

## **UTILIZZO DEL PROGRAMMA**

Il seguente programma implementa un sistema simulato di streaming di musica. La comunicazione avviene tra un server e più client.

## **PROTOCOLLO**

Il server quando viene avviato va in ascolto sulla porta 12345.

Il client interagisce con il server tramite i seguenti comandi: LIST, PLAY, STOP, RESUME, EXIT

### **LIST**

CLIENT: invia il comando LIST.

SERVER: restituisce al client una lista contenente le canzoni disponibili da scaricare dal server, indicizzate numericamente.

### **PLAY <n>**

CLIENT: invia il comando PLAY <n>, dove n corrisponde all'indice della canzone da riprodurre. Attende l'invio del file musicale scelto che memorizzerà poi in una directory creata all'avvio del client.

SERVER: trasmette al client il file corrispondente alla canzone scelta e simula la riproduzione di questa.

Dopo aver ottenuto l'indice della canzone scelta estraendolo dal comando in arrivo, verifica prima se il comando è valido, poi invia il file al client e aggiorna lo stato di riproduzione.

### **STOP**

CLIENT: invia il comando STOP.

SERVER: Interrompe la riproduzione della canzone.

Verifica se il server sta riproducendo una canzone; se positivo, interrompe la riproduzione.

## **RESUME**

CLIENT: invia il comando RESUME.

SERVER: riprende la riproduzione della canzone.

Verifica che il brano non sia attualmente in riproduzione; se positivo, aggiorna lo stato del brano e riprende la riproduzione.

## **EXIT**

CLIENT: invia il comando EXIT.

SERVER: interrompe la connessione con il client.

### **MusicClient:**

La classe MusicClient gestisce i seguenti eventi:

- Connessione al server e gestione dell'interazione con il server
- Gestione del download delle canzoni dal server (creazione di una cartella dedicata a memorizzare le canzoni, ricezione del file)
- Ricezione delle risposte dal server

### **MusicServer**

La classe MusicServer gestisce i seguenti eventi:

- Ricerca della directory dove sono contenute le canzoni da trasmettere
- Memorizzazione delle informazioni inerenti ai client connessi (creazione di una lista, creazione di handler per i client, creazione di thread separati per gestire i client, inizializzazione del socket dei client)
- Esecuzione dei comandi inviati dai client (LIST, PLAY <n>, STOP, RESUME, EXIT)
- Trasmissione dei file contenenti le canzoni richieste dai client

<https://github.com/tommyspano/songlist.git>