



**UNIVERSIDADE FEDERAL DE OURO PRETO**  
Departamento de Computação e Sistemas – DECSI  
Prof. Filipe Ribeiro

**RELATÓRIO ATIVIDADE PRÁTICA DE PROGRAMAÇÃO**  
Redes de Computadores I – Turma: Sistemas de Informação

Alunos: Arthur Mendonça Feu, Marco Túlio Madeira e Daniel Ângelo Moraes

**João Monlevade**  
**2023**

## INTRODUÇÃO:

Como parte da ementa da disciplina de Redes de Computadores, foi solicitado a realização de uma atividade prática com foco na utilização de Sockets e Threads. O arquivo TXT com as orientações do professor está presente na mesma pasta Zip que contém este relatório.

## DIVISÃO DE TAREFAS

O grupo apresentado optou por dividir as tarefas de forma mista. Cada aluno contribuiu ativamente na implementação dos códigos. O aluno Daniel Ângelo trabalhou no desenvolvimento de uma interface para executar os programas em Python, inicialmente em parceria com Marco Túlio e posteriormente com Arthur Feu. Ele também auxiliou nos códigos de implementação, fornecendo instruções, dicas e suporte para debug. Os outros dois alunos, Marco e Arthur, foram responsáveis pela implementação básica dos protocolos TCP e UDP, respectivamente.

A construção do relatório e arquivos de documentação foi uma tarefa realizada em conjunto por todos os membros da equipe. Além disso, foi utilizado um repositório Git para facilitar a colaboração e a dinâmica de trabalho da equipe, disponível em: ([https://github.com/marco-madeira/REDES\\_SOCKET](https://github.com/marco-madeira/REDES_SOCKET)).

## EXECUÇÃO DA PRÁTICA

Para realizar a implementação da prática, foram criados inicialmente 4 arquivos básicos, que posteriormente foram aprimorados para a contemplação dos pontos extras. Isso incluiu a implementação de uma interface gráfica com botões para selecionar o arquivo a ser enviado pelo protocolo e a funcionalidade de abrir o arquivo no cliente. Ao final, obtivemos um arquivo *cliente*, um arquivo *servidor* e o *transfer\_data*, que é responsável por implementar as atividades dos protocolos. Essa estrutura foi replicada para cada um dos protocolos, resultando em um total de 6 arquivos.

## PROTOCOLO DE CONTROLE DE TRANSMISSÃO (TCP)

O protocolo TCP é um dos mais confiáveis e amplamente utilizado para a transferência de arquivos pela internet, devido a sua capacidade de garantir a entrega de dados dentro do pacote. O protocolo, implementa mecanismo de retransmissão automática de pacotes perdidos. Se um pacote não for recebido corretamente, o TCP solicitará a retransmissão desse pacote específico, garantindo que todos os dados sejam entregues corretamente, mesmo em condições de rede instáveis.

Uma das dificuldades enfrentadas pelo grupo na implementação do TCP, foi o tamanho do buffer, que às vezes não suportava enviar todo arquivo. Para solucionar esse problema foi criado uma condição para aumentar esse buffer, dependendo do arquivo que era solicitado.

## PROTOCOLO DE DATAGRAMAS DO USUÁRIO (UDP)

A transferência de arquivos utilizando o protocolo UDP apresenta limitações significativas devido à falta de garantias de entrega confiável de dados. Dados enviados por UDP podem ser perdidos, chegar fora de ordem, ou corrompidos, o que compromete a integridade e a precisão dos arquivos transferidos. No UDP, os dados são divididos em datagramas e transmitidos sem estabelecer uma conexão prévia entre as partes envolvidas. Isso resulta em uma transmissão mais rápida e eficiente em comparação com o TCP, pois não há sobrecarga de controle de conexão.

Na implementação do UDP, uma das dificuldades enfrentadas foi o travamento da interface do usuário durante a operação. Para contornar esse problema, o grupo optou pelo uso de threads, permitindo que a transferência de dados ocorresse em segundo plano, enquanto a interface permanecia responsiva. Além disso, outro desafio enfrentado foi a perda de pacotes, mencionada anteriormente. A estratégia adotada consistiu em implementar um mecanismo de confirmação de recebimento: Após receber um pacote, o receptor enviava uma confirmação ao remetente. Se o remetente não recebesse a confirmação dentro de um tempo limite, ele reenviaria o pacote. Isso garantia que o remetente tivesse conhecimento da entrega bem-sucedida de cada pacote.

## COMPARATIVO e CONCLUSÃO

Levando em consideração a atividade prática realizada, é possível atribuir diferenças consideráveis entre os dois protocolos citados. Além disso, utilizaremos os arquivos base fornecidos pelo professor (*small*, *medium*, *large*) para montar uma pequena base de dados comparativa:

Tempo de execução e transferência em milissegundos por arquivo

Protocolo	Small (1001 ln)	Medium (10001 ln)	Large (100001 ln)
TCP	0.07	0.24	2.02
UDP	0.04	0.25	1.88

Em termos apenas de velocidade de transmissão, o protocolo TCP é superior ao protocolo UDP. Essa diferença se dá pela quantidade de processos realizados na implementação de cada um e fica mais significativa apenas para o maior arquivo.

Contudo, é interessante também avaliar as diferenças que vão além da velocidade de transferência dos dados, como a confiabilidade de cada um e o controle de fluxo. Como explicado anteriormente, o protocolo TCP é mais confiável e possui mais processos para a transferência de pacotes. Além disso, a ausência de controle de fluxo no UDP pode levar a problemas de sobrecarga de rede, principalmente em redes com qualidade variável. Já o TCP

ajusta automaticamente a taxa de transferência de dados com base na capacidade da rede, evitando sobrecargas.

Em última análise, nosso grupo chegou à conclusão de que o protocolo TCP é melhor avaliado para situações que exigem maior segurança e confiabilidade na transferência de pacotes, enquanto por outro lado, o protocolo UDP será fortemente recomendado para situações que exigem velocidade e não levam em consideração a confiabilidade no ato da transferência de dados.