


| | | |
|---|--|---------------------------------|
|  | UNIVERSIDADE FEDERAL DA PARAÍBA | |
| | CENTRO DE INFORMÁTICA | |
| | Disciplina | Linguagem de Programação I |
| | Semestre | 2018.2 |
| | Professor | Bruno Jefferson de Sousa Pessoa |

PROJETO DA DISCIPLINA

1. Introdução

O projeto consiste na implementação de um programa para troca de mensagens, denominado *ZapRaiz*, utilizando os conceitos e as tecnologias apresentados durante o curso.

2. Definição do problema

O *ZapRaiz* é um aplicativo em modo texto para troca de mensagens entre usuários que simula as funcionalidades do conhecido programa *WhatsApp*. O programa deverá ser estruturado em vários módulos e usar a API de sockets da linguagem C para comunicação entre processos. O *ZapRaiz* executará em dois modos: **cliente** e **servidor**. No modo cliente, o usuário poderá iniciar uma conversa com um contato ou grupo de contatos. Já no servidor, ele só poderá enviar uma mensagem após o recebimento de uma mensagem anterior. O modo de execução deverá ser indicado no parâmetro argv (ex.: `./zap_raiz servidor`). As seguintes funcionalidades devem ser implementadas.

- **Gerenciamento de contas:** Na primeira vez que o *ZapRaiz* for executado, o usuário deverá criar uma conta, fornecendo dados como nome, usuário e senha. Essas informações deverão ser armazenadas em um arquivo denominado **contas.txt**. Ao longo da execução do programa, o usuário também poderá editar e excluir contas.
- **Login:** Uma vez criada sua conta, o usuário deverá logar no sistema usando suas credenciais. Caso os dados digitados correspondam com aqueles gravados no arquivo de contas, o login será efetuado com sucesso.
- **Gerenciamento de contatos:** Ao logar no sistema, o *ZapRaiz* proverá um conjunto de funcionalidades que darão suporte à troca de mensagens. O gerenciamento de contatos consiste nas operações de adicionar, editar e excluir um contato. As informações básicas de um contato são: nome, telefone e IP. Tais informações deverão ser armazenadas em um arquivo com o seguinte formato: **contatos_<usuário>.txt** (ex.: `contatos_bruno.txt`).
- **Gerenciamento de grupos:** Assim como no gerenciamento de contatos, o *ZapRaiz* deverá possibilitar a criação, edição e exclusão de grupos. As informações básicas de um grupo são: nome do grupo, nomes e IP's dos participantes. Tais dados serão armazenados em um arquivo no formato: **grupos_<usuário>.txt** (ex.: `grupos_bruno.txt`).

- **Iniciar uma conversa:** Ao executar o *ZapRaiz* no modo **cliente**, o usuário poderá iniciar uma conversa com um contato ou com um grupo. Caso escolha enviar uma mensagem para um grupo, todos os seus participantes deverão receber o texto digitado.
- **Participar de uma conversa:** No modo de execução **servidor**, o usuário só poderá enviar uma mensagem após o recebimento de uma mensagem anterior. Nesse modo passivo de execução, não há a possibilidade de enviar mensagem para um grupo. A mensagem será enviada obrigatoriamente para o contato que iniciou a conversa.

3. Sugestão de telas

A seguir será apresentado um conjunto de telas para ilustrar as funcionalidades do *ZapRaiz* e servir como sugestão de projeto de interface.

| | |
|---|--|
| <pre>> ./zap_raiz cliente ***** Seja bem-vindo ao ZapRaiz, o melhor programa de troca de mensagens do planeta! ***** 1-Criar conta 2-Sair</pre> | <pre>ZapRaiz 1-Gerenciar contas 2-Login 3-Sair</pre> |
| <pre>ZapRaiz 1-Gerenciar contas 2-Gerenciar contatos 3-Gerenciar grupos 4-Iniciar conversa 5-Sair</pre> | <pre>ZapRaiz 1-Gerenciar contas 2-Gerenciar contatos 3-Gerenciar grupos 4-Participar de uma conversa 5-Sair</pre> |
| <pre>ZapRaiz 1-Criar conta 2-Editar conta 3-Excluir conta 4-Voltar</pre> | <pre>ZapRaiz 1-Criar grupo 2-Editar grupo 3-Excluir grupo 4-Voltar</pre> |

ZapRaiz

1-Adicionar contato
2-Editar contato
3-Excluir contato
4-Voltar

4. Instruções gerais

- O presente projeto deve ser desenvolvido em grupo de no máximo **dois** alunos.
- O projeto deverá ser enviado no dia **02/05/2019** e apresentado no dia seguinte.
- Deverá ser entregue um arquivo zipado, contendo o código fonte, no formato descrito a seguir:
 - LP2-Projeto-grupo.zip
 - Ex.: LP2-Projeto-jose-maria.zip
- Os seguintes itens serão avaliados: execução, interface, código-fonte, modularização e apresentação. A última terá uma avaliação individual para cada aluno.

5. Instruções para implementação

- O projeto deve ser desenvolvido na linguagem C.
- A solução deverá usar ponteiros, strings, funções, passagem de parâmetros via argc/argv, modularização, estruturas e processamento de arquivos.
- A transmissão mensagens deve ser realizada através dos protocolos TCP/UDP (Sugestão: UDP).