UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO

Arthur Ramos Lima
Mateus da Silva Batista
Ullyanne Julia Freire Patriota

Relatório do Projeto de Redes de Computadores 2021.2

Maceió - AL
2022

1 Projeto

O projeto consiste na implementação via sockets de um bate-papo no terminal. Os usuários mandam e recebem mensagens no chat geral. Foi utilizado conceitos de sockets, protocolo de transporte e outros assuntos referentes à disciplina. A aplicação é inteiramente programada em Python 3.

1.1 Como executar

É necessário clonar o repositório do projeto utilizando o comando git clone, acessar o diretório src, executar server.py para atuar como servidor e client.py como cliente em diferentes instâncias do terminal.

A seguir, os comandos:

```
$ git clone https://github.com/Arthur-r19/Projeto-de-Redes.git
$ cd Projeto-de-Redes/src/
```

Servidor

```
$ python3 server.py
```

Clientes

```
$ python3 client.py
```

2 Funcionalidades

2.1 Envio e recebimento de mensagem

A principal utilidade da aplicação é a troca de mensagem pelos usuários que é realizada via terminal.

2.2 Comandos

Envie "!HELP"para receber uma mensagem com todos os comandos disponíveis.
Envie "!CLOSE"para sair do bate-papo.
Envie "!CLEAR"para limpar o bate-papo.

2.3 Timestamp

Todas as mensagens enviadas no bate-papo possuem registro de hora de envio.

3 Implementação

A aplicação foi desenvolvida na linguagem de programação Python 3 utilizando as bibliotecas threading, socket, random, entre outras.

Foram implementadas as funções start e handle_client, ambas presentes no arquivo server.py. A função start serve para escutar e aceitar as conexões dos usuários, e fica rodando em loop em um processo distinto. Após a aprovação da conexão, o cliente é gerenciado pela função handle_client, que é executada em outra thread com o intuito de permitir a conexão de múltiplos clientes.

No arquivo client.py as principais funções são start, connect, listen_messages e send. A função start é utilizada para conectar-se ao servidor e manter essa conexão até o usuário solicitar saída. A função listen_messages escuta mensagens digitadas no terminal do cliente e as envia formatadas para todos os clientes conectados através da função send quando é pressionado ENTER.

4 Dificuldades

O principal bug encontrado em nossa aplicação se dá quando múltiplos usuários estão enviando mensagens: quando um usuário envia mensagem, tudo que os demais usuários digitaram no terminal, mas não enviaram, é ocultado da tela e salvo em um buffer. Para enviar essa mensagem, basta dar enter, porém como o usuário não consegue visualizá-la, não é intuitivo.

A principal dificuldade foi limpar o terminal após enviar a mensagem, quando se digita no terminal, a mensagem enviada continuava na tela o que era um problema, pois a mensagem ficava duplicada para remetente.

Além disso, podemos adicionar a formatação de texto no terminal como mais uma dificuldade que tivemos na implementação.

5 Futuras implementações

Visando aprimorar nosso programa e torná-lo mais amigável, pensamos em funcionalidades como a criação de bate-papos privados, comando *!ONLINE* para mostrar os clientes conectados no momento, salas para grupos, vote-kick, melhoria na interface, envio de emojis através de comandos, implementação de um sistema de log, expulsão de usuários inativos, punições para spam, viabilização de conexão em computadores diferentes, exportação da conversa em arquivo .txt e controle de spam.

Pontua-se que é interessante criar um controle de usuários para limitar a quantidade de pessoas no bate-papo. Quando se conecta ao servidor, é necessário um processo para lidar com o sistema final, o que pode ser um problema se houver muitas requisições simultâneas.