



**Instituto Politécnico Nacional**  
**Escuela Superior de Cómputo**



## **Proyecto Streaming**

**Aplicaciones para comunicaciones en red**

**García Vera Jared Alberto**  
**Muñoz Primero Elías**  
**Sánchez García Francisco Nicolas**

**Fecha: 17/Enero/2020**

# Planteamiento del proyecto

Lo que se planea construir es una aplicación que permite compartir videos (principalmente películas) con diferentes usuarios con el uso de las redes peer to peer. Donde tendremos un catálogo del cual podremos seleccionar la transmisión que más nos guste y dentro de cada transmisión podremos encontrar un chat para conversar con otros usuarios que estén conectados a la misma sala de transmisión. Adicionalmente, se tienen opciones para hacer un chat privado, para solo leer a tus conocidos en una transmisión pública; y también existe la posibilidad de crear una transmisión privada, donde solo podrán acceder las personas con quienes el usuario que transmita se les haya compartido el código de la transmisión.

Para la creación de este proyecto se deberán usar alguno de los protocolos que a continuación se presentan.

## RTP

Es un protocolo de transporte en tiempo real, que provee una comunicación punto a punto para información con características de tiempo real, como puede ser la transmisión de audio y video. Este protocolo se maneja por tipos de carga, numeración de secuencia, tiempos de envío y monitoreo de llegada. Al necesitar velocidad, hace uso del protocolo de transporte UDP, con el que podemos hacer uso de multicast para la distribución de datos a múltiples puntos de destino simultáneamente.

Por sí solo RTP no provee de métodos para garantizar la calidad o incluso la llegada de información a las diferentes conexiones. Es por esto que hace uso de otros protocolos de control como es RTCP y RTSP.

En el caso del presente proyecto, el equipo en cuestión decidió implementar el protocolo RTCP a través de la biblioteca VLC para java en su versión 2.1.0 para sistemas operativos windows 10 de 64 bits.

# RTCP

## Características

- Se encapsula sobre UDP (TCP no sirve para aplicaciones de tiempo real).
- Usa puertos de usuario para cada medio que se transfiere.
- Puede enviar tramas generadas por cualquier algoritmo de Codificación: H261, MPEG-1, MPEG-2.
- Puede usarse con direcciones de destino unicast o multicast.
- Identifica los orígenes del tráfico, lo que permite reencapsular agrupando tráfico a mitad de camino.
- Incorpora marcas de tiempo para cada medio:
- Para sincronización intra-flujo (eliminar jitter).
- Para sincronización intermedios (coincidencia audio/vídeo).
- Incluye números de secuencia para detectar pérdidas dentro de un flujo.

## Cabecera

- V (versión), 2 bits: los primeros dos bits identifican la versión del protocolo.
- P (padding), 1 bit: el siguiente bit identifica el padding. Informa que los datos de RTP llevan un "relleno" para completar un bloque de cierto tamaño. El último byte en el mensaje UDP dice de qué tamaño es el padding.
- X (extensión), 1 bit: indica si a continuación viene una cabecera de extensión.
- CC (CSRC count), 4 bits: número de identificadores CSRC que siguen a la cabecera fija.
- M (marker), 1 bit: está indicada para señalar elementos especiales como salirse de los límites.
- PT (payload type), 7 bits: formato de la información que se transporta para que lo interprete la aplicación.
- Número de secuencia, 16 bits: se incrementa en uno por cada paquete que se envía y sirve para que el receptor detecte pérdidas de paquetes.
- Timestamp, 32 bits: tiempo en el que se muestra el primer octeto de los datos transmitidos en el paquete.
- SSRC, 32 bits: identifica la fuente del paquete.
- CSRC, 32 bits: esta información es introducida por los mezcladores para indicar que han contribuido a modificar la información

0..1	2	3	4..7	8	9..15	16..31
V	P	X	CC	M	PT	Numero de Secuencia
Marca de Tiempo						
Identificador de fuente de sincronización						
Identificador de fuente de contribuyente						
...						

Figura 3. Encabezado del paquete RTP.

# Tecnologías Utilizadas

Para la implementación del presente proyecto, se utilizó la tecnología java con los siguientes complementos y bibliotecas:

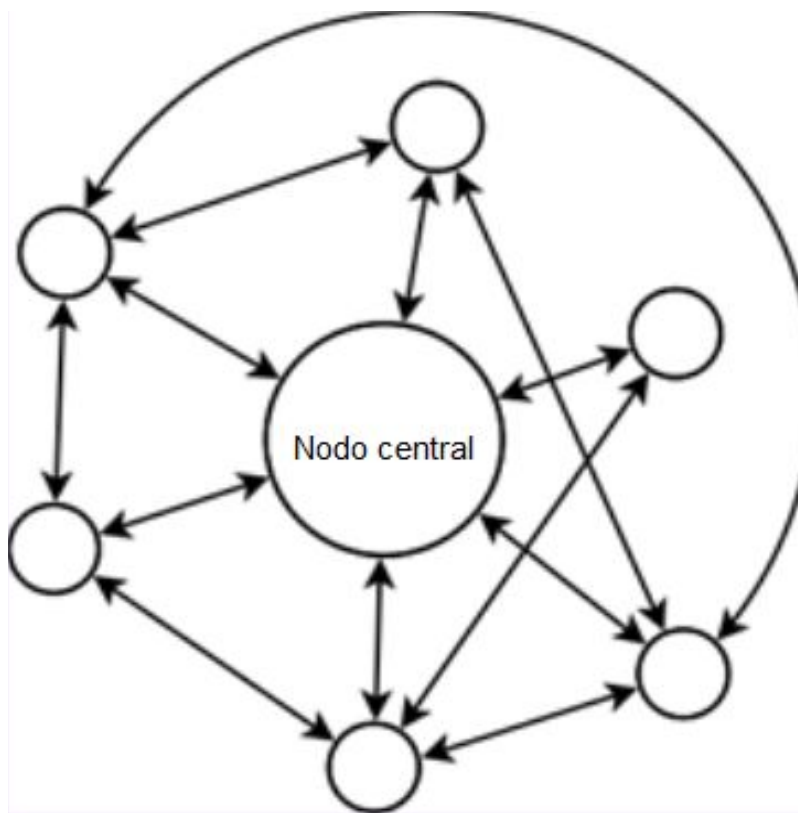
- Java Development Kit en su versión 14
- Apache Netbeans Versión 12
- Protocolo RTCP
- Sockets de Datagrama con el protocolo UDP (RTCP)
- Biblioteca de VLC para java en su versión 2.1.0 de 64 Bits
- Biblioteca JNA (complemento de VLC) para sincronizar el streaming

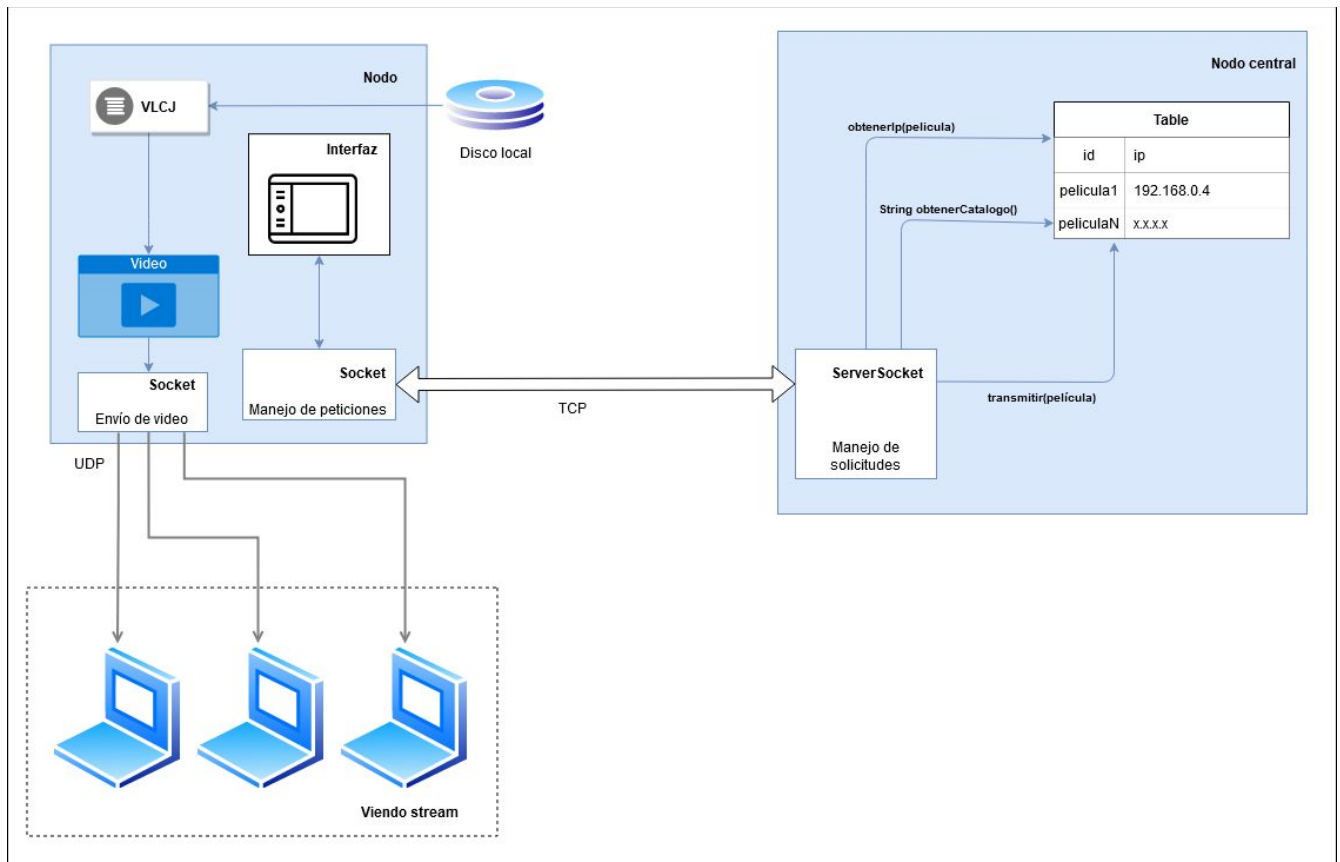


# Estructura Lógica

Características generales del proyecto:

- Será una aplicación de **escritorio**.
- Topología peer to peer híbrida:
- El nodo se encarga de **transmitir** la película.
- El servidor central/supernodo únicamente tiene información del usuario y de cada sala o transmisión. **No se suben los archivos de las películas.**
- El usuario sólo puede transmitir **una película a la vez**.
- El **chat no tiene restricciones**.
- La transmisión **no tiene controles de reproducción**





# Interfaz Gráfica

**Catalogo de Peliculas**

nico

Selecciona que pelicula quieres ver

Peliculas Disponibles:

🔍

Toy Story  
Toy Story 2  
Toy Story 3  
Toy Story 4  
Mi villano favorito 1  
Mi villano favorito 2  
Gool  
La momia  
Hercules  
Mulan  
Soul  
Grandes Heroes  
UP

¿Quieres transmitir una película?

Transmitir

**Inicio de sesion**

## Datos del nodo Central

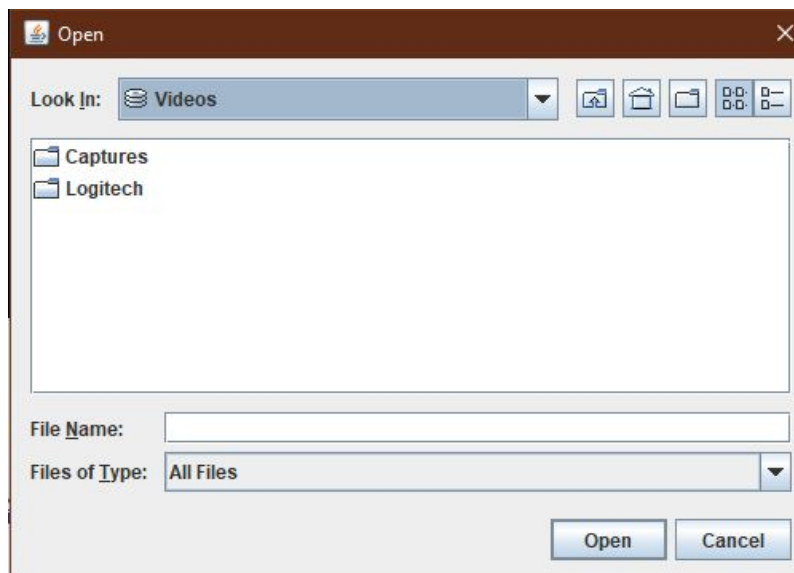
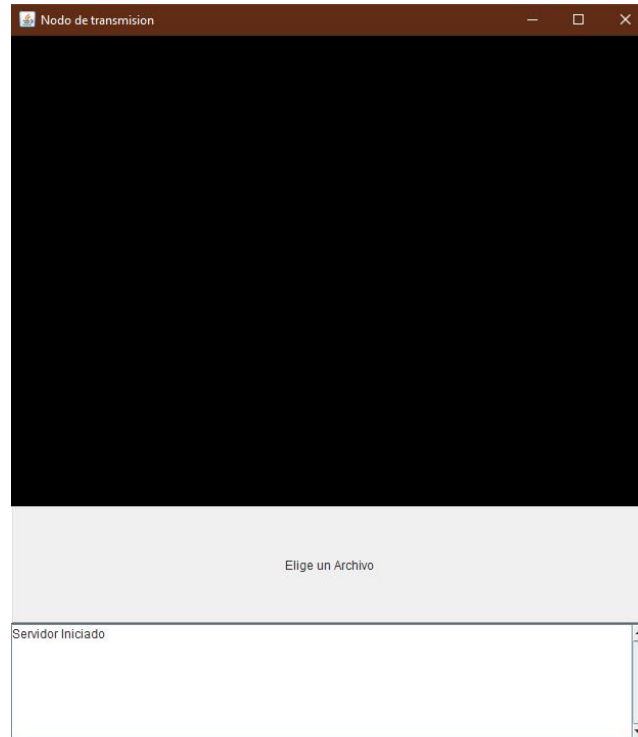
IP:

Puerto:

Username:

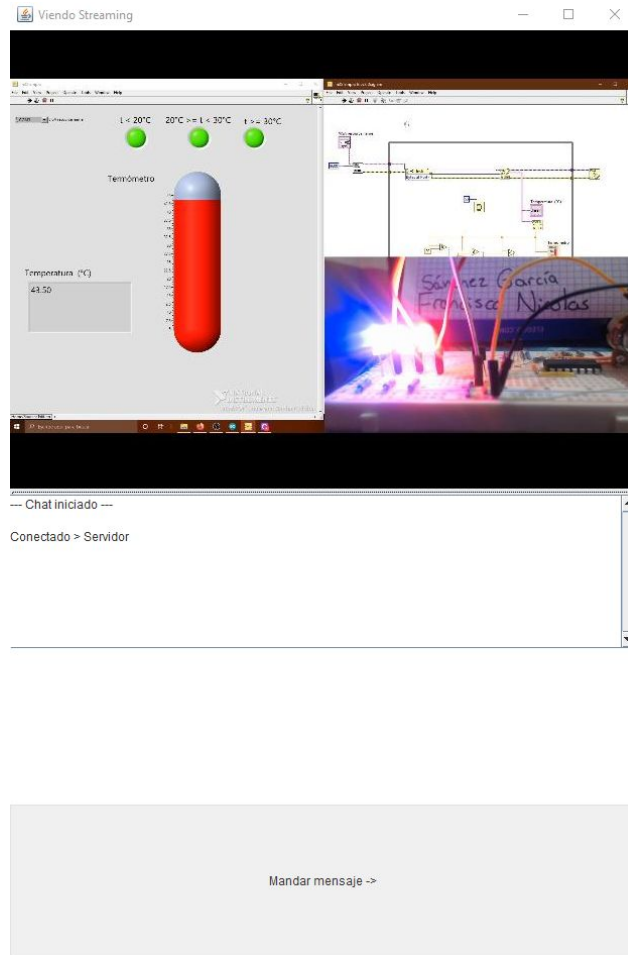
Entrar

## Nodo Que realiza el Streaming





## Nodo que se une a un streaming



## Bibliografía

- [https://en.wikipedia.org/wiki/Real\\_Time\\_Streaming\\_Protocol](https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol)
- [https://github.com/thodde/Video\\_Streaming](https://github.com/thodde/Video_Streaming)
- <https://github.com/mutaphore/RTSP-Client-Server>
- <https://soporte.qloudea.com/hc/es/articles/115003659065--Qu%C3%A9-es-TCP-IP-#:~:text=El%20conjunto%20de%20protocolos%20TCP,crea%20un%20flujo%20de%20datos>
- <http://www.cs.columbia.edu/~hgs/rtsp/draft/draft-ietf-mmusic-rtsp-03.html>
-