

Universidade de Brasília



FUNDAMENTOS DE REDES DE COMPUTADORES - 2021/2

Criando ambientes virtuais de conversação com uso de
sockets

Grupo

Nome	Matrícula
Denniel William Roriz Lima	17/0161871
Edvan Barreira Gomes Junior	17/0032591
Philippe Rosa Serafim	16/0141842

Professor

Fernando William Cruz

Sumário

1.	Introdução	3
2.	Objetivos	3
3.	Metodologias	3
3.1.	Metodologia organizacional	3
3.2.	Encontros	4
4.	Descrição da solução	4
5.	Execução do projeto	5
6.	Conclusão	6
7.	Referências bibliográficas	7

1. Introdução

O TCP/IP é um conjunto de protocolos de envio e recebimento de pacotes de dados entre dispositivos estabelecendo uma conexão entre um ponto de origem e um ponto de destino. Os dados são transmitidos em forma de pacote pois dessa forma, caso ocorra algum problema durante o trajeto, evita-se o reenvio da mensagem inteira. Esses pacotes de dados devem passar por quatro camadas antes de chegar no destino: camada de datalink, camada de internet, camada de transporte e camada de aplicação.

Criado pelo Departamento de Defesa dos Estados Unidos, o TCP/IP tem o objetivo de facilitar a transmissão de dados entre dispositivos de maneira segura e precisa. Outra vantagem dessa arquitetura é sua alta velocidade, permitindo sua aplicação para a solução de problemas do cotidiano.

Uma aplicação para o protocolo TCP/IP é a criação de uma rede que permita a comunicação de computadores. Uma vez que essa rede é criada, ela pode ser usada por usuários para trocar vários tipos de dados, como por exemplo, mensagens de texto, áudio e vídeo.

2. Objetivos

Desenvolver uma aplicação de rede, usando protocolo TCP/IP que envolvam gerência de diálogo. A aplicação deve permitir a criação de salas de bate papo virtuais com nome e limite máximo de participantes. Além disso, os participantes podem entrar nas salas usando identificadores, trocar mensagens com os demais membros e sair para liberar espaço para outros usuários.

3. Metodologia

3.1. Metodologia organizacional

Para o desenvolvimento da aplicação, foram organizadas três etapas:

- **Planejamento:** nesta etapa, os requisitos da aplicação são definidos;
- **Desenvolvimento:** uma vez que as atividades foram distribuídas, a equipe começou a etapa de desenvolvimento, que consiste na produção da aplicação. Nesta etapa foi aplicado a técnica do pair programming.
- **Montagem do relatório e slides:** a terceira e última etapa consiste na montagem do relatório, descrevendo o projeto e as etapas concluídas, e a elaboração do slide que será utilizado na apresentação.



3.2. Encontros

Todos os encontros foram realizados de forma remota com o auxílio da ferramenta [Discord](#). Ao total, foram 7 encontros, que foram divididos entre ‘teóricos’ e ‘práticos’. Nos encontros teóricos, realizados nos dias 22, 23 e 24 de abril, no primeiro dia foi decidida a tecnologia e metodologia que seria utilizada, enquanto os outros dois dias foram utilizadas para pesquisa de bibliografia e referências relacionadas ao tema, ao conteúdo e a especificação do projeto.

Os dias 25, 27 e 28 de abril foram utilizados para codificação da maior parte do projeto, codificação essa realizada através de ‘pareamentos triplos’, onde todos os integrantes estavam presentes, havendo um piloto e dois co-pilotos. Dessa forma foi possível contornar a falta de familiaridade do time com a biblioteca e com o tema proposto.

O último dia, 03 de maio, foi utilizado para confecção do relatório, do slide e pequenas melhorias no código do projeto.

4. Descrição da solução

Conforme especificado, o projeto foi realizado com python e algumas bibliotecas que serão abordadas adiante no relatório. A pasta do projeto é composta por seis arquivos python e o README.md, que contém as instruções para a execução da aplicação. Dos arquivos, tem-se:

- app.py: Parte responsável por apresentar o menu e todas as funções chamadas no menu (listar sala, criar sala, entrar em uma sala). Nela foram utilizados as bibliotecas *os* e *threading*, a primeira utilizada para realizar funções relacionadas ao sistema operacional e a segunda utilizada para gerar as threads utilizadas no projeto.
- client.py: Onde é especificado os atributos e métodos da classe Client, bem como a parte responsável por realizar a conexão tcp, enviar mensagens na sala e receber mensagens. Além das bibliotecas já mencionadas, também foram utilizadas a *socket* e a *sys*. Socket para disponibilizar acesso às interfaces utilizadas na conexão tcp.
- main.py: Realiza a configuração das portas para inicialização do servidor, do cliente e as portas utilizadas. A biblioteca random foi utilizada para que a escolha das portas utilizadas pelos clientes ocorra de forma menos enviesada.
- room.py: Definição da classe room e dos métodos referentes à sala ativa. Nesse arquivo é possível observar o bind responsável pela conexão da sala ao servidor, bem como todos os métodos responsáveis por realizar a criação, exclusão e controle de acesso à sala.
- server.py: Além da instanciação da sala, é responsável pelo controle das mensagens recebidas no servidor, bem como verificar os apelidos dos integrantes e realizar a criação do host para conexões.

- user.py: Definição da classe de usuário, contendo seus respectivos atributos.

5. Execução do projeto

Para executar a aplicação, é necessário clonar o repositório e acessar o repositório. O repositório do projeto pode ser acessado clicando [aqui](#).

Uma vez dentro do diretório, é possível executar os comandos (Tabela 1) para levantar o servidor e executar a aplicação, permitindo a criação das salas de bate papo e acessos dos usuários (Imagem 1). É importante ressaltar que a aplicação só pode ser iniciada uma vez que o servidor já tenha sido iniciado. Os requisitos necessários para executar a aplicação podem ser encontrados no arquivo README.md.

Com a execução da aplicação, o usuário pode tanto criar uma sala, definindo para ela um nome e capacidade máxima de participantes, quanto acessar uma sala já criada, informando um apelido (Imagem 2). Uma vez acessada uma sala, um usuário pode enviar e receber mensagens ou deixar a sala.

Comando	Função
python server.py	Inicia o servidor. Deve ser executado antes de iniciar a aplicação.
python main.py	Executa a aplicação. Deve ser executado com o servidor ativo.

Tabela 1: Comandos para execução do projeto

```
eddie@Edvan:~/FRC-2021.2$ python3 main.py
1 - Criar uma sala
2 - Entrar em uma sala
0 - Sair do programa
Selecione uma opção: 0
eddie@Edvan:~/FRC-2021.2$
```

Imagem 1: Menu inicial da aplicação

```
eddie@Edvan:~/FRC-2021.2$ python3 main.py
1 - Criar uma sala
2 - Entrar em uma sala
0 - Sair do programa
Selecione uma opção: 1
Escreva o nome da sua sala de bate papo: Fundamento de Redes
Qual vai ser o limite máximo de participantes? 7
Para entrar no bate papo deve primeiro digitar seu apelido:
Aluno 1
Bem vindo ao bate papo Fundamento de Redes!
Aluno 2 entrou na sala!
<Aluno 2>: Bom dia!
Bom dia!
<Aluno 2>: Está gostando de cursar esta disciplina?
Sim, mas preciso estudar muito!
<Aluno 2>: É verdade!
<Aluno 2>: Até mais
Aluno 2 saiu do bate papo!
```

Imagem2: Execução da aplicação

6. Conclusão

Algumas dificuldades foram encontradas durante o desenvolvimento do projeto, sendo que o primeiro deles foi descobrir uma forma de criar e manter as salas de bate papo ativas.

Outro problema enfrentado pela equipe foi encontrar uma maneira de permitir diálogos com áudio e vídeo na aplicação. Devido ao alto nível de complexidade combinado com a falta de experiência sobre o assunto, tal funcionalidade não pôde ser implementada.

A aplicação desenvolvida cumpre os requisitos de criar salas de bate papo, listar salas ativas, permitir o ingresso de participantes em salas criadas, permitir a troca de mensagens entre membros de uma mesma sala e também permitir que um usuário deixe uma sala.



7. Referências bibliográficas

Simple Chat Room using Python. Disponível em:

<https://www.geeksforgeeks.org/simple-chat-room-using-python/>. Acesso dia 23 de abril de 2022.

socket - Low-level networking interface. Disponível em:

<https://docs.python.org/3/library/socket.html>. Acesso em: 22 de abril de 2022.

JUNGES, Renata. Bate-Papo. Disponível em:

<https://github.com/rejunes/Bate-Papo>. Acesso dia 23 de abril de 2022.

JENNINGS, Nathan. Disponível em: <https://realpython.com/python-sockets/>. Acesso dia 24 de abril de 2022.