



Estruturas de dados dinâmicos utilizando classes (lista encadeada) - Manipulação de pilhas e filas

Equipe:

Davy Silva Ferreira - 119210349
Joseane dos Santos Silva - 119210750
Kleiton Oliveira Santos - 119210964
Ravânia Luciano Martildes - 119210485

Disciplina: Técnicas de Programação

Docente: Marcus Salerno

Período Suplementar 2020.3

Estrutura de Dados

Estrutura de Dados : Conceitos

❖ **Eficiência de memória**



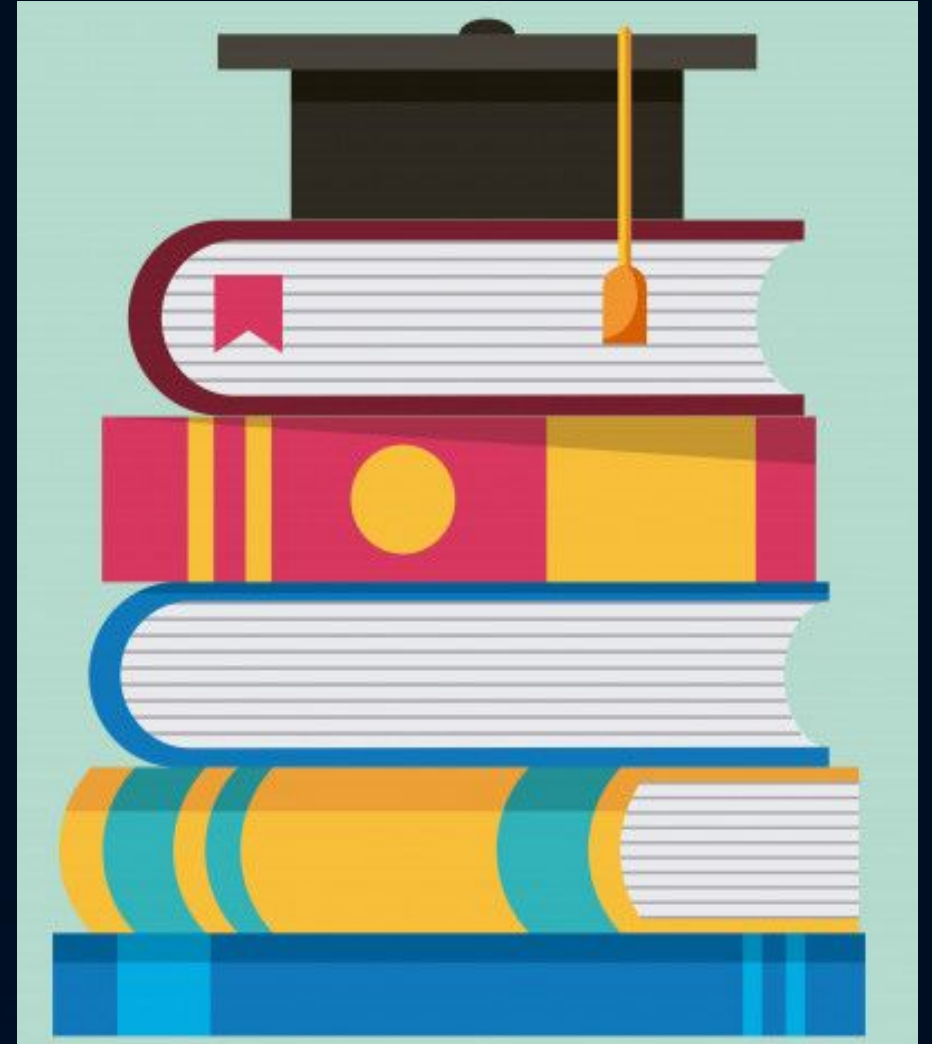
Estrutura de Dados : Exemplos

- Array (vetores e matrizes)
- Registro
- Union
- Lista encadeada
- Árvore binária
- Grafo
- Pilha
- Fila



Pilhas: Conceitos

- ❖ Pilhas são estruturas de dados que armazenam os elementos em um formato sequencial, empilhando um item acima do outro

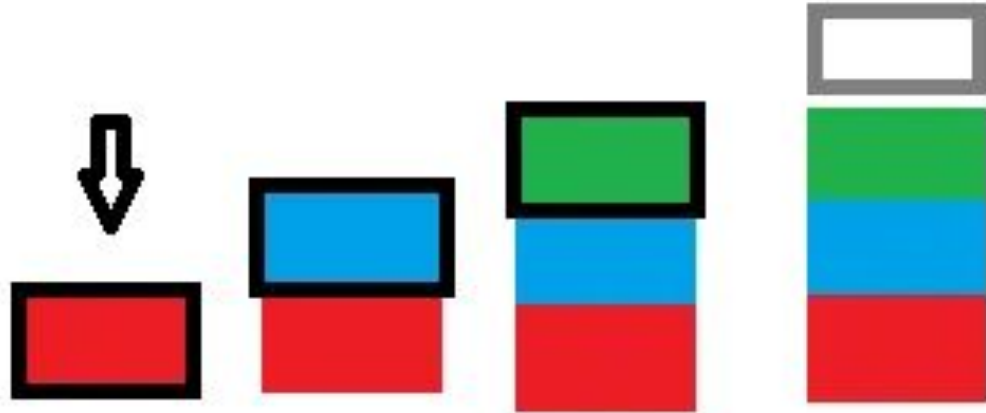


Pilhas: Funcionamento

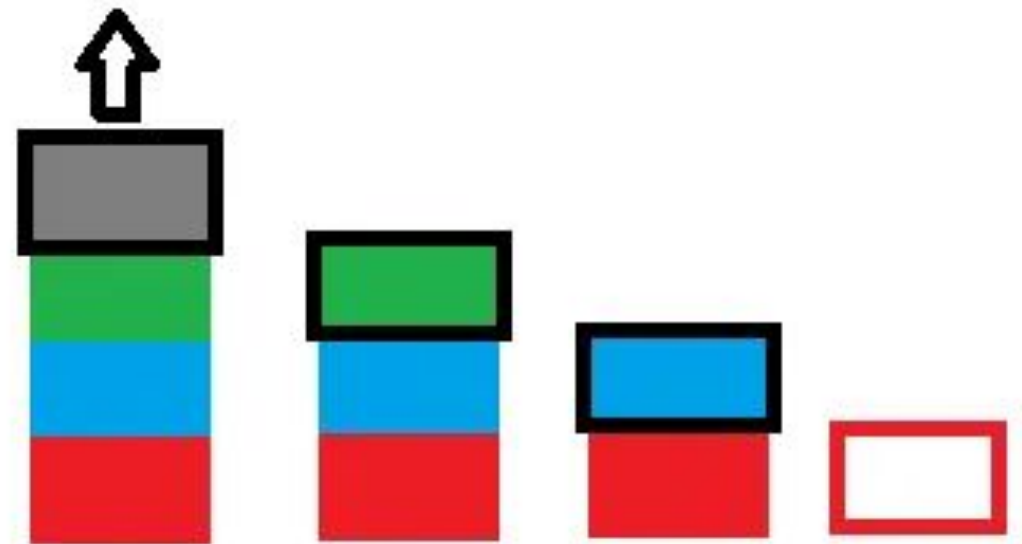


Pilhas: Last-In, First-Out – LIFO

O Último a Entrar...
(Last In...)



É o Primeiro a Sair.
(First Out)



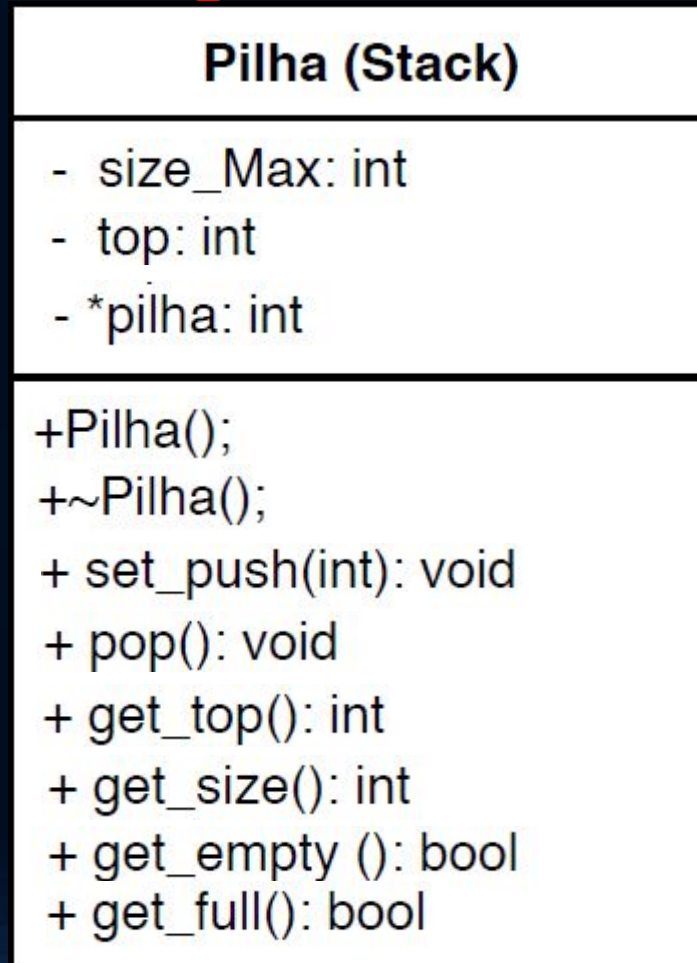
Pilhas: Aplicações

- **Gerenciamento de Memória em Tempo de Compilação**
- **Operações como desfazer e refazer em aplicações**
- **Controle de navegação em browsers**
- **Endereçamento de instruções em microprocessadores**
- **Análise de expressões aritméticas**



Implementação de uma pilha utilizando o conceito de classes e alocação dinâmica

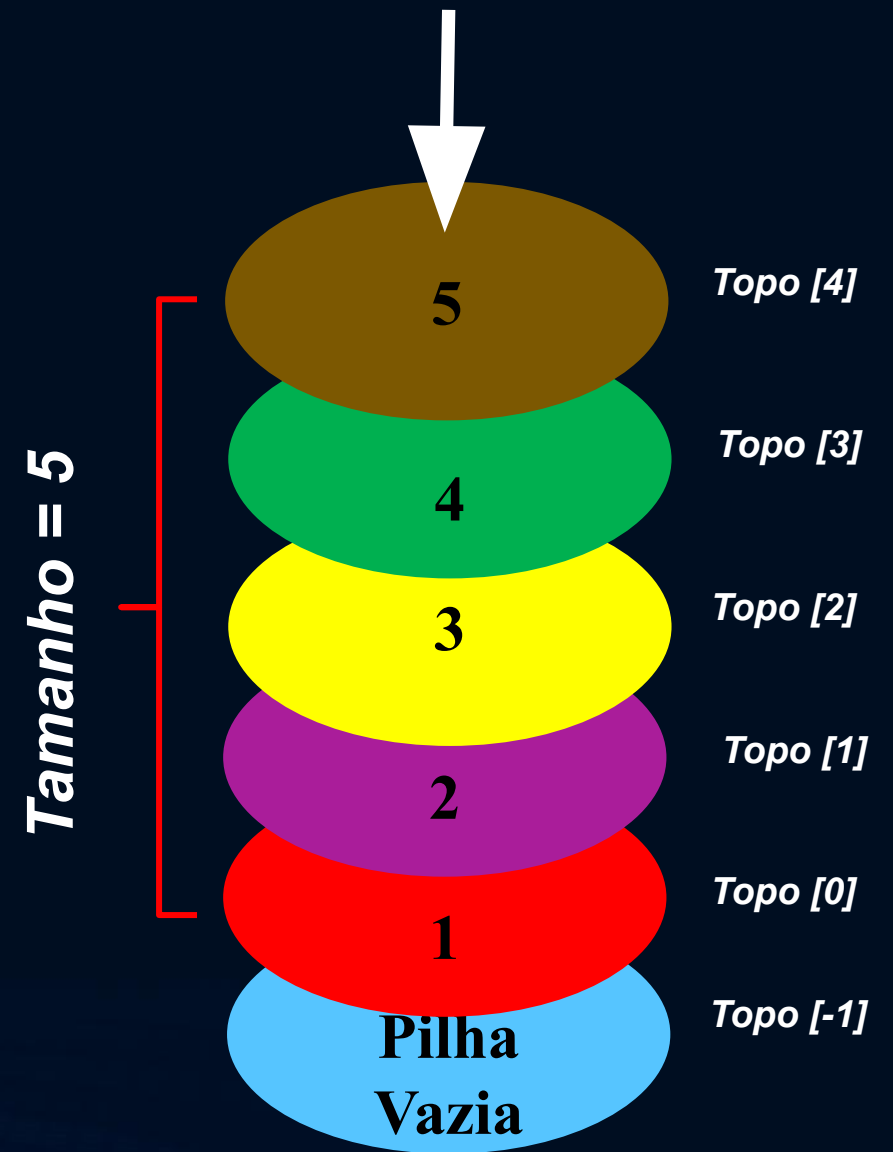
Diagrama UML



```

1  #include <iostream>
2  #include<cstdlib>
3  #include <Locale>
4
5  using namespace std;
6
7  #include "Pilha.cpp"
8
9  = int main() {
10
11     setlocale(LC_ALL,"Portuguese");
12
13     Pilha numero;
14
15  =   if (numero.get_empty()){
16
17       cout << endl<< "A pilha está vazia!" << endl << endl;
18   }
19
20     cout << "Tamanho: " << numero.get_size() << endl;
21
22     numero.set_push(1);
23     numero.set_push(2);
24     numero.set_push(3);
25     numero.set_push(4);
26     numero.set_push(5);
27
28     cout << endl << "Elemento do topo: " << numero.get_top();
29     cout << endl << "Tamanho: " << numero.get_size() << endl;
30

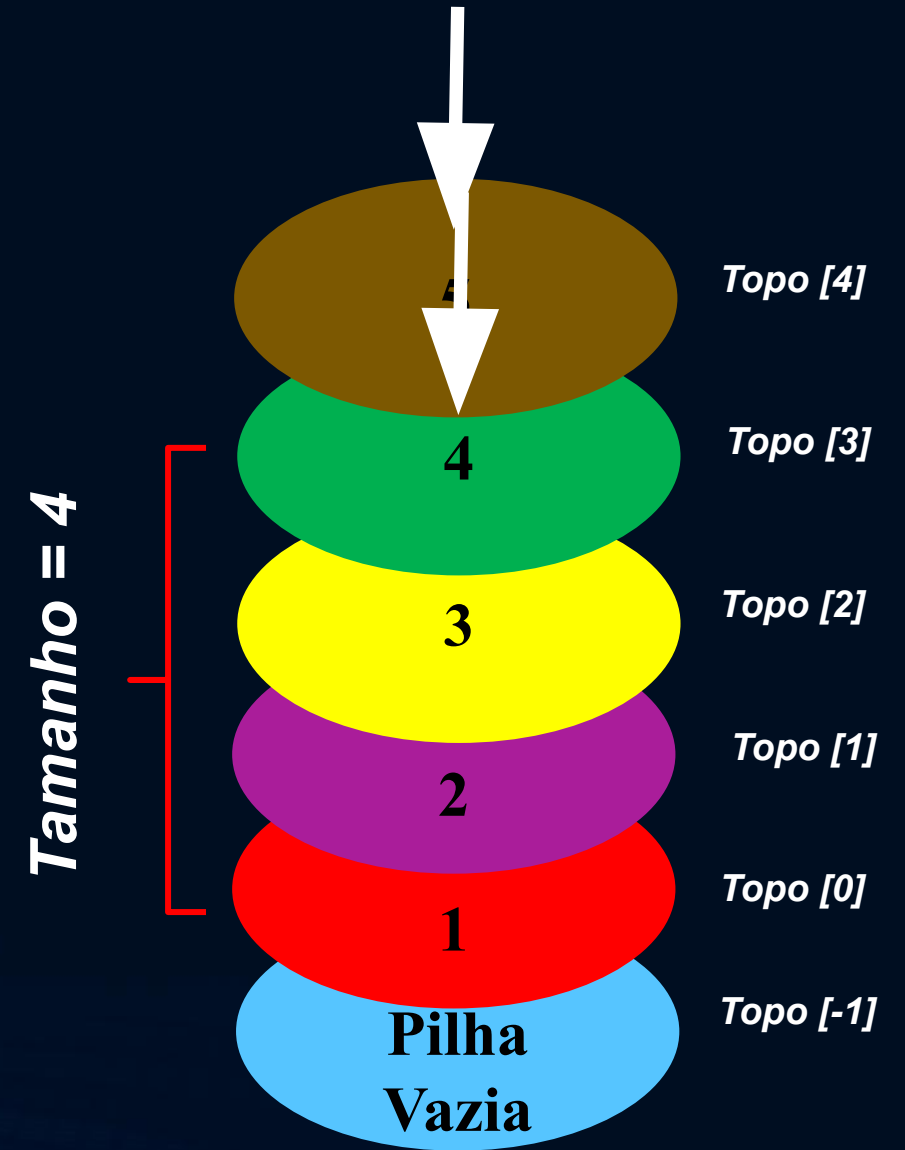
```



```

28 cout << endl << "Elemento do topo: " << numero.get_top();
29 cout << endl << "Tamanho: " << numero.get_size() << endl;
30
31 numero.pop();
32
33 cout << endl << "Elemento do topo: " << numero.get_top();
34 cout << endl << "Tamanho: " << numero.get_size() << endl;
35
36 cout << endl << "Elementos da pilha: ";
37
38 while (!numero.get_empty()){
39     cout << numero.get_top() << " ";
40     numero.pop();
41 }
42
43 cout << endl;
44
45 if (numero.get_empty()){
46     cout << endl << "A pilha está vazia!" << endl << endl;
47 }
48
49 system("pause");
50 return 0;
51
52 }

```



Pilhas : Container/Biblioteca <stack>

- ❖ Inclusão da biblioteca

`#include <stack>`

- ❖ Definição da pilha

`stack<tipo da pilha> nome da pilha;`

Pilhas : Container/Biblioteca <stack>

- ❖ **push ();** - Adiciona elementos
- ❖ **pop ();** - Remove elementos
- ❖ **empty ();** - Verifica se a pilha está vazia.
- ❖ **top();** - Topo da pilha
- ❖ **size();** - Tamanho da pilha

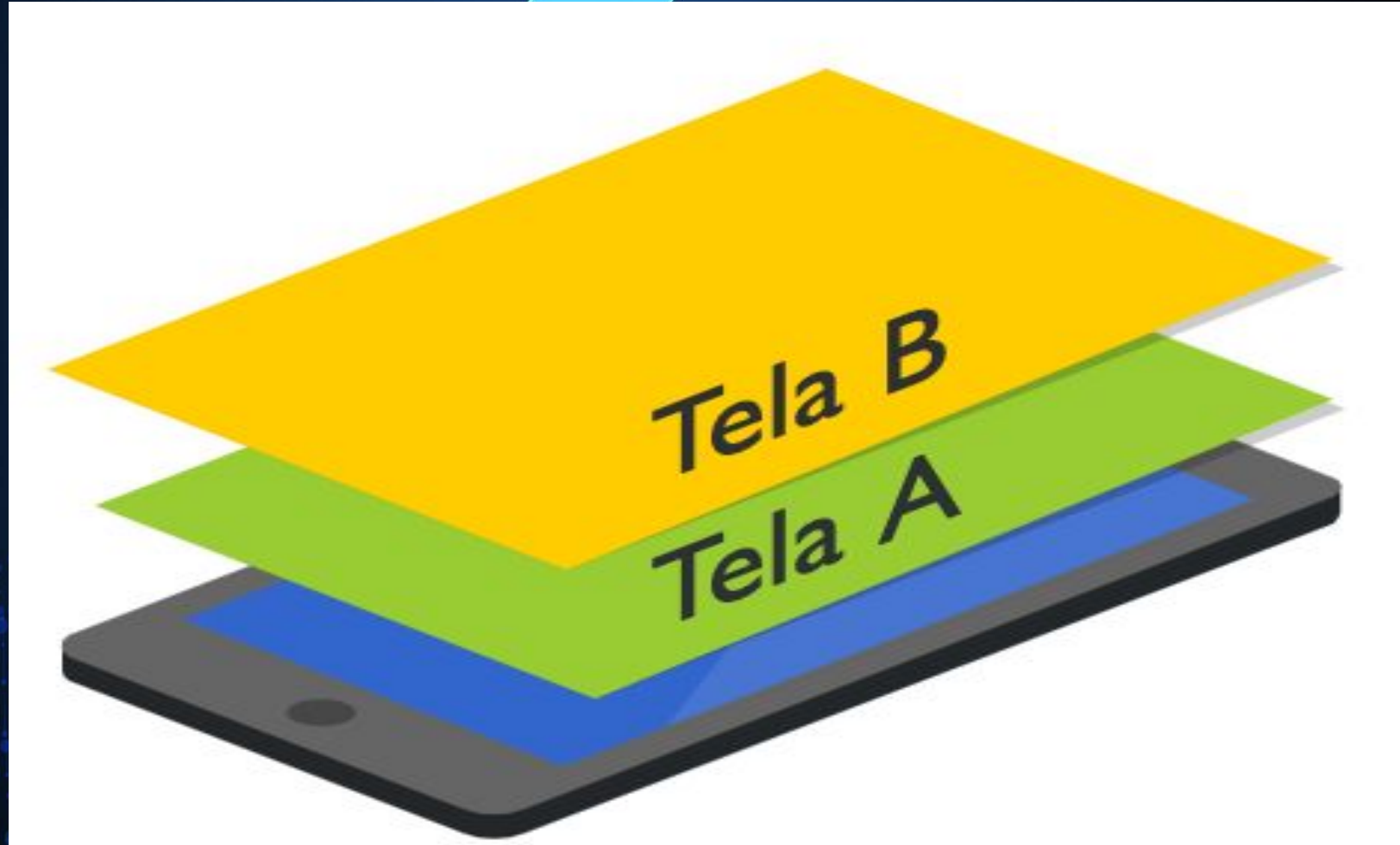


Pilhas: Aplicações

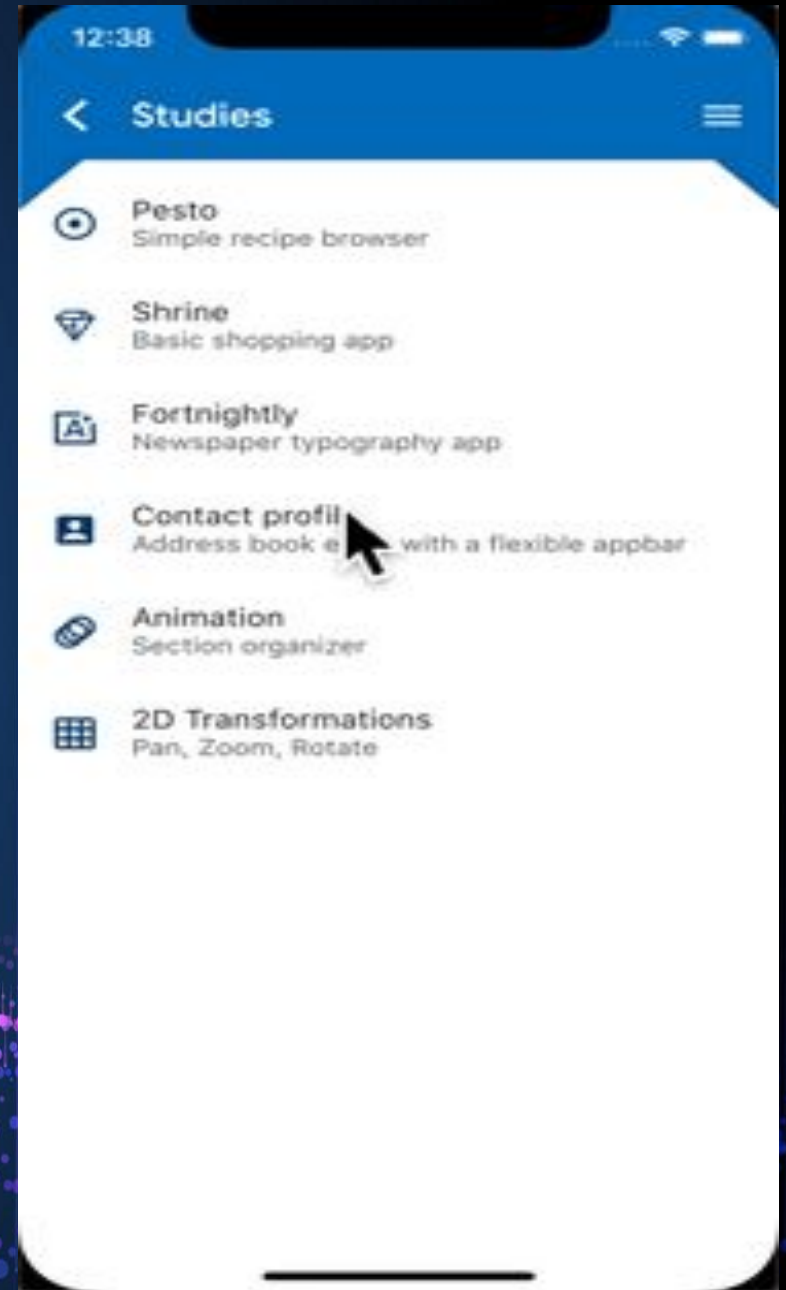
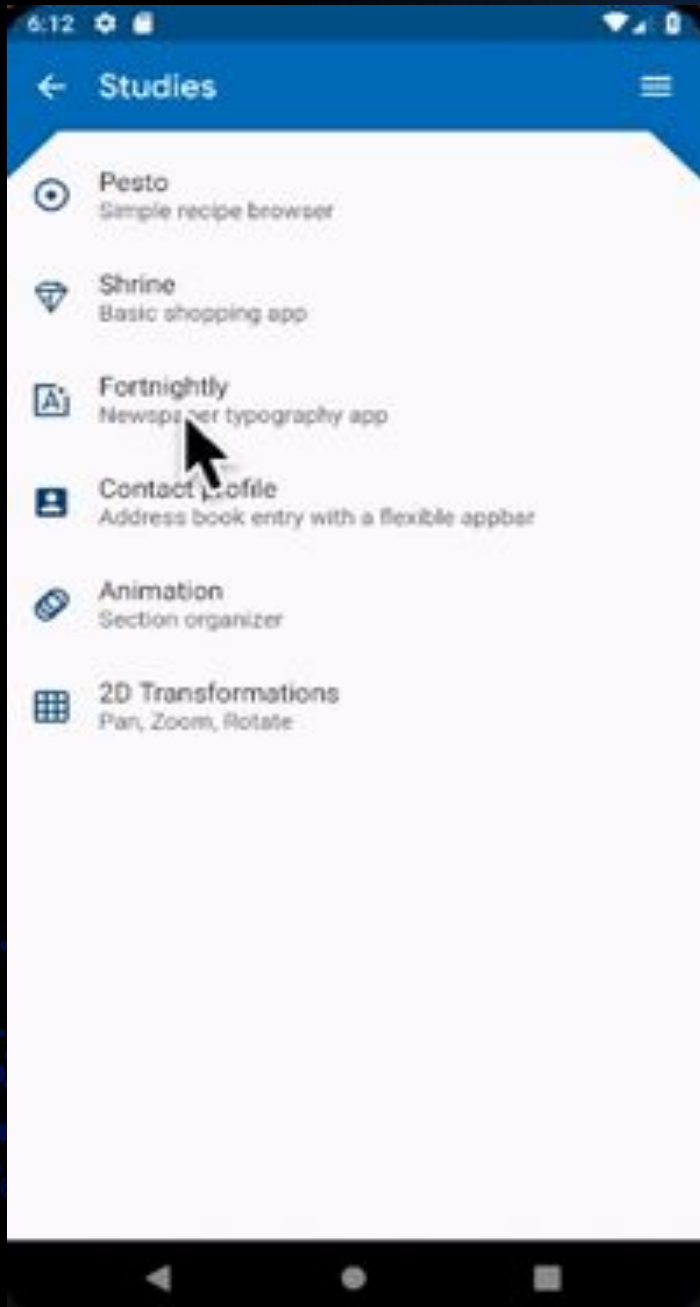
FLUTTER



FLUTTER



FLUTTER



The background is a deep blue gradient. On the left, there is a faint grid of small squares. On the right, there are several concentric, curved lines that create a sense of depth and movement, resembling a tunnel or a stylized eye.

Filas

Filas: Conceitos

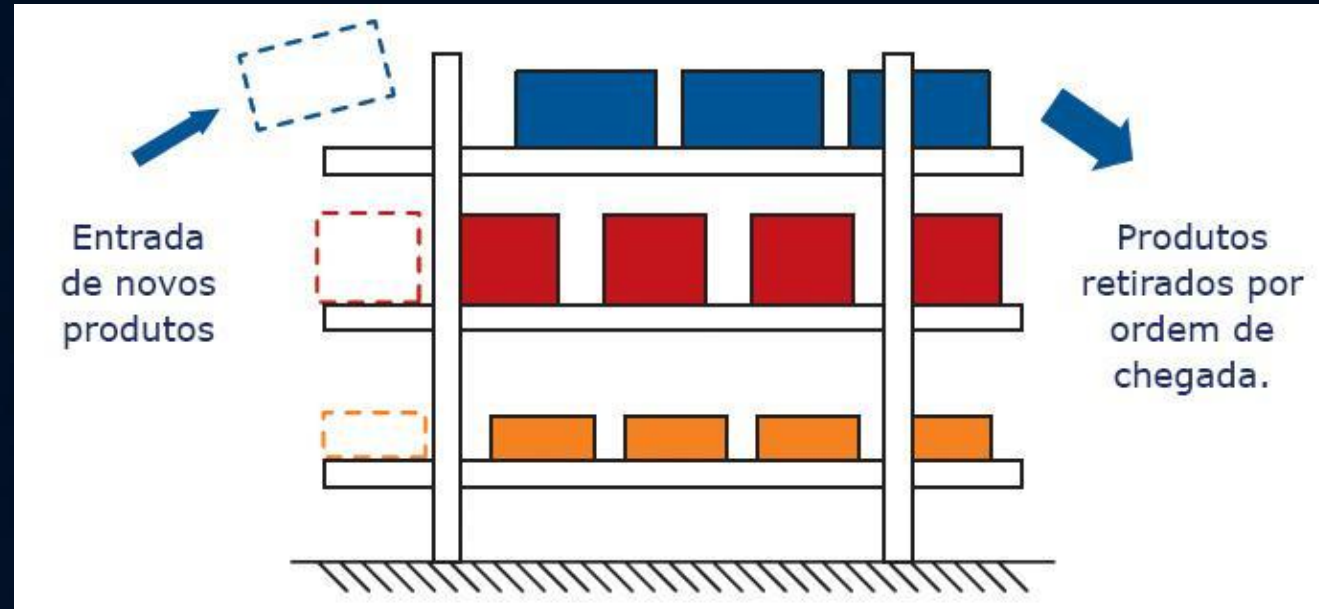
- ❖ Filas são estruturas de dados conhecidas como *primeiro a entrar*, *primeiro a sair* e armazenam os elementos em um formato sequencial.



Filas: Funcionamento



Pilhas: First-In, First-Out – FIFO



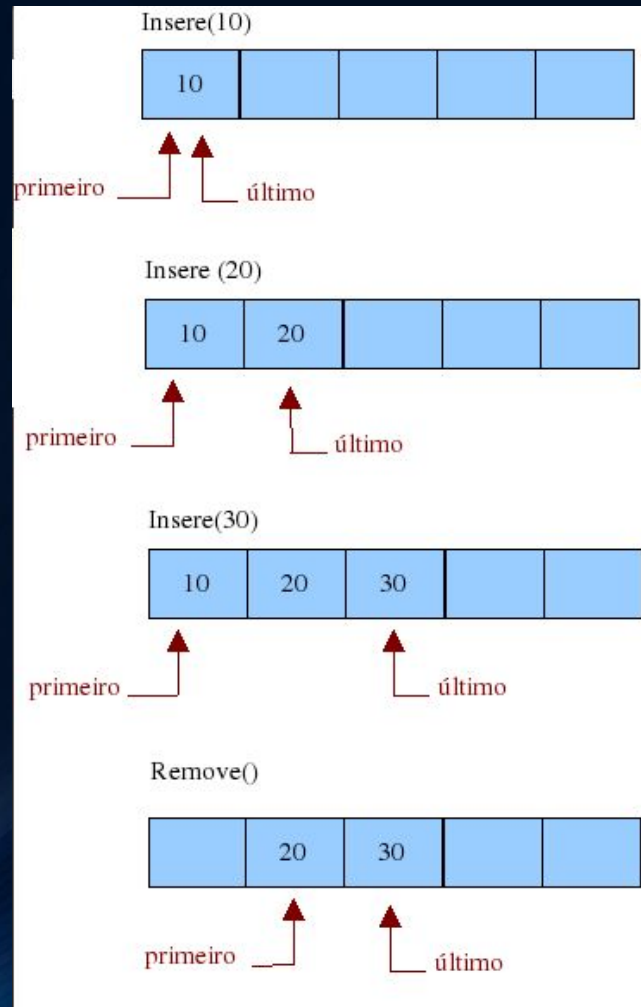
Pilhas: Aplicações

- Computadores com um único processador
- Spooling de impressão

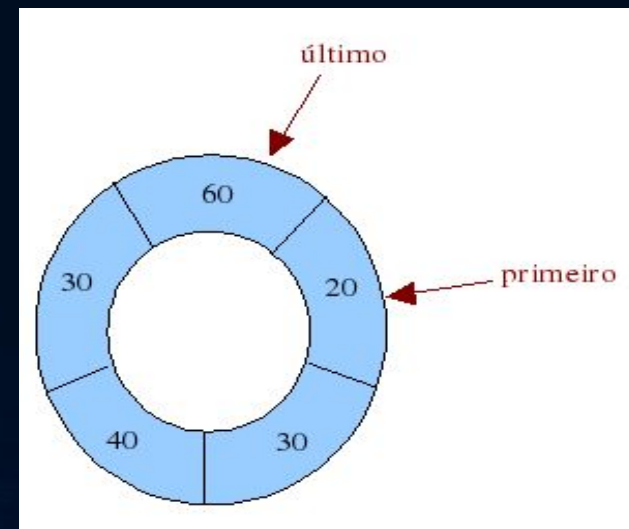
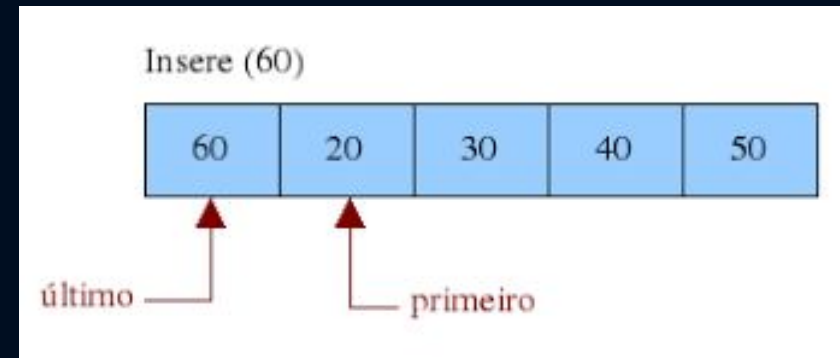


Filas: Estáticas

Linear Sequential

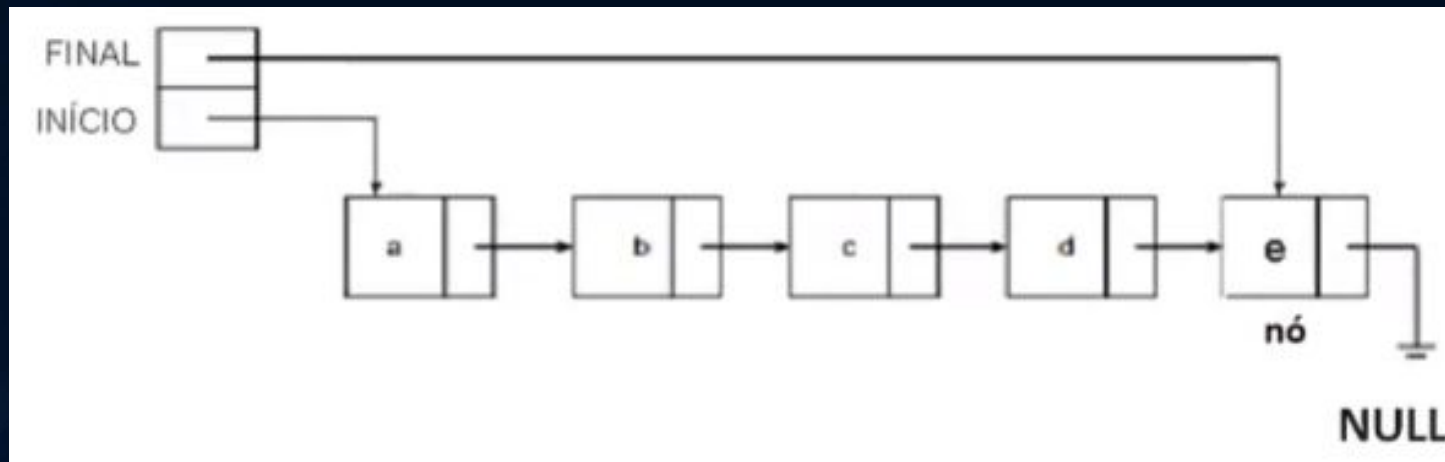
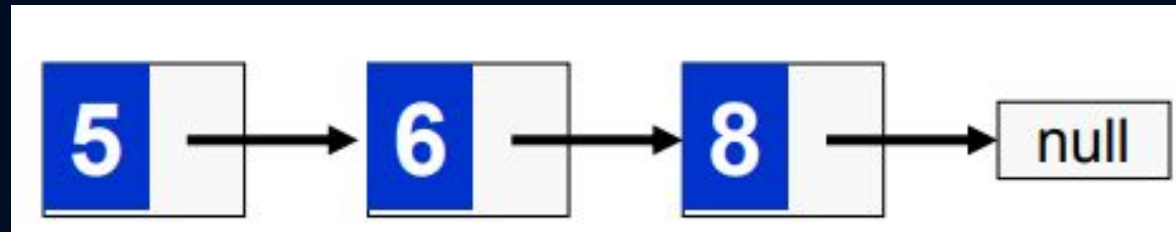


Circular



Filas: Dinâmicas

Lista encadeada



Filas: Container/Biblioteca <queue>

- ❖ Inclusão da biblioteca

`#include <queue>`

- ❖ Definição da fila

`queue<tipo da fila> nome da fila;`

Filas : Container/Biblioteca <queue>

- ❖ **push ();** - Adiciona elementos
- ❖ **pop ();** - Remove elementos
- ❖ **empty ();** - Verifica se a fila está vazia.
- ❖ **front();** - Elemento da frente
- ❖ **size();** - Tamanho da fila
- ❖ **back();** - Último elemento



MELO, Rubens de. Flutter para iniciantes. Navegação, [S. l.], p. 1, 6 ago. 2020. Disponível em: <https://www.flutterparainiciantes.com.br/basico/navegacao>. Acesso em: 16 nov. 2020.

REIS , Fábio dos. Estruturas de Dados. O que são Estruturas de Dados, Bóson Treinamentos em Ciência e Tecnologia, p. 1, 4 mar. 2017. Disponível em: <http://www.bosontreinamentos.com.br/estruturas-de-dados/estruturas-de-dados/>. Acesso em: 14 nov. 2020.i

RANGEL, W. Celes e J. L. Pilhas. Estruturas de Dados –PUC-Rio, [S. l.], p. 1-8, 10 ago. 2008. Disponível em: <https://www.ic.unicamp.br/~ra069320/PED/MC102/1s2008/Apostilas/Cap11.pdf>. Acesso em: 14 nov. 2020.

Deitel, H.M., 1945- . C++ : como programar / H.M. Deitel, P.J. Deitel ; tradução Edson Furmankiewicz ; revisão técnica Fábio Lucchini. — São Paulo: Pearson Prentice Hall, 2006.