

CENTRO UNIVERSITÁRIO DA FEI

BRENNO TONDATO DE FARIA

ATIVIDADE 2 - PEL 216

São Bernardo do Campo

2019

BRENNO TONDATO DE FARIA

ATIVIDADE 2 - PEL 216

Realatório da Atividade 2 proposta da disciplina Programação Científica (PEL216) ministrada pelo Prof. Dr. Reinaldo Bianchi

São Bernardo do Campo

2019

RESUMO

Estrutura de dados são largamente utilizadas em uma grande variedade de sistemas. Porém em um ambiente de desenvolvimento é necessário considerar alguns pontos para se obter um código eficiente. Um dos pontos que se deve considerar é o tipo de estrutura que se vai utilizar e além disso se a estrutura será estática ou dinâmica. No caso de uma estrutura dinâmica, esta pode ser em uma única estrutura ou pode ser implementada em uma estrutura ligada. Este trabalho tem o objetivo de implementar as estrutura Pilha, Fila e Lista de forma ligada, utilizando conceitos de Programação Orientada a Objetos (POO).

Keywords: Estrutura de Dados, Pilha, Fila, Lista, Ligada

LISTA DE ILUSTRAÇÕES

Ilustração 1 – Inserção em Lista Ligada	5
Ilustração 2 – Remoção em Lista Ligada	6
Ilustração 3 – Diagrama de Classes	7
Ilustração 4 – Resultados Slack	9
Ilustração 5 – Resultados Queue	10

SUMÁRIO

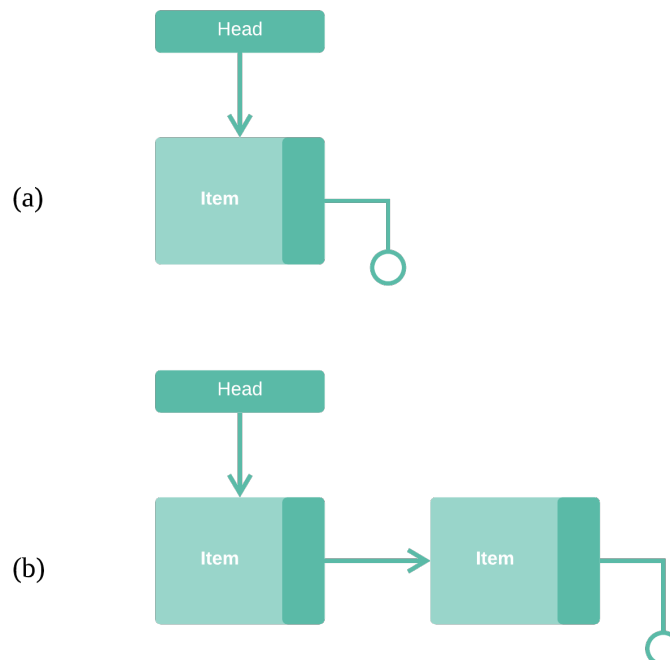
1	Conceitos Fundamentais	5
2	Metodologia	7
3	Experimentos	8
4	Conclusão	11
	REFERÊNCIAS	12

1 CONCEITOS FUNDAMENTAIS

A estrutura mais importante de se considerar neste trabalho é a lista Ligada [Ref]. Esta estrutura consiste em uma série de nós e estes nós são ligados entre si através de referências para os nós seguintes. Um componente importante é o ponteiro *Head*. Este ponteiro é

A inserção de novos elementos na lista é dada a partir do apontamento do último elemento da lista para o endereço de memória novo elemento. A Figura 1 ilustra a ideia.

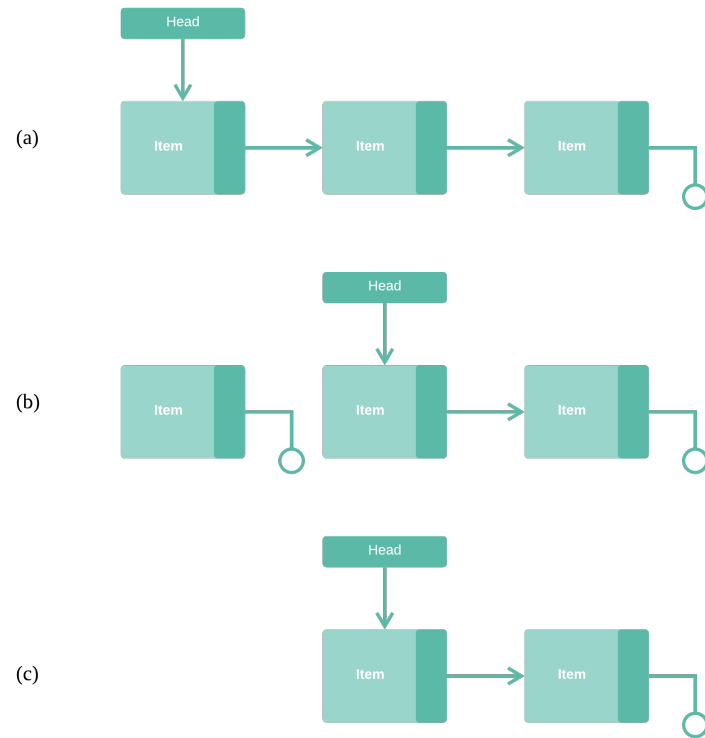
Figura 1 – Inserção em Lista Ligada



(Do Autor, 2019)

A remoção de um elemento da lista ocorre eliminando a referência ao elemento de seu nó antecessor e desalocando este elemento da memória do sistema. A Figura 2 ilustra a ideia. é importante notar que no caso a remoção seja realizada no primeiro elemento é necessário que o ponteiro *Head* seja incrementado para o seu nó sucessor.

Figura 2 – Remoção em Lista Ligada



(Do Autor, 2019)

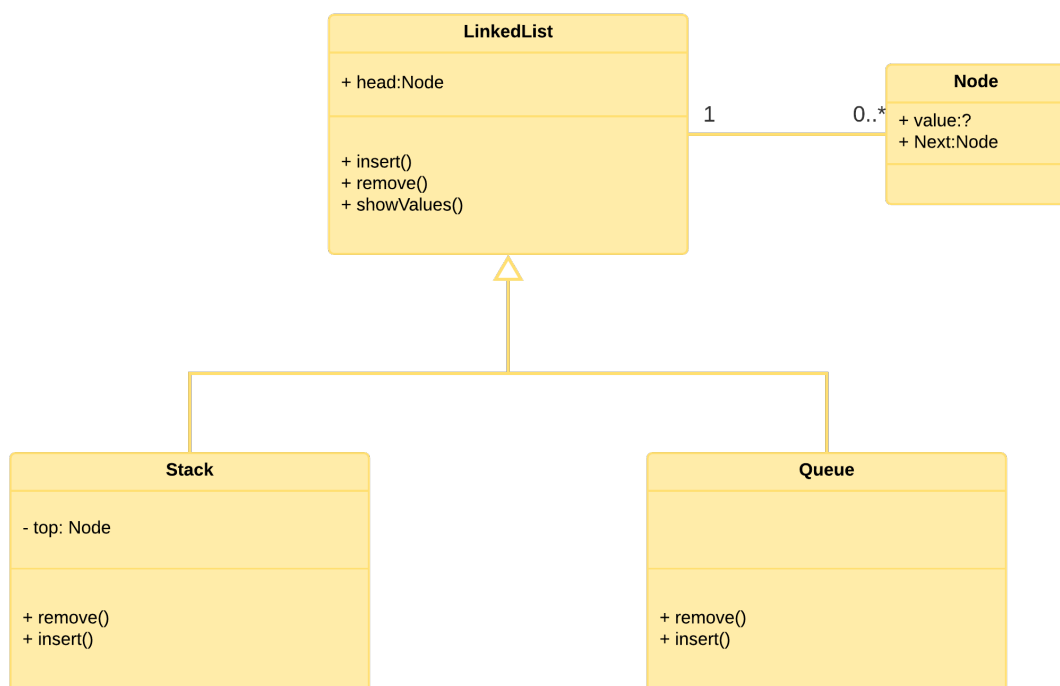
2 METODOLOGIA

Para a implementação das estruturas foi desenvolvido um código em linguagem de programação Swift (APPLE, 2019). Para a criação das classes foi necessário realizar uma sobrecarga de operadores, que nesta linguagem é realizada estendendo um protocolo [cite] chamado "Equatable". A extensão deste protocolo permite que os operadores "==" , >=" , <=" e "!=" , sejam utilizados por tipos de dados genéricos.

Inicialmente a classe *Node* é criada. Esta classe possui duas informações. a primeira é um valor sem tipo definido, que mais tarde será tratado como um tipo genérico, e uma variável do tipo *Node* que armazena uma referência para o próximo nó da lista.

As estruturas criadas consistem na Classe *LinkedList* que implementa as funções básicas, sendo estas a inserção, remoção e uma função para mostrar os elementos da estrutura. Outras classes criadas são a da *Stack* e da *Queue*, é importante notar que estas estruturas herdam a classe *LinkedList* para utilizar suas funções, porém estas classes fazem uma sobreposição da função de remoção a fim de caracterizar a determinada estrutura. A Figura 3 a seguir ilustra as classes criadas.

Figura 3 – Diagrama de Classes



(Do Autor, 2019)

3 EXPERIMENTOS

Os experimentos se concentram em uma série de operações de inserção e remoção, afim de atestar o cerreto funcionamento da lista ligaada, tanto para a estrutura Fila, quanto para à estrutura Pilha.

Para a estrutura pilha observou-se o correto sequenciamento de informação. a Figura 4 apresenta à cópia da tela apresentada pelo programa.

Para a estrutura fila observou-se o correto sequenciamento de informação. a Figura 5 apresenta à cópia da tela apresentada pelo programa.

Figura 4 – Resultados Slack

```
===== Stack Tests =====  
-----  
The item is banana  
The item is Picles  
The item is azeitonas  
The item is Brenno  
The item is Batata  
The item is Camunhão  
The item is Azul  
The item is Cereja  
And the top list is Optional("Cereja")  
  
-----  
The item is banana  
The item is Picles  
The item is azeitonas  
The item is Brenno  
The item is Batata  
The item is Camunhão  
The item is Azul  
And the top list is Optional("Azul")  
  
-----  
The item is banana  
The item is Picles  
The item is azeitonas  
The item is Brenno  
The item is Batata  
The item is Camunhão  
And the top list is Optional("Camunhão")  
  
-----  
The item is banana  
The item is Picles  
The item is azeitonas  
The item is Brenno  
The item is Batata  
And the top list is Optional("Batata")  
  
-----  
The item is banana  
The item is Picles  
The item is azeitonas  
The item is Brenno  
And the top list is Optional("Brenno")
```

(Do Autor, 2019)

Figura 5 – Resultados Queue

```
===== Queue Tests =====  
-----  
The item is banana  
The item is Picles  
The item is azeitonas  
The item is Brenno  
The item is Batata  
The item is Camunhão  
The item is Azul  
The item is Cereja  
And the top list is Optional("banana")  
  
-----  
The item is Picles  
The item is azeitonas  
The item is Brenno  
The item is Batata  
The item is Camunhão  
The item is Azul  
The item is Cereja  
And the top list is Optional("Picles")  
  
-----  
The item is azeitonas  
The item is Brenno  
The item is Batata  
The item is Camunhão  
The item is Azul  
The item is Cereja  
And the top list is Optional("azeitonas")  
  
-----  
The item is Brenno  
The item is Batata  
The item is Camunhão  
The item is Azul  
The item is Cereja  
And the top list is Optional("Brenno")  
  
-----  
The item is Batata  
The item is Camunhão  
The item is Azul  
The item is Cereja  
And the top list is Optional("Batata")  
  
-----  
The item is Camunhão  
The item is Azul  
The item is Cereja  
And the top list is Optional("Camunhão")
```

(Do Autor, 2019)

4 CONCLUSÃO

O projeto elaborou um *software* para compreender o funcionamento de uma estrutura de dados ligada por ponteiros, afim de compreender os fundamentos desta estrutura. Foram elaboradas duas estruturas ligadas, seguindo os princípios de Fila (FIFO) e Pilha (LIFO). Pode-se observar o correto funcionamento da estrutura e compreender seus fundamentos observando-se o seu resultados.

REFERÊNCIAS

APPLE. **Swift**. 2019. Disponível em: <<https://developer.apple.com/swift/>>.