

**UNIVERSIDADE FEDERAL DE SÃO CARLOS
CAMPUS SÃO CARLOS
JOÃO VITOR BRANDÃO MOREIRA - 407496
LUCAS OLIVEIRA DAVID – 407917
THIAGO FARIA NOGUEIRA - 407534**

DOCUMENTAÇÃO TRABALHO 2 - ESTRUTURAS DE DADOS

SÃO CARLOS – SP
2012

1. INTRODUÇÃO

Este trabalho consiste na criação e teste das seguintes estruturas de dados estudadas ao longo do curso: fila, lista duplamente encadeada com header, cadastro e árvore binária de busca. Desde o início, definimos como objetivo a implementação por meio de Orientação a Objetos, o que nos direcionou para a linguagem de programação C++. É interessante frisar, desde já, que todas as estruturas anteriores possuem um elemento em comum: o nó. Além disso, todas as classes foram implementadas utilizando *template* afim de criar estruturas de dados genéricas.

2. ESTRATÉGIA DE IMPLEMENTAÇÃO

Criamos primeiramente a classe *Node*, isto é, Nó, que tem os seguintes atributos: **T item**; **Node *esq**; **Node *dir**, e definimos operações de construção (com e sem parâmetro) e destruição. Tal classe seria, a partir daí, utilizada por todas as outras implementações (listas e árvores, por exemplo).

Fizemos a lista duplamente encadeada com header, também tendo em mente a reutilização de código. Essa estrutura não tem regras para inserção ou remoção, portanto fizemos métodos de inserção e remoção genéricos, além de métodos para calcular o tamanho da estrutura, imprimir e fazer busca.

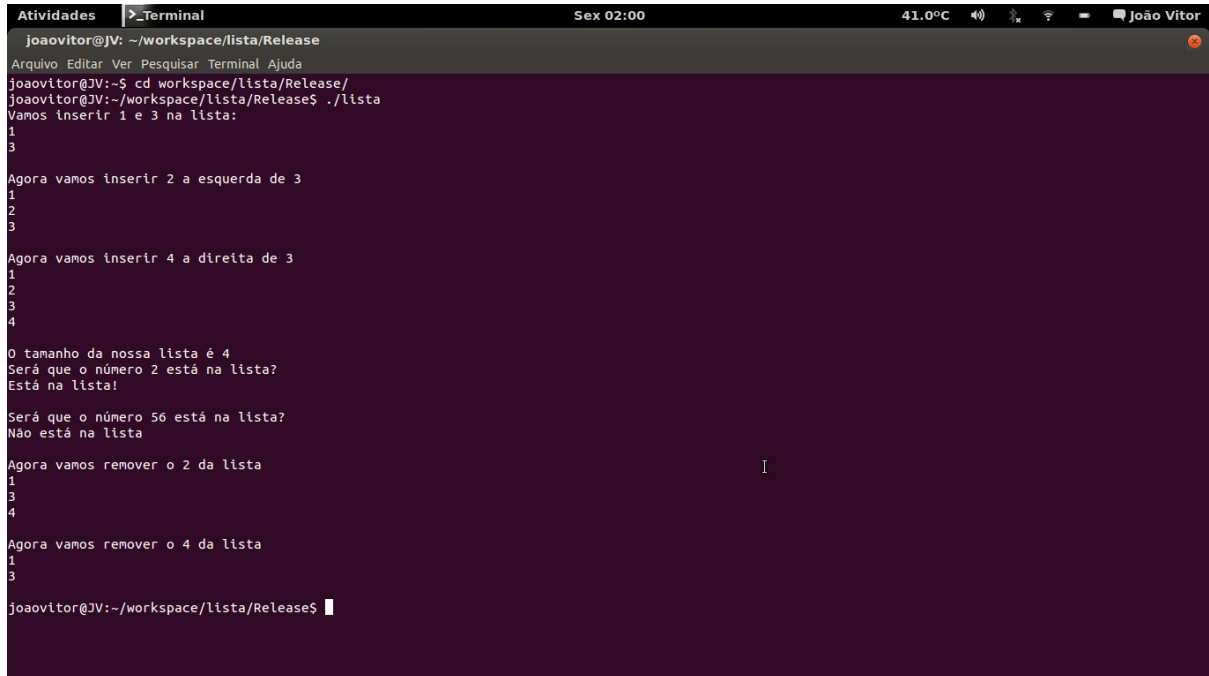
As classes Cadastro e Lista herdam de forma privada da classe Lista. Mas porque? Justamente para poder utilizar os métodos da Lista para implementar as operações inerentes à cada classe sem que os métodos da Lista fiquem acessíveis fora destas classes. Em outras palavras, utilizamos métodos da Lista para construir métodos do Cadastro e da Fila, mas os métodos da Lista não podem ser chamados fora do escopo destas classes. Assim o conceito de TAD não é quebrado.

A classe *Tree* (que é, essencialmente, uma árvore binária de busca), por sua vez, possui um único atributo: **Node<T> *root**, que é sua raiz. Mais importante, este mesmo possui os métodos básicos para o comportamento da árvore: **inserir**, **remover**, **buscar**, **percurso em ordem**, **pré-ordem** e **pós-ordem**, **busca**, **vazio?**, etc. Os métodos **remove** (que segue o padrão adotado em sala de aula, isto é, durante a remoção de um elemento com dois filhos, o maior elemento da sub-árvore esquerda é transferido para a raiz dessa mesma árvore, que é o elemento que queríamos remover inicialmente. Posteriormente, o nó do elemento transferido é removido recursivamente, evitando assim uma duplicação na árvore) e as **listagens ordenadas** foram implementados *recursivamente*.

Todas as estruturas foram construídas com a utilização de *template*, a fim de proporcionar uma generalização do uso das estruturas.

3. PRINT-SCREENS

3.1. Lista

A terminal window titled 'Atividades' and 'Terminal' showing a series of commands and their outputs. The user is in the directory ~/workspace/lista/Release. The script ./lista performs several operations on a list: it inserts 1 and 3, then 2 to the left of 3, then 4 to the right of 3. It then checks if 2 and 56 are in the list, and finally removes 2 and 4. The list state is printed after each major step.

```
joaovitor@JV: ~/workspace/lista/Release
Arquivo Editar Ver Pesquisar Terminal Ajuda
joaovitor@JV:~$ cd workspace/lista/Release/
joaovitor@JV:~/workspace/lista/Release$ ./lista
Vamos inserir 1 e 3 na lista:
1
3

Agora vamos inserir 2 a esquerda de 3
1
2
3

Agora vamos inserir 4 a direita de 3
1
2
3
4

O tamanho da nossa lista é 4
Será que o número 2 está na lista?
Está na lista!

Será que o número 56 está na lista?
Não está na lista

Agora vamos remover o 2 da lista
1
3
4

Agora vamos remover o 4 da lista
1
3

joaovitor@JV:~/workspace/lista/Release$
```

3.2. Fila

```
Atividades | _Terminal | Sex 01:57 | 40.5°C | João Vitor
joaovitor@JV: ~/workspace/fila/Release
Arquivo Editar Ver Pesquisar Terminal Ajuda
3 - Ver tamanho da fila
4 - Procurar elemento na fila
5 - Exibir fila
0 - Sair do programa
3
0 tamanho da fila é 2
1 - Inserir inteiro na fila
2 - Remover inteiro da fila
3 - Ver tamanho da fila
4 - Procurar elemento na fila
5 - Exibir fila
0 - Sair do programa
4
Entre com o valor a ser procurado
5
5 está na fila
1 - Inserir inteiro na fila
2 - Remover inteiro da fila
3 - Ver tamanho da fila
4 - Procurar elemento na fila
5 - Exibir fila
0 - Sair do programa
4
Entre com o valor a ser procurado
10
10 não está na fila
1 - Inserir inteiro na fila
2 - Remover inteiro da fila
3 - Ver tamanho da fila
4 - Procurar elemento na fila
5 - Exibir fila
0 - Sair do programa
0
Saindo...
joaovitor@JV:~/workspace/fila/Release$
```

3.3. Cadastro

```
Atividades | >_Terminal Sex 02:15 38.5°C João Vitor
joaovitor@JV: ~/workspace/cadastro/Release
Arquivo Editar Ver Pesquisar Terminal Ajuda

1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
1
Entre com o nome da pessoa a ser cadastrada
João

1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
1
Entre com o nome da pessoa a ser cadastrada
Thiago

1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
1
Entre com o nome da pessoa a ser cadastrada
Lucas

1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
1
```

```
Atividades | >_Terminal Sex 02:18 38.0°C João Vitor
joaovitor@JV: ~/workspace/cadastro/Release
Arquivo Editar Ver Pesquisar Terminal Ajuda
Entre com o nome da pessoa a ser cadastrada
Lucas

1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
5

Relação de pessoas cadastradas:
João
Lucas
Thiago

1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
2
Entre com o nome da pessoa a ser removida
Lucas

1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
5

Relação de pessoas cadastradas:
João
Thiago
```

```
Atividades | _Terminal Sex 02:22 39.5°C João Vitor
joaovitor@JV: ~/workspace/cadastro/Release
Arquivo Editar Ver Pesquisar Terminal Ajuda
1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
3
Total de pessoas cadastradas: 2
1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
4
Entre com o nome a ser procurado
João
João está cadastrado
1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa
4
Entre com o nome a ser procurado
Lucas
Lucas não está cadastrado
1 - Inserir no cadastro
2 - Remover do cadastro
3 - Ver quantidade de pessoas cadastradas
4 - Procurar nome no cadastro
5 - Exibir todos as pessoas cadastrados
0 - Sair do programa

```

3.4. Árvore



```
C:\Users\lucas\Desktop\Arvore\bin\Debug\Arvore.exe

Programa teste para a implementacao da Arvore Binaria de Busca!
Sera que ela esta vazia?
Sim! Obviamente ne? Ela acabou de ser criada...

Inserindo alguns elementos:
Inseriu 1!
Inseriu 2!
Inseriu 5!
Inseriu 900!
Inseriu 10!
Falha ao inserir 1! Claro, ele jaa esta na pilha!

Perguntando se ela esta vazia de novo...
Nao! Obviamente ne? Inserimos alguns elementos...

Em ordem:
1 2 5 900 10
Pre-ordem:
1 2 5 10 900
Pos-ordem:
10 900 5 2 1

Tentando remover o 102301392: Ops! Nao removeu!

Tentando remover o 5: Removi o numero!

Em ordem:
1 2 900 10
Process returned 0 (0x0)  execution time : 0.038 s
Press any key to continue.

C:\Users\lucas\Desktop\Arvore\bin\Debug\Arvore.exe

Programa teste para a implementacao da Arvore Binaria de Busca!
Sera que ela esta vazia?
Sim! Obviamente ne? Ela acabou de ser criada...

Inserindo alguns elementos:
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!

Perguntando se ela esta vazia de novo...
Nao! Obviamente ne? Inserimos alguns elementos...

Em ordem:
1000 -3922 45302 23421 12932 87552
Pre-ordem:
-3922 1000 12932 23421 45302 87552
Pos-ordem:
-3922 12932 23421 87552 45302 1000

Tentando remover o 102301392: Ops! Nao removeu!
Tentando remover o 5: Ops! Nao removeu!

Em ordem:
1000 -3922 45302 23421 12932 87552
Process returned 0 (0x0)  execution time : 0.036 s
Press any key to continue.
```



```
C:\Users\lucas\Desktop\Arvore\bin\Debug\Arvore.exe
Sera que ela esta vazia?
Sim! Obviamente ne? Ela acabou de ser criada...

Inserindo alguns elementos:
Inseriu 10!
Inseriu 20!
Inseriu 5!
Inseriu 1!
Inseriu 0!
Inseriu 100!

Em ordem:
10 5 1 0 20 100

Pre-orden:
0 1 5 10 20 100

Pos-orden:
0 1 5 100 20 10

Tentando remover o 102301392: Ops! Nao removeu!

Tentando remover o 5: Removi le numero!

Em ordem:
10 1 0 20 100

Process returned 0 (0x0)   execution time : 0.017 s
Press any key to continue.
```

```
C:\Users\lucas\Desktop\Arvore\bin\Debug\Arvore.exe

Programa teste para a implementacao da Arvore Binaria de Busca!

Sera que ela esta vazia?
Sim! Obviamente ne? Ela acabou de ser criada...

Inserindo alguns elementos:
Inseriu 1!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Inseriu!
Falha ao inserir 1! Claro, ele jaa esta na pilha!

Perguntando se ela esta vazia de novo...
Nao! Obviamente ne? Inserimos alguns elementos...

Em ordem:
1 2 5 3 900 33 7 9 10 1000

Pre-orden:
1 2 3 5 7 9 10 33 900 1000

Pos-orden:
3 10 9 7 33 1000 900 5 2 1

Tentando remover o 102301392: Ops! Nao removeu! Tem certeza que ele esta na arvore?
Tentando remover o 900: Removi o numero!
Tentando remover o 1: Removi o numero!

Em ordem:
2 5 3 33 7 9 10 1000

Process returned 0 (0x0)   execution time : 0.029 s
Press any key to continue.
```

4. CONCLUSÃO

Observamos as vantagens da implementação de estruturas genéricas, uma vez que foi definido a utilização da **lista duplamente encadeada com header** para a implementação do trabalho 3. Também ficou evidente as semelhanças e diferenças entre as estruturas utilizadas: o **Cadastro** e **Fila** são classes criadas a partir de uma *herança* da classe **Lista**, embora estes mesmos mostrem conceitos distintos de manipulação. Tal herança exprime o conceito de “reaproveitamento de código” de nosso trabalho.

