

**Nombre:** Hans Buker Gutiérrez.

**Código Fuente:** (Ver Anexo: CódigoFuente.txt)

Para efecto de mayor entendimiento se dejaron fuera los métodos nombreDeUsuario y ventanaSiguiente debido a que estos métodos solo contienen el diseño visual de la interfaz de usuario.

Se adjuntan las otras clases y en específico de PanelMenuInicial los listener.

**Clase Tarea1Redes:** Esta clase contiene al método de ejecución main, además de las variables de inicialización de la conexión al servidor TCP con el protocolo AVIUBB.

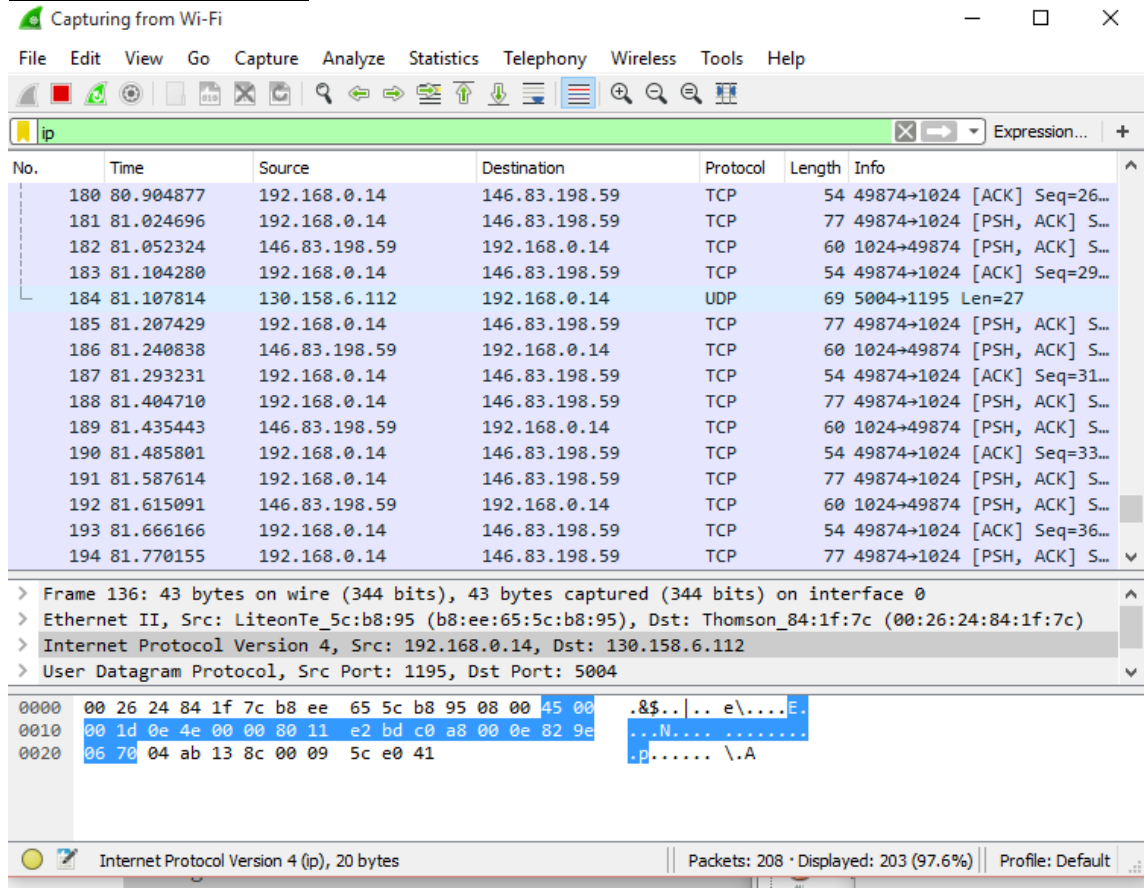
**Clase VentanaMenuInicial:** Esta clase no es de mayor relevancia debido a que es utilizada netamente para la llamada de la interfaz de usuario.

**Clase PanelMenuInicial:** Esta clase hace uso de los elementos visuales que conformaran la interfaz de usuario junto a las funcionalidades que ofrecen los ActionListener, los cuales utilizaremos para realizar las respectivas funcionalidades del protocolo.

- **public void pulsarBotonIniciarSesion(ActionEvent e):**
  - Este método se encargara de ejecutar las tareas relacionadas con NEW\_USER, junto a sus respectivas restricciones como lo es ingresar un nombre vacío o solo un espacio.
- **public void enviarMensaje(ActionEvent e):**
  - Este método se encarga de ejecutar las tareas relacionadas con SET, junto a su respectiva restricción de ingresar un nombre de usuario y un mensaje a la vez.
- **public void obtenerMensaje(ActionEvent e):**
  - Este método se encargara de las tareas relacionadas con GET, junto a la restricción de pedir un nombre de usuario para poder ser ejecutada.
- **public void obtenerUltimoUsuario(ActionEvent e):**
  - Este método se encargara de las tareas relacionadas con LAST, junto a la restricción de pedir un nombre de usuario para poder ser ejecutada.
- **public void pulsarBotonSalir(ActionEvent e):**
  - Este método da la funcionalidad de salir del programa y enviar el mensaje FIN al servidor, tiene la restricción de tener que pedir un nombre de usuario al servidor antes de finalizar el programa.
- **Dificultades.**
  - Comprendí la complejidad de mantener la conexión siempre activa, la cual al principio me resulto bastante difícil debido a que el programa solo se estaba ejecutando desde el constructor, por eso tuve que implementar una forma de llamar constructores anidados entre sí, con la peculiaridad de siempre ir pasando los datos de conexión declarados al principio del programa, por un golpe de suerte por así decirlo, logre enviar un mensaje al servidor por medio de uso de un botón, por lo cual

decidí trabajar en base a lo primero que me resulto para poder así finalizar el programa.

- **Aprendizajes:** Aprendí a utilizar el protocolo TCP fragmentado en varias clases, lo cual tiene un nivel elevado de complejidad ya que, esta conexión es muy fácil de perder, siempre hay que mantener la conexión viva por así decirlo entre medio de todas las clases.
- **Captura en Wireshark:**



Envié un mensaje una sucesiva cantidad de veces para que este pueda ser visto claramente.