RFC-Redes2023

Projeto Redes - BCC-IFG 2023 (Aluno: Gabriel Oliveira Braga)

Descrição: Projeto de Curso Bacharelado de Ciências da Computação, Instituto Federal de Goiás - Campus Anápolis

Disciplina: Redes de Computadores - 02/2023

Link: Link desse projeto no github

Comandos e Tutorial

Como iniciar e utilizar o Chat Server!

Iniciando o Servidor

Primeramente se inicia o servidor rodando o ChatServer.main()

```
& 'pasta_ate_o_executavel_java.exe' '-cp' 'ChatServer'
```

Exemplo utilizando o Visual Studio Code

```
& 'C:\Program Files\Java\jdk-17\bin\java.exe'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\gabri\AppData\Roaming
\Code\User\workspaceStorage\ab7295dbf81058516ef039c606cb423d\redhat.java\jdt_ws\RFC-
Redes2023_dc8c22c8\bin' 'ChatServer'
```

Tutorial para abrir porta do computador

Em seguida iniciaremos os cliente, pode ser na mesma máquina para teste, ou em máquinas difentes, para isso é necessário liberar a porta no firewall, para que os demais computadores tenha acesso ao computador servidor.

Tutorial de Windowns via Comando TechExpert Tips

Tutorial de Windowns via Interface Gestor Tecnico

Tutorial de Linux Via Comando

Iniciando o Cliente

Com o servidor já iniciado, agora ele está em modo de espera por novas conexões, como o codigo é feito em multi threads, fica a critério do sistema operacional dar suporte e limitar a quantidade máxima de conexões.

Nota:

```
Caso tenha alterado a porta de conexão do servidor [Linha 28 do ChatServer]
(https://vscode.dev/github/GabrielOBraga/RFC-Redes2023/blob/main/ChatServer.java#L28),
é necessário alterar também na [Linha 24 do ChatCliente](https://vscode.dev/github
/GabrielOBraga/RFC-Redes2023/blob/main/ChatClient.java#L24)
```

o Comando para iniciar o ChatCliente segue o mesmo padrão

```
& 'pasta_ate_o_executavel_java.exe' '-cp' 'ChatClient'
```

Exemplo utilizando o Visual Studio Code

```
& 'C:\Program Files\Java\jdk-17\bin\java.exe'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\gabri\AppData\Roaming
\Code\User\workspaceStorage\ab7295dbf81058516ef039c606cb423d\redhat.java\jdt_ws\RFC-
Redes2023_dc8c22c8\bin' 'ChatClient'
```

Ao iniciar será pedido pelo nome do usuário logado, não possui nenhum sistema de verificação de autenticade, nomes duplicados, ou senhas, para isso é recomendado um sistema de arquivos, onde é armazenado o login e uma senha de usuário, ao ser cadastrado a primeira vez no sistema. o uso de banco de dados também possibilita essa funcionalidade.

Comandos

O comando de ajuda é o **/help**, onde retorna uma descrição amigável, sobre os outros 3 comandos do sistema!

```
    ____________________________________________________________________
    Comandos para se comunicar o BragaZap!

    /logout - caso você queira sair do Zap!
    /users - para você ver quais dos seus colegas estão online!
    /changeuser - para você ser um ninja e alterar seu nome de usuário!
    /help - cuidado, recursividade encontrada! Error: /help
    ____________________________________________________________________
```

Para deslogar com segurança do chat, se utiliza o comando **/logout**

Possui um comando onde ele retorna o nome de todos os usuários que estão ativos no chat, para visualizar use o comando **/users**

Como adição sem utilidade bem definida, mas apenas para fins de possibilidades, foi adicionado o comando **/changeuser** que permite o usuário alterar seu nome, caso tenha escrito errado ao entrar no chat, e deseja fazer a troca

```
O comando avisa todos os outros usuários ativos, que foi alterado o nome de usuário
```

Descrição do Projeto

Trabalho sobre camada de aplicação.

Resumo: Desenvolver um cliente e um servidor ou híbrido (cliente/servidor) que realize um serviçodefinido por você. Utilize a linguagem Java.

Detalhes:

```
1. Trabalho em dupla;

2. Definição de um mini RFC sobre o protocolo definido, sintaxe e semântica, consultar exemplos de RFC: [Hypertext Markup Language - 2.0](https://datatracker.ietf.org/doc/html/rfc1866) (2 pontos);

    2.1 Seu RFC deverá descrever todas as palavras chaves utilizadas, sua sintaxe e semântica dentro doserviço desenvolvido por você. Mínimo de 20 palavras/tokens de controle. (1 pontos);

3. Utilizar a abordagem de múltiplos clientes e um servidor. (3 pontos);

4. Exibir durante uma apresentação de 20 minutos o mini RFC, o servidor e o cliente em operação. Permitir que o professor tenha uma cópia do cliente para conectar ao servidor simultaneamente. Os dois alunos devem apresentar e destacar sua contribuição no projeto. Apresentação é do sistema rodando e não de slides. (4 pontos);
```

Data entrega e apresentação: 27/10/2023

Referencias Utilizadas

Essas são as classes e interfaces importadas no código:

java.io.BufferedReader: Para ler os dados de entrada do cliente.

java.io.IOException: Para tratar exceções de E/S (entrada/saída).

java.io.InputStreamReader: Para criar um InputStreamReader a partir de uma InputStream.

java.io.PrintWriter: Para escrever dados de saída para o cliente.

java.net.ServerSocket: Para criar o socket do servidor e aguardar conexões dos clientes.

java.net.Socket: Para representar o socket de comunicação entre o servidor e o cliente.

java.util.ArrayList: Para armazenar a lista de clientes e membros do grupo.

java.util.HashMap: Para armazenar os usuários logados e os grupos.

java.util.List: Para representar listas de objetos.

java.util.Map: Para representar mapeamentos de chave-valor.…