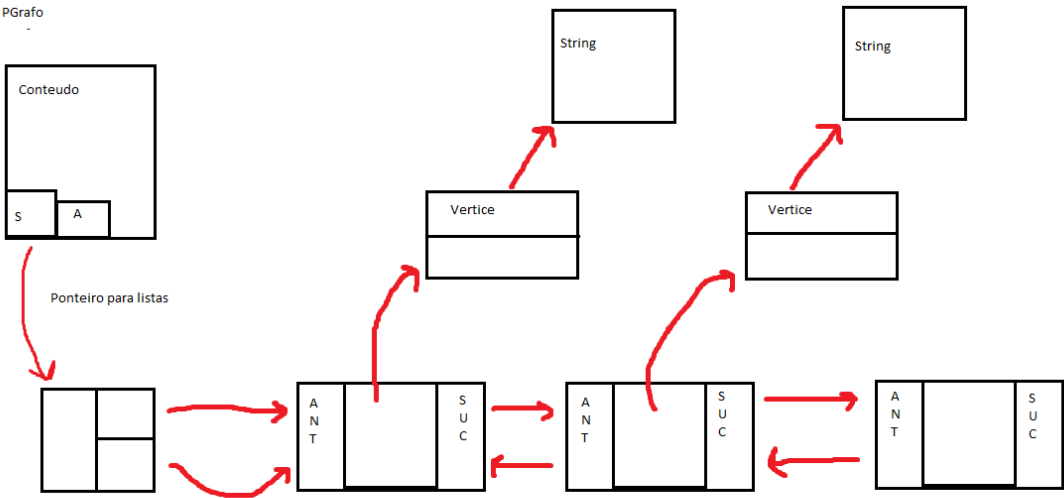
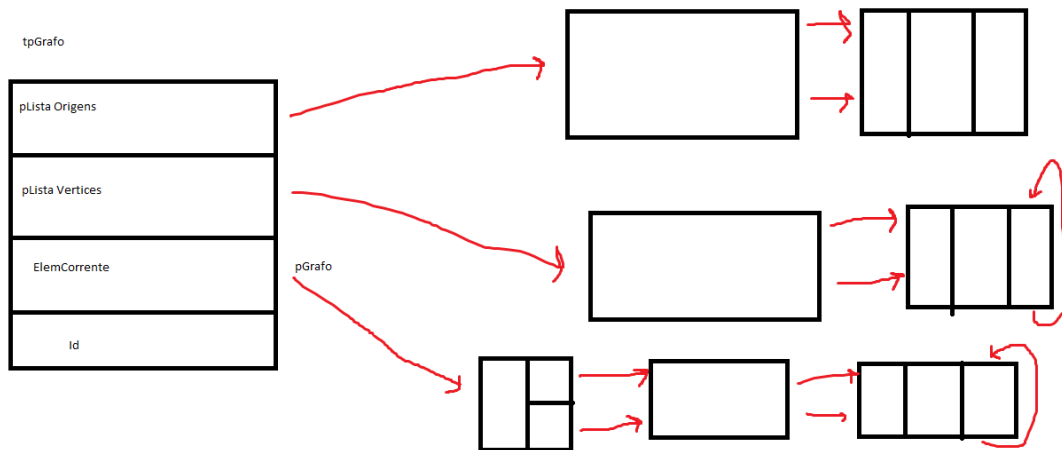


Modelo Físico





### Assertivas Estruturais

Lista: Se  $\text{Nó} \rightarrow \text{Próximo} \neq \text{NULL}$ , então  $\text{Nó} \rightarrow \text{Próximo} \rightarrow \text{Anterior} = \text{Nó}$ .

Se  $\text{Nó} \rightarrow \text{Anterior} \neq \text{NULL}$ , então  $\text{Nó} \rightarrow \text{Anterior} \rightarrow \text{Próximo} = \text{Nó}$ .

Se um Nó for o primeiro da lista, então  $\text{Nó} \rightarrow \text{Anterior} = \text{NULL}$

Se um Nó for o último da lista, então  $\text{Nó} \rightarrow \text{Próximo} = \text{NULL}$

Se um Nó não for nem o primeiro nem o último da lista, então  $\text{Nó} \rightarrow \text{Anterior} \neq \text{NULL}$  e  $\text{Nó} \rightarrow \text{Próximo} \neq \text{NULL}$ .

### Grafo:

Se um vértice tem lista de arestas vazia, ele não deve constar em nenhuma lista de arestas.

Todo vértice contém um, e somente um, valor.

Todo vértice deve estar na lista de vértices do grafo.

Cada aresta na lista de arestas de um vértice pode estar ligada somente a um vértice.

Um vértice pode ter mais de um sucessor.

Um vértice pode ter mais de um antecessor.

Um vértice pode estar ligado a nenhuma aresta.

Assertivas de entrada:

GRA\_criarGrafo: Ponteiro válido para cabeça do grafo

GRA\_criarAresta: Ponteiro válido para grafo, id válido para a aresta e uma string de tipo

GRA\_irVertice: Ponteiro válido para grafo e id válido

GRA\_obterCorr: Ponteiro válido para grafo e string válida

GRA\_destruirGrafo ( GRA\_tppGrafo parmGrafo )

GRA\_destruirAresta ( GRA\_tppGrafo Grafo, `int` antecessor, `int` sucessor )

VER\_CriarVertice : Id válido

VER\_InserirValor : Ponteiro válido para vértice e uma string válido

VER\_DestroyVertice : Ponteiro válido para vértice

VER\_ObterValor : Ponteiro válido para vértice e uma string válido

VER\_ObterID : Ponteiro válido para vértice

Assertivas de saída:

GRA\_criarGrafo: Cabeça do grafo criada com sucesso

GRA\_criarAresta: Lista de sucessores e antecessores atualizada

GRA\_irVertice: Vértice corrente atualizado

GRA\_obterCorr: Valor corrente obtido

GRA\_destruirGrafo ( GRA\_tppGrafo parmGrafo )

GRA\_destruirAresta ( GRA\_tppGrafo Grafo, `int` antecessor, `int` sucessor )

VER\_CriarVertice : Ponteiro válido para vértice

VER\_InserirValor : Ponteiro válido para vértice com a string inserida

VER\_DestroyVertice : Memória do ponteiro liberada

VER\_ObterValor : String com nome do vértice recebido

VER\_ObterID : Id do vértice retornado