

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Fancybook

André Pessanha, Bianca Fragoso, Bianca Gomes

Dezembro 2017

Orientador: Prof. Alessandro Garcia

Resumo

Este documento descreve o desenvolvimento de um sistema na linguagem C que consiste num gerenciador de uma rede de relacionamentos chamada Fancybook.

Contents

1	Especificação dos Requisitos	4
1.1	Requisitos Funcionais:	4
1.2	Requisitos Não-Funcionais:	6
2	Modelo Conceitual da Arquitetura	7
2.1	Visão Expandida das Interfaces	8
3	Modelo Físico	9
3.1	Exemplo Modelo Físico	10
4	Assertivas Estruturais	11
4.1	Lista	11
4.2	Grafo	12
4.3	Grafo com instrumentação	12
5	Modelo Físico do Grafo Instrumentado	13
6	Modelo de Interação: MoLIC	14

1 Especificação dos Requisitos

1.1 Requisitos Funcionais:

1. A rede de relacionamentos será representada por um grafo conexo não dirigido. Um grafo é um conjunto de nós interligados, onde cada nó é chamado de vértice e cada interligação é chamada de aresta. Na rede, cada vértice representará um usuário e cada aresta representará um relacionamento entre usuários.
2. Cada usuário deve informar obrigatoriamente: Id, nome, data de nascimento, email e cidade de moradia. O usuário é uma pessoa que utilizará a rede de relacionamentos e essas informações devem ser fornecidas no momento da inclusão.
 - 2.1. O Id será o identificador de cada usuário. Não será permitido mais de um usuário com o mesmo Id.
 - 2.2. A entrada do email terá um limite de 50 caracteres e será validado através de uma expressão regular.
 - 2.3. Para serem válidos, o Id, nome e a cidade de moradia devem ser fornecidos em caracteres alfabéticos maiúsculos e terão um limite de 50 caracteres cada. Com exceção do Id, nomes compostos e sobrenomes devem ser separados por um único espaço.
 - 2.4. A entrada da data de nascimento deve ser fornecida no formato DD/MM/AAAA.
3. A interface da rede de relacionamentos será um menu contendo opções para o usuário interagir e navegar pela rede.
 - 3.1. Após selecionar a opção "Sair", a aplicação será encerrada.
4. A rede permite incluir, alterar e remover usuários, assim como seus relacionamentos.
 - 4.1. Após selecionar a opção "Incluir usuario" e fornecer os dados necessários citados em [2], o usuário será incluso na rede de relacionamentos.
 - 4.2. Após selecionar a opção "Alterar usuario" e selecionar um id, será possível alterar os seguintes dados: Nome, Data de nascimento, email e cidade de moradia. Não será possível alterar o id.

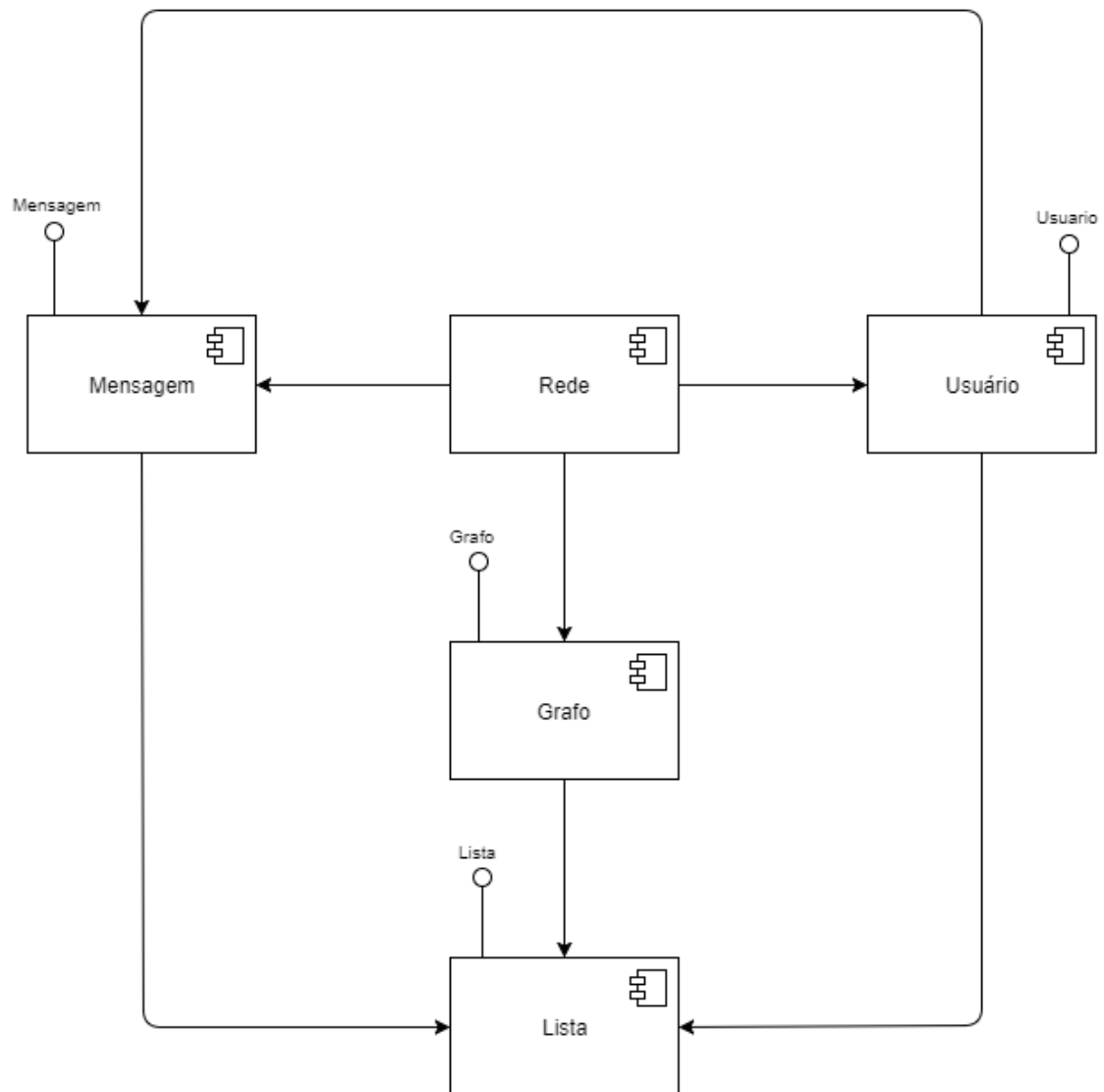
- 4.3. Após selecionar a opção "Remover usuario" e informar um id. O usuário será removido da rede, assim como seus relacionamentos.
- 4.4. Após selecionar a opção "Remover relacionamento" e informar dois ids de usuário. O relacionamento entre ambos será removido.
- 5. A rede possibilita a troca de mensagem entre usuários. Cada usuário poderá enviar mensagem para um usuário específico ou para todos os usuários da rede. Não será permitido que um usuário envie mensagem para si mesmo.
 - 5.1. Após selecionar a opção "Enviar mensagem" seguido da opção "Para um" será solicitado o id do usuário destino e logo em seguida será solicitado ao usuário remetente fornecer o texto da mensagem.
 - 5.2. Após selecionar a opção "Enviar mensagem" seguido da opção "Para todos" será solicitado o texto da mensagem e a mesma será enviada à todos os usuários da rede.
 - 5.3. Após selecionar a opção "Enviar mensagem" seguido da opção "Cancelar mensagem" a mensagem será cancelada.
 - 5.4. Cada mensagem terá um limite máximo de 300 caracteres.
- 6. A rede permite que um usuário envie solicitações de amizade para outros usuários que não estejam presentes em sua lista de amigos. Não será possível enviar a solicitação para si mesmo.
 - 6.1. Após selecionar a opção "Incluir relacionamento" e fornecer dois ids de usuário. Será estabelecido um novo relacionamento entre ambos.
- 7. A rede permite exibir todos os dados de um usuário específico.
 - 7.1. Após selecionar a opção "Exibir dados usuario" e fornecer um id, será exibido todos os dados correspondentes ao usuário selecionado.
- 8. A rede permite exibir as mensagens entre dois usuários e os relacionamentos de um usuário específico.
 - 8.1. Após selecionar a opção "Exibir mensagens" e fornecer dois ids, será exibido todas as mensagens trocadas entre ambos.

- 8.2. Após selecionar a opção "Exibir relacionamentos" e fornecer um id, será exibido o id de todos os usuários que possuem relacionamento com o usuário fornecido.
9. Todos os dados de entradas fornecidos no momento da inclusão e da alteração serão validados pela rede, assim como as opções de seleção e navegação da interface da rede. Caso alguma entrada não seja válida, uma mensagem será exibida informando que o dado é inválido e será possível digitar novamente.

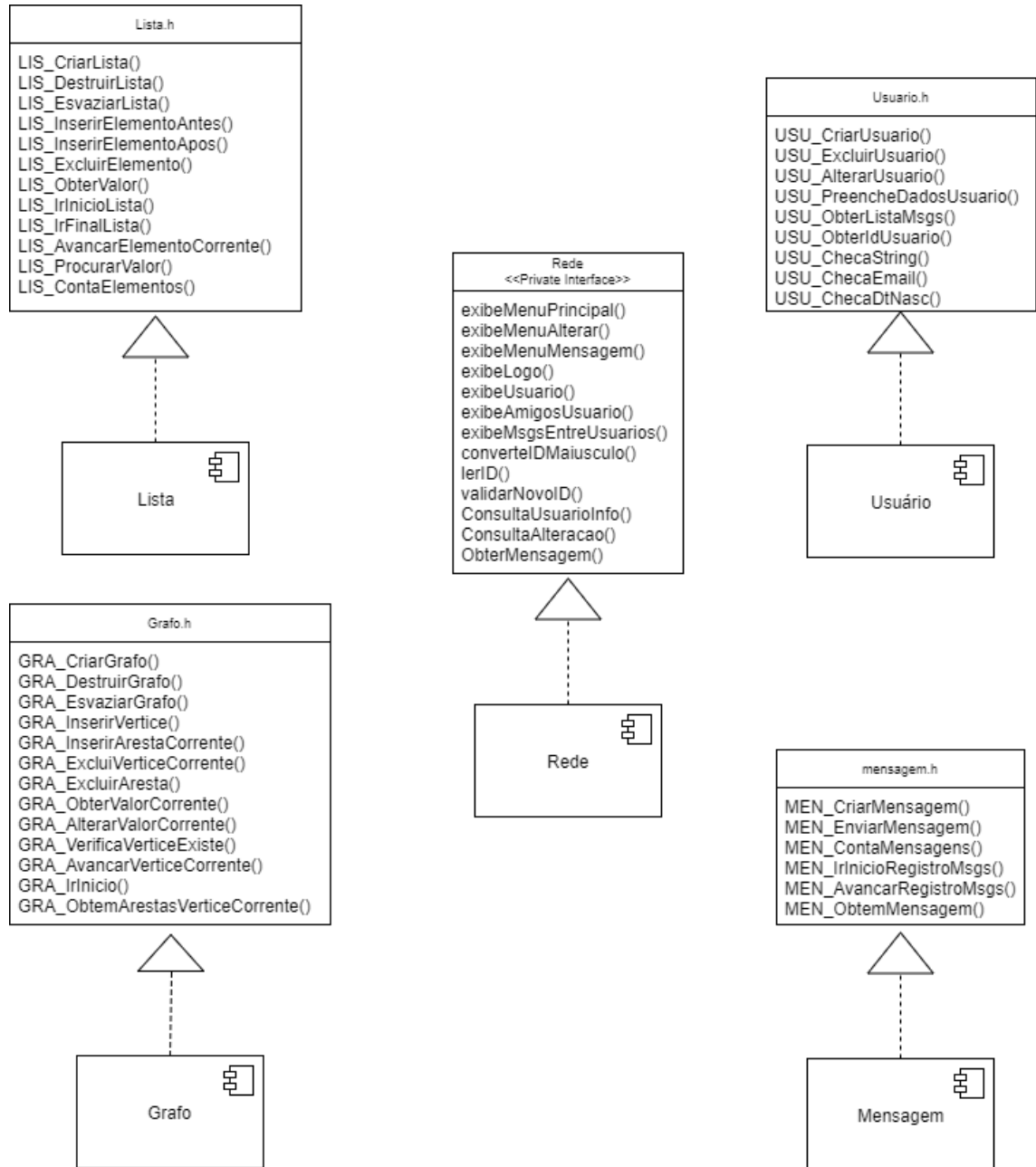
1.2 Requisitos Não-Funcionais:

1. O sistema deve ser implementado na linguagem de programação C.
2. A aplicação deve ser desenvolvida para sistema Windows 10.
3. A aplicação poderá ser utilizada 24h por dia e 7 dias por semana.
4. A interação com o usuário será feita através de um prompt de comando que representará a interface da rede de relacionamentos e os dados serão inseridos pelo teclado.
5. Os dados da rede serão armazenados em memória e vão persistir enquanto a aplicação estiver em execução.
6. A navegação pela interface da rede, assim como o processamento de cada solicitação do usuário deve ser executado em no máximo 4 segundos.
7. Com o auxílio do Arcabouço de teste automatizado, todos os módulos serão testados individualmente, cada função dos módulos será testada em diferentes circunstâncias garantindo a confiabilidade do sistema.
8. Os módulos e as funções devem ser desenvolvidos utilizando os padrões de faltas e documentação, apresentados no curso, garantindo assim que o programa seja de fácil entendimento e manutenção.
9. Com o objetivo de acelerar o processo de reutilização de projeto, implementação e teste, deve-se maximizar a reutilização de módulos. Serão reutilizados alguns módulos do arcabouço, como por exemplo o módulo Lista, **CESPDIN**, **CONTA**, etc.

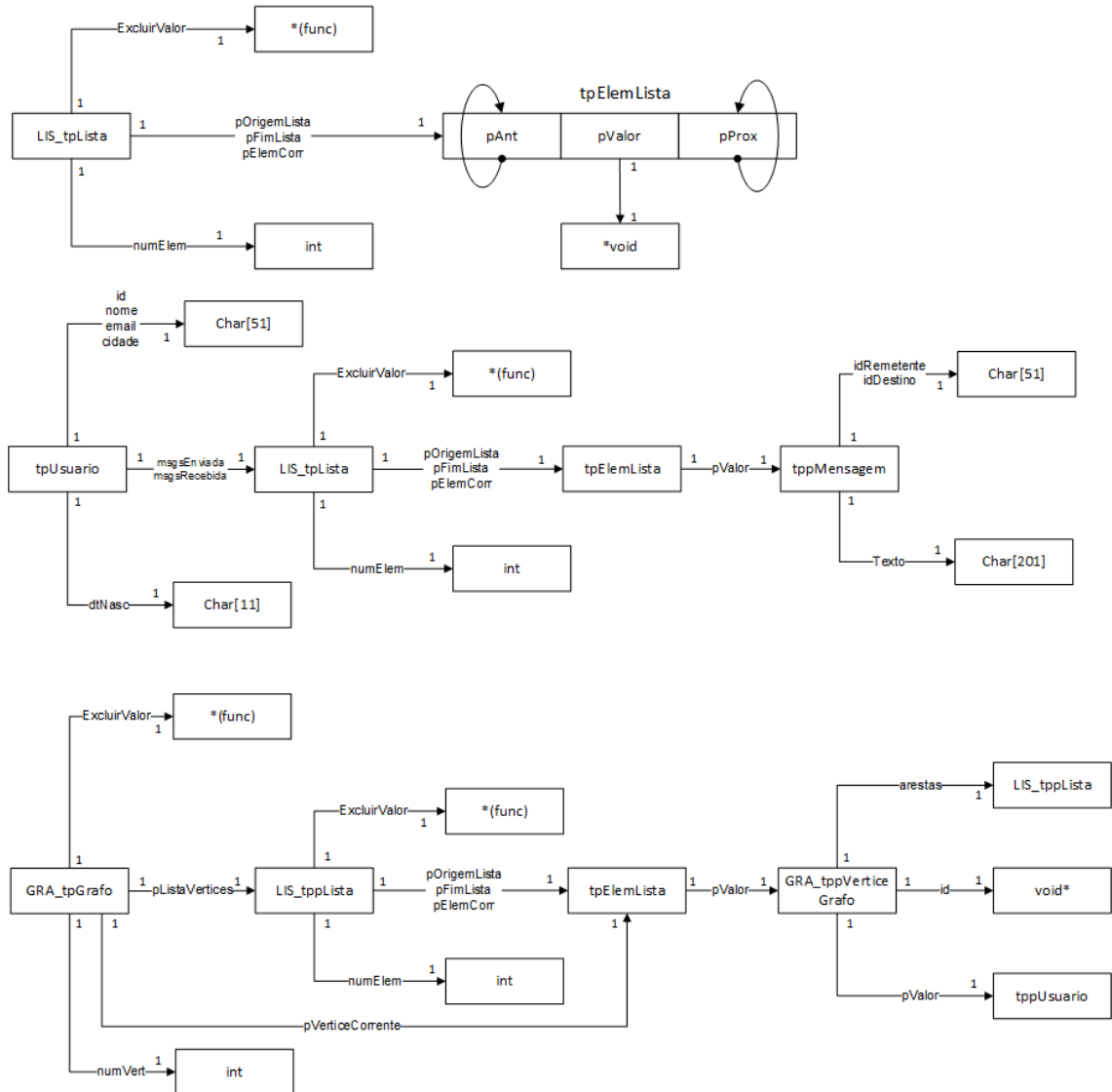
2 Modelo Conceitual da Arquitetura



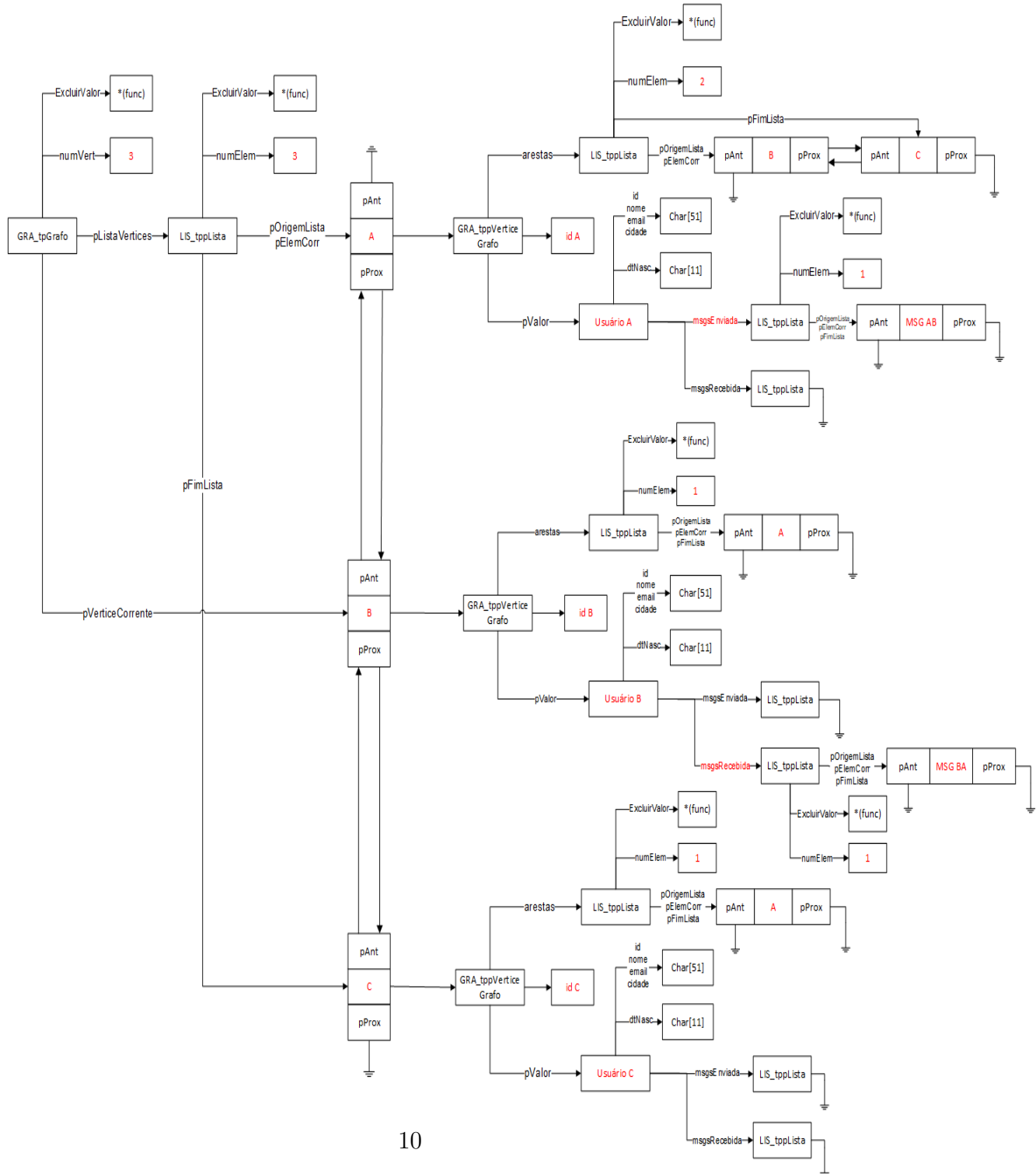
2.1 Visão Expandida das Interfaces



3 Modelo Físico



3.1 Exemplo Modelo Físico



4 Assertivas Estruturais

4.1 Lista

Seja lista um ponteiro para estrutura cabeça da lista

Seja no um ponteiro para estrutura nó da lista

Se $no \rightarrow pProx \neq NULL$, então $no \rightarrow pProx \rightarrow pAnt == no$

Se $no \rightarrow pAnt \neq NULL$, então $no \rightarrow pAnt \rightarrow pProx == no$

Se $lista \rightarrow numElem == 0$, então

- $lista \rightarrow pElemCorr == NULL$
- $lista \rightarrow pOrigemLista == NULL$
- $pFimLista == NULL$

Se $lista \rightarrow numElem > 0$, então $lista \rightarrow pElemCorr \neq NULL$

Se $lista \rightarrow numElem == 1$, então

- $lista \rightarrow pElemCorr == lista \rightarrow pOrigemLista$
- $lista \rightarrow pOrigemLista == lista \rightarrow pFimLista$

4.2 Grafo

Seja grafo um ponteiro para estrutura cabeça do grafo

Seja vertice ponteiro para estrutura vértice do grafo

Se grafo->pListaVertices->numElem > 0, então grafo->pVerticeCorrente != NULL

grafo->pVerticeCorrente sempre irá pertencer a grafo->pListaVertices

Se grafo->pListaVertices->numElem == 1, então:

grafo->pVerticeCorrente == grafo->pListaVertices->pOrigemLista

Se grafo->numVert == 0, então:

- grafo->pVerticeCorrente == NULL
- grafo->pListaOrigens == NULL

Se grafo->numVert > 0, então grafo->pVerticeCorrente != NULL

4.3 Grafo com instrumentação

Seja pGrafo um ponteiro para estrutura cabeça do grafo

Seja pVertice ponteiro para estrutura vértice do grafo

Se pGrafo->numVert > 0 então pGrafo->pVerticeCorrente != NULL

Se pGrafo->numVert == 0 então pGrafo->tamTotal == 0

Se pGrafo->numVert == 1 então pGrafo->tamTotal == pGrafo->pVerticeCorrente->tamValor

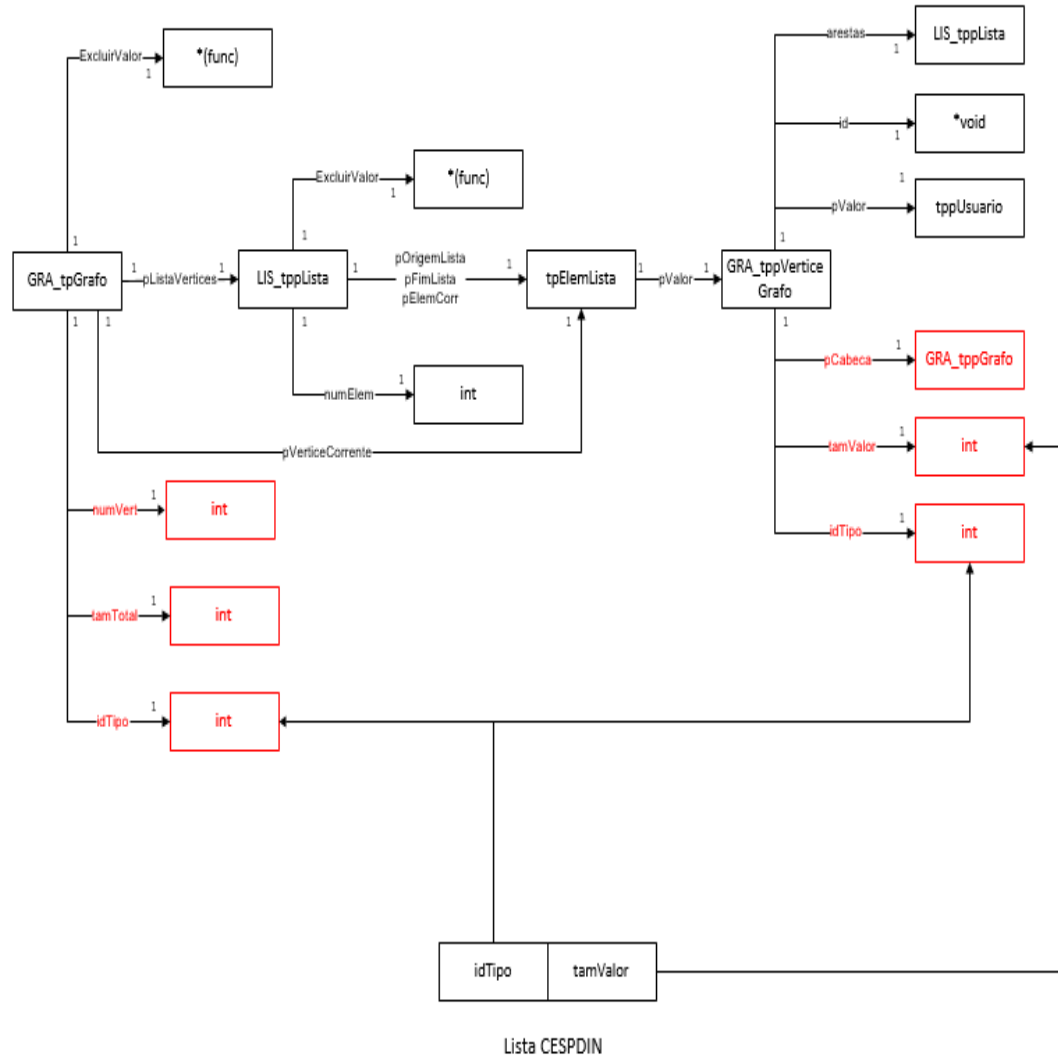
Se pGrafo->numVert > 1 então pGrafo->tamTotal == \sum pGrafo->pVerticeCorrente->tamValor (Somatório do tamValor de todos o vértices do Grafo)

Se pGrafo->pVerticeCorrente != NULL então pGrafo->pVerticeCorrente->pCabeca == pGrafo

Se pGrafo != NULL então pGrafo->idTipo == GRA_TipoEspacoCabeca

Se pGrafo->pVerticeCorrente != NULL então pGrafo->pVerticeCorrente->idTipo == GRA_TipoEspacoVertice

5 Modelo Físico do Grafo Instrumentado



6 Modelo de Interação: MoLIC

