

INSTITUTO FEDERAL
BAIANO



Estrutura de Dados

Pilhas Dinâmicas

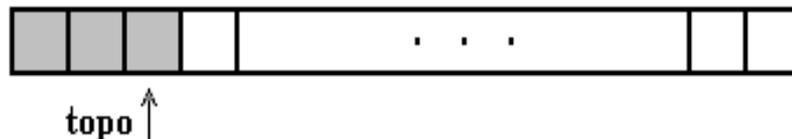
Prof. José Honorato Ferreira Nunes

honoratonunes@gmail.com

<http://softwarelivre.org/zenorato>

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.
- 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**.



Implementação de pilha com lista

- ▶ Quando o número máximo de elementos que serão armazenados na pilha não é conhecido, devemos implementar a pilha usando uma estrutura de dados dinâmica, no caso, empregando uma lista encadeada.
- ▶ Os elementos são armazenados na lista e a pilha pode ser representada simplesmente por um ponteiro para o primeiro nó da lista.

Pilhas

:: Implementação de pilha com estruturas

```
typedef struct {  
    float      info;  
    struct No* anterior;  
} No;
```

```
typedef struct {  
    No* topo;  
} Pilha;
```

Pilhas

:: Operações básicas

- ▶ Criar uma estrutura de pilha;
- ▶ Inserir um elemento no topo (push);
- ▶ Remover o elemento do topo (pop);
- ▶ Verificar se a pilha está vazia;
- ▶ Liberar a estrutura de pilha

Pilhas

:: Criar uma estrutura de pilha

```
Pilha* cria(void)
{
    Pilha *p;
    p = (Pilha*) malloc(sizeof(Pilha));
    p->topo = NULL;
    return p;
}
```

Pilhas

:: Inserir o elemento do topo – **push()**

```
Pilha* push(Pilha *p, float v)
{
    No* aux;
    aux = (No*) malloc(sizeof(No));
    aux->info = v;
    aux->anterior = p->topo;
    p->topo = aux;
    return aux;
}
```

Pilhas

:: Remover o elemento do topo – **pop()**

```
float pop(Pilha *p)
{
    float v;
    No* aux;
    if (vazia(p))
    {
        printf("Pilha vazia.");
        exit(1); /*aborta o programa*/
    }
    v      = p->topo->info;
    aux    = p->topo;
    p->topo = aux->anterior;
    free(aux);
    return v;
}
```

Pilhas

:: Verificar se a pilha está vazia

```
int vazia(Pilha *p)
{
    return (p->topo == NULL);
}
```

Pilhas

:: Liberar a estrutura de pilha

```
void libera(Pilha *p)
{
    No* q = p->topo;
    while (q != NULL)
    {
        No *t = q->anterior;
        free(q);
        q = t;
    }
    free(p);
}
```

Pilhas

:: Imprime estrutura de pilha

```
void imprime (Pilha* p) {  
    No* q;  
    for (q=p->topo; q!=NULL; q=q->anterior)  
        printf("%4.2f\n",q->info);  
}
```

Pilhas

:: Teste

```
main(){
    Pilha* p = cria();
    push(p,20.0);
    push(p,20.8);
    push(p,20.3);
    push(p,44.5);
    push(p,33.3);
    push(p,20.9);
    imprime (p);
    printf ("Retirado: %4.2f\n", pop(p));
    printf ("Retirado: %4.2f\n", pop(p));
    printf ("Configuracao da fila:\n");
    imprime (p);
    libera (p);
    system("pause");
}
```

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.