Arquitetura de Módulos

Trabalho 4 - Rede de Relacionamentos INF 1301 - PUC-RIO

Grupo:
Gabriel Cantergiani
Gabrielle Bradenburg
Wellington Bezerra

Modelo de Interface INTERFACE Componentes Grafo Controle Perfil PERFIL GRAFO CONTROLADOR Mensagem Lista MENSAGEM LISTA

Interfaces

CONTROLADOR **PERFIL MENSAGEM** LISTA **GRAFO INTERFACE** CON criarPerfil() PER criaPerfil() MEN CriarMensagem() GRA CriarGrafo() LIS CriarLista() INT MenuPrincipal() CON buscarPerfil() PER destruirPerfil() MEN ObterTexto() LIS DestruirLista() GRA DestruirGrafo() INT MostrarMensagem() CON MostrarPerfil() PER compararPerfil() MEN ObterID() LIS InserirElemAntes() GRA IrVertice() INT MostrarPerfil() CON ExcluirPerfil() PER_mostrarPerfil() MEN ObterRemetente() LIS_InserirElemApos() GRA_ObterValor CON AlterarPerfilNome() PER obterEmail() MEN_ObterDestinatario() LIS_ExcluirElemento() GRA CriarVertice() CON_AlterarPerfilCidade() PER obteNome() MEN CompararMensagem() LIS ObterValor() GRA excluirVertCorr() CON AlterarPerfilNasc() PER obterCidade() MEN_DesativarRemetente() LIS IrInicioLista() GRA_CriarAresta() CON AlterarPerfilGenero() PER obterNascimento() MEN DesativarDestinatario() LIS IrFinalLista() GRA excluirAresta() CON CriarAmizade() PER_obterGenero() LIS_AvancarElemCorr() GRA IrVizinho() CON ExcluirAmizade() PER AlterarNome() LIS_ProcurarPorConteudo() CON_BuscarAmizades() GRA AvancarVizinho() PER AlterarCidade() LIS_EsvaziarLista() GRA_AlterarVertCorr() CON EnviarMensagem() PER_AlterarNascimento() LIS ProcurarValor() CON_CarregarHistorico() PER_AlterarGenero() PER_EnviarMensagem() PER BuscarMsgEnviada() PER BuscarMsgRecebida() **PERFIL MENSAGEM** CONTROLADOR **GRAFO INTERFACE** LISTA

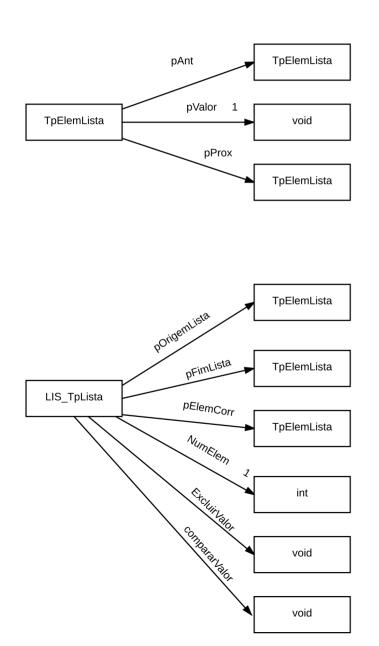
Modelagem Física

Trabalho 4 - Rede de Relacionamentos INF 1301 - PUC-RIO

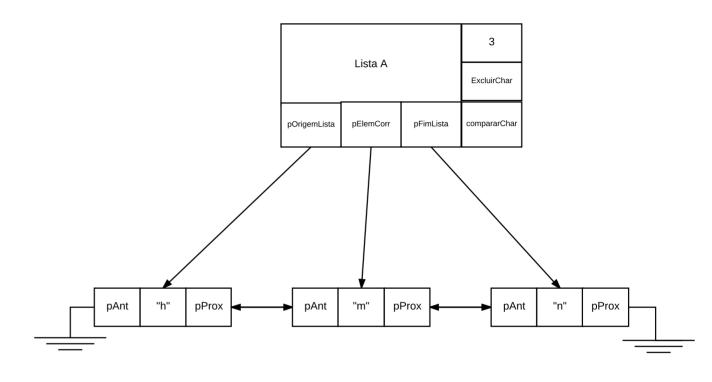
Grupo:
Gabriel Cantergiani
Gabrielle Bradenburg
Wellington Bezerra

LISTA DUPLAMENTE ENCADEADA

Modelo Físico



Exemplo Físico

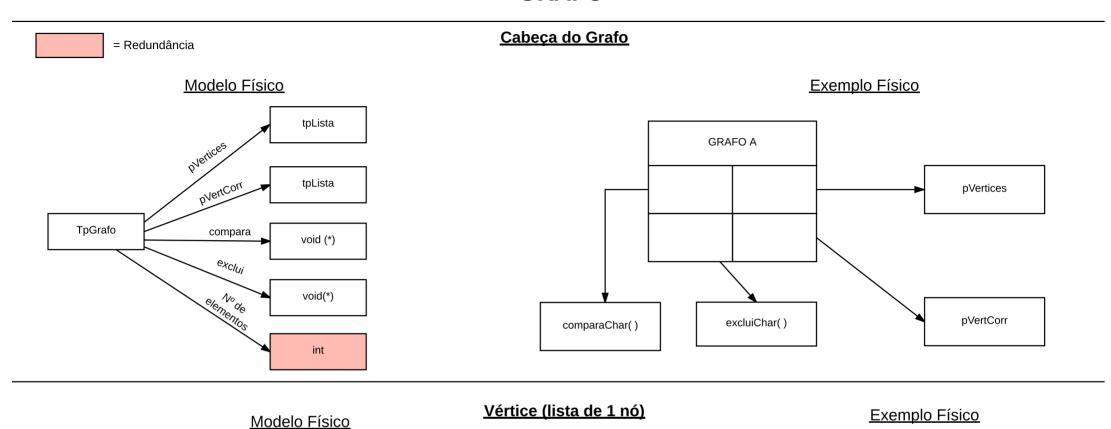


Assertivas

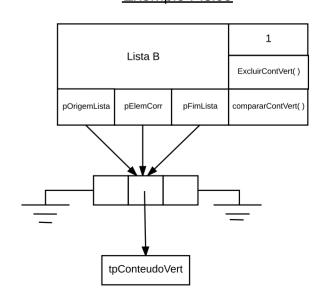
Para todo elemento Elem da lista:

- se Elem-> pAnt != NULL, então Elem -> pAnt -> pProx == Elem
- se Elem -> pProx != NULL, então Elem -> pProx -> pAnt == Elem
 - se Elem é o primeiro da lista, então Elem -> pAnt == NULL
 - -se Elem é o último da lista, então Elem -> pProx == NULL

GRAFO



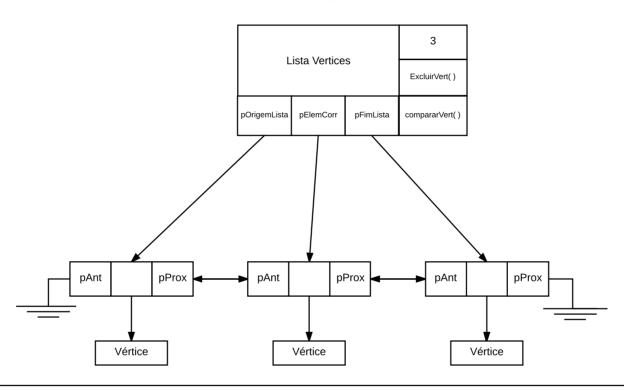
pElemCorr tpElemLista pFimLista numElem 1 int TpElemLista pValor 1 tpConteudoVert tpConteudoVert void (*)

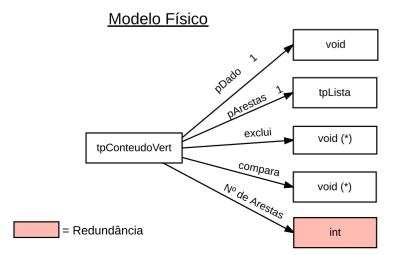


GRAFO (continuação)

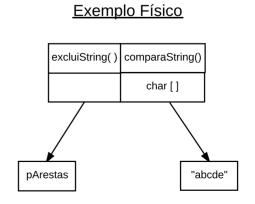
Lista Vértices

Exemplo Físico





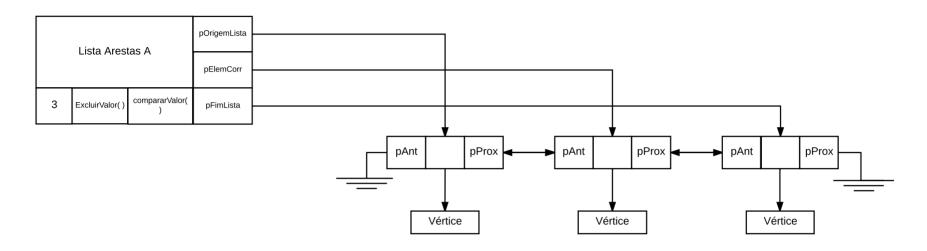
Conteúdo do Vértice



GRAFO (continuação)

Exemplo Físico

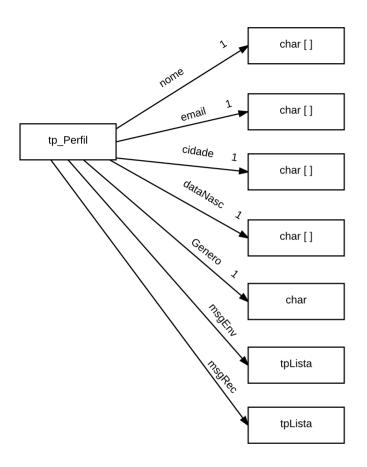
Lista Arestas



<u>Assertivas de Grafo</u>

- Se a Lista Vértices é não vazia, então tpGrafo -> pVerticeCorrente != NULL .
- -Se pVertices -> numElem ==1 , então tpGrafo -> pVerticeCorrente == tpGrafo -> pOrigem -> pElemCorr .
- Se ElemLista == tpGrafo -> pVerticeCorrente , então ElemLista pertence à tpGrafo > pVertices .
- Se a Lista Vértices é não vazia, então sempre existe um elemento (ElemLista) pertencente à Lista Vértices tal que tpGrafo -> pOrigem == ElemLista .
- Se a lista de arestas do vértice X possui um elemento cujo valor é o vértice Y, então a lista de arestas do vértice Y deve possuir um elemento cujo valor é o vértice X. (Grafo não-dirigido)
- A lista de arestas do Vértice X NÃO PODE conter um elemento cujo valor é o próprio elemento X
- -O valor armazenado no vértice deve ser do mesmo tipo para todos os vértices

Modelo Físico

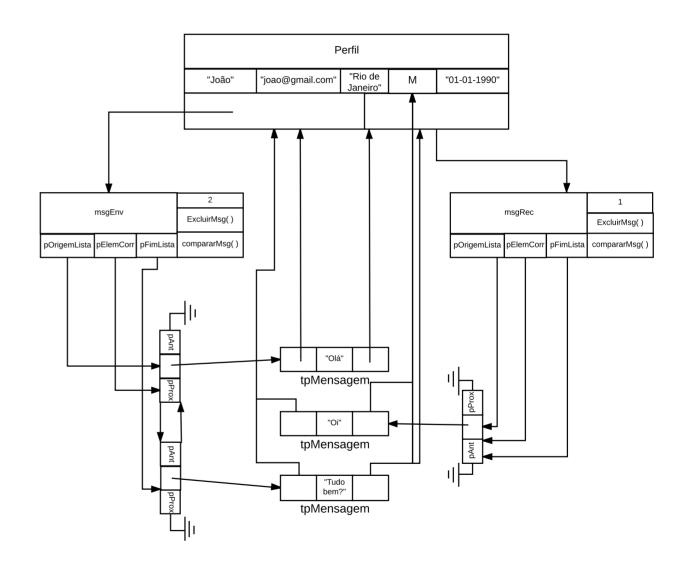


Assertivas de PERFIL

- Para todos os perfis cadastrados no sistema, não pode haver dois perfis com a mesma string no campo "email".
- Nenhum dos campos da estrutura Perfil pode ter valor nulo.
 - Nenhuma das strings pode ser vazia.

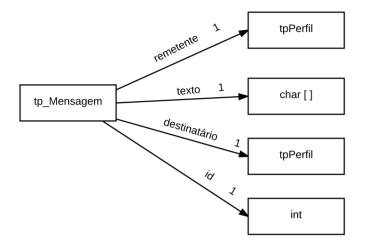
PERFIL

Exemplo Físico



MENSAGEM

Modelo Físico



Assertivas de MENSAGEM

- Duas mensagens não podem ter o mesmo identificador (id)
- O id de uma mensagem de ser maior que 1.
- -Os valores Remetente e Destinatário devem ser diferentes de NULL
- O campo texto de ser diferente de NULL e a string não pode ser vazia.

Exemplo Físico

