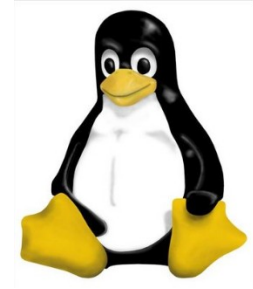


INSTITUTO FEDERAL  
BAIANO



# LP II – Estrutura de Dados

---

## Pilhas Estáticas Sequenciais

*Prof. José Honorato Ferreira Nunes*

**honoratonunes@softwarelivre.org**

**<http://softwarelivre.org/zenorato/honoratonunes>**

# Pilhas - conceitos

---

Pilhas são listas onde a inserção de um novo item ou a remoção de um item já existente se dá em uma única extremidade, no topo.

Dada uma pilha  $P = ( a(1), a(2), \dots, a(n) )$ , dizemos que  $a(1)$  é o elemento da base da pilha;  $a(n)$  é o elemento topo da pilha; e  $a(i+1)$  está acima de  $a(i)$ .

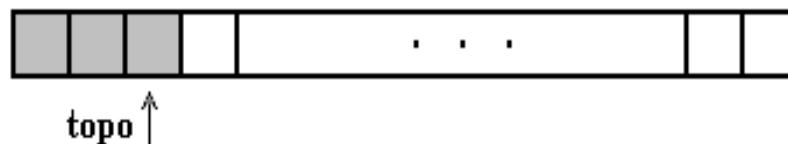
Pilhas são também conhecidas como listas LIFO (last in first out).

# Pilhas

---

Pilha é um caso especial de lista ordenada, na qual as inscrições e retiradas são feitas numa extremidade com denominação de **topo**, cujo critério de acesso é denominado de LIFO – Last In First Out.

Qualquer elemento só pode ser inserido e removido por uma extremidade da estrutura denominada **topo**.



# Declaração da Estrutura Pilha

---

A declaração da estrutura se dá mediante criação de novos tipos ou da criação de uma estrutura cujos membros podem ser um array ou vetor para armazenar os elementos da pilha e topo para informar o local de inserção, acesso e remoção do elemento da pilha.

```
#define m 10
struct pilha {
    int topo;
    char elem[m];
};
```

# Pilha Estáticas

---

Para trabalhar com a estrutura pilha, deve-se definir qual o tamanho e quais os membros, bem como os tipos de dados que a pilha irá armazenar.

Para empilhar o primeiro elemento na estrutura, primeiro deverá executar a função de inicialização da pilha.

# Operações Primitivas de Pilhas

---

- ✓ Criar a pilha;
- ✓ Inicializar a pilha;
- ✓ Empilhar um elemento;
- ✓ Acessar o elemento do topo da pilha;
- ✓ Desempilhar um elemento.

# Criar a pilha

---

Antes de começar a carregar a pilha com dados, há a necessidade de declarar a estrutura e inicializá-la.

```
//criação da pilha
```

```
#define m 10  
struct pilha {  
    int topo;  
    char elem[m];  
};
```

## Inicialização da pilha

---

O processo de inicialização é realizado por uma função de iniciar que atribui o valor de -1 a variável topo.

```
void iniciaPilha(struct pilha *P){  
    P->topo = -1;  
}
```



# Empilhar um elemento

---

Essa função recebe a pilha e o valor a ser empilhado como parâmetros. Devemos verificar se ainda há espaço no vetor para mais um elemento!

```
void empilhar(struct pilha *P, char x){
    if (P->topo == m-1)
        printf("Pilha Cheia");
    else{
        P->topo++;
        P->elem[P->topo]=x;
    }
}
```

# Acessar o elemento do topo da pilha

---

A função Topo() retorna o valor da variável topo da pilha sem desempilhar.

```
char topo(struct pilha *P){
    char x;
    if (P->topo == -1)
        printf("Pilha Vazia");
    else{
        x = P->elem[P->topo];
        return(x);
    }
}
```

# Desempilhar um elemento

---

Essa função elimina o elemento que está no topo da pilha.

```
void desempilha(struct pilha *P){
    if (P->topo == -1)
        printf("Pilha Vazia");
    else{
        P->topo--;
    }
}
```

# Atividade

---

Faça um programa que, utilizando as funções criadas nos exemplos dessa aula crie uma pilha P e exiba o seguinte menu de opções:

## **EDITOR DE PILHA**

- 1 - EMPILHAR
- 2 - DESEMPILHAR
- 3 - EXIBIR ELEMENTO DO TOPO
- 4 - EXIBIR A PILHA
- 5 - ESVAZIAR A PILHA
- 6 - SAIR

DIGITE SUA OPÇÃO:

# Bibliografia Básica

---

- ❑ SILVA, Osmar Quirino da. **Estruturas de Dados e Algoritmos Usando C – Fundamentos e Aplicações.** Rio de Janeiro, Ed: Ciência Moderna, 2007.
- ❑ MANZANO, Wilson Y. Yamaturni-São Paulo-SP. **Lógica estruturada para programação de computadores,** Ed. Érica 1997 e 2001.
- ❑ MORAES, Celso Roberto. **Estruturas de Dados e Algoritmos.** Ed. Érica, São Paulo.
- ❑ LOPES, Anita. **Introdução à programação.** Rio de Janeiro: Campus, 2002.

# Bibliografia Complementar

---

- ❑ BENEDUZZI, Humberto M. e METZ, João A. **Lógica e Linguagem de Programação – Introdução ao Desenvolvimento de Software** (1ª edição). Editora do Livro Técnico, 2010
- ❑ FORBELLONE, A. L. V. e Eberspacher, H. F. **Lógica de Programação - a Construção de Algoritmos e Estruturas de Dados** (3ª edição). Pearson, 2005
- ❑ CORMEN, Thomas H. et. al. **Algoritmos: Teoria e Prática**. Editora Campus, 2002.
- ❑ ZIVIANI, Nivio. **Projeto de Algoritmos**. Editora Nova Fronteira, 2004.
- ❑ SEBESTA, Robert W. **Conceitos de Linguagens de Programação**. Bookman, 2001.