

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

**Redes de computadoras**

**Práctica 3 “Mi trama”**

ALUMNO:

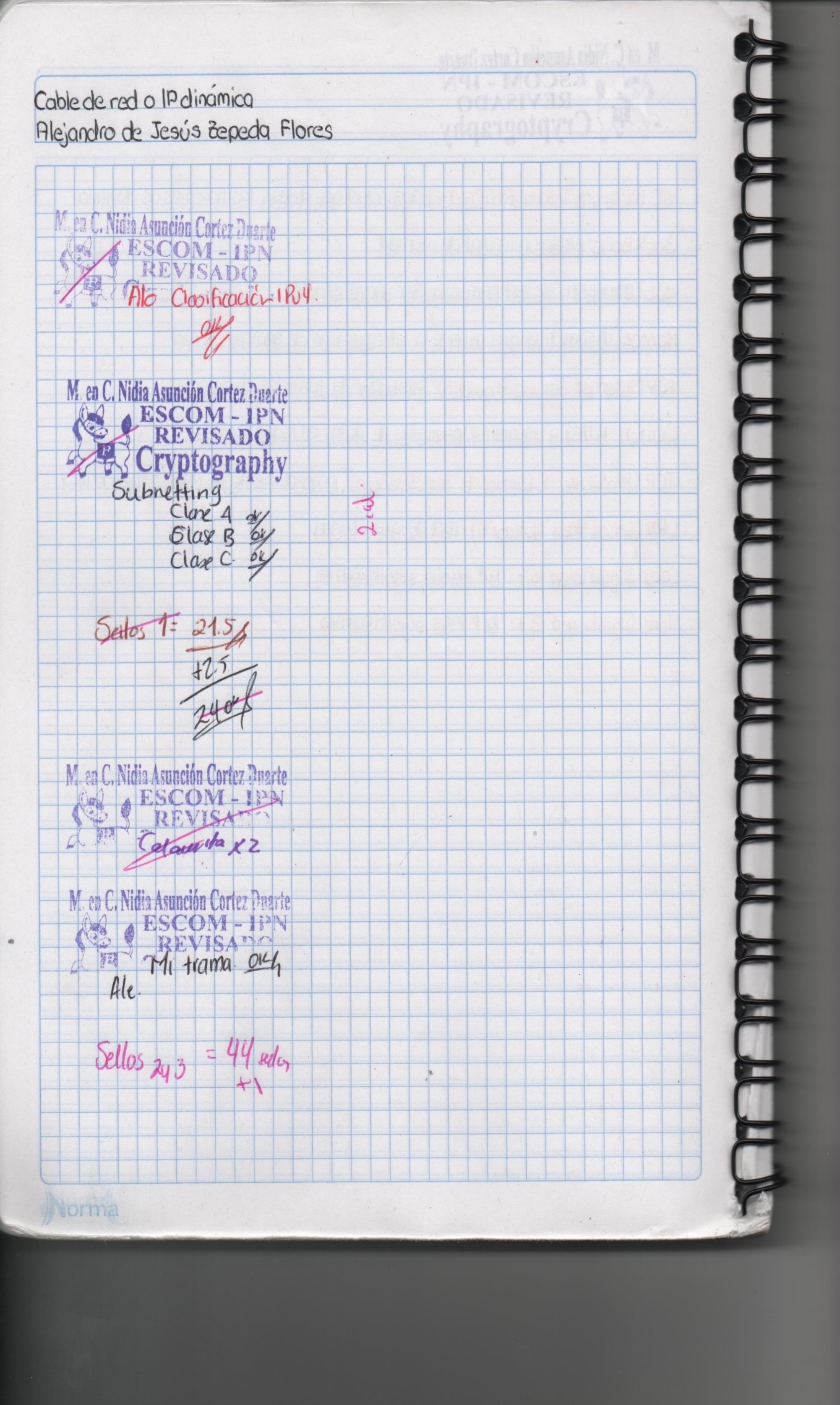
ZEPEDA FLORES ALEJANDRO DE JESÚS

PROFESOR:

M. en C. NIDIA ASUNCIÓN CORTEZ DUARTE

g

DICIEMBRE 2018



# **Índice**

Contenido

[**Índice** 2](#_Toc531538654)

[**Objetivo** 3](#_Toc531538655)

[**Material y equipo** 3](#_Toc531538656)

[**Marco teórico** 3](#_Toc531538657)

[**Resultados** 4](#_Toc531538658)

[**Conclusiones** 6](#_Toc531538659)

[**Código** 6](#_Toc531538660)

[**Bibliografía** 9](#_Toc531538661)

# **Objetivo**

Desarrollar un programa en lenguaje C capaz de analizar una trama mediante el protocolo implementado, dicho programa recibirá una trama en forma de arreglo hexadecimal y analizará cada una de las 4 capas para imprimir las características del protocolo.

# **Material y equipo**

* Equipo de cómputo con el sistema operativos Windows
* Entorno de desarrollo (Sublime Text)
* Compilador para programas en C (GCC)

# **Marco teórico**

La trama es la unidad de datos que utiliza la capa de enlace. Más correctamente, según el estándar ISO/OSI, debería denominarse LPDU (Link Protocol Data Unit). Las tramas generadas por la capa de enlace de datos viajan desde un nodo hasta otro que esta conectado al mismo medio físico de comunicación que el anterior. Una trama nunca salta de una subred a otra, sino que son el vehículo para transmitir datos que pueden estar viajando más allá de una subred, para atravesar un medio físico determinado.

La trama facilita la sincronización en la comunicación entre las entidades de la cada de Enlace, que consiste en la localización del comienzo y el final del bloque de información transmitido. Además, el protocolo ha de permitir la transmisión de cualquier tipo de datos que no deben de ser confundidos con información de control del protocolo, aunque su codificación coincida.

En una transmisión sincrónica se requiere de un nivel de sincronización adicional para que el receptor pueda determinar dónde está el comienzo y el final de cada bloque de datos. Normalmente, una trama constará de una cabecera, datos y cola. En la cola suele estar algún chequeo de errores. EN la cabecera habrá campos de control de protocolo y en la parte de datos es la que se quiere transmitir.

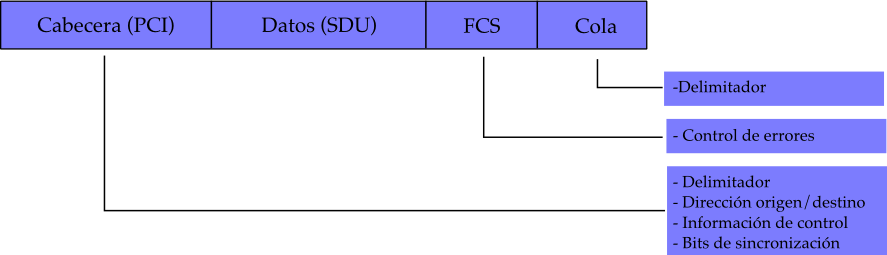


Imagen 1. Estructura básica de la trama

# **Resultados**

Para analizar los resultados, primero debemos mostrar la trama que se tiene que analizar. En el arreglo de caracteres trama, se encuentra declarada en formato hexadecimal.

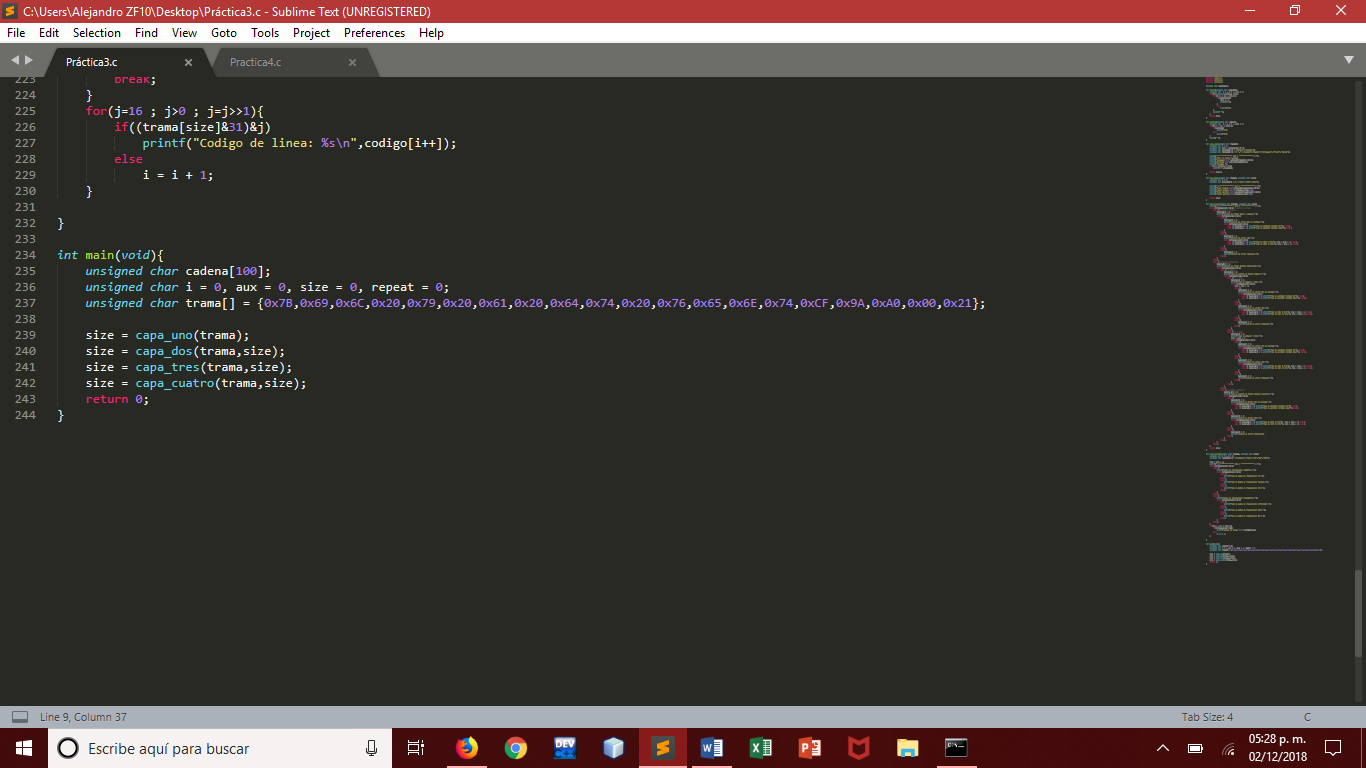


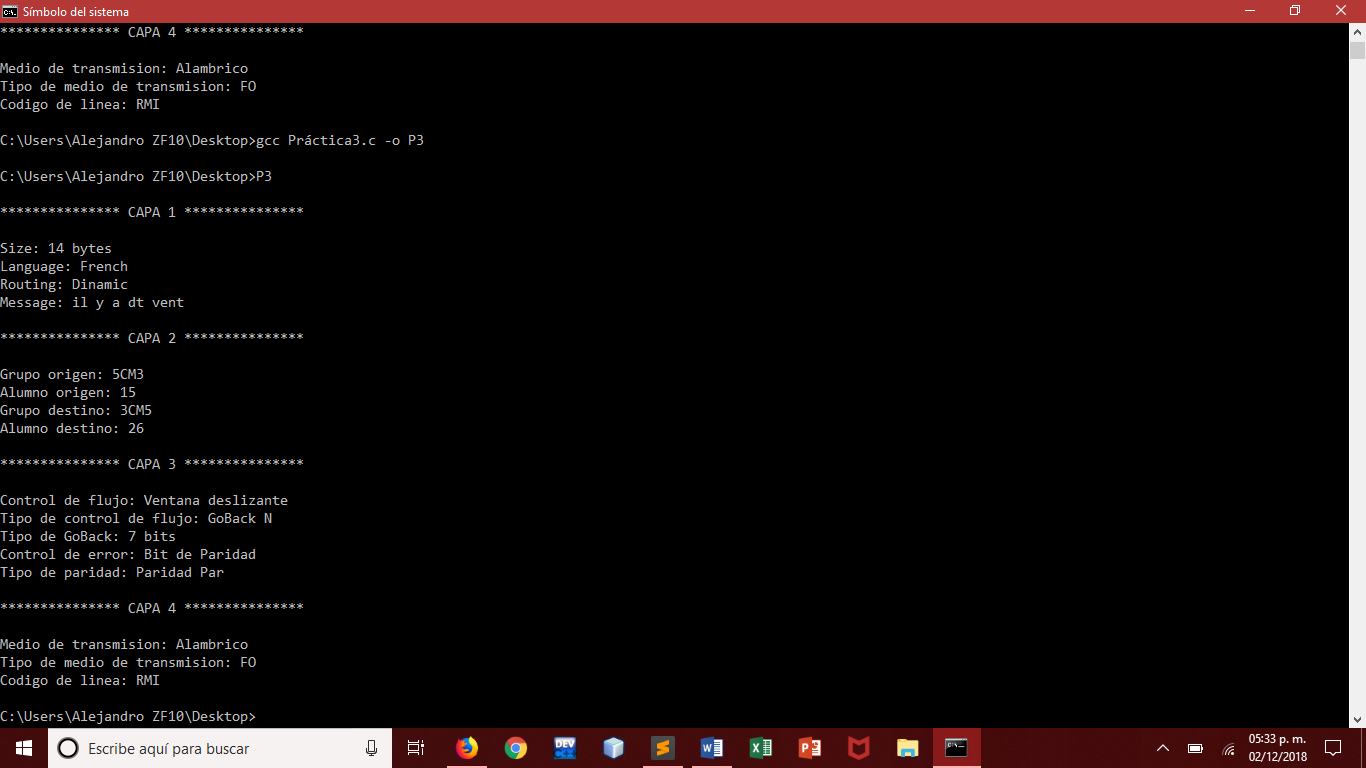
Imagen 2. Declaración de la trama

Además de la imagen anterior, podemos rescatar que la trama es analizada capa por capa y de manera individual, evitando errores como de posicionamiento o análisis incorrecto dentro de la trama.

El análisis de la trama es relativamente, de manera sencilla, ya que la tarea fundamental del programa es analizar cada una de las posiciones de la misma, determinar en cual se encuentra y aplicar el análisis de la capa correspondiente según su posición.

Un dato relevante y fundamental dentro de nuestro análisis, es el tamaño de la trama (Ver imagen 2). El análisis se basa en esta variable, ya que es la limitante para nuestro ciclo que recorrerá la trama.

Sin este dato, no sería imposible, pero se dificultaría el análisis, ya que indica la pauta para el corrimiento.

 Imagen 3. Impresión de protocolo

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Unsigned char size = 0; |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | Unsigned char trama[]; |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

Tabla 1. Mapa de memoria inicial

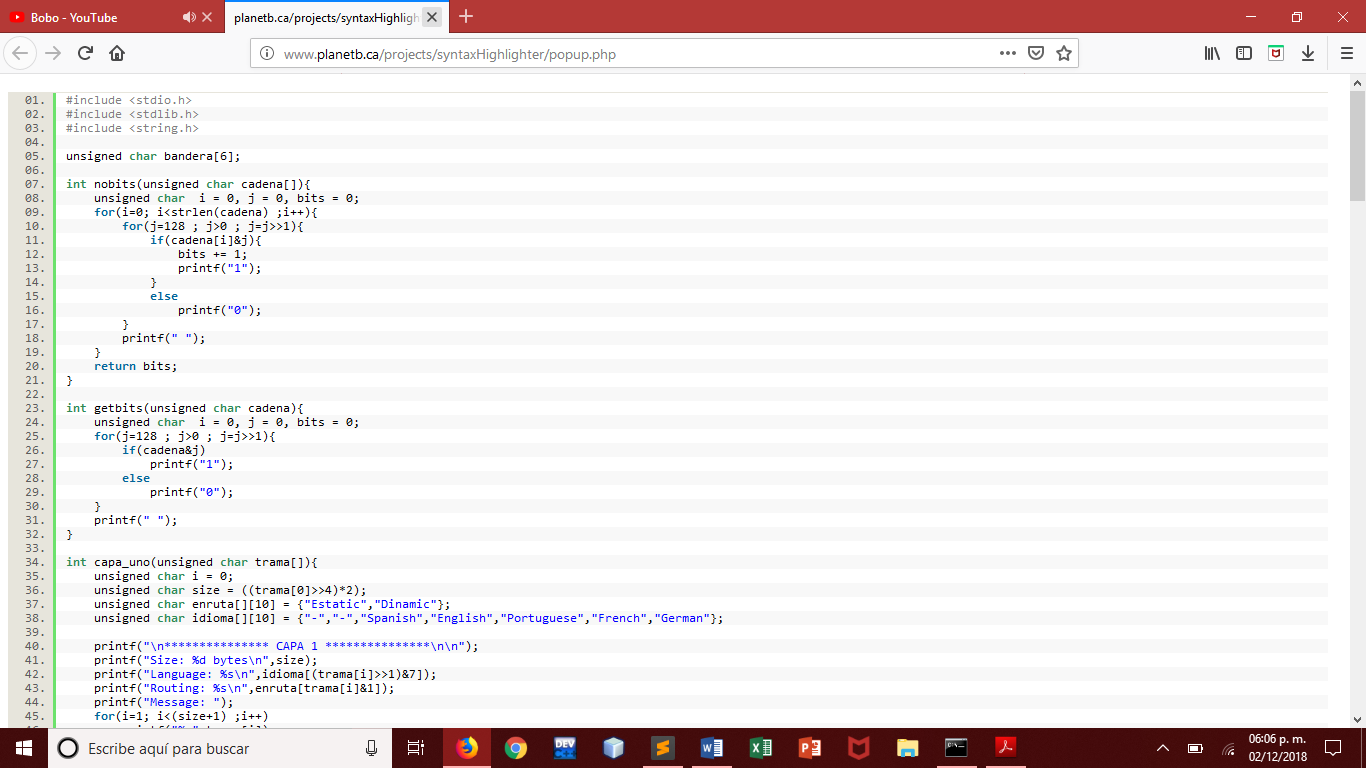
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | Unsigned char trama[]; |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

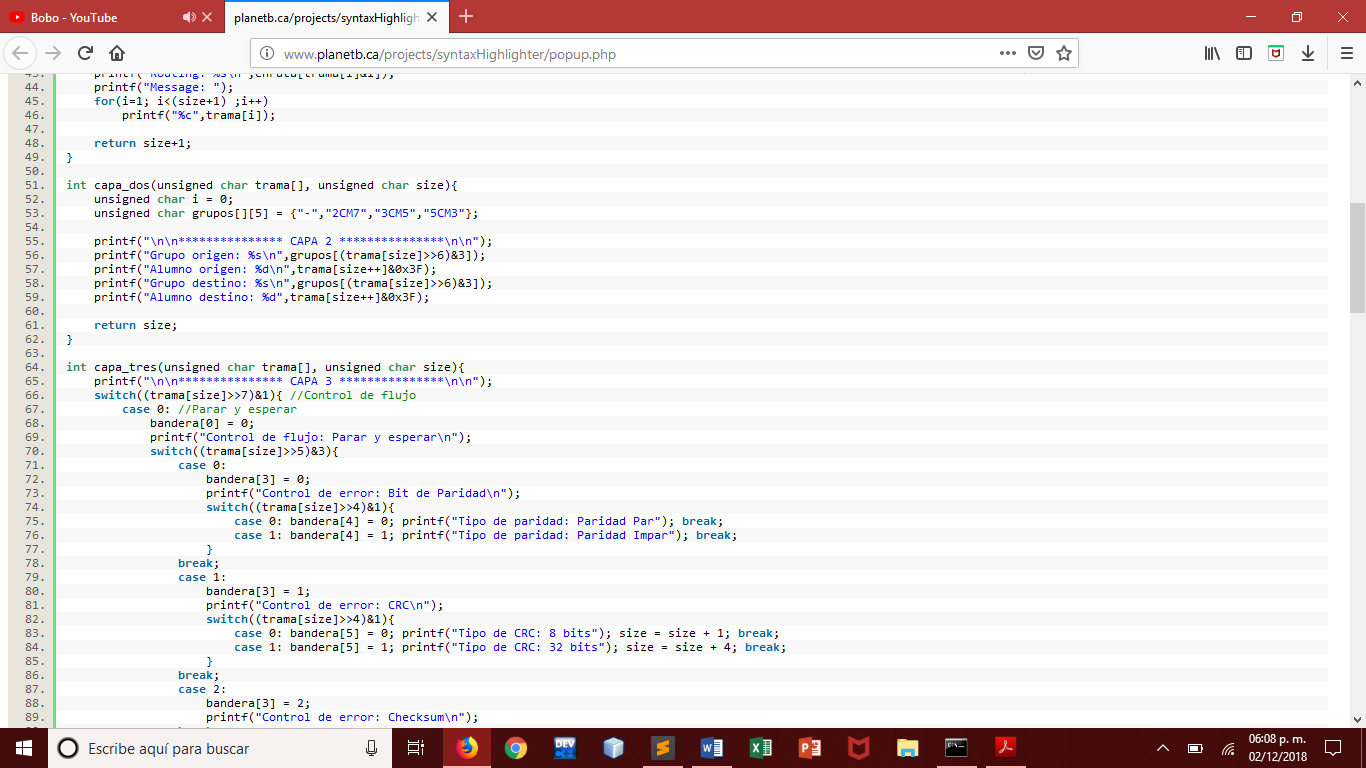
Tabla 2. Prueba de escritorio

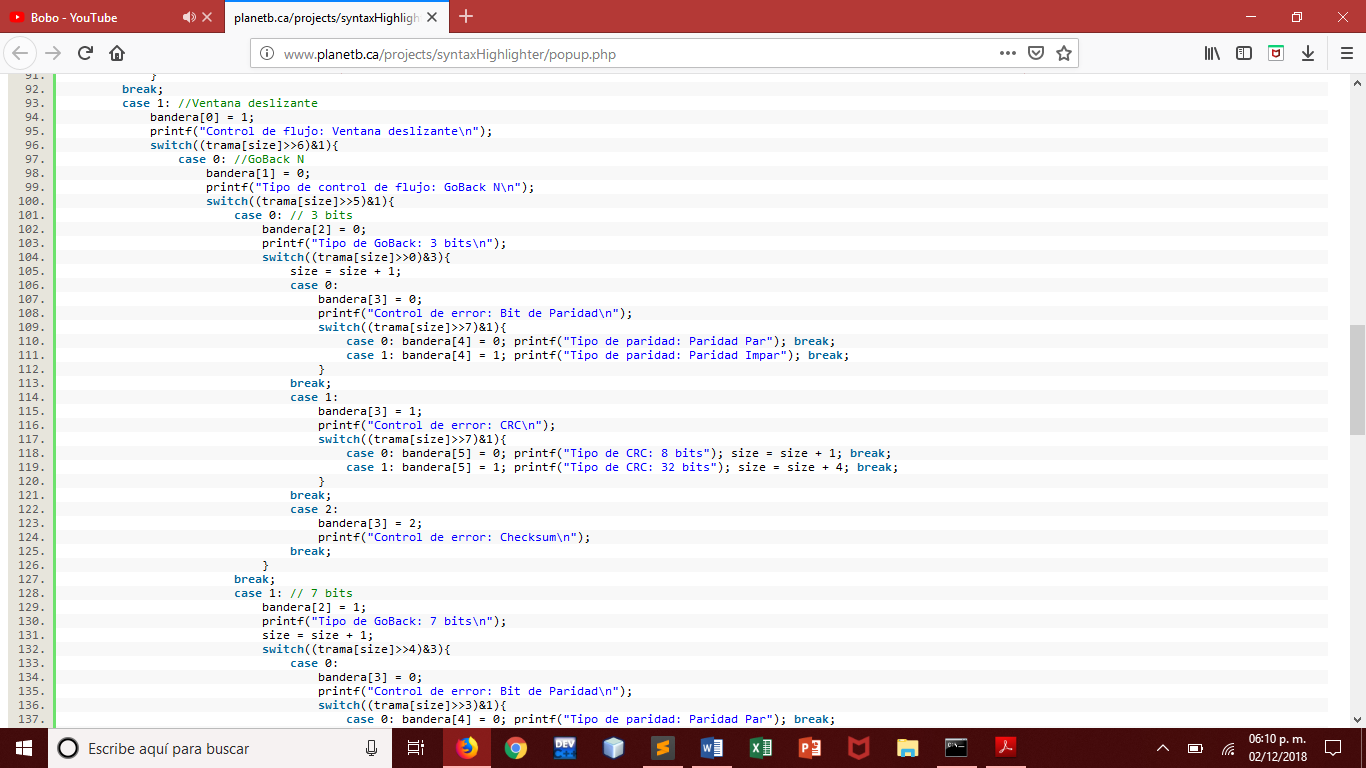
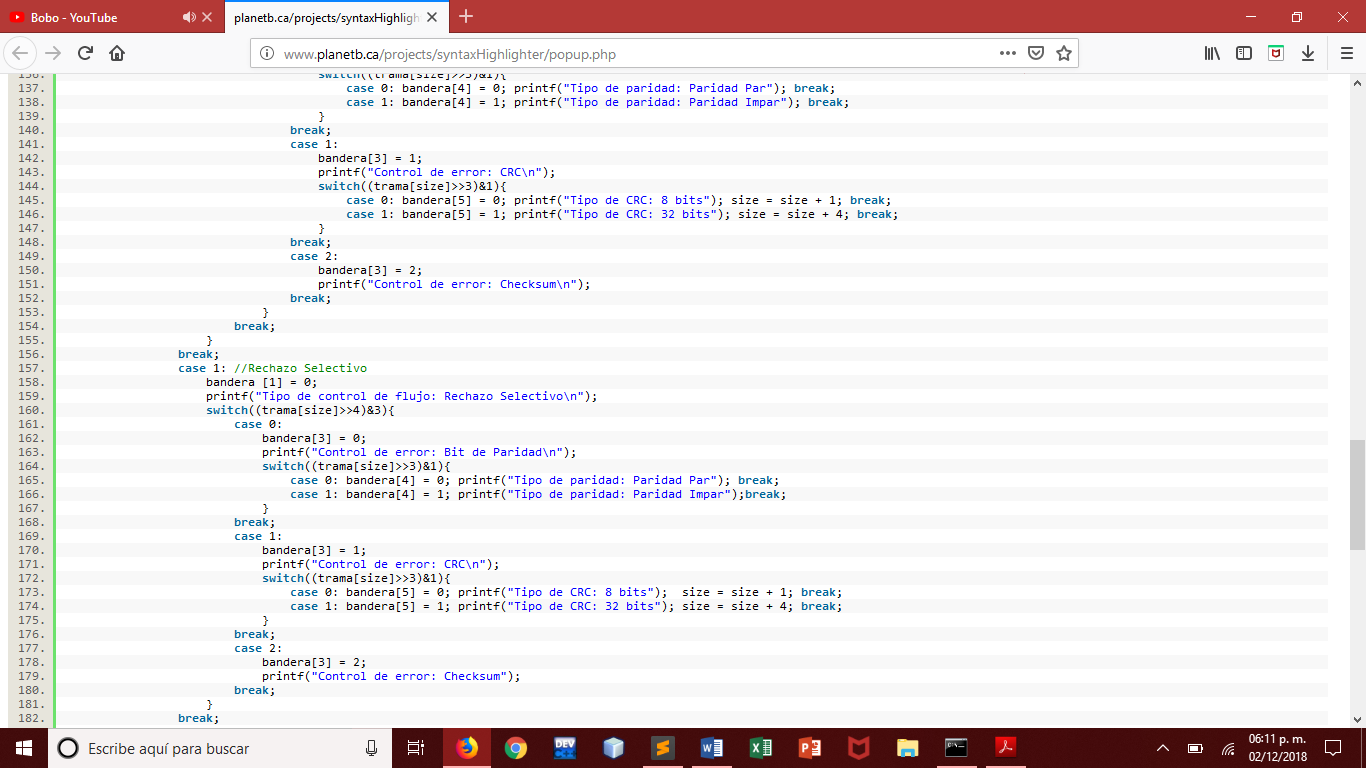
# **Conclusiones**

El objetivo de la capa de enlace es lograr la comunicación entre 2 máquinas que estén conectadas fluya y sea libre. Para lograr este objetivo, se tienen que montar bloques de información, denominados tramas, para poder transmitir la información, gestionar o detectar la corrección de errores y ocuparse del control de flujo entre equipos. De ahí la importancia de la trama, para facilitar la transmisión de la información.

# **Código**





# **Bibliografía**

* S/A. (2003). Capa de Enlace. Oviedo: Universidad de Oviedo.
* Sloman, M; Kramer, J. (1987) Distributed Systems and Computer Networks. Prentice-Hall