

# Relatório Redes 2 - Entrega 2 de Trabalho

**Aluno: Bruno Vasconcelos Xavier**

## 1. Introdução

Este relatório detalha a implementação de um sistema de streaming de música baseado no modelo Peer-to-Peer (P2P), utilizando sockets para comunicação entre os nós. O objetivo do sistema é permitir que dispositivos (nós) atuem como clientes e servidores simultaneamente, compartilhando e requisitando arquivos de áudio de maneira eficiente.

## 2. Objetivo

O objetivo principal foi criar uma aplicação que possibilite:

- Registro de nós em um servidor central.
- Consulta de nós conectados e arquivos disponíveis.
- Transferência de arquivos de áudio diretamente entre os nós usando UDP, permitindo reprodução contínua durante o download.

## 3. Arquitetura do Sistema

O sistema é composto por dois principais componentes:

- **Servidor Central:**
  - Registra e gerencia a lista de nós ativos e seus arquivos.
  - Fornece informações sobre nós e arquivos disponíveis para consulta.
  - Remove nós quando solicitada a desconexão.
- **Nós P2P:**
  - Atuam como clientes ao se registrar e consultar o servidor.
  - Funcionam como servidores ao atender requisições de outros nós, transmitindo arquivos via UDP.

## 4. Tecnologias e Protocolos Utilizados

- **Linguagem:** Python, devido à sua simplicidade e bibliotecas robustas.
- **TCP:** Utilizado para comunicação confiável entre os nós e o servidor central.
- **UDP:** Implementado para transferência de arquivos, otimizando o streaming contínuo.
- **Multithreading:** Permite que múltiplas conexões sejam processadas simultaneamente, tanto no servidor central quanto nos nós P2P.

## 5. Funcionalidades do Sistema

## 5.1 Servidor Central

1. **Registro de Nós:**
  - Armazena endereço IP, porta e lista de arquivos compartilhados por cada nó.
  - Retorna mensagens de confirmação ou notifica se o nó já está registrado.
2. **Consulta:**
  - Envia a lista de nós conectados e arquivos disponíveis ao cliente que requisitar.
3. **Remoção de Nós:**
  - Remove o registro de nós quando solicitado, liberando espaço para novas conexões.

## 5.2 Nós P2P

1. **Registro no Servidor:**
  - Envia endereço IP, porta e lista de arquivos ao servidor central.
2. **Consulta de Arquivos:**
  - Obtém a lista de nós ativos e seus arquivos compartilhados do servidor central.
3. **Transferência de Arquivos via UDP:**
  - Atende requisições de outros nós, transmitindo o arquivo solicitado em pacotes UDP.
  - Inicia a reprodução do arquivo no cliente assim que os pacotes começam a ser recebidos.
4. **Notificação de Erros:**
  - Informa ao cliente quando um arquivo requisitado não está disponível.

## 6. Casos de Uso Detalhados

1. **Registro de Nó:**
  - **Cenário:** Um nó deseja ingressar no sistema.
  - **Processo:** O nó envia seu endereço e lista de arquivos ao servidor central.
  - **Resultado Esperado:** Registro bem-sucedido e confirmação enviada ao nó.
2. **Consulta de Arquivos:**
  - **Cenário:** Um nó deseja acessar a lista de arquivos disponíveis.
  - **Processo:** O nó envia uma solicitação ao servidor central e recebe a lista atualizada.
  - **Resultado Esperado:** Lista recebida e exibida no nó solicitante.
3. **Requisição de Arquivo:**
  - **Cenário:** Um nó deseja obter um arquivo específico de outro nó.
  - **Processo:** O nó solicitante se conecta diretamente ao nó com o arquivo e inicia a transferência via UDP.
  - **Resultado Esperado:** Transferência bem-sucedida com reprodução contínua.
4. **Encerramento de Conexão:**
  - **Cenário:** Um nó deseja sair do sistema.
  - **Processo:** O nó envia uma solicitação de desconexão ao servidor central.
  - **Resultado Esperado:** Nó removido com sucesso e confirmação enviada.

## 7. Testes Realizados

### 7.1 Registro de Cliente

- **Objetivo:** Verificar o registro de novos nós.
- **Resultado:** Todos os registros foram bem-sucedidos, com confirmações enviadas corretamente.

### 7.2 Consulta de Arquivos

- **Objetivo:** Testar a obtenção da lista de nós e arquivos do servidor central.
- **Resultado:** Listas retornadas corretamente conforme o estado atual do servidor.

### 7.3 Transferência de Arquivo

- **Objetivo:** Validar a transferência de arquivos entre nós via UDP.
- **Resultado:** Arquivos transferidos com sucesso e reprodução iniciada automaticamente.

### 7.4 Notificação de Erro

- **Objetivo:** Testar o comportamento ao requisitar arquivos inexistentes.
- **Resultado:** Mensagens de erro foram enviadas corretamente ao cliente solicitante.

## 8. Atualizações para entrega 2

- Adição do suporte ao protocolo UDP:
  - Implementação da funcionalidade de envio de arquivos usando UDP nos nós P2P para streaming contínuo durante a transferência de dados.
- Modificação no comportamento do nó cliente:
  - Criação de lógica para permitir que o cliente requisitasse arquivos informando a porta de recebimento.
  - Ajustes no cliente para iniciar a reprodução dos dados assim que começam a chegar.
- Tratamento de erros:
  - Inserção de uma funcionalidade no servidor para verificar a disponibilidade de arquivos requisitados e retornar mensagens adequadas caso o arquivo não esteja presente.
- Gerenciamento de desconexões:

- Adição de lógica para que os nós clientes possam solicitar a remoção segura do registro no servidor central.
  
- Integração das novas funcionalidades no fluxo geral:
  - Atualização de mensagens e fluxos de entrada/saída para contemplar a interação completa entre registro, consulta e transferência.

## **9. Conclusão**

O sistema P2P para streaming de música foi implementado com sucesso, atendendo aos requisitos propostos. Ele permite registro, consulta e transferência de arquivos de forma eficiente e escalável. Melhorias futuras podem ampliar a segurança e usabilidade, tornando-o mais robusto e acessível para um público maior.