado), "T-Unnumbered Respuesta %15s", UnumerR[((T[C1]>>2)&3)|((T[C1]>>3)&28)]);
60           }
61         }
62       }
```

```

61         else if (!(T[15]&1))
62         {
63             //printf("Comando %d", (T[C1]>>2)&3) | ((T[C1]>>3)&28);
64             sprintf(resultado+strlen(resultado), "T-Unnumerated Comando    %13s",
        UnumerC[((T[C1]>>2)&3) | ((T[C1]>>3)&28)]);
65         }
66     }
67     }
68     else sprintf(resultado+strlen(resultado), "T-Unnumerated Sin PF");
69
70
71
72     break;
73
74
75
76     }
77     return 1;
78 }
79 else return 0;
80 }
81
82 bool AnalizarTrama(byte T[][250]){
83
84     for (int i = 0; i < NoTramas; ++i)
85     {
86         if (!(Analyzer(&T[i][0], i))){
87
88             return 0;
89         }
90     }
91     return 1;
92 }
93
94 }
95
96
97 int main(int argc, char const *argv[]) {
98
99     byte T[33][250]={
100     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,
101     0x7f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 1
102     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1,
103     0x73,0x81,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 2
104     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf0,
105     0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 3
106     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
107     0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 4
108     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
109     0x00,0x01,0x0e,0x00,0xff,0xef,0x19,0x8f,0xbc,0x05,0x7f,0x00,0x23,0x00,0x7f,0x23}, //Trama 5
110     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x12,0xf0,0xf0,0x00,0x03,0x0e,0x00,0xff,0xef,0x19,0x8f,
111     //Trama 6
112     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 7
113     //Trama 8
114     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0xac,0xf0,0xf0,0x02,0x02,0x0e,0x00,0xff,0xef,0x16,0x04,
115     //Trama 9
116     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 10
117     //Trama 11
118     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,0x01,0x04,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 12
119     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x91,0xf0,0xf0,0x04,0x04,0x0e,0x00,0xff,0xef,0x16,0x0c,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 13
120     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 14
121     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 15
122     {0x00,0x02,0xb3,0x9c,0xff,0xef,0x16,0x0c,0x00,0x00,0x00,0x28,0x00,0x28,0x00,0x7f,0x23}, //Trama 16
123     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 17
124     //Trama 18
125     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 19
126     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 20
127     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 21
128     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 22
129     {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 23
130     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 24
131     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 25
132     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 26
133     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 27
134     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 28
135     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 29
136     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 30
137     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 31
138     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 32
139     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 33
140     {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,0x01,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 34

```



```

141 | 0x01,0x0d,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, //Trama 24
142 | {0x03,0x00,0x00,0x00,0x00,0x01,0x00,0x04,0xae,0x44,0x4d,0x02,0x00,0x8b,0xf0,0xf0,0x03,0x2c,0x00,0xff,0xef,0x08,0x00,0x00,
    | //Trama 25
143 | {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0,
    | 0x0c,0x0a,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 26
144 | {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x35,0xf0,0xf0,
    | 0x0c,0x0e,0x0e,0x00,0xff,0xef,0x16,0x04,0x00,0x00,0x00,0x00,0x28,0x00,0x7f,0x23}, //Trama 27
145 | {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
    | 0x0e,0x0d,0x0e,0x00,0xff,0xef,0x14,0x00,0x00,0x00,0x28,0x00,0x00,0x00,0x7f,0x23}, //Trama 28
146 | {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
    | 0x01,0x11,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 29
147 | {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x12,0xf0,0xf0,
    | 0x10,0x0d,0x0e,0x00,0xff,0xef,0x18,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7f,0x23}, //Trama 30
148 | {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x04,0xf0,0xf1,
    | 0x01,0x13,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 31
149 | {0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x03,0xf0,0xf0,
    | 0x53,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}, //Trama 32
150 | {0x00,0x02,0xb3,0x9c,0xdf,0x1b,0x00,0x02,0xb3,0x9c,0xae,0xba,0x00,0x03,0xf0,0xf1,
    | 0x73,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}; //Trama 33
    | };
151 |
152 |
153 |
154 |
155 |
156 |
157 |
158 |
159 |
160 |
161 |     if(AnalizarTrama(T)){
162 |         printf("%s\n",resultado);
163 |     }
164 |
165 |     return 0;
166 | }

```

## 9. Mapa de memoria

## Referencias

- [1] Axel Ernesto Moreno Cervantes *Redes de Computación*. ESCOM, 2018.
- [2] Nidia Cortez. *Redes de Computadoras* ESCOM, 2018.