

Trabalho Chat

15/0081197- Myllena de Almeida Prado

¹Dep. Ciência da Computação – Universidade de Brasília (UnB)
Transmissão de Dados

{myllena.almeidap}@gmail.com

Resumo. *No trabalho foi desenvolvido um sistema de chat que possui um servidor e clientes. O desenvolvimento consistiu na implementação de dois códigos diferentes: um para o servidor e outro para o cliente, e a comunicação entre eles utiliza TCP.*

necessário que a mensagem chegue até os usuários para se ter uma conversa e é preciso que todo o conteúdo também chegue para ter sentido nessa comunicação entre participantes do chat.

1. Introdução

Com o desenvolvimento da internet e a necessidade de comunicação entre os seres humanos os chats adentraram o dia a dia das pessoas cada vez mais. O objetivo de seu uso pode ser para uma venda entre uma empresa e um cliente, ou para uma video conferência de uma empresa que possui integrantes em vários lugares do planeta. Além disso, o seu uso mais comum hoje é a conversa entre pessoas comuns, substituindo telefonemas.

O funcionamento dos chats consiste em um servidor que armazena informações sobre conversas, usuários e entre outras, e clientes que contactam o servidor para conversar com outros usuários. Nesse trabalho foi desenvolvido um servidor que possui as características de armazenar informações do usuário e permitir que os participantes do chat conversem entre si, e também foi desenvolvido o código do cliente desse chat. O trabalho foi feito em Python e o protocolo de camada de transporte utilizado foi o TCP. Esse protocolo possui a características de entrega confiável de pacotes o que é muito importante em chats, pois é

2. Desenvolvimento

2.1. O chat

O trabalho consistiu em dois códigos desenvolvidos em Python. Um código foi para o servidor e o outro para o cliente. A comunicação entre os dois utilizou o TCP. Além disso foi utilizado threads para que o servidor pudesse atender a várias requisições de vários clientes ao mesmo tempo e para que o cliente pudesse receber mensagens de outros usuários no momento que fosse mandado pelo servidor.

2.2. Servidor

No servidor a cada conexão criada por um novo cliente é feita uma nova thread. Ao se estabelecer a conexão o servidor espera por uma requisição de cadastramento do usuário. Após feito o cadastro há o aguardo de uma nova requisição do cliente entre as funções do chat que são: Buscar informações de sala ou participantes, entrar ou sair de uma sala, criar sala, apagar sala ou enviar mensagem. O protocolo pode ser visualizado na figura 1.

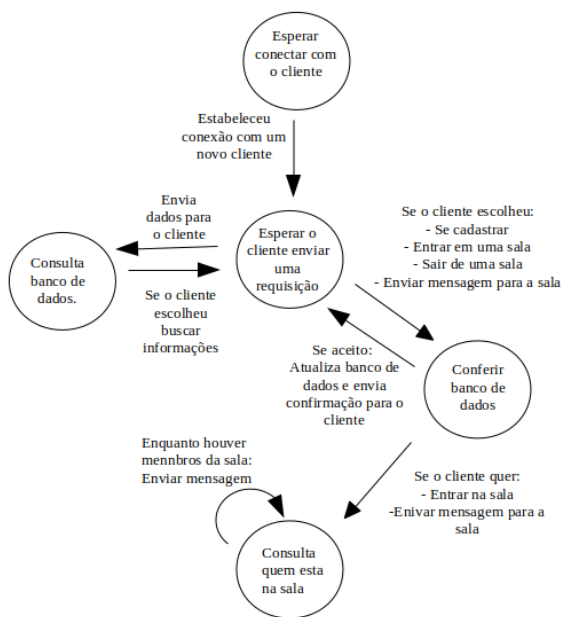


Figura 1. Protocolo da camada de aplicação do servidor.

2.3. Cliente

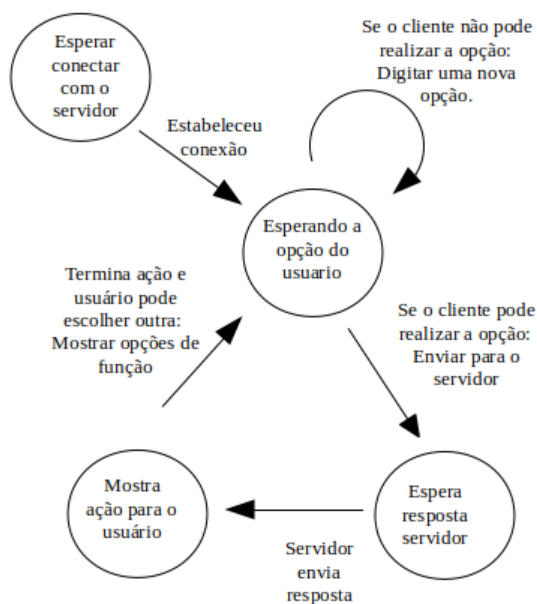


Figura 2. Protocolo da camada de aplicação do cliente.

No cliente duas threads são utilizadas. Uma para se comunicar com o servidor e requisitar as funções e outra para ficar escutando o servidor caso chegue alguma mensagem do grupo. A requisição

é feita primeiro pelo usuário que seleciona um número de 1 a 5 correspondente ao que deseja fazer. Esse número é passado para o servidor, o cliente então aguarda a resposta do servidor e após isso mostra na tela a resposta. Isso é visto na figura 2

2.4. Funções

2.4.1. Cadastro

A primeira função do chat é o cadastro. Assim que o cliente se conecta ele informa um nome que será usado no chat. Esse nome é passado para o servidor e nele é utilizada uma estrutura do tipo dicionário para associar o nome ao socket do cliente. Essa estrutura foi utilizada por ter que usar o nome como uma chave para achar o socket correspondente. A operação de cadastro do cliente é mostrado na figura 3.

```

myllena@myllena-Inspiron-3421: ~/PycharmProjects/TD
myllena@myllena-Inspiron-3421:~/PycharmProjects/TD$ python Servidor.py
Conecction sucefful
False
myllena@myllena-Inspiron-3421: ~/PycharmProjects/TD
myllena@myllena-Inspiron-3421:~/PycharmProjects/TD$ python Cliente.py
Bem vindo! Para se cadastrar no chat escolha um nome
Nome: João

-----
Escolha o numero de uma opcao abaixo:
1- Buscar Informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas

Opcao:

```

Figura 3. Cadastro no chat.

2.4.2. Buscar informações das salas ou de participantes

O usuário pode obter duas informações sobre o chat: as salas existentes e o nome dos participantes do grupo em que se encontra. Para se obter as informações da sala o cliente faz uma requisição para o servidor que envia a lista das salas e o número de participantes, depois da resposta do servidor a lista é colocada na tela para o usuário, como visto na figura 4.

```

Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas

Opcao:1
Digite 1 para saber as salas existente e o numero de participante
Digite 2 para saber os usuarios de sua sala atual
opcao:1
Salas publicas:
Salas, participantes:
('Grupo', 0)
('Sala', 0)

```

Figura 4. Buscar informações sobre as salas.

Na busca pelos participantes o cliente faz a requisição para o servidor. Esse último percorre o banco de dados buscando pelos participantes e forma uma lista com todos os participantes da sala, depois envia ela para o cliente. Por fim é postada na tela todos os nomes dos participantes da sala, como é visto na figura 5. Foi feito uma lista com nomes dos usuários por ser mais fácil de enviar para o cliente e postar na tela.

```

Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas
5- Envio de arquivos e de mensagens de texto

Opcao:1
Digite 1 para saber as salas existente e o numero de participante
Digite 2 para saber os usuarios de sua sala atual
opcao:2
Participantes:
['Maria', 'Joao']

```

Figura 5. Buscar informações sobre os usuários na sala atual.

2.4.3. Entrar ou sair de uma sala

O usuário possui a opção de entrar e sair de apenas uma sala por vez. Assim para entrar em uma sala é preciso conferir se o usuário não está em nenhuma. Caso não esteja o servidor armazena o socket e a sala em uma lista e aceita a requisição do cliente. Após isso é mostrado na tela a mensagem de bem vindo a sala, como mostrado na figura 6. A estrutura de lista foi escolhida por não ser possível ter chaves únicas como no caso do cadastro do usuário, assim a associação é feita entre um elemento da lista e seu próximo para associar a sala a

um usuário.

```

Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas
5- Envio de arquivos e de mensagens de texto

Opcao:2
Digite 1 para entrar em uma sala e 2 para sair de uma sala:
opcao:1
Salas, participantes:
('Grupo', 0)
('Gato', 0)
Digite o nome da sala desejada:
Grupo
Bem vindo a sala:
Grupo

```

Figura 6. Entrar em uma sala.

A opção de sair só ocorre se o usuário estiver em uma sala. Após a requisição o servidor exclui o socket e o nome da sala do banco de dados. Após isso é mostrado para o cliente que ele saiu da sala, como é visto na figura 7.

```

Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas
5- Envio de arquivos e de mensagens de texto

Opcao:2
Digite 1 para entrar em uma sala e 2 para sair de uma sala:
opcao:2
Voce saiu da sala

```

Figura 7. Sair de uma sala.

2.4.4. Criar sala

No chat também tem a opção criar sala. A sala criada é pública e qualquer usuário pode entrar nela. Para criar uma sala o usuário digita o nome de uma sala e o servidor associa a sala a um número 0 de participantes. A opção criar sala é mostrada na figura 8

```

-----
Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas

Opcao:3
Digite um nome para a sala:
Sala:Grupo

```

Figura 8. Criar uma sala.

2.4.5. Apagar sala

Além de criar salas o usuário pode apagá-las. Para executar esse ação a sala a ser deletada deve estar vazia, caso contrário o usuário não consegue deletá-la. Ao deletar a sala o servidor exclui a sala e o número de participantes do banco de dados e envia uma resposta positiva ao cliente de que a sala foi deletada. A opção para o cliente é mostrada na figura 9

```

-----
Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas
5- Envio de arquivos e de mensagens de texto

Opcao:4
Salas, participantes:
('Grupo', 0)
('Gato', 0)
Apenas salas sem participantes podem ser deletadas, escolha uma acima vazia:
sala:Gato
Sala deletada

```

Figura 9. Apagar uma sala.

2.4.6. Enviar mensagem

O principal do chat é a conversa entre dois ou mais usuários e há esse opção no trabalho. Para se implementar essa função foi necessário um thread também no cliente para que esse sempre fique escutando o servidor. A cada resposta do servidor é comparado o nome da sala do cliente com a resposta e quando forem iguais é mostrado na tela a mensagem vinda do grupo. Essa função foi a mais difícil de fazer no início por não ter sido usado thread, porém após considerar essa ideia a implementação foi mais tranquila.

```

-----
Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas
5- Envio de arquivos e de mensagens de texto

Opcao:5
Digite a mensagem a ser enviada:
Olá, João

-----
Escolha o numero de uma opcao abaixo:
1- Buscar informacoes sobre as salas
2- Entrar em e sair de uma sala existente
3- Criacao de sala publicas
4- Apagar salas
5- Envio de arquivos e de mensagens de texto

Opcao:Maria Entrou na Sala.
Maria : Olá, João

```

Figura 10. Mandar mensagens para o grupo.

2.4.7. Arquivo log

As requisições e as respostas do servidor e cliente foram armazenadas em arquivo log. Ele contém a hora da requisição, o socket do cliente que requisitou, o tipo de opção pedido e a resposta do servidor.

3. Discussão e Conclusão

Com o trabalho foi possível entender melhor como funciona uma comunicação entre servidor e cliente e suas dificuldades. Uma das maiores dificuldades encontrada foi o tempo de comunicação entre eles, por conta disso ainda há problemas inesperados no chat. Esse problemas são que em algumas requisições o cliente trava e caso seja fechado e aberto um novo cliente a função funciona novamente, a conclusão para isso é que ainda há conflito nos tempos de envio e recebimento entre cliente e servidor. Por mais que tenha sido usado a função `time.sleep` a requisição ainda não funciona às vezes. Outra dificuldade encontrada foi fechar o programa do cliente, devido a thread do cliente não é possível fechar o programa do cliente no terminal do linux utilizando o comando `ctrl + c`, mesmo que a opção de sair do chat não tenha sido pedido foi detectado esse problema durante a implementação do chat. Mesmo tendo essas dificuldades o chat ainda funciona em todas as opções desde de que não haja perda

de informação entre cliente e servidor de
vido a falta de sincronização entre os dois.

[2] http://www3.ifrn.edu.br/~jurandy/fdp/doc/aprenda-python/capitulo_08.html

Referências

- [1] Kurose Ross, Redes de Computadores e a Internet – Uma abordagem top-down, a edição, editora Pearson.
- [6] <https://wiki.python.org.br/SocketBasico>