

UNIVERSIDAD POLITÉCNICA SALESIANA
COMPUTACIÓN
SISTEMAS DISTRIBUIDOS

Nombre: Jhonny Velasco

Asunto: Informe Taller U2 Streaming

OBJETIVO

Desarrollar un sistema distribuido cliente-servidor en Python que permita la reproducción remota de videos, aplicando los conceptos fundamentales de sistemas distribuidos, programación de redes y comunicación entre procesos.

Competencias Desarrolladas

- **Programación de Sockets:** Creación y gestión de conexiones de red
- **Arquitectura Cliente/Servidor:** Diseño de sistemas distribuidos
- **Manejo de Concurrencia:** Uso de hilos para múltiples clientes
- **Protocolos de Comunicación:** Diseño de protocolo simple para transferencia de datos
- **Integración de Sistemas:** Conexión entre red y reproducción multimedia

Desarrollo servidor

Librerías:

```
import socket
import os
import threading
```

Videos cargados localmente:

```
def __init__(self, host='localhost', port=7777):
    self.host = host
    self.port = port
    self.videos = {
        '1': 'video1.mp4',
        '2': 'video2.mp4',
        '3': 'video3.mp4',
        '4': 'video4.mp4'
    }
```

Método para reproducción de video:

```
def handle_client(self, connection, address):
    print(f"Cliente conectado: {address}")
    try:
        while True:
            connection.send(self.create_menu().encode())
            data = connection.recv(1024).decode().strip()
            if not data or data == '5':
                break
            if data in self.videos:
```

```

video = self.videos[data]
if os.path.exists(video):
    connection.send(f"PLAY:{video}".encode())
    print(f"Enviando: {video} a {address}")
else:
    connection.send(b"ERROR:Video no encontrado")
else:
    connection.send(b"ERROR:Opcion No valida")
finally:
    connection.close()

```

Desarrollo cliente

Librerías:

```

import socket
import os
import subprocess

```

Método para reproducción de video:

```

def play_video(self, video_path):
    try:
        print(f"Reproduciendo: {video_path}")
        subprocess.Popen(['xdg-open', video_path])
        return True
    except Exception as e:
        print(f"Error: {e}")
        return False

```

Ejecución

Primero tenemos que integrar el proyecto a un entorno virtual con el comando:

source venv/bin/activate

```

/EjemploSockets$ source venv/bin/activate
(venv) alumno@LN1-MAQ21:~/Documentos/sistemasDistribuidos
/EjemploSockets$

```

Y ejecutar ambos scripts

```

(venv) alumno@LN1-MAQ21:~/Documentos/sistemasDistribuidos
/EjemploSockets$ python3 servervideo.py
Servidor en localhost:7777
Cliente conectado: ('127.0.0.1', 35582)

```

Ejecución servidor

```
(venv) alumno@LN1-MAQ21:~/Documentos/sistemasDistribuidos/EjemploSockets$ python3 clientvideo.py
Conectado a localhost:7777

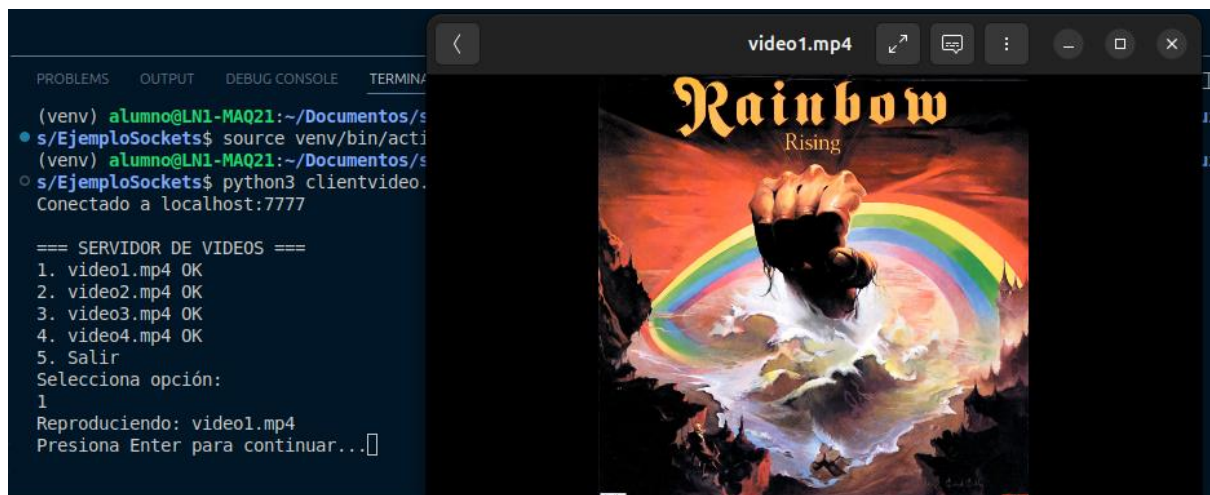
=== SERVIDOR DE VIDEOS ===
1. video1.mp4 OK
2. video2.mp4 OK
3. video3.mp4 OK
4. video4.mp4 OK
5. Salir
Selecciona opción:

```

Ejecución cliente

Resultados

Reproducción de cualquier video del menú



Conclusión:

El objetivo académico se cumplió integrando múltiples áreas del conocimiento en una aplicación funcional que sirve como demostración tangible de los principios de los sistemas distribuidos como Pyrot4 u otra librería que permita la reproducción remota de multimedia.