

Framework de Estrutura de Dados

Gianluca Matos Klaumann¹; Rodrigo Curvêllo²

¹ Estudante de Graduação em Bacharelado em Ciência da Computação, IFC - Campus Rio do Sul. E-mail: gianlucamk04@gmail.com

² Orientador, Professor EBTT, IFC - Campus Rio do Sul. E-mail: rodrigo.curvello@ifc.edu.br

RESUMO

Estrutura de Dados é uma maneira de organizar e armazenar dados em um computador para que possam ser acessados e manipulados de maneira eficiente. Dentro deste campo, existem as Listas Simplesmente Encadeadas, Listas Circulares e Listas Duplamente Encadeadas, que são maneiras distintas de ordenar esses dados e serão os tipos analisados e implementados dentro deste trabalho. Desta forma, será abordado tanto a implementação com alguns métodos básicos para manipular as listas utilizando de Orientação à Objetos, princípios do SOLID (*Single Responsibility Principle; Open Close Principle; Liskov Substitution Principle; Interface Segregation Principle; Dependency Inversion Principle*) e Padrões de Projeto (*Design Patterns*), quanto uma ilustração delas com o uso da biblioteca JavaFX para melhor visualização da estrutura da respectiva lista.

Palavras-chave: Estrutura de Dados; Listas; Listas Encadeadas; Listas Duplamente Encadeadas.

INTRODUÇÃO

Uma estrutura de dados é uma maneira de organizar e armazenar dados em um computador para que possam ser acessados e manipulados de maneira eficiente. Desta forma, a escolha da estrutura de dados adequada pode impactar significativamente o desempenho e a complexidade de algoritmos e sistemas computacionais. As principais características das estruturas de dados incluem a forma como os dados são organizados, as operações suportadas, e a eficiência das operações em termos de tempo e espaço.

Uma lista encadeada é uma estrutura de dados linear composta por uma sequência de elementos, onde cada elemento (ou nó) contém dois componentes principais: um valor (ou dado) e uma referência (ou ponteiro) para o próximo nó na sequência. Dentro das listas encadeadas, é abordado no projeto dois tipos: a lista simples e a lista circular.

Uma lista encadeada simples é uma lista onde cada nó aponta apenas para o próximo nó. A navegação é unidirecional, o que significa que você pode percorrer a lista apenas na direção do final. Enquanto isso, a lista encadeada circular é uma variação da lista encadeada onde o último nó aponta de volta para o primeiro nó, formando um ciclo. Assim, não há um "fim" no sentido tradicional.

Além das formas citadas, o framework trabalha com listas duplamente encadeadas, que são listas onde cada nó tem duas referências: uma para o próximo nó e outra para o nó anterior. Isso permite navegação bidirecional, tornando operações como inserção e remoção mais flexíveis.

Desta forma, ao entender os tipos de listas abordados, é possível observar como cada uma pode ser útil dependendo da ocasião, assim, o projeto traz versatilidade para a implementação do tipo de lista desejado em um outro trabalho e a ilustração destas para melhor visualização da estrutura da lista no momento.

PROCEDIMENTOS METODOLÓGICOS

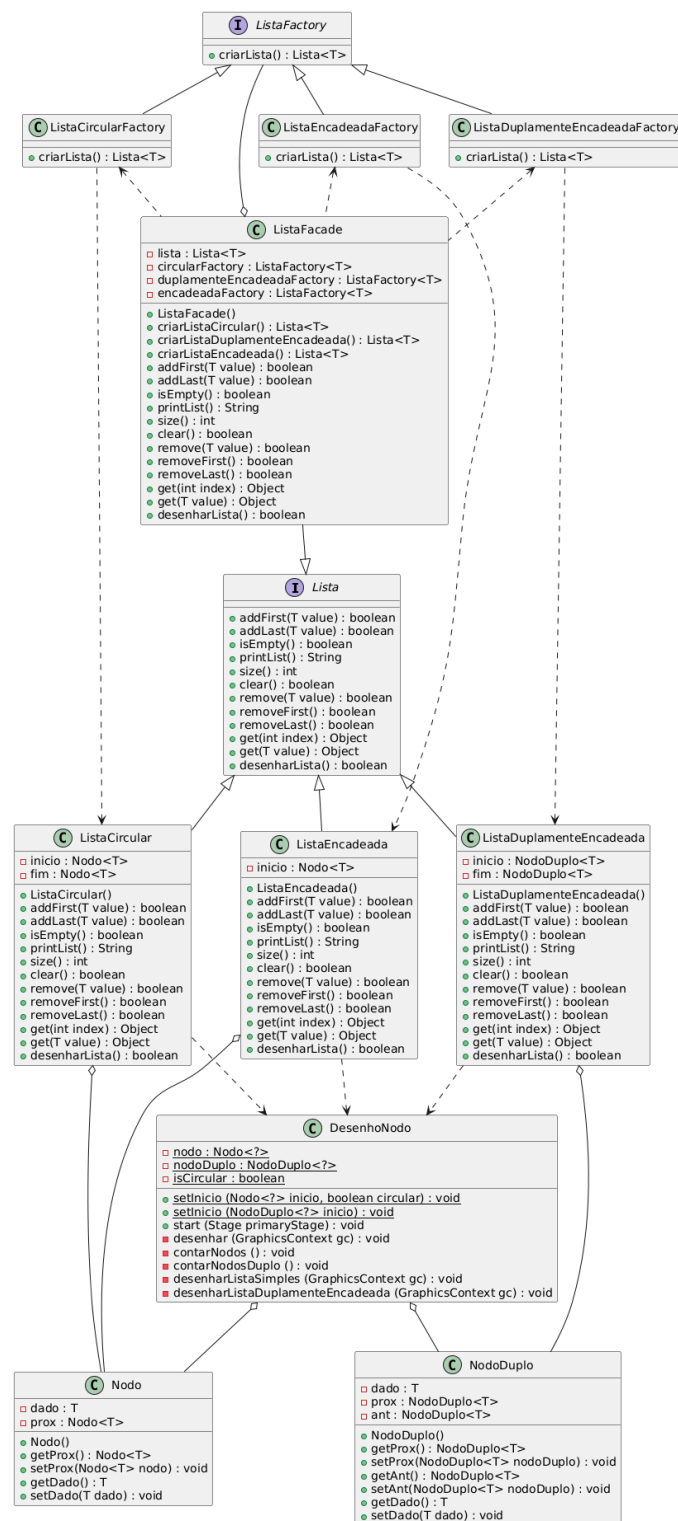
Para a criação do Framework, foi necessário conhecimento de Orientação à Objetos, do SOLID, de Padrões de Projeto e de desenvolvimento em Java. Dentro do trabalho, foram utilizados principalmente dois padrões, Factory Method e Facade. O Factory Method é um Padrão de Projeto Criacional que consiste em uma interface para criar objetos em uma superclasse, mas permite que subclasses alterem o tipo de objetos que serão criados. Já o Facade é um Padrão de Projeto Estrutural que fornece uma interface simplificada para um conjunto de interfaces em um subsistema.

Dentro do Projeto o uso de Factory é para a criação das diferentes listas e o Facade é para centralizar as funcionalidades que serão utilizadas pelo usuário do Framework a fim de facilitar o uso deste no software desejado.

Para saber aplicar de forma adequada os conceitos citados, teve-se um estudo sobre Estrutura de Dados e Listas com o intuito de entender suas definições e suas implementações em diferentes meios. Dessa forma, foi possível criar um Framework capaz de criar listas para a finalidade escolhida pelo usuário, tornando o trabalho mais genérico para atender a demanda deste.

Por fim, para a criação da ilustração das listas, necessitou-se de um estudo sobre como criar desenhos dentro do Java de maneira eficiente e menos custosa. Assim, a maneira que melhor atendia os requisitos foi com a utilização da biblioteca JavaFX, que possui uma vasta gama de funcionalidades que ajudam na personalização da ilustração que deseja ser criada. Dessa forma, foi possível desenhar a estrutura das diferentes listas implementadas pelo Framework e apresentá-las ao usuário.

Abaixo está como ficou o Diagrama de Classe do projeto após todos estes procedimentos:



RESULTADOS E DISCUSSÃO

A partir da realização do Framework, foi possível implementar as funcionalidades desejadas com êxito, contando tanto com a criação das listas com alguns métodos

padrões quanto com a ilustração da estrutura da lista originada. Para evidenciar o uso, serão apresentados três exemplos, um de cada tipo implementado de lista, mostrando a inserção dentro da lista e o desenho desta.

O primeiro exemplo é de uma lista simplesmente encadeada do tipo Number que armazena e ilustra o seguinte:

```
ListaFacade<Number> lista = new ListaFacade<>();  
lista.criarListaEncadeada();  
lista.addFirst(1.5);  
lista.addFirst(2.9);  
lista.addFirst(5.8);  
lista.desenharLista();
```

Figura 01 - Inserção na Lista Simplesmente Encadeada

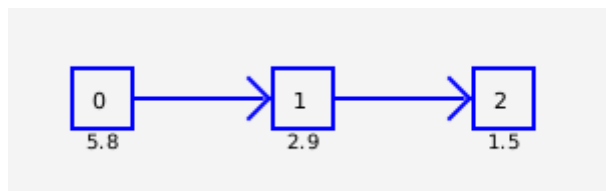


Figura 02 - Ilustração da Lista Simplesmente Encadeada

O segundo exemplo é de uma lista encadeada circular do tipo String que armazena e ilustra o seguinte:

```
ListaFacade<String> listaCirc = new ListaFacade<>();  
listaCirc.criarListaCircular();  
listaCirc.addFirst("c");  
listaCirc.addFirst("b");  
listaCirc.addFirst("a");  
listaCirc.desenharLista();
```

Figura 03 - Inserção na Lista Encadeada Circular

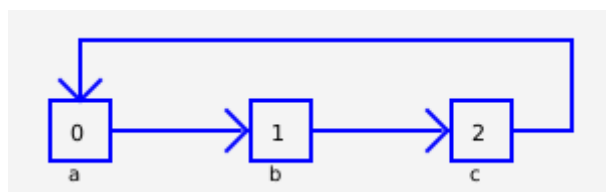


Figura 04 - Ilustração da Lista Encadeada Circular

O terceiro exemplo é de uma lista duplamente encadeada do tipo Integer que armazena e ilustra o seguinte:

```
ListaFacade<Integer> listaDupla = new ListaFacade<>();
listaDupla.criarListaDuplamenteEncadeada();
listaDupla.addFirst(3);
listaDupla.addFirst(2);
listaDupla.addFirst(1);
listaDupla.desenharLista();
```

Figura 05 - Inserção na Lista Duplamente Encadeada

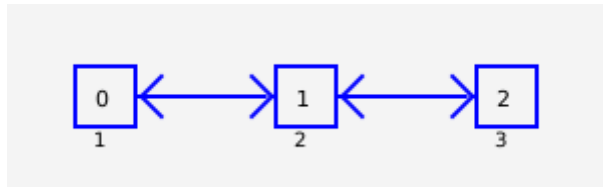


Figura 06 - Ilustração da Lista Duplamente Encadeada

A partir dos exemplos é possível visualizar a função de desenhar as listas e como que esta pode ser útil tanto para uma apresentação mais amigável da lista quanto para fins de estudo, pois com a ilustração da lista, é bem mais fácil de entender seu conceito e como funciona o encadeamento das informações bem como sua utilidade para estruturar os dados armazenados de maneira eficiente.

Além dos métodos demonstrados, existem outros que também possuem sua utilidade quando se quer utilizar estas listas mais na prática. O Framework oferece o básico que se é necessário para manipular as listas da maneira que for melhor para o usuário e seu projeto.

CONSIDERAÇÕES FINAIS

O desenvolvimento deste framework de Estruturas de Dados, que implementa e ilustra Listas Simplesmente Encadeadas, Listas Circulares e Listas Duplamente Encadeadas, permitiu uma compreensão mais profunda sobre a aplicação dos conceitos de Orientação à Objetos, dos princípios do SOLID e dos Padrões de Projeto.

Além disso, a utilização dos padrões de projeto, como Factory Method e Facade, trouxe flexibilidade e simplicidade ao framework, permitindo sua fácil adaptação para diferentes necessidades. A visualização das listas por meio da ilustração feita com a biblioteca JavaFX, por sua vez, proporcionou uma ferramenta eficaz para fins educacionais e práticos, facilitando o entendimento da estrutura e do comportamento das listas encadeadas.

O framework desenvolvido se mostrou uma solução eficiente e versátil, que pode ser incorporada em projetos futuros onde a manipulação de listas encadeadas é necessária. As metodologias empregadas garantiram que o projeto atendesse aos requisitos propostos, resultando em uma ferramenta útil tanto para profissionais quanto para estudantes que buscam uma melhor compreensão das estruturas de dados e sua aplicação.

Portanto, o resultado final do projeto demonstra uma boa aplicação de listas e uma boa maneira de visualizar sua estrutura para um melhor entendimento destas. Assim, o Framework se demonstra eficiente tanto para uma utilização prática quanto teórica para o uso de listas.