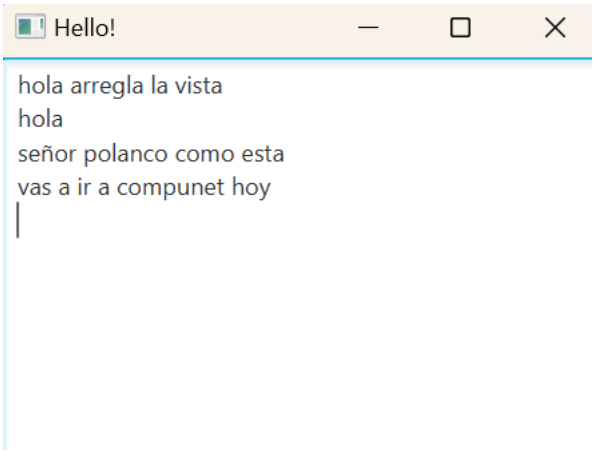
Ricardo Andrés Chamorro Martínez A00399846
Diego Armando Polanco Lozano A00399926

Capturas:

Como se puede ver, si aparecen los mensajes y su contenido




```
   875 92.604342    192.168.105.70          192.168.105.150      UDP        47 5001 → 5001 Len=5
   971 119.844299   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
   972 120.928564   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
   977 121.891417   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
   981 122.909422   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
  1022 131.515523   192.168.105.70          192.168.105.150      UDP        47 5001 → 5001 Len=5
  1050 146.058675   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
  1056 146.990554   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
  1106 159.778046   192.168.105.70          192.168.105.150      UDP        66 5001 → 5001 Len=24
  1520 191.405992   192.168.105.70          192.168.105.150      UDP        65 5001 → 5001 Len=23
```

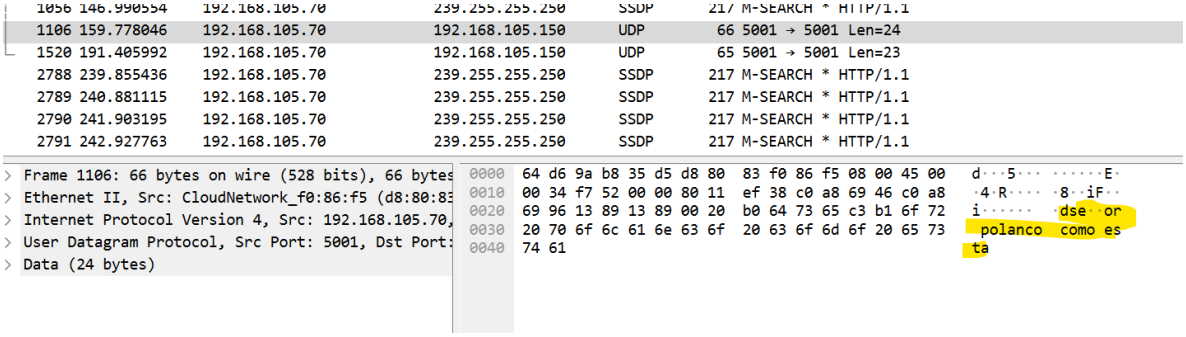
```
> Frame 875: 47 bytes on wire (376 bits), 47 bytes
> Ethernet II, Src: CloudNetwork_f0:86:f5 (d8:80:83
> Internet Protocol Version 4, Src: 192.168.105.70,
> User Datagram Protocol, Src Port: 5001, Dst Port:
> Data (5 bytes)
```

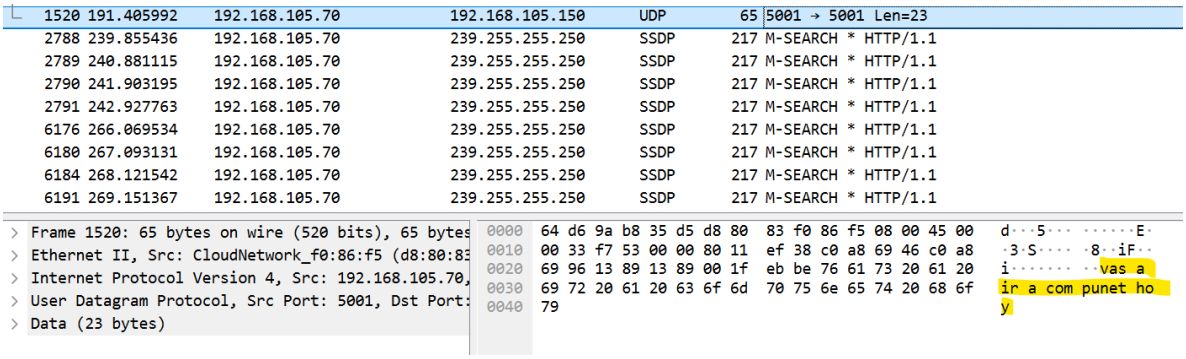
```
0000  64 d6 9a b8 35 d5 d8 80  83 f0 86 f5 08 00 45 00   d···5··· ·····E·
0010  00 21 f7 50 00 00 80 11  ef 4d c0 a8 69 46 c0 a8   ·!·P···· ·M··iF··
0020  69 96 13 89 13 89 00 0d  8f c3 68 6f 6c 61 20      i····· ··hola
```

```
   977 121.891417   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
   981 122.909422   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
  1022 131.515523   192.168.105.70          192.168.105.150      UDP        47 5001 → 5001 Len=5
  1050 146.058675   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
  1056 146.990554   192.168.105.70          239.255.255.250      SSDP      217 M-SEARCH * HTTP/1.1
  1106 159.778046   192.168.105.70          192.168.105.150      UDP        66 5001 → 5001 Len=24
  1520 191.405992   192.168.105.70          192.168.105.150      UDP        65 5001 → 5001 Len=23
```

```
> Frame 1022: 47 bytes on wire (376 bits), 47 bytes
> Ethernet II, Src: CloudNetwork_f0:86:f5 (d8:80:83
> Internet Protocol Version 4, Src: 192.168.105.70,
> User Datagram Protocol, Src Port: 5001, Dst Port:
> Data (5 bytes)
```

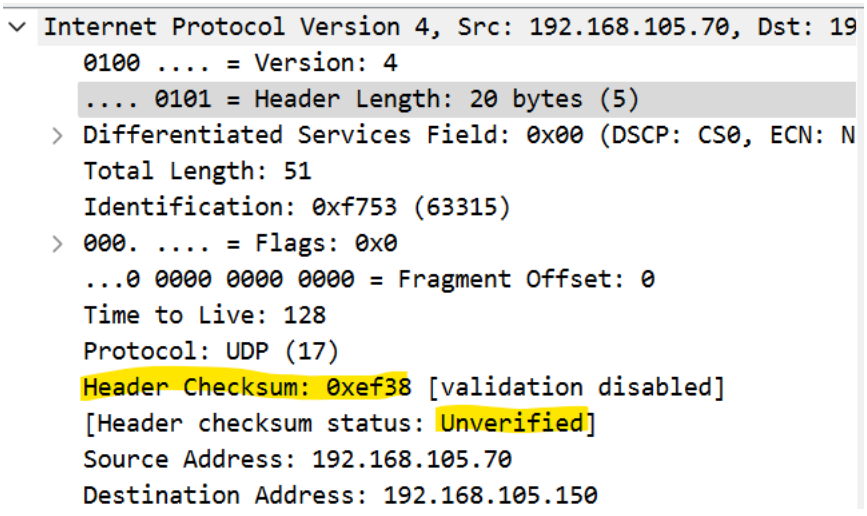
```
0000  64 d6 9a b8 35 d5 d8 80  83 f0 86 f5 08 00 45 00   d···5··· ·····E·
0010  00 21 f7 51 00 00 80 11  ef 4c c0 a8 69 46 c0 a8   ·!·Q···· ·L··iF··
0020  69 96 13 89 13 89 00 0d  8f c3 68 6f 6c 61 20      i····· ··hola
```





El checksum:

Del IP



Del UDP:

```
User Datagram Protocol, Src Port: 5001, Dst Port: 5001
    Source Port: 5001
    Destination Port: 5001
    Length: 31
    Checksum: 0xebbe [unverified]
    [Checksum Status: Unverified]
    [Stream index: 14]
  > [Timestamps]
    UDP payload (23 bytes)
```

Ambos sirven para la integridad de los archivos.

¿Qué patrones de diseño/arquitectura aplicaría al desarrollo de un programa basado en red como este?

Patrón Observer (Observador):

Facilita la actualización de la interfaz de usuario cuando se reciben nuevos mensajes de la red sin acoplar estrechamente la lógica de red con la presentación.

Patrón Singleton:

Útil para manejar conexiones de red compartidas, asegurando que solo exista una instancia de la conexión que gestione todas las operaciones de envío y recepción.

Patrón Factory (Fábrica):

Permite crear diferentes tipos de conexiones o mensajes (por ejemplo, TCP o UDP) sin cambiar el código que utiliza estas clases.

Arquitectura MVC (Modelo-Vista-Controlador):

Ayuda a mantener separadas la lógica de negocio y la presentación, facilitando el mantenimiento y la escalabilidad del código.

Patrón Strategy (Estrategia):

Permite cambiar dinámicamente el protocolo de comunicación o el método de manejo de datos sin modificar las clases que los utilizan.

Ricardo Andrés Chamorro Martínez A00399846 Computación en internet 1
Diego Armando Polanco Lozano A00399926

Investiguen que modificaciones son necesarias para implementar este mismo sistema, pero para la comunicación TCP en java.

TCP es un protocolo orientado a la conexión, lo que implica que debe establecerse una conexión estable entre el cliente y el servidor antes de que se pueda transferir cualquier dato. A diferencia de UDP, que es sin conexión, TCP asegura la transmisión confiable de los datos mediante confirmaciones, retransmisiones y control de flujo. Las modificaciones incluirían el uso de las clases ServerSocket y Socket para manejar las conexiones, y el uso de flujos de entrada y salida (InputStream y OutputStream) para enviar y recibir datos de manera continua. Además, sería necesario manejar el cierre adecuado de las conexiones una vez que la comunicación termine

¿Qué utilidades de codificación o seguridad agregaría al código?:

- Encriptación: Para la privacidad
- Sesiones: Que cierren automáticamente
- Limitar el numero de intentos fallidos