



Protocolo para bate-papo

Renan Rodolfo da Silva
São José - 26/02/2019

Sumário

| | | |
|----------|-----------------------------|----------|
| 1 | Introdução | 3 |
| 2 | Especificações | 4 |
| 3 | Manual de utilização | 5 |
| 4 | API do Chat | 6 |

1 | Introdução

Neste relatório será apresentado as especificações do protocolo de bate-papo referente a tarefa da matéria de Projeto de Protocolo, assim como o manual de utilização.

2 | Especificações

Serviço: Transferência de mensagens puramente textuais entre usuários através de sockets.

Ambiente de execução: Servidor gera uma conexão TCP utilizando IPV4 entre usuários que desejam ser conectados, gerenciando a transferência de mensagens entre os usuários por meio de sockets providos da API BSD Sockets, sendo assim, um modelo cliente-servidor, onde possibilita o envio de mensagens privadas ou públicas.

Vocabulário: Possui os comandos :Mensagem pública, Nome_Destino:Mensagem privada, {quit}, e os códigos de erro 01,02,03,04 e 05.

- 01 - Esse apelido já está em uso
- 02 - Destino da mensagem privada não existe
- 03 - Cancelamento de conexão forçado
- 04 - Servidor de destino é Inválido
- 05 - Endereço IP do servidor é inválido”

Codificação: Puramente textual(ASCII)

Comportamento:

Comportamento lado Servidor:

1. Recebe um endereço IP e porta válidos onde deve ficar aguardando conexões de clientes.
2. Quando algum cliente se conecta, mantém a conexão até que o usuário encerre a aplicação.
3. Solicita a aplicação do cliente uma identificação para poder usar o chat.
4. Recebe tudo o que cliente digita, fazendo assim o gerenciamento das mensagens, encaminhando para o destino correto, caso esteja disponível, ou alertando que o destino não está mais disponível.

Comportamento lado cliente.

1. Recebe um endereço IP e porta válidos referente ao servidor onde deve se conectar.
2. Deve inserir um apelido(identificação) para conseguir se conectar e usar o chat
3. Pode enviar uma mensagem privada ou pública e sair do chat quando quiser.

3 | Manual de utilização

Manual de uso do lado servidor:

1. Necessário executar primeiro o arquivo servidor.py com alguma IDLE python
2. Em seguida é necessário informar um endereço IP que seja válido em sua rede para que a aplicação possa ficar aguardando conexões de clientes, assim como deve informar uma porta de escuta, caso não seja um endereço válido, irá mostrar erro 05 e deverá ser reiniciado a aplicação. Se deixar o endereço de IP em branco será atribuído o IP 127.0.0.1.
3. Assim que algum cliente se conectar, será mostrado o endereço IP e a porta do cliente que acabou de se conectar, e irá solicitar uma identificação ao mesmo.
4. Após aprovada a identificação, o servidor mantém a conexão com o cliente e possibilita a ele enviar mensagens privadas usando o comando Destino(apelido):Mensagem ou mandar para todos que estiverem online com :Mensagem pública.
5. Assim que o cliente enviar uma mensagem privada, ele irá receber uma mensagem "Mensagem enviada" se o destino estiver correto e online, caso contrário informará erro 02.
6. O cliente que recebe uma mensagem privada, receberá uma mensagem de "Mensagem recebida de: Cliente(Apelido de quem mandou).
7. Servidor encerra a conexão com o cliente assim que ele fechar a aplicação cliente.py ou digitar quit, informando para todos os usuários conectados que esse cliente saiu.

Manual de uso do lado cliente:

1. Assim que o servidor já estiver em execução, execute o arquivo cliente.py com alguma IDLE python
2. Será solicitado ao usuário um endereço IP e uma porta, que deve ser o endereço ao qual foi definido na aplicação servidor.py para aguardar conexões de clientes, caso endereço esteja errado, será informado erro 04, ao contrário, informará que a conexão foi estabelecida.
3. Assim que conexão estiver estabelecida, será mostrado uma listagem com todos os usuários que estão conectados, para caso queira mandar uma mensagem privada a alguém, o envio da mensagem está descrito no item 4 do manual do servidor.
4. Caso queira sair do chat, basta digitar quit ou fechar a aplicação.

4 | API do Chat

O intuito da API seria possibilitar que qualquer usuário consiga utilizar o servidor e o cliente já criado, para criar uma interface para o chat, por exemplo, ou usar apenas para trocas de mensagens textuais mesmo, mas ainda não foi implementado. A API disponibilizaria as seguintes funções.

Conecta_Servidor(): Nessa função o usuário deve informar um endereço de IP e uma porta ao qual o servidor irá ficar aguardando novas conexões, e armanezaria o endereço do socket do cliente.

Conecta_cliente(): O usuário passaria como parâmetro o endereço IP e a porta do servidor onde desejaria se conectar para entrar no sistema de chat.

Conecta_cliente.enviar_Msg(Destino,mensagem): : Quando cliente se conecta, é apresentado para o usuário a lista das pessoas online, portanto, ele poderia passar como parâmetro dessa função o Destino e a mensagem.

Desconecta_Servidor(): Caso precisar sem encerrado a aplicação do servidor, seria necessário chamar a função DesconectaServidor(quit) para tal.

Desconecta_cliente(): O mesmo acontece quando o cliente deseja encerrar um conexão existente com o servidor