

Programação 1

Pilhas

Rafael Vargas Mesquita

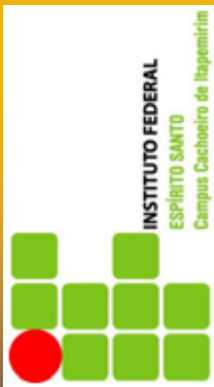


linguagem

Pilhas

◆ Definição

- A pilha é uma das estruturas de dados mais simples. Nela todo acesso a seus elementos é feito pelo topo. Assim, quando inserimos um novo elemento ele passa a ser o elemento do topo e o único elemento que pode ser removido da pilha é o do topo.
- As pilhas são muito utilizadas na programação devido a sua simplicidade, sendo inclusive implementadas em hardware pela maioria das máquinas modernas.



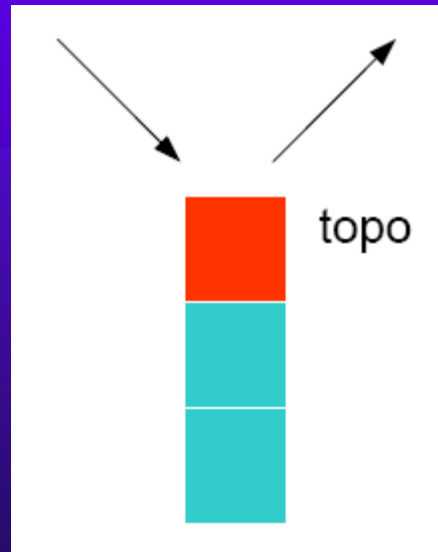


linguagem

Pilhas

♦ Representação Gráfica

- Uma pilha pode ser graficamente representada pela figura a seguir:





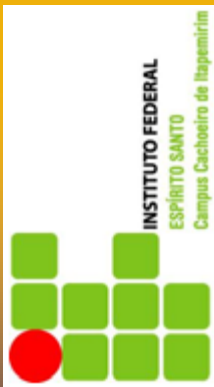
linguagem

Pilhas

◆ Operações Básicas

– As operações básicas de uma pilha são:

- Push (empilhar): inserir um novo elemento na pilha.
- Pop (desempilhar): retirar o elemento do topo da pilha.



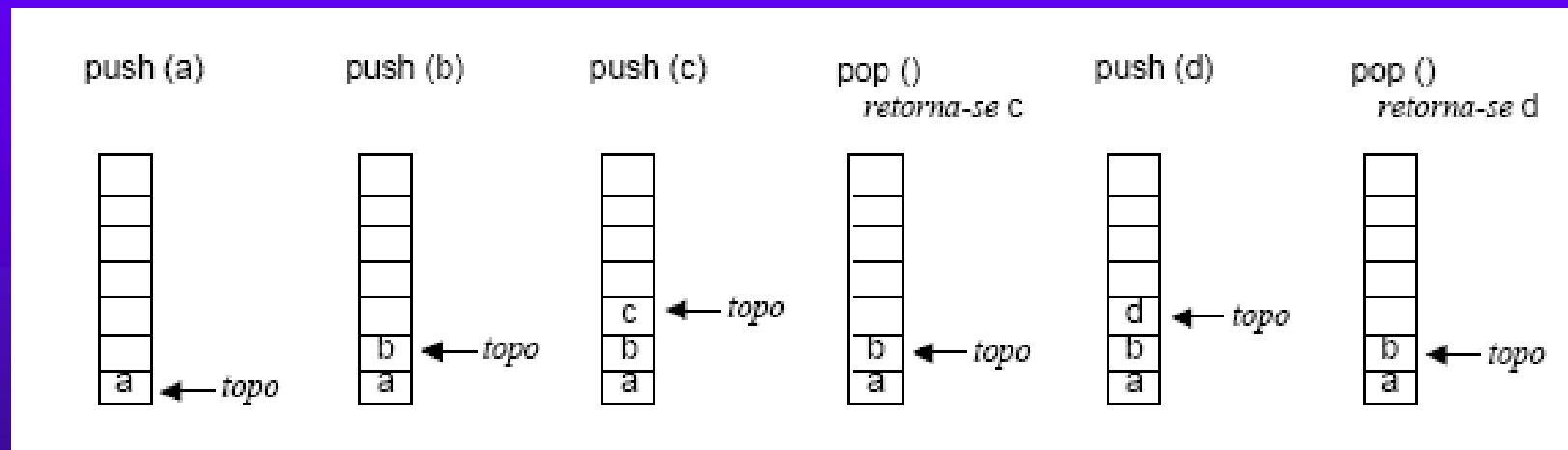


linguagem

Pilhas

◆ Operações Básicas

- A seguinte figura ilustra as operações de Push e Pop:





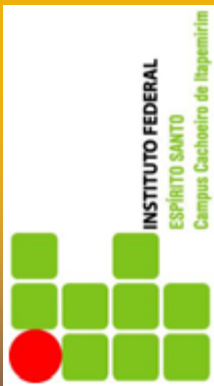
linguagem

Pilhas

♦ Formas de Implementar uma Pilha

- Podemos implementar uma pilha de várias formas diferentes, por exemplo:
 - Pilha implementada com vetor.
 - Pilha implementada com lista simplesmente encadeada.

Obs.: Vamos abordar a implementação com lista



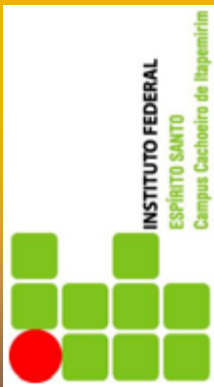


linguagem

Pilhas

♦ Funcionalidades de uma Pilha

- Para utilizarmos uma pilha é necessário implementar as seguintes funcionalidades:
 - – inicializa: inicializa a pilha.
 - – push: insere um elemento na pilha.
 - – pop: retira um elemento da pilha
 - – vazia: determina se a pilha está vazia.
 - – libera: desaloca os elementos da pilha.





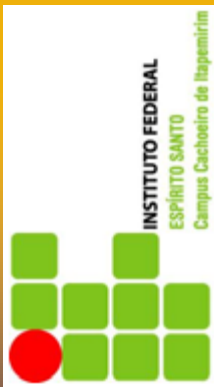
linguagem

Pilha – Implementação com Lista

◆ Declaração

```
typedef struct no {  
    int info;  
    struct no* prox;  
} No;
```

```
typedef struct pilha {  
    No* prim;  
} Pilha;
```





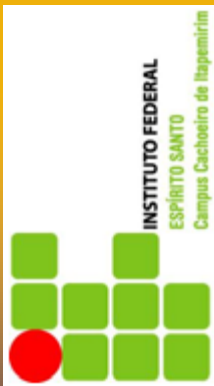
linguagem

Pilhas

◆ Funções

– Função de Inicialização de uma Pilha

```
/* função de inicialização: retorna pilha vazia */  
Pilha* inicializa () {  
    Pilha *nova = (Pilha *) malloc(sizeof(Pilha));  
  
    /* Inicializa os dados */  
    nova->prim = NULL;  
    return nova;  
}
```





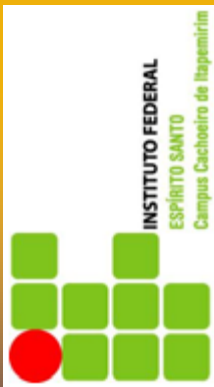
linguagem

Pilhas

◆ Funções

– Função de Inserção de Elementos na Pilha

```
/* inserção no início */  
void push (Pilha *p, int v){  
    No *novo = (No*) malloc(sizeof(No));  
    novo->info = v;  
    novo->prox = p->prim;  
  
    p->prim = novo;  
}
```





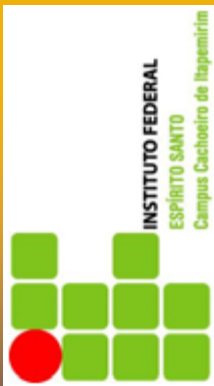
linguagem

Pilhas

◆ Funções

– Função de Remoção de Elemento da Pilha

```
int pop (Pilha* p){  
    No* novo_prim;  
    No* removido;  
    int valor;  
  
    removido = p->prim;  
    novo_prim = p->prim->prox;  
  
    valor = p->prim->info;  
    p->prim = novo_prim;  
    free(removido);  
    return valor;  
}
```





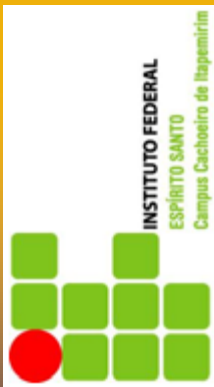
linguagem

Pilhas

◆ Funções

– Função de Impressão de uma Pilha

```
/* função imprime: imprime valores dos elementos */  
void imprime (Pilha *p){  
    No *aux;  
  
    for (aux = p->prim; aux != NULL; aux = aux->prox)  
        printf("\t\tInfo = %d\n", aux->info);  
}
```





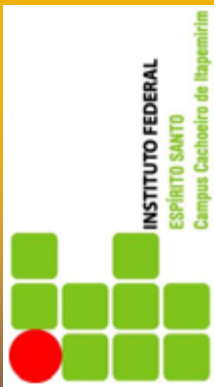
linguagem

Pilhas

◆ Funções

– Função de Verificação de Pilha Vazia

```
/* função vazia: retorna 1 se vazia ou 0 se não vazia */  
int vazia (Pilha *p){  
    return (p->prim == NULL);  
}
```





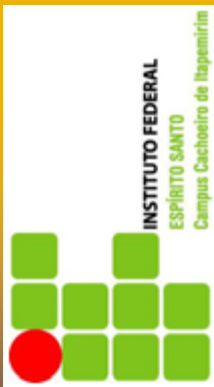
linguagem

Pilhas

◆ Funções

– Função de Liberação de uma Pilha

```
void libera (Pilha* p){  
    Pilha* aux = p;  
    while (aux != NULL) {  
        /* guarda referência para o próximo elemento */  
        Pilha* t = aux->prox;  
        free(aux); /* libera a memória apontada por aux */  
        aux = t; /* faz p apontar para o próximo */  
    }  
}
```





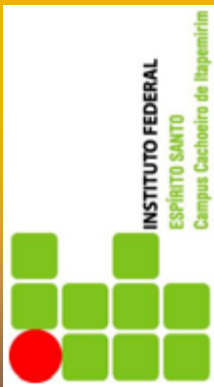
Pilhas

◆ Funções

- A seguir um pequeno trecho de código utilizando as funcionalidades descritas até esse slide:

```
#include <stdio.h>

int main(){
    Pilha *pilha;
    pilha = inicializa();
    push(pilha, 10);
    push(pilha, 20);
    if (! Vazia(pilha) )
        printf("O primeiro número é: %f \n", pop(pilha));
    if (! Vazia(pilha) )
        printf("O segundo número é: %f \n", pop(pilha));
    libera(pilha);
}
```





linguagem

Bibliografia

- ♦ SANTOS, Henrique José. Curso de Linguagem C da UFMG, apostila.
- ♦ FORBELLONE, André Luiz. Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados. São Paulo: MAKRON, 1993.

