



**INSTITUT  
O  
POLITÉC  
NICO  
NACIONA  
L  
ESCUELA  
SUPERIO  
R DE  
CÓMPUT  
O**





**INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO**



## **REDES DE COMPUTADORAS**

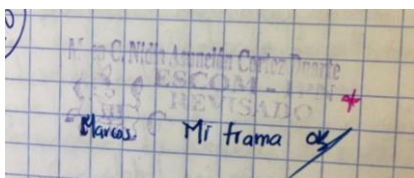
### **“MI TRAMA-ESTRUCTURA”**

#### **Abstract**

In this report we will see an structure of the frame, where you can find out the different parts on the basic frame. By the way, here you will check the languages, routers, ID and line code.

**By:**

**MARCOS OSWALDO VÁZQUEZ MORENO**



**Professor:**

**MSc. NIDIA ASUNCIÓN CORTEZ DUARTE**

**DICIEMBRE 2018**

1	Mensaje	n bytes						
	Idioma	3 bits	010	Español				
			011	Inglés				
			100	Portugués				
			101	Francés				
110	Alemán							
2	Enrutamiento	1 bit	0	Estático				
			1	Dinámico				
	Id origen/destino	1 byte	01	2CM7				
			10	3CM5				
11	5CM3							
3	Control de Flujo	2-10 bits	0	Parar y esperar				
			1	Ventana deslizante		0	Go Back N	{
					1	Rechazo Selectivo		
	Control de Error	4-35 bits	00	Bit de paridad		0	Par	FCS
					1	Impar		
			01	CRC		0	8 bits	
					1	32 bits		
11	Checksum 32 bits							
4	Medio	3 bits	0 Alámbrico		01	Fibra Óptica		
					10	Cable Coaxial		
					11	UTP		
			1 Inalámbrico		00	Infrarrojo		
					01	WiFi		
					10	NFC		
	Código de línea	6 bits	0	Unipolar				
			0	Polar				
			0	RZ				
			0	NRZ				
			0	AMI				
0			Manchester					

```

1.  /*
2.  VAZQUEZ MORENO MARCOS OSWALDO
3.  REDES DE COMPUTADORAS
4.  EJERCICIO 1
5.  CLASIFICACION DE CLASES EN UNA DIRECCION IP, MASCARA DE RED, TIPO, RANGO DE HOST,
    BROADCAST, REPETIR
6.  */
7.
8.  #include<stdio.h>
9.  #include<stdlib.h>
10.
11. //int validacion(int ip);
12.
13. int main(){
14.
15.
16. //arreglo de cuatro octetos
17. unsigned char IP[4]= {0,0,0,0}, MR[4]= {0,0,0,0};
18. char a= 'y';
19. do{
20.
21.     printf("ingrese direccion ip valida: \n");
22.     scanf("%u.%u.%u.%u",&IP[0], &IP[1], &IP[2], &IP[3]);
23.     fflush(stdin);
24. //validacion para entrar al distinto tipo de casos
25. if(IP[0] & 128) //clase A
26.     {
27.         if(IP[0] & 64) //clase B
28.         {
29.             if(IP[0] & 32) //clase C
30.             {
31.                 if(IP[0] & 16) //clase D
32.                 {
33.                     if(IP[0] & 8) //clase E
34.                     {
35.                         printf("Clase E\n");
36.                     }
37.                     else
38.                     {
39.                         printf("Clase E\n");
40.                     }
41.                 }
42.                 else
43.                 {
44.                     printf("Clase D\n");
45.                 }
46.             }
47.             else
48.             {
49.                 printf("-----\n");
50.                 printf("CLASE C\n");
51.                 MR[0]= 255; //VALORES DE LS OCTETOS DE CADA MASCARA DE RED
52.                 MR[1]= 255;
53.                 MR[2]= 255;
54.                 MR[3]= 0;
55.                 if(IP[3]==255){
56.                     printf("TIPO BROADCAST\n");
57.                 }
58.                 else if(IP[3]==0){

```

```

59.             printf("TIPO RED\n");
60.         }
61.         else
62.             printf("TIPO HOST\n");
63.         printf("MASCARA DE RED: %u.%u.%u.%u\n", MR[0],MR[1], MR[2], MR[3])
;
64.         printf("RED: %u.%u.%u.%u\n", IP[0]&MR[0], IP[1]&MR[1], IP[2]&MR[2]
, IP[3]&MR[3]);
65.         printf("BROADCAST: %u.%u.%u.%u\n", IP[0] | (unsigned char)~MR[0],
IP[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], IP[3] | (unsigned ch
ar)~MR[3]); //CASTEO DE ENTERO A UNSIGNED CHAR
66.         printf("RANGO DE HOST: %u.%u.%u.%u A %u.%u.%u.%u\n", IP[0] & MR[0]
, IP[1] & MR[1], IP[2] & MR[2], (IP[3] &MR[3])+1, IP[0] | (unsigned char)~MR[0], I
P[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], (IP[3] | (unsigned ch
ar)~MR[3])-1);
67.     }
68. }
69. else
70. {
71.     printf("-----\n");
72.     printf("CLASE B\n");
73.     MR[0]= 255; //VALORES DE LS OCTETOS DE CADA MASCARA DE RED
74.     MR[1]= 255;
75.     MR[2]= 0;
76.     MR[3]= 0;
77.     if(IP[3]==255 && IP[2]==255){
78.         printf("TIPO BROADCAST\n");
79.     }
80.     else if(IP[3]==0 && IP[2]==0){
81.         printf("TIPO RED\n");
82.     }
83.     else
84.         printf("TIPO HOST\n");
85.     printf("MASCARA DE RED: %u.%u.%u.%u\n", MR[0],MR[1], MR[2], MR[3]);
86.     printf("RED: %u.%u.%u.%u\n", IP[0]&MR[0], IP[1]&MR[1], IP[2]&MR[2], IP
[3]&MR[3]);
87.     printf("BROADCAST: %u.%u.%u.%u\n", IP[0] | (unsigned char)~MR[0], IP[1]
| (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], IP[3] | (unsigned char)~
MR[3]);
88.     printf("RANGO DE HOST: %u.%u.%u.%u A %u.%u.%u.%u\n", IP[0] & MR[0], IP
[1] & MR[1], IP[2] & MR[2], (IP[3] &MR[3])+1, IP[0] | (unsigned char)~MR[0], IP[1]
| (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], (IP[3] | (unsigned char)~
MR[3])-1);
89.     }
90. }
91. else
92. {
93.     printf("-----\n");
94.     printf("CLASE A\n");
95.     MR[0]= 255; //VALORES DE LS OCTETOS DE CADA MASCARA DE RED
96.     MR[1]= 0;
97.     MR[2]= 0;
98.     MR[3]= 0;
99.     if(IP[3]==255 && IP[2]==255 && IP[1]==255){
100.         printf("TIPO BROADCAST\n");
101.     }
102.     else if(IP[3]==0 && IP[2]==0 && IP[1]==0){
103.         printf("TIPO RED\n");
104.     }
105.     else
106.         printf("TIPO HOST\n");

```

```

107.
108.         printf("MASCARA DE RED: %u.%u.%u.%u\n", MR[0], MR[1], MR[2], MR[3]
);
109.         printf("RED: %u.%u.%u.%u\n", IP[0] & MR[0], IP[1] & MR[1], IP[2] &
MR[2], IP[3] & MR[3]);
110.         printf("BROADCAST: %u.%u.%u.%u\n", IP[0] | (unsigned char)~MR[0],
IP[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], IP[3] | (unsigned ch
ar)~MR[3], IP[4] | (unsigned char)~MR[4]);
111.         printf("RANGO DE HOST: %u.%u.%u.%u A %u.%u.%u.%u\n", IP[0] & MR[0]
, IP[1] & MR[1], IP[2] & MR[2], (IP[3] & MR[3])+1, IP[0] | (unsigned char)~MR[0], I
P[1] | (unsigned char)~MR[1], IP[2] | (unsigned char)~MR[2], (IP[3] | (unsigned ch
ar)~MR[3])-1);
112.     };
113.
114.     printf("desea repetir el proceso (Y/N)\n");
115.     scanf("%c", &a);
116.     fflush(stdin);
117.
118.     }while(a == 'Y' || a == 'y');
119.     return 0;
120.
121.     }

```

A continuación, en la imagen 1 se muestra la salida del código y su ejecución.

```

C:\Users\marco\Documents\ESCOM\QUINTO SEMESTRE\REDES\pr8ctica1.exe
Ingrese direccion ip valida:
128.12.12.12
-----
CLASE B
TIPO HOST
MASCARA DE RED: 255.255.0.0
RED: 128.12.0.0
BROADCAST: 128.12.255.255
RANGO DE HOST: 128.12.0.1 A 128.12.255.254
desea repetir el proceso (Y/N)

```

Imagen 1. Captura mi Trama.