



## Redes de Computadores – Turma A – Trabalho 1

**Professor:** Marcos F. Caetano <mfcaetano@unb.br>

**Monitor:**

**Resumo:** Implementação de uma aplicação de troca de mensagens multi-utilizador em tempo real.

# 1 Introdução

Um dos primeiros sistemas desenvolvidos para troca de mensagens em tempo real foi o Internet Relay Chat(IRC). O IRC foi originalmente concebido por Jarkko Oikarinen em 1988. Desde o seu começo na Finlândia, tem sido usado em mais de 60 países de todo o mundo. Apesar de inicialmente pretender ser um substituto do programa "talk", típico em sistemas UNIX, acabou por se tornar em muito mais do que isso.

O IRC é um sistema de troca de mensagens multi-utilizador em tempo real, onde as pessoas convivem em "canais"(locais virtuais, normalmente com um tópico de conversação), para falar em grupo ou em privado. O IRC continua em constante evolução e a forma como as coisas funcionam numa determinada altura pode não corresponder à forma como funcionam na semana seguinte. Cada vez que se entra no IRC, é aconselhável que se leia a "Message Of The Day"(MOTD) para se conhecerem as modificações introduzidas.

O IRC ganhou a sua fama internacional durante 1991, com a Guerra do Golfo Pérsico, onde constantes atualizações de informação apareciam nesta rede, provenientes de todo o mundo, e uma enorme percentagem de utilizadores se reunia num único canal para ouvir (ou melhor, ler) os últimos relatos. Teve um uso idêntico durante o golpe contra Boris Yeltsin, em Setembro de 1993, em que utilizadores do IRC de Moscovo emitiam relatos em direto acerca da evolução da situação.

O acesso ao serviço de IRC requer a utilização de um programa do tipo "cliente", que o liga à rede de IRC, através de um outro programa chamado "servidor". Existem diversas RFC's sobre a implementação de tal aplicação, porém neste trabalho uma versão simplificada e modificada será apresentada.

# 2 Descrição e implementação

O sistema de chat será composto por dois programas distintos, um **Cliente** e um **Servidor**. O programa **Cliente** deve prover ao usuário uma plataforma para envio de comandos, mensagens, arquivos ao servidor. Como também o recebimento de mensagens e arquivos oriundos de outros usuários. O programa Servidor deve suportar múltiplos usuários conectados simultaneamente recebendo e respondendo comandos, e encaminhando as mensagens e arquivos aos respectivos grupos e usuários.

## 2.1 Cliente

O programa **Cliente** é o canal pelo qual o usuário acessa os diversos grupos disponíveis no **Servidor**, ele exibe todas as mensagens enviadas dentro de um grupo, recebe e encaminha ao **Servidor** comandos, arquivos e mensagens dos usuários. Ele deve ser capaz de lidar com todas as demandas do usuário ao mesmo tempo em que recebe mensagens e arquivos que estão sendo difundidos dentro dos grupos, executando essas duas tarefas de forma paralela.

Assim que o programa é aberto ele deverá exibir uma tela com a "Message Of The Day"(MOTD), contendo o dia e horário local, além de uma lista com todos os comandos disponíveis. **Tal mensagem deve ser solicitada ao programa Servidor e não armazenada de forma fixa no código do**

**programa Cliente.** Em seguida o programa deve aguardar por um comando do usuário para prosseguir com sua execução.

Este programa não suporta comunicação individual entre os usuários, para se comunicarem eles devem estar sempre conectados a um grupo. **Antes de se conectarem a um grupo os usuário só poderão listar os grupos disponíveis, criar um novo grupo, excluir um grupo existente ou se juntar a um grupo.** Entretanto os usuários poderão enviar mensagens privadas uns aos outros através do grupo de chat, usando um comando específico a mensagem enviada por ele só será entregue ao usuário de destino sem que os outros membros do grupo vejam o seu conteúdo.

Todo **usuário** é identificado unicamente através de um *nickname*, ou seja não há duas pessoas com o mesmo *nickname*. O mesmo vale para os **Grupos**, não existem dois grupos com o mesmo nome. O *nickname* contém no máximo 50 caracteres sem espaços.

### 2.1.1 Grupos

Cada grupo de chat possui um único **administrador**, ele é sempre usuário que criou o grupo e não pode ser modificado posteriormente. O nome do Grupo também não pode ser modificado e contém no máximo 50 caracteres sem espaços. Os grupos de chat só podem ser excluídos pelo seu respectivo administrador. Além disso, os grupos só podem ser removidos quando não há nenhum usuário conectado ao grupo. Uma tentativa de remoção contendo usuários ativos deve ser notificada e negada pelo servidor.

Quando conectados a um grupo de chat os usuários podem enviar mensagens e arquivos, que serão encaminhados a todos os outros usuários conectados aquele mesmo grupo. Além disso um usuário pode ter acesso a lista de todos os membros conectados, esta informação é pertinente pois existe a possibilidade do envio de mensagens privadas dentro do ambiente do grupo de chat.

## 2.2 Servidor

O **Servidor** é programa responsável por receber as mensagens, comandos e arquivos advindos da aplicação **Cliente**, dando a estes o seu devido tratamento. Vários Clientes estarão conectados a ele nos mais diversos grupos. É papel do **Servidor** encaminhar corretamente as mensagens para cada grupo destinatário sem interromper o processamento de outros grupos.

O Servidor faz todos os encaminhamentos em tempo real, não armazenando qualquer mensagem. Contudo, para facilitar a sua implementação, os arquivos enviados pelos clientes nos grupos poderão ser armazenados no Servidor por um período determinado de tempo.

Ele também deve manter registros das atividades que desempenha na forma de *Logs de dados*. Toda mensagem, comando ou arquivo encaminhado por ele deve ter seus metadados (usuário de origem, grupo de destino, hora, data) registrados para futuras verificações.

## 2.3 Lista de Comandos

Aqui estão descritos os comandos que os usuários da aplicação terão a sua disposição:

- Gerais - Devem funcionar em todos os ambientes do programa.

**/help** - Exibe a Mensagem do Dia (MOTD) que traz consigo um texto de ajuda sobre todas as funcionalidades suportadas nesta versão do programa.

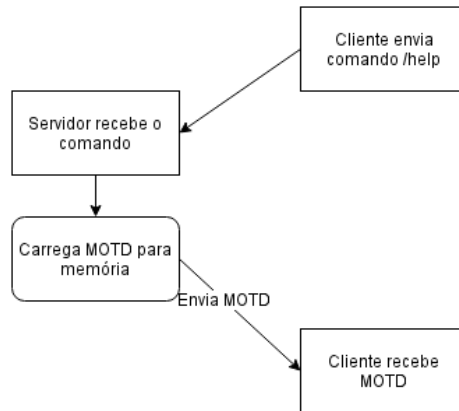


Figura 1: Help Fluxograma

**/nick nome** - Troca o *nickname* do usuário pelo argumento nome passado juntamente ao comando

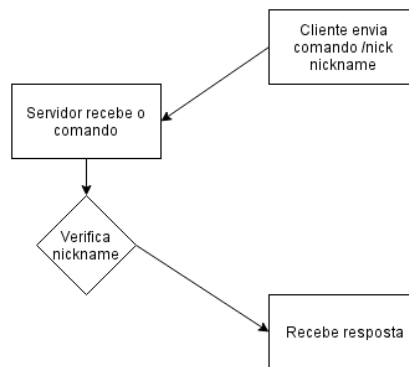


Figura 2: Nick Fluxograma

**/leave** - Quando na tela inicial, fecha o programa. Dentro de algum grupo, deixa o grupo e retorna a tela inicial.

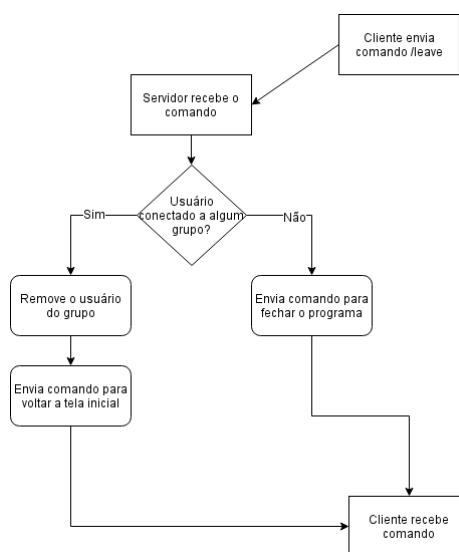


Figura 3: Leave Fluxograma

**/list** - Quando na tela inicial listam todos os grupos disponíveis no servidor bem como quantos usuários estão conectados aquele grupo. Dentro de algum grupo listam todos os usuários presentes na conversa.

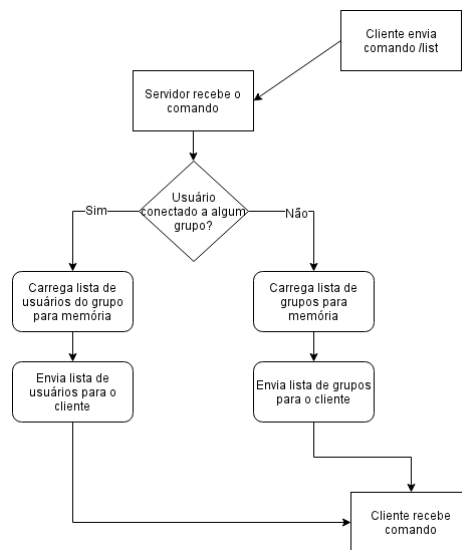


Figura 4: List Fluxograma

- Comandos de Grupos - Comando que são utilizados no modo inicial do programa para gerenciamento de grupos.

**/join grupo** - Comando para entrada em um grupo, quando executado o programa Cliente deve entrar no grupo descrito pelo parâmetro.

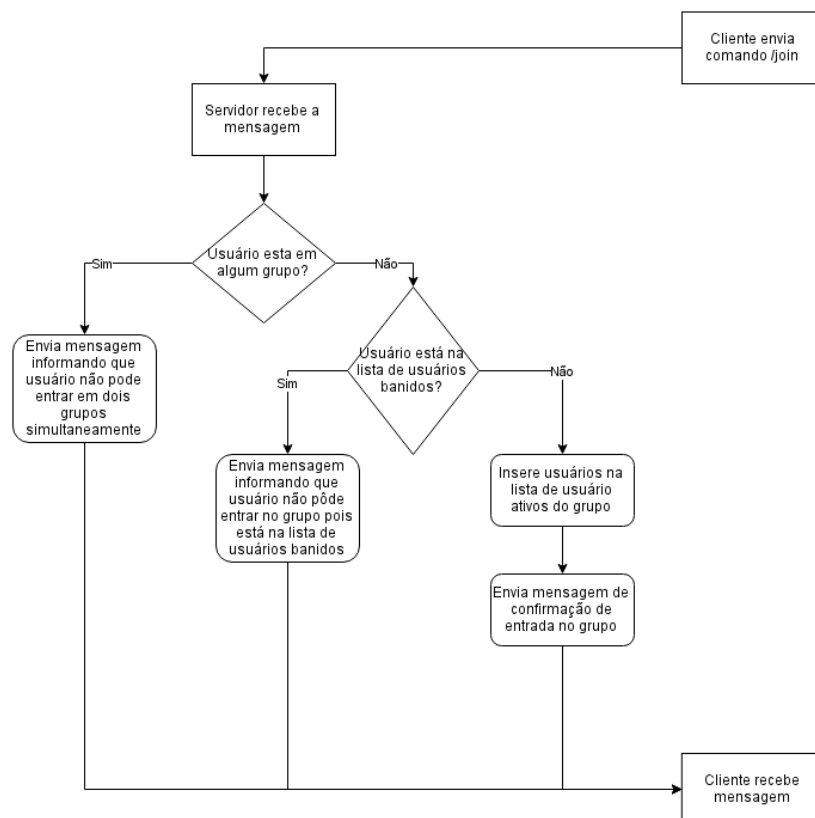


Figura 5: Join Fluxograma

**/create grupo** - Cria um grupo com o nome do parâmetro.

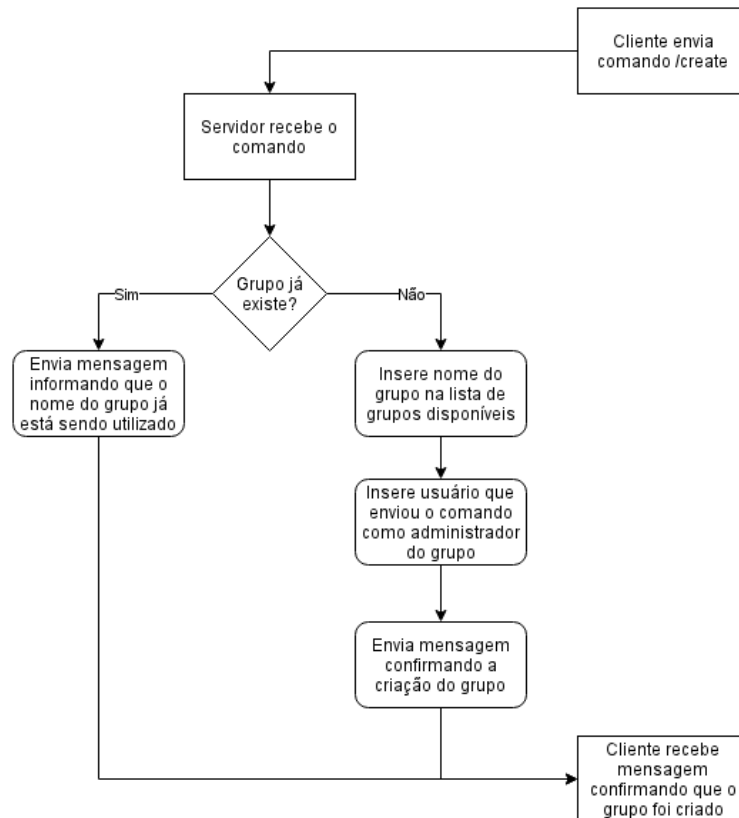


Figura 6: Create Fluxograma

**/delete grupo** - Deleta o grupo com o nome descrito pelo parâmetro. Lembrando que somente o administrador do grupo pode deletá-lo e também que só é possível deletar um grupo que não possui nenhum membro ativo.

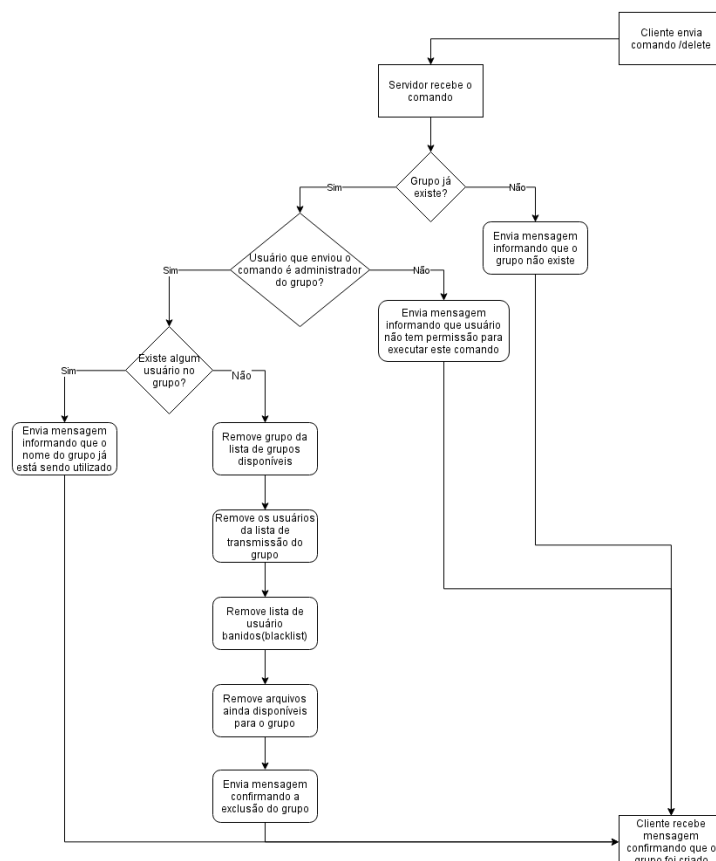


Figura 7: Delete Fluxograma

- Chat - Comando que só devem funcionar dentro das conversas em grupo

**/away** - Informa que o usuário está indisponível para conversa. Quando executado deve enviar uma mensagem a todos os membros do grupo informando que o usuário que executou o comando está temporariamente fora do chat. Caso algum usuário envie uma mensagem privada ele deve receber uma resposta automática de que o usuário está temporariamente indisponível.

Enquanto esta indisponível o usuário continua a receber mensagens endereçadas aquele grupo. Quando o usuário enviar uma mensagem ao grupo de chat ele deixa de estar indisponível e uma mensagem deve ser enviada a todos os usuários informando que o usuário deixou de estar indisponível.

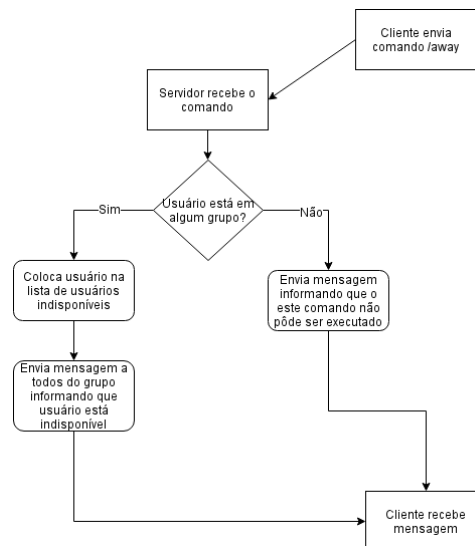


Figura 8: Away Fluxograma

O comando away também gera alteração quando uma mensagem é enviada por um usuário seja ela a todos os membros do grupo ou privada. Uma verificação deve ser feita para constatar se o usuário estava indisponível antes de enviar a mensagem, se isto aconteceu primeiro deve-se enviar uma mensagem informando que usuário deixou de estar indisponível.

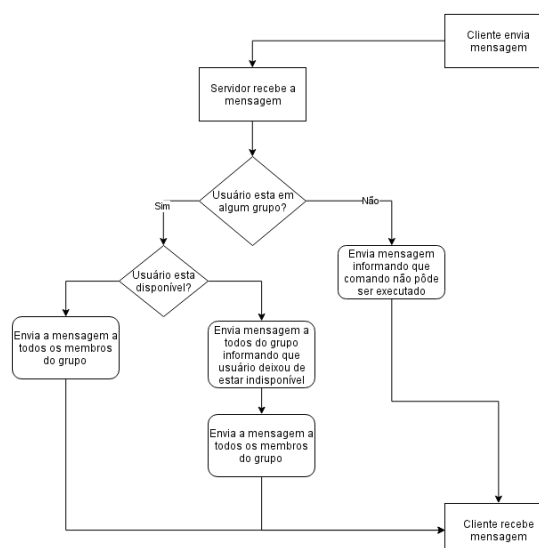


Figura 9: Mensagem Fluxograma

**/msg nick mensagem** - Envia mensagem somente ao usuário especificado pelo parâmetro nick. A mensagem deve ser enviada pelo mesmo chat do grupo, porém somente ao usuário especificado.

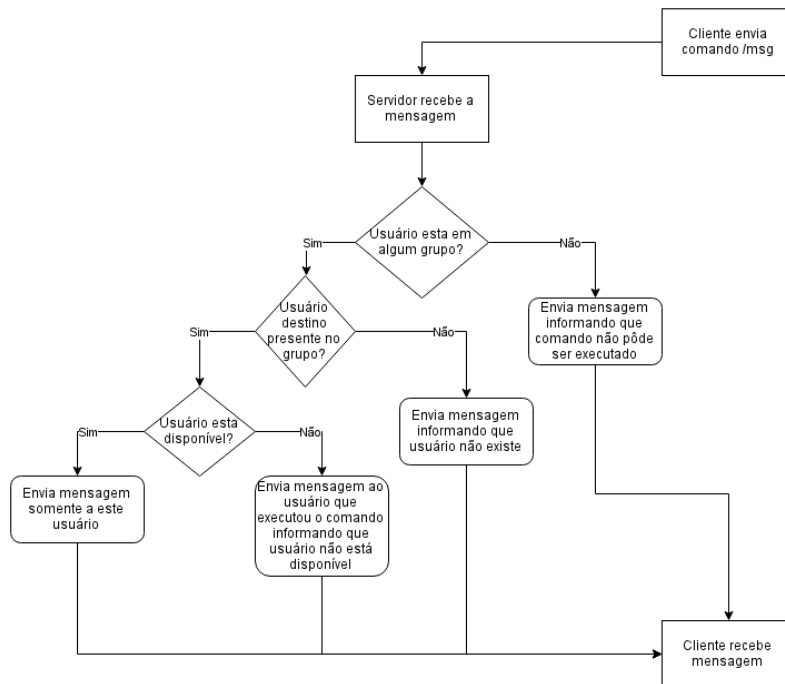


Figura 10: Msg Fluxograma

**/ban nick** - Expulsa de forma permanente o usuário especificado pelo parâmetro nick do grupo em questão. Somente o administrador do grupo pode executar este comando.

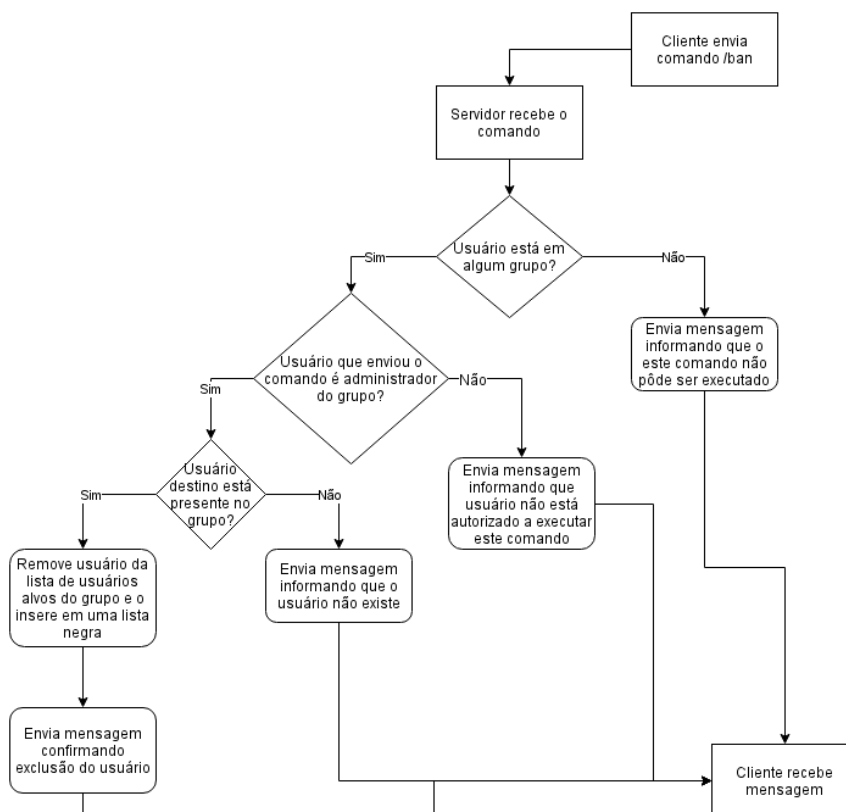


Figura 11: Ban Fluxograma

**/kick nick** - Desconecta o usuário especificado pelo parâmetro nick do grupo em questão. Somente o administrador do grupo pode executar este comando. O usuário pode entrar novamente no grupo sem nenhuma restrição.

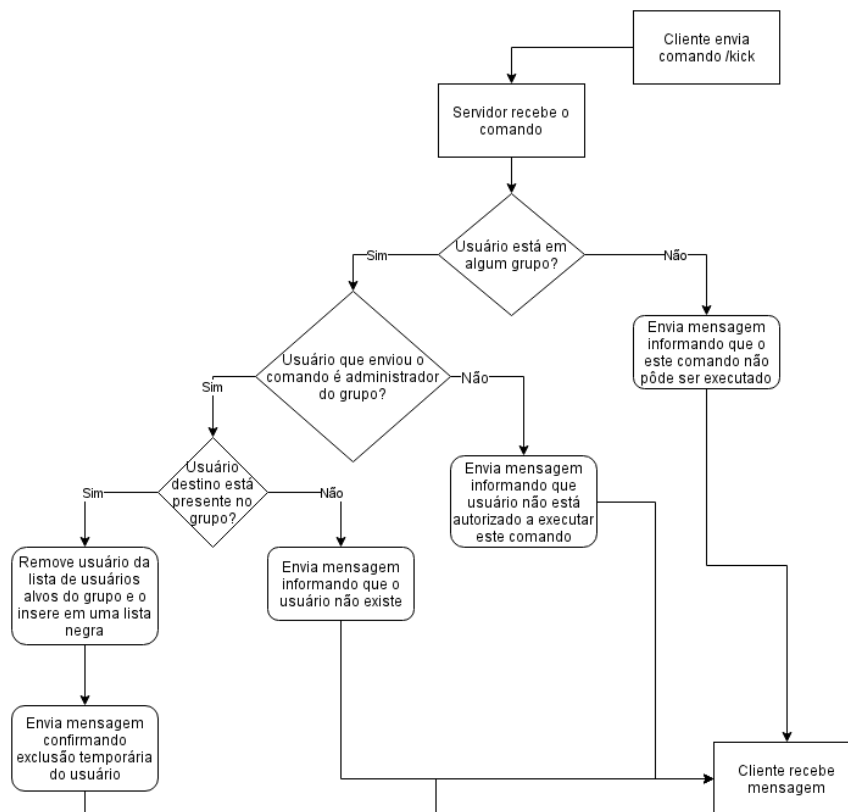


Figura 12: Kick Fluxograma

**/clear** - Limpa a janela de conversa.

**/file caminho\_para\_arquivo** - Envia um arquivo ao servidor para ser difundido posteriormente. O arquivo a ser enviado deve ser passado como parâmetro. Assim que o arquivo é enviado uma mensagem deve ser enviada a todos os membros do grupo informado que um novo arquivo está disponível para o grupo, lembrando que o arquivo deve permanecer disponível por um tempo determinado e não indefinidamente.

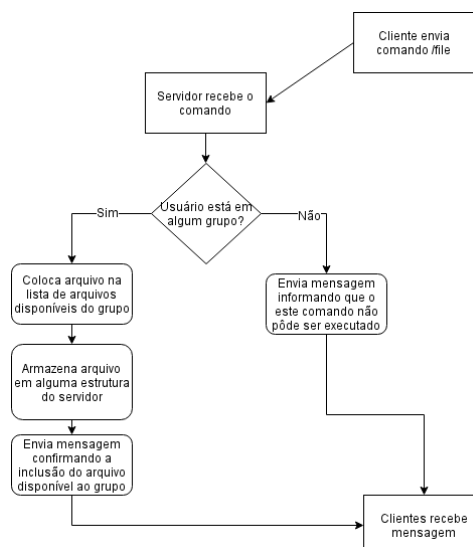


Figura 13: File Fluxograma



**/list\_files** - Lista todos os arquivos disponíveis naquele grupo.

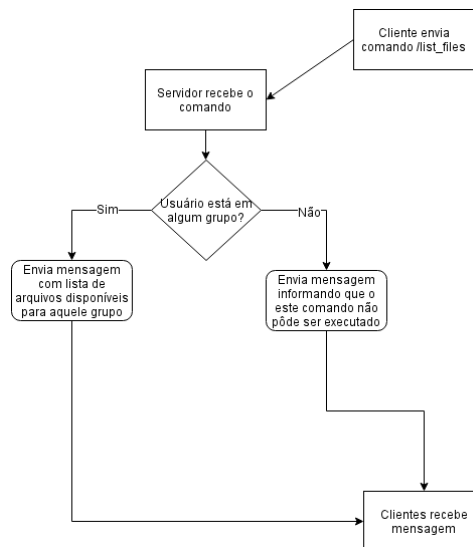


Figura 14: List File Fluxograma

**/get\_file nome\_arquivo** - Obtém o arquivo descrito pelo parâmetro. Caso o arquivo não exista mais na lista de arquivos disponíveis deve mostrar mensagem de erro.

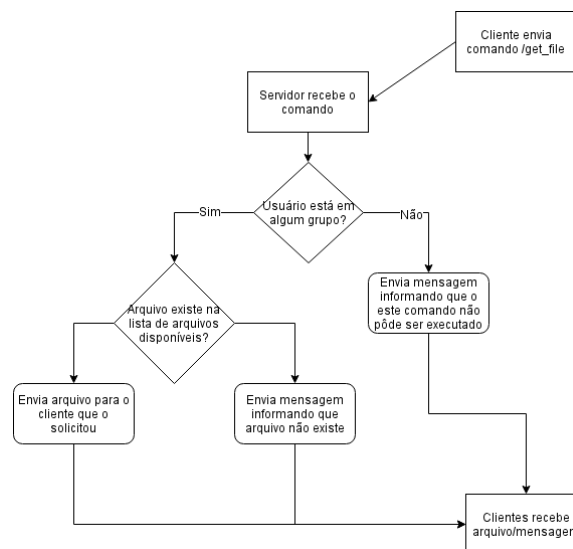


Figura 15: Get File Fluxograma

## 3 Funcionamento do sistema

Exemplos de funcionamento do sistema serão apresentados a seguir:

### 3.1 Tela inicial

```
Chat Redes de Computadores - versão 1.0
Mensagem do Dia (MOTD)
-----Comandos disponíveis-----
/help - Exibe esta tela de ajuda
/nick nome - Altera o nickname do usuário pelo parâmetro nome
/leave - Sai do ambiente atual para o anterior. Quando dentro de um chat volta a tela inicial, na tela inicial sai do programa
/list - Lista os grupos disponíveis no Servidor. Dentro do Chat lista todos os usuários na conversa
/join grupo - Acessa o grupo especificado
/create grupo - Cria um novo grupo
/delete grupo - Exclui um grupo existente
/away - Informa que usuário esta temporariamente indisponível
/msg nick mensagem - Envia uma mensagem privada a um usuário descrito por nick
/ban nick - Exclui usuário definitivamente do grupo
/kick nick - Expulsa usuário temporariamente do grupo
/clear - Limpa todas as mensagens da tela
/file caminho_para_arquivo - Envia arquivo ao grupo
/list_files - Lista todos os arquivos disponíveis no grupo
/get_file nome_arquivo - Obtem o arquivo disponível
-----
Seja muito bem-vindo a este Servidor
Digite um comando para continuar:
█
```

Figura 16: Tela Inicial

### 3.2 Trocando nickname

```
Chat Redes de Computadores - versão 1.0
Mensagem do Dia (MOTD)
-----Comandos disponíveis-----
/help - Exibe esta tela de ajuda
/nick nome - Altera o nickname do usuário pelo parâmetro nome
/leave - Sai do ambiente atual para o anterior. Quando dentro de um chat volta a tela inicial, na tela inicial sai do programa
/list - Lista os grupos disponíveis no Servidor. Dentro do Chat lista todos os usuários na conversa
/join grupo - Acessa o grupo especificado
/create grupo - Cria um novo grupo
/delete grupo - Exclui um grupo existente
/away - Informa que usuário esta temporariamente indisponível
/msg nick mensagem - Envia uma mensagem privada a um usuário descrito por nick
/ban nick - Exclui usuário definitivamente do grupo
/kick nick - Expulsa usuário temporariamente do grupo
/clear - Limpa todas as mensagens da tela
/file caminho_para_arquivo - Envia arquivo ao grupo
/list_files - Lista todos os arquivos disponíveis no grupo
/get_file nome_arquivo - Obtem o arquivo disponível
-----
Seja muito bem-vindo a este Servidor
Digite um comando para continuar:
\nick estudante_rc
Olá estudante_rc, seu nickname foi alterado com sucesso!
Digite um comando para continuar:
estudante_rc ->█
```

Figura 17: Comando /nick

### 3.3 Comando /list página inicial

```
Chat Redes de Computadores - versão 1.0
Mensagem do Dia (MOTD)
-----Comandos disponíveis-----
/help - Exibe esta tela de ajuda
/nick nome - Altera o nickname do usuário pelo parâmetro nome
/leave - Sai do ambiente atual para o anterior. Quando dentro de um chat volta a tela inicial, na tela inicial sai do programa
/list - Lista os grupos disponíveis no Servidor. Dentro do Chat lista todos os usuários na conversa
/join grupo - Acessa o grupo especificado
/create grupo - Cria um novo grupo
/delete grupo - Exclui um grupo existente
/away - Informa que usuário esta temporariamente indisponível
/msg nick mensagem - Envia uma mensagem privada a um usuário descrito por nick
/ban nick - Exclui usuário definitivamente do grupo
/kick nick - Expulsa usuário temporariamente do grupo
/clear - Limpa todas as mensagens da tela
/file caminho_para_arquivo - Envia arquivo ao grupo
/list_files - Lista todos os arquivos disponíveis no grupo
/get_file nome_arquivo - Obtem o arquivo disponível
-----
Seja muito bem-vindo a este Servidor
Digite um comando para continuar:
estudante_rc ->/list
-----Grupos disponíveis-----
- Estudantes_de_redes_unidos
- Discussao_trabal
- PDS
- off-topic
estudante_rc ->█
```

Figura 18: Comando /list

### 3.4 Criando um novo grupo

```
Chat Redes de Computadores - versão 1.0
Mensagem do Dia (MOTD)
-----Comandos disponiveis-----
/help - Exibe esta tela de ajuda
/nick nome - Altera o nickname do usuário pelo parâmetro nome
/leave - Sai do ambiente atual para o anterior. Quando dentro de um chat volta a tela inicial, na tela inicial sai do programa
/list - Lista os grupos disponiveis no Servidor. Dentro do chat lista todos os usuários na conversa
/join grupo - Acessa o grupo especificado
/create grupo - Cria um novo grupo
/delete grupo - Exclui um grupo existente
/away - Informa que usuário esta temporariamente indisponível
/msg nick mensagem - Envia uma mensagem privada a um usuário descrito por nick
/ban nick - Exclui usuário definitivamente do grupo
/kick nick - Expulsa usuário temporariamente do grupo
/clear - Limpa todas as mensagens da tela
/file caminho_para_arquivo - Envia arquivo ao grupo
/list files - Lista todos os arquivos disponiveis no grupo
/get_file nome_arquivo - Obtem o arquivo disponível
-----
Seja muito bem-vindo a este Servidor
Digite um comando para continuar:
estudante_rc -> /create Novo_grupo_rc
Olá estudante_rc, seu grupo Novo_grupo_rc foi criado com sucesso! Use /join Novo_grupo_rc para entrar
estudante_rc -> /list
-----Grupos disponiveis-----
- Estudantes_de_redes_unidos
- Discussao_trabl
- PDS
- off-topic
- Novo_grupo_rc
estudante_rc -> █
```

Figura 19: Comando /create

### 3.5 Entrando em um grupo

```
estudante_rc -> /join Novo_grupo_rc
-----Bem vindo ao grupo Novo_grupo_rc-----
estudante_rc -> █
```

Figura 20: Comando /join

### 3.6 Exemplo chat em grupo

```
estudante_rc -> Novo_usuario entrou no grupo
estudante_rc -> █
```

Figura 21: Exemplo chat

```
estudante_rc -> Novo_usuario entrou no grupo
Novo_usuario -> Olá
estudante_rc -> Oi tudo bem?
Novo_usuario -> Tudo e com você?
estudante_rc -> Estou bem
Novo_usuario -> :)
estudante_rc -> █
```

Figura 22: Exemplo chat

### 3.7 Comando /away

```
-----Bem vindo ao grupo Novo_grupo_rc-----
estudante_rc -> /away
Servidor -> Usuário estudante_rc está temporariamente indisponível.
estudante_rc -> █
```

Figura 23: Exemplo comando /away

### 3.8 Comando /msg

```
estudante_rc -> /join Novo_grupo_rc
-----Bem vindo ao grupo Novo_grupo_rc-----
estudante_rc -> /msg Novo_usuario_rc Teste mensagem privada
estudante_rc -> (Novo_usuario_rc) Teste mensagem privada
estudante_rc -> █
```

Figura 24: Exemplo comando /msg

### 3.9 Comando /ban

```
estudante_rc -> /join Novo_grupo_rc
-----Bem vindo ao grupo Novo_grupo_rc-----
estudante_rc -> /ban Novo_usuario_rc
Servidor -> estudante_rc expulsou Novo_usuario_rc de forma permanente
estudante_rc -> █
```

Figura 25: Exemplo comando /ban

### 3.10 Comando /kick

```
estudante_rc -> /join Novo_grupo_rc
-----Bem vindo ao grupo Novo_grupo_rc-----
estudante_rc -> /kick Novo_usuario_rc
Servidor -> estudante_rc removeu Novo_usuario_rc de forma temporária
estudante_rc -> █
```

Figura 26: Exemplo comando /kick

### 3.11 Comandos /file

```
estudante_rc -> /join Novo_grupo_rc
-----Bem vindo ao grupo Novo_grupo_rc-----
estudante_rc -> /file arquivo_teste.txt
Servidor -> Um arquivo foi adicionado - arquivo_teste.txt
Servidor -> Digite /get_file arquivo_teste.txt para obtê-lo
estudante_rc -> /list_files
-----Arquivos disponíveis-----
- arquivo_teste.txt -> estudante_rc
estudante_rc -> █
```

Figura 27: Exemplo comando /file

## 4 Resumo

De forma sintética o sistema implementado neste trabalho deve:

- Possuir um programa Cliente que se comunica com um programa Servidor para envio de mensagens e arquivos.
- A única forma de comunicação dos usuários se dá a partir de grupos de chat
- Os grupos de chat devem aceitar e executar corretamente todas as funções descritas na sessão 2.3
- Cada grupo é unicamente identificado por um nome de no máximo 50 caracteres
- Cada usuário é unicamente identificado por um nome de no máximo 50 caracteres
- Uma vez logado em um grupo um usuário recebe as mensagens de todos os usuários conectados
- Quando utilizado o comando `/msg` envia uma mensagem que só será vista pelo usuário destinatário
- Os arquivos enviados a um grupo podem ser armazenados no Servidor mas somente de forma temporária
- Quando executado o comando `/get_file` transfere o conteúdo do arquivo armazenado no Servidor a máquina do usuário que o executou
- O programa Cliente deve sempre consultar as informações de mensagem do dia, usuário, grupo e arquivos junto ao Servidor.

## 5 Relatório

Um relatório final do projeto deve ser apresentado. Este relatório deve conter:

- Apresentação teórica sobre comunicação TCP e protocolo IRC, incluindo os conceitos envolvidos neste trabalho. Exemplos deverão ser apresentados;
- Documento apresentando a arquitetura do sistema desenvolvido;
- Explicação da arquitetura produzida e da relação entre os principais componentes;
- Doxygen de todo o código produzido;
- *Screenshots* e explicação do funcionamento de todas as funcionalidades implementadas.

## 6 Avaliação

A avaliação consiste em 2 etapas:

- Código e funcionamento do projeto (80%).
  - O código e o relatório deverão ser submetidos no Moodle até o dia **30/10 as 23:55h**;
  - A apresentação do trabalho deverá ser agendada via tópico específico que será aberto no fórum da disciplina. As apresentações acontecerão no período de **31/10 a 4/11** e **7/11 a 9/11**. As apresentações serão feitas fora do horário de aula;
  - Não serão aceitos trabalhos enviados fora do prazo;

– Trabalhos não apresentados também não serão considerados.

- Nota do Relatório (20%).

A descrição detalhada da correção e nota de cada um dos pontos estará a disposição dos alunos na divulgação dos resultados.

## 7 Observações

As seguintes observações deverão ser consideradas pelos alunos que forem fazer o trabalho:

- O trabalho deve rodar **OBRIGATORIAMENTE** na plataforma GNU/Linux;
- O programa pode ser feito em qualquer linguagem;
- No relatório do trabalho deve conter as instruções para compilação do código e a relação de todas as bibliotecas utilizadas;
- O relatório deverá ser entregue **OBRIGATORIAMENTE** no formato de arquivo PDF;
- Códigos copiados (entre alunos ou retirados da Internet) serão considerados "cola" e todos os alunos envolvidos ganharão nota zero;
- Dúvidas sobre o trabalho deverão ser tiradas **NO FÓRUM DE DÚVIDAS DO AMBIENTE APRENDER**. Desta forma, dúvidas comuns e esclarecimentos poderão ser respondidos uma única vez para toda a turma.