

Rede de relacionamentos - Especificação de Requisitos

PUC-Rio

INF1301 3WA

Data: 7 de Outubro de 2017

Alunos: Gabriel Brito, Gabrielle Brandenburg, Wellington Bezerra

SOBRE O PROGRAMA

O programa a ser desenvolvido (em linguagem C) se trata de uma simulação de uma rede de relacionamentos, no qual apenas um usuário faz uso da rede, adicionando perfis fictícios e conectando perfis entre si para criar amizades. Quando dois perfis estão conectados, o usuário pode executar uma ação no qual ele digita uma mensagem e informa qual é o remetente e qual é o destinatário da mensagem (simulando o envio de uma mensagem de um perfil para o outro).

O usuário também poderá executar uma ação que irá mostrar o histórico de mensagens entre dois perfis ao informar o email dos dois. Para cada mensagem, será mostrado junto o nome do perfil que enviou a mensagem, como se fosse o histórico de um chat. Também será possível buscar e mostrar todas as informações sobre um perfil cadastrado, e para isso é necessário informar o e-mail do perfil para executar esta ação. A última funcionalidade do programa será listar todas as amizades de um determinado perfil, e ao executar este comando, é necessário informar o e-mail do perfil em questão.

Nenhuma informação inserida no sistema, seja um novo perfil, nova amizade ou nova mensagem, poderá ser excluído do programa após ser cadastrada. O programa não fará persistência dos dados inseridos, ou seja, após finalizar a sua execução, todas as informações cadastradas serão perdidas.

As ações que podem ser executadas pelo usuários estarão disponíveis através de um menu (especificado neste documento), que irá guiá-lo durante o uso do programa.

PARTE 1: REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

LEGENDA

[RF]: requisito funcional

[RNF]: requisito não funcional

I. PERFIL

1. [RF] Deve ser possível criar um perfil contendo as seguintes informações:

- nome[caracteres]: obrigatório
- data de nascimento [caracteres]: obrigatório
- sexo [caractere 'F' ou 'M']: obrigatório
- cidade [caracteres]: obrigatório
- e-mail [caracteres]: obrigatório

1.1 [RF] A informação "e-mail" deve ser única para cada perfil criado.

1.2 [RF] Se houver algum campo cadastrado de forma incorreta, o programa deve avisar ao usuário que ocorreu um erro ao cadastrar o perfil como resultado. O cadastro nesse caso não é concluído e o usuário deve solicitá-lo novamente.

1.3. [RNF] O limite para os campos de cadeia de caracteres é de 100 caracteres.

1.4 [RF] Deve ser avisado que o perfil foi cadastrado com sucesso.

1.5 [RNF] O nome pode ser apenas o primeiro nome ou o nome completo.

1.6 [RNF] O programa não irá verificar se o e-mail existe ou se o formato é válido.

2. [RNF] Não deve ser possível remover um perfil. O usuário que gerencia todos os perfis não tem a opção de remover um perfil já criado.

3. [RNF] Não deve ser possível editar um perfil. O usuário que gerencia todos os perfis não tem a opção de editar um perfil.

4. [RF] Deve ser possível criar uma amizade entre dois perfis.

4.1 [RF] As seguintes informações são necessárias para conectar os dois perfis: e-mail do perfil 1 e e-mail do perfil 2

4.2 [RNF] A amizade é criada automaticamente, ou seja, não é necessário nenhum tipo de aprovação.

4.3 [RF] Uma mensagem de sucesso ou insucesso da criação de amizade será mostrada na tela como resultado.

5. [RNF] Não deve ser possível desfazer uma amizade. O usuário que gerencia todos os perfis não tem a opção de excluir uma amizade.

6. [RF] Deve ser possível buscar e visualizar as informações de um perfil.

6.1 [RF] É necessário digitar apenas o número de e-mail do perfil para buscar suas informações.

6.2 [RF] Todas as informações cadastradas no perfil serão exibidas na tela.

7. Deve ser possível buscar e visualizar a lista de amizades de um perfil.

7.1 [RF] É necessário digitar apenas o e-mail do perfil para buscar sua lista de amigos.

7.2 [RF] Para cada amigo do perfil, deve ser exibido: nome, e-mail.

II. MENSAGENS

1. [RF] Deve ser possível um perfil enviar uma mensagem para outro perfil, desde que exista um relacionamento entre ambos.

1.1 [RF] As seguintes informações são necessárias para enviar a mensagem: e-mail do perfil 1 (remetente), texto da mensagem [caracteres], e-mail do perfil 2 (destinatário).

1.2 [RNF] Deve haver um identificador único [número inteiro] para cada mensagem enviada.

1.3 [RF] Caso os perfis não sejam amigos, uma mensagem de erro aparece como resultado na tela.

2. [RF] Deve ser possível carregar o histórico de mensagens recebidas e enviadas entre dois perfis, ordenadas da mais recente para a mais antiga.

2.1 [RF] As seguintes informações são necessárias para carregar o histórico de mensagens: e-mail do perfil 1 e e-mail do perfil 2

2.2 [RF] Cada mensagem visualizada na tela deve seguir o seguinte padrão: "[nome do perfil remetente]:", "[texto da mensagem]", linha em branco

2.3 [RF] Caso não haja mensagens entre os perfis ou eles não sejam amigos, nada será exibido.

3. [RNF] Não deve ser possível excluir mensagens enviadas/recebidas.

III. INTERFACE

1. [RF] Deve ser mostrado um menu único com todas as opções de ação do programa, identificados por um número, da seguinte forma:

- (1) Criar novo perfil
- (2) Criar uma nova amizade
- (3) Enviar uma mensagem
- (4) Carregar histórico de mensagens
- (5) Buscar perfil
- (6) Listar amigos de perfil

1.1 [RF] O usuário deverá entrar com o número da opção acessada para acessá-la

1.2 [RF] Após selecionar uma opção, o usuário entra com os dados necessários (solicitados na tela) para a opção escolhida e após a resposta do programa, o menu inicial volta a aparecer na tela e aguarda a escolha de uma nova solicitação.

1.3 [RF] Após uma ação ser concluída, o resultado (especificado nos requisitos anteriores) daquela ação é mostrado na tela antes de voltar ao menu.

IV. MODERADOR

1. [RNF] A rede de relacionamentos possui um único usuário, que é o moderador que insere as informações no programa

1.1 [RNF] O moderador não assume o "papel" de nenhum perfil, ou seja, o usuário que está interagindo com o programa não entra em um perfil específico para adicionar um amigo, enviar uma mensagem ou qualquer outra ação que o programa possa fazer.

V. CORRETUDE

1. [RNF] Todos os módulos devem ser testados individualmente, utilizando o arcabouço de testes fornecido junto com uma boa massa de testes, onde cada função dos módulos é testada especificando-se as condições de entrada e saída, bem como as condições de estado do módulo.

2. [RNF] O cliente deve assegurar a validade sintática e semântica dos dados transmitidos ao servidor, assim como o servidor deve garantir o mesmo dos dados retornados ao cliente.

VI. REUSO

[RNF] De forma a acelerar o processo de reutilização de projeto e implementação (e teste), deve-se maximizar a reutilização de módulos. Para isso, os módulos desenvolvidos de Lista, Grafo, Perfil e Mensagem são genéricos, podendo ser aplicados em outras soluções. O módulo lista pode utilizar qualquer tipo de valor para os seus elementos, assim como o grafo.

VII. MANUTENIBILIDADE

[RNF] Todas funções e módulos deverão ser desenvolvidos utilizando padrões de documentação,, garantindo assim que o programa seja de fácil manutenção. Todas as funções de cada módulo deverão ser documentadas, explicitando entradas e possíveis saídas. A nomeação das funções e condições de retorno deverão seguir os padrões de sintaxe definidos.

VIII. ROBUSTEZ

[RNF] O programa deve ser resistente à entradas incorretas do usuário (regras definidas anteriormente controlam essas entrada).

PARTE 2: REQUISITOS TÉCNICOS

GRAFO

Um grafo é formado por dois conjuntos. Um conjunto de elementos chamados vértices e um conjunto de elementos chamados arestas. Cada aresta está associada a um par de vértices.

1. Cada perfil será representado por um vértice do grafo.

1.1 Cada vértice será armazenado em uma lista com somente um nó, a Lista Vértice

1.2 Este nó guardará todas as informações referentes ao perfil.

1.3 O nó também guardará uma referência para a Lista Arestas deste vértice, especificada no item 3.1 .

2. Todos os vértices do grafo (armazenados como descrito em 1.1) serão referenciados por nós de uma nova lista, a Lista Vértices.

2.1 A Lista Vértices ancora todos os vértices existentes, representando todas as possíveis origens do grafo.

3. Cada relacionamento entre perfis será representado por uma aresta entre vértices do grafo.

3.1 Estas arestas serão armazenadas em uma lista, a Lista Arestas.

3.2 Nesta lista (3.1), cada nó possuirá uma referência para os outros vértices com quem este possui uma ligação (aresta).

4. Todas as listas do grafo, incluindo Lista Vértice, Lista Vértices e Lista Arestas, possuem uma estrutura Cabeça de Lista.

4.1 Esta estrutura guarda uma referência para o início da lista além de uma referência para o nó corrente.

5. Cada vértice (perfil) pode ter arestas conectadas a nenhum ou todos os vértices.

5.1 No caso de um vértice com nenhuma aresta, a sua Lista Arestas será uma lista vazia.

6. Um vértice (perfil) não pode estar conectado (possuir uma aresta) com ele mesmo.

7. O Grafo possuirá uma estrutura principal, a Cabeça do Grafo.

7.1 Esta estrutura guardará a referência para a Lista Vértices, onde é possível acessar todos os vértices do grafo.

7.2 Também guardará uma referência para o vértice corrente do grafo.

