Algoritmos
e
Estrutura de Dados

Profª. Angela Abreu Rosa de Sá, Drª.

Contato: angelaabreu@gmail.com

Dinâmica

AGOSTO / 2022

D	S	T	Q	Q	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

SETEMBRO / 2022

D	S	T	Q	Q	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	25
25	26	27	28	29	30	

Avaliação 1

OUTUBRO / 2022

D	S	T	Q	Q	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

NOVEMBRO / 2022

D	S	T	Q	Q	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Avaliação 2

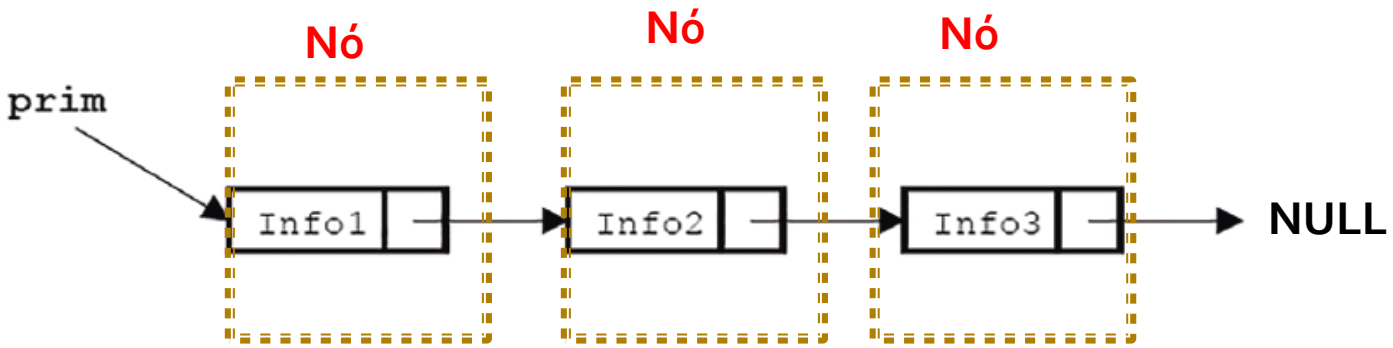
Algoritmos e Estrutura de Dados

Sumário

Algoritmos e Estrutura de Dados

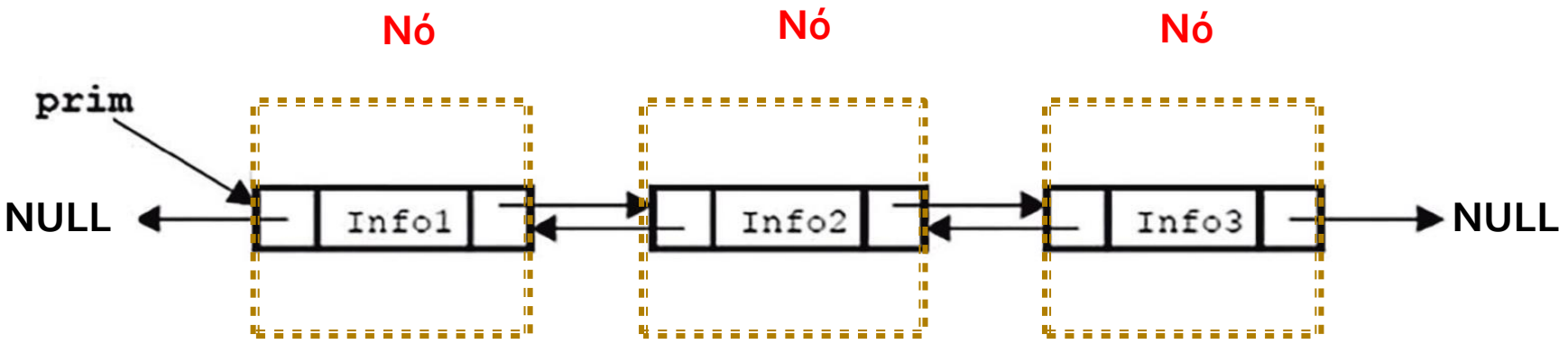
Sumário

Lista Encadeada




```
struct noLista
{
   int informacao;
   struct noLista *proximo;
};
```

Lista Duplamente Encadeada




```
struct noLista
{
   struct noLista *NoAnterior;
   int informacao;
   struct noLista *proximoNo;
};
```

Fila

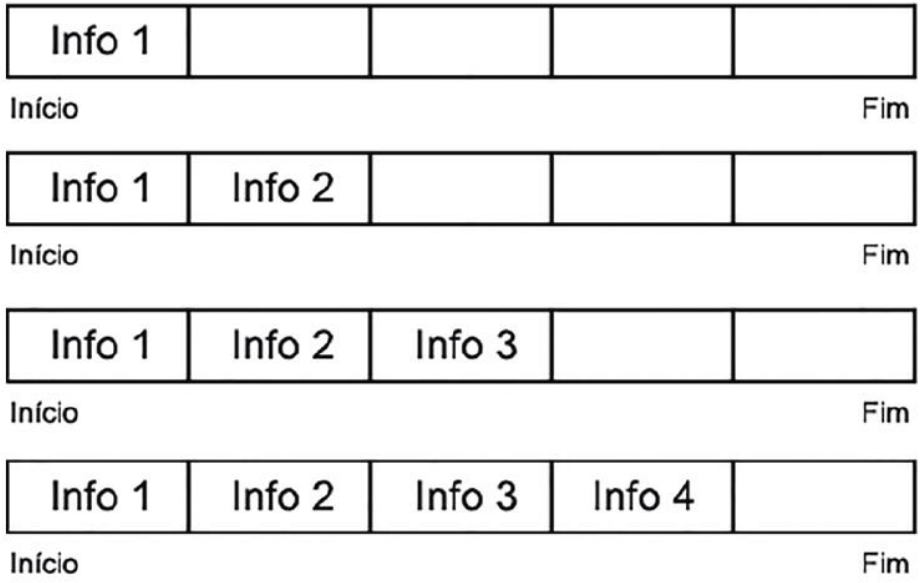
Fila é a representação de um conjunto de elementos no qual podemos *remover esses elementos* por uma extremidade chamada de inicio da fila. Já a outra extremidade, onde são inseridos os elementos, e conhecida como final da fila.

Figura 2.7 | Fila em guichê de aeroporto

Fila

Inserindo elementos na Fila

Figura 2.8 | Entrada de elemento na fila pelo seu final.

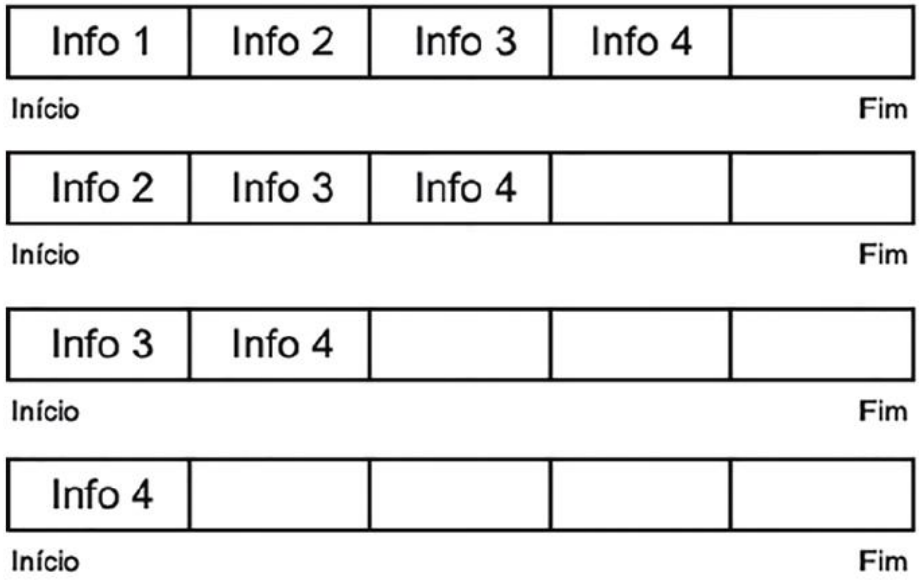


FIFO – ***First In, First Out*** **(o primeiro a entrar, é o primeiro a sair)**

Fila

Removendo elementos da Fila

Figura 2.9 | Saída de elemento pelo início da fila



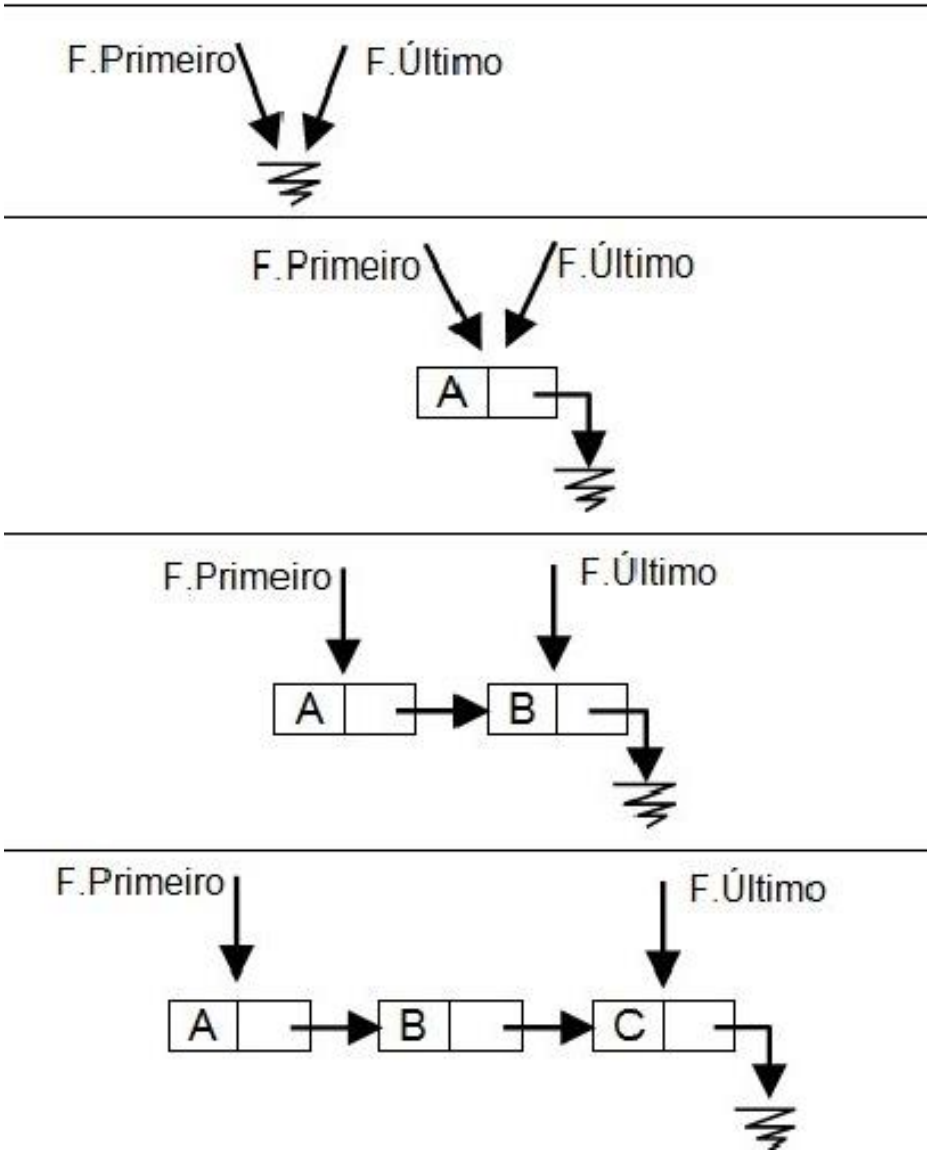
FIFO – ***First In, First Out*** **(o primeiro a entrar, é o primeiro a sair)**

Definição da estrutura da FILA

```c
#include <stdio.h>
#include <stdlib.h>

struct noFila
{
  struct noFila *NoAnterior;
  int informacao;
  struct noFila *proximoNo;
};
```

Fila em Lista Encadeada (DINÂMICA)



Operações com FILAS

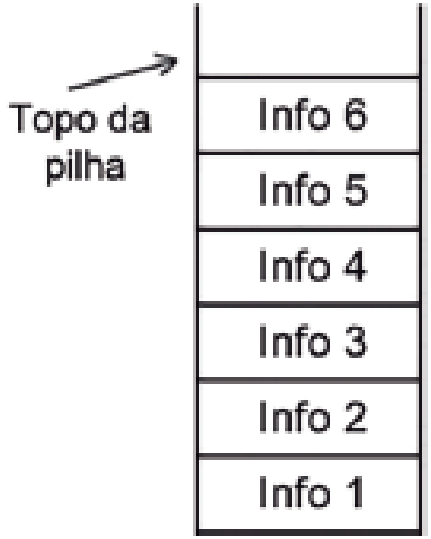
- **Inserir elementos na Fila** insere sempre no final

- **Remover elemento da Fila** remove sempre do início

- **Saber quantos elementos tem na Fila**

Pilha

Uma pilha tem como definição básica um conjunto de elementos que permite a **inserção e a remoção** de elementos em apenas uma das extremidades da estrutura denominada topo da pilha



Pilha

Os elementos inseridos em uma pilha possuem uma sequencia de Inserção: O **primeiro elemento que entra** na pilha só pode ser **removido por ultimo**, após todos os outros elementos serem removidos.

LIFO – *Last In, First Out* **(o último a entrar, é o primeiro a sair)**
FILO – *First In, Last Out* **(o primeiro a entrar, é o último a sair)**

Pilha

Inserindo elementos na Pilha

Figura 2.4 | Inserindo elemento na pilha

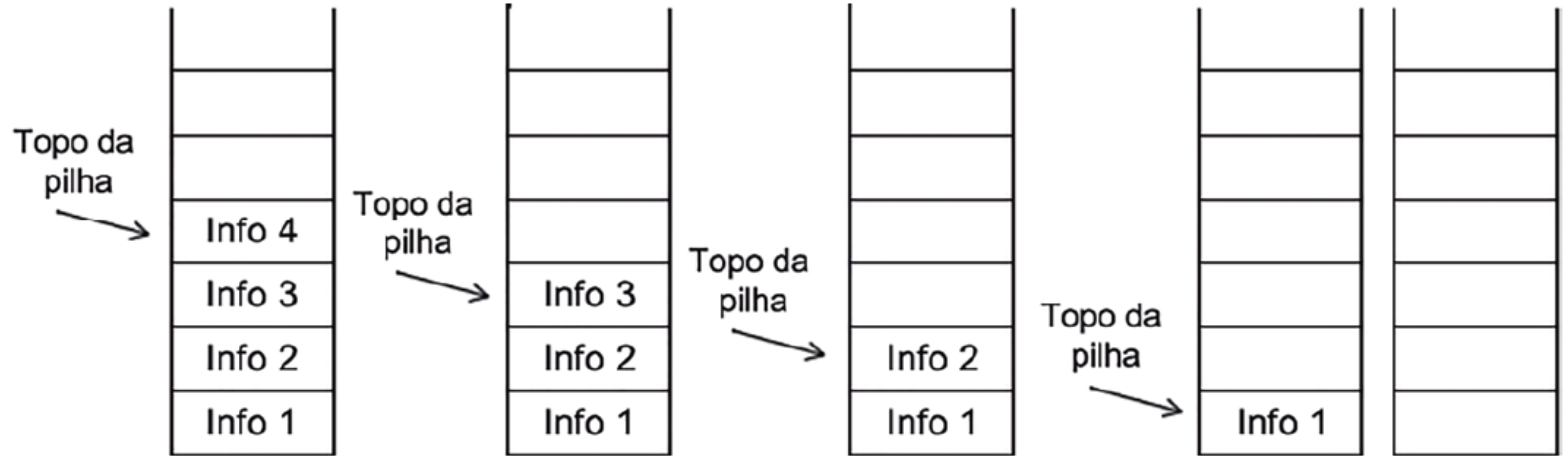
Fonte: elaborada pelo autor

LIFO – *Last In, First Out* **(o último a entrar, é o primeiro a sair)**
FILO – *First In, Last Out* **(o primeiro a entrar, é o último a sair)**

Pilha

Removendo elementos da Pilha

Figura 2.5 | Removendo um elemento da pilha



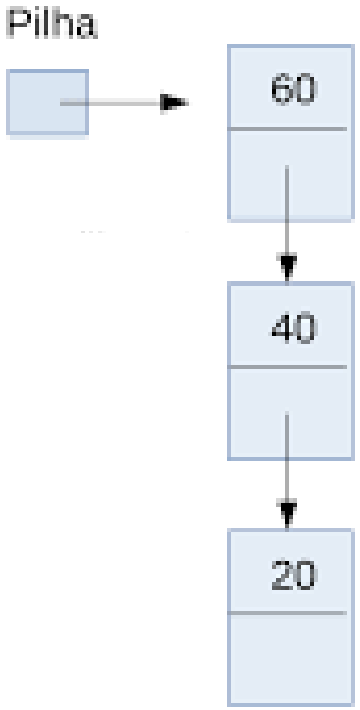
LIFO – *Last In, First Out* **(o último a entrar, é o primeiro a sair)**
FILO – *First In, Last Out* **(o primeiro a entrar, é o último a sair)**

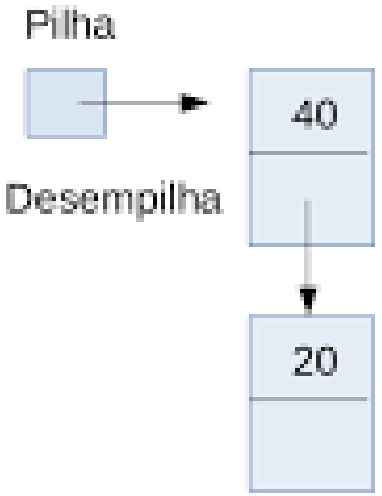
Definição da estrutura da PILHA

```c
#include <stdio.h>
#include <stdlib.h>

struct noPILHA
{
   struct noPILHA *NoAnterior;
   int informacao;
   struct noPILHA *proximoNo;
};
```

Pilha em Lista Encadeada (Dinâmica)





Operações com PILHAS

- **EMPILHAR (PUSH)** → **Inserir elementos na Pilha -** insere sempre no início

- **DESEMPILHAR (POP)** → **Remover elemento da Pilha -** remove sempre do início

- **Saber quantos elementos tem na Pilha**

Exemplo

2) Armazene numa Pilha os seguintes dados de 10 pessoas: **idade e sexo**. Após a leitura dos dados, imprima o conteúdo da Pilha.

Em seguida, **desempilhe os dados** e mostre apenas as informações das pessoas do sexo Feminino.

```c
#include <stdio.h>
#include <stdlib.h>

struct noPILHA
{
   struct noPILHA *NoAnterior;
   int idade;
   char sexo;
   struct noPILHA *proximoNo;
};
```

```c
struct noPILHA* Empilhar(struct noPILHA* InicioPILHA, int idade,  char sexo)
{

    //alocar memória para o novo Nó da pilha
    struct noPILHA* novoNo = (struct noPILHA*) malloc(sizeof(struct noPILHA));

    //O campo NoAnterior será NULL, pois agora ele será o primeiro da pilha
    novoNo->NoAnterior = NULL;
    //Inserir as informação para o novo nó
    novoNo->idade = idade;
    novoNo->sexo = sexo;
    //Apontar o campo "próximo" do novo nó para o local que o InicioDaPilha apontava
    novoNo->proximoNo = InicioPILHA;


    if(InicioPILHA != NULL) // se a pilha não estiver vazia, ligar o nó anterior do início da p
       InicioPILHA->NoAnterior = novoNo;

    return novoNo;
}
```

```c
struct noPILHA* Desempilhar (struct noPILHA* topoPILHA)
{
    struct noPILHA* PercorrePilha = topoPILHA; //cópia do início da pilha

    if (PercorrePilha == NULL ) //não tem elemento para ser removido
        return topoPILHA;

    //Remover o primeiro da Fila
    topoPILHA = PercorrePilha->proximoNo; //atualizar o início da pilha
    topoPILHA->NoAnterior = NULL; // o primeiro da pilha  aponta para NULL


    free(PercorrePilha); //liberar região de memória do nó removido

  return topoPILHA;
}
```

```c
void ImprimirPilha(struct noPILHA  *topoPILHA)
{
    struct noPILHA *NoAtual = topoPILHA; //copiar o endereço do primeiro nó da pilha

    while (NoAtual != NULL) //percorrer a pilha até encontrar o último nó = NULL
    {
        printf("\n %d  ",NoAtual->idade); //mostrar a informação do nó
        NoAtual = NoAtual->proximoNo; //apotar para o próximo nó da pilha
    }

    printf("\n NULL");

}
```

```c
int main()
{   struct noPILHA *topoPILHA = NULL;


    printf("*** INSERIR INFORMAÇÕES NA PILHA **** \n");
    int idade;
    char nome[30];
    char sexo;
    for(int i = 0; i < 5; i++)
    {
            printf("\n -- Digite a idade:");
            scanf("%d", &idade);

            printf("\n -- Digite a sexo:");
            scanf("%s", &sexo);

            topoPILHA = Empilhar (topoPILHA, idade,  sexo);
    }



   printf("\n\n *** ELEMENTOS DA PILHA  **** \n");
   //imprimir o conteúdo da pilha Feminino
   ImprimirPilha(topoPILHA);
```

```
    //desempilhar e empilhar as pilhas Feminino e Masculino
    while(topoPILHA!=NULL)
    {
        if (( topoPILHA->sexo == 'F') ||( topoPILHA->sexo == 'f'))
        {
            printf("\n\n -- SEXO: %c e idade %d ",topoPILHA->sexo, topoPILHA->idade);

        }

            //desempilhar
         topoPILHA = Desempilhar(topoPILHA);
    }


}
```

Exercícios 😀

Muito Obrigada!

Profª. Angela Abreu Rosa de Sá, Drª.

Contato: angelaabreu@gmail.com