

# Projeto de Chat Room

Fundamentos de Redes de Computadores

# Integrantes do Grupo

- Bernardo Chaves Pissutti - 190103302
- Matheus Soares Arruda - 190093480
- João Victor Correia de Oliveira - 190089792
- Paulo Henrique de Oliveira Resende - 190094257

# Introdução

Esse trabalho mostra o projeto ChatRoom de FRC, que permite mensagens em tempo real com conexões e conversas simultâneas. Usando arquitetura TCP/IP e sockets de rede, a comunicação é eficiente e segura.

Explicaremos a metodologia da equipe, organização e roteiros do projeto. Descreveremos suas características e funcionalidades. Concluiremos com resultados positivos, desafios e melhorias futuras.

# Metodologia

## **Encontros realizados:**

Em 20 de junho, definimos escopo e requisitos da aplicação de chat. Cada membro da equipe recebeu uma tarefa específica no GitHub.

Em 21 de junho, revisamos o progresso de cada membro, compartilhamos dificuldades e soluções e discutimos melhorias para o código. Realizamos uma integração parcial das implementações.

Em 23 de junho, na 3ª reunião, resolvemos problemas da fase de integração, identificando conflitos no código e definindo a estrutura do relatório final.

# Metodologia

## **Encontros realizados:**

Em 24 de junho, revisamos o projeto, finalizamos as implementações e testamos o chat. Discutimos os resultados.

Em 29 de junho, resumimos as informações e os resultados em um relatório final. Cada membro contribuiu com suas próprias experiências e conclusões sobre o projeto. Revisamos o documento em conjunto para garantir a coerência das informações.

Usamos o GitHub para trabalho em equipe. Cada membro teve sua própria branch. Após avaliação e escolha da melhor, usamos para implementações e relatório final.

# Solução

A solução encontrada pelo grupo para implementar corretamente o trabalho, foi criar uma matriz com dimensões de `qtdSala` x `qtdUsuarios`. Dessa maneira, quando um cliente se conecta ao servidor, o usuário insere seu nome e a sala à qual deseja se conectar, e envia essas informações ao servidor. Ao receber esses dados, o servidor os armazena em dois vetores: um vetor para as salas e outro vetor para as informações dos usuários. Onde cada índice do vetor das salas contém um vetor com as informações dos usuários conectados à essa sala, formando assim uma matriz `qtdSala` x `qtdUsuarios`. Esses vetores têm tamanho equivalente aos limites e usuários por sala permitidos pela aplicação.

# Conclusão

Neste trabalho construímos uma aplicação de chat room, na qual cada membro do grupo contribuiu com uma funcionalidade na sua própria branch no github. Após avaliar os progressos individuais, integramos os resultados na branch main.

A solução encontrada pelo grupo foi um armazenamento dos dados das salas e usuários em uma matriz, assim as mensagens dos usuários poderiam ser encaminhadas somente aos usuários na mesma sala.

Em suma, esse projeto trouxe a oportunidade de aprendermos na prática os fundamentos da implementação de uma arquitetura cliente-servidor.

# Conclusão

Neste trabalho construímos uma aplicação de chat room, na qual cada membro do grupo contribuiu com uma funcionalidade na sua própria branch no github. Após avaliar os progressos individuais, integramos os resultados na branch main.

A solução encontrada pelo grupo foi um armazenamento dos dados das salas e usuários em uma matriz, assim as mensagens dos usuários poderiam ser encaminhadas somente aos usuários na mesma sala.

Em suma, esse projeto trouxe a oportunidade de aprendermos na prática os fundamentos da implementação de uma arquitetura cliente-servidor.



# Referência

Linux Man Pages. Select(2) - Linux manual page. Disponível em:  
<https://man7.org/linux/man-pages/man2/select.2.html>. Acesso em: 20 de junho.

Comunicação entre processos. PUCRS. Disponível em:  
<https://www.inf.pucrs.br/~gustavo/disciplinas/ppd/material/aula-sockets.pdf>.  
Acesso em 29 de junho.