



Pilha Encadeada

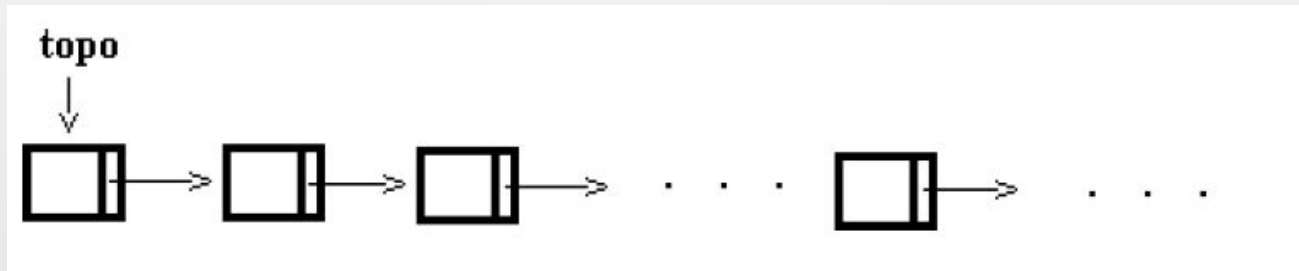


Implementação de Pilhas

- Encadeada
 - Pilha é representada pelo elemento topo
 - Cada nó armazena
 - Informação (dados)
 - Ponteiro para próximo elemento



Pilha Encadeada





Operações

- PUSH – Insere no topo da pilha
- POP – Remove o topo da pilha
- Top – Pega o elemento do topo (sem remover)
- Tamanho;
- Verificar se está vazia (empty)
- Inicializar



Exemplo para Pilha com header

```
typedef struct{  
    char nome[100];  
    char email[150];  
    char fone[15];  
} contato;
```

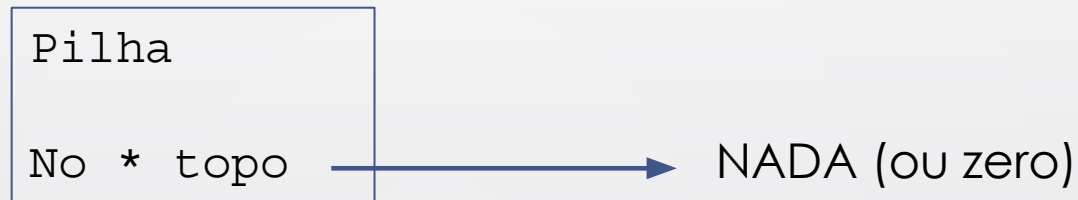
```
typedef struct{  
    contato c;  
    struct no* prox;  
} no;
```

```
typedef struct{  
    no *topo;  
} pilha;
```



Exemplo para Pilha com header

```
typedef struct{  
    char nome[100];  
    char email[150];  
    char fone[15];  
} contato;
```



```
typedef struct{  
    contato c;  
    struct no* prox;  
} no;
```

PILHA VAZIA

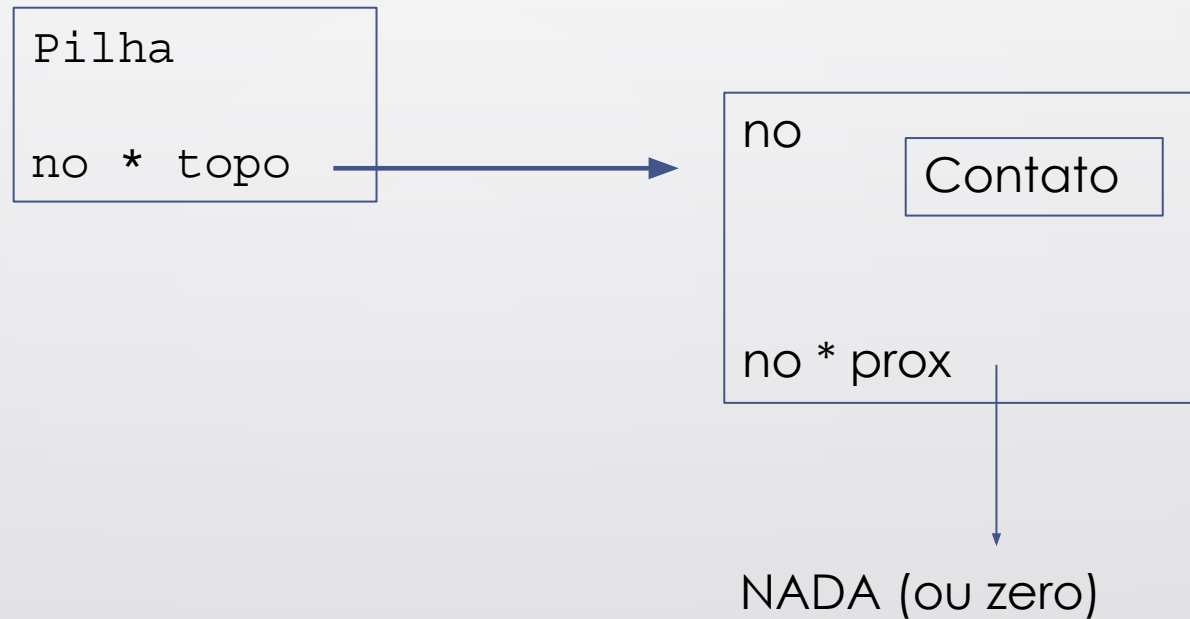
```
typedef struct{  
    no *topo;  
} pilha;
```

Exemplo para Pilha com header

```
typedef struct{
    char nome[100];
    char email[150];
    char fone[15];
} contato;
```

```
typedef struct{
    contato c;
    struct no* prox;
} no;
```

```
typedef struct{
    no *topo;
} pilha;
```



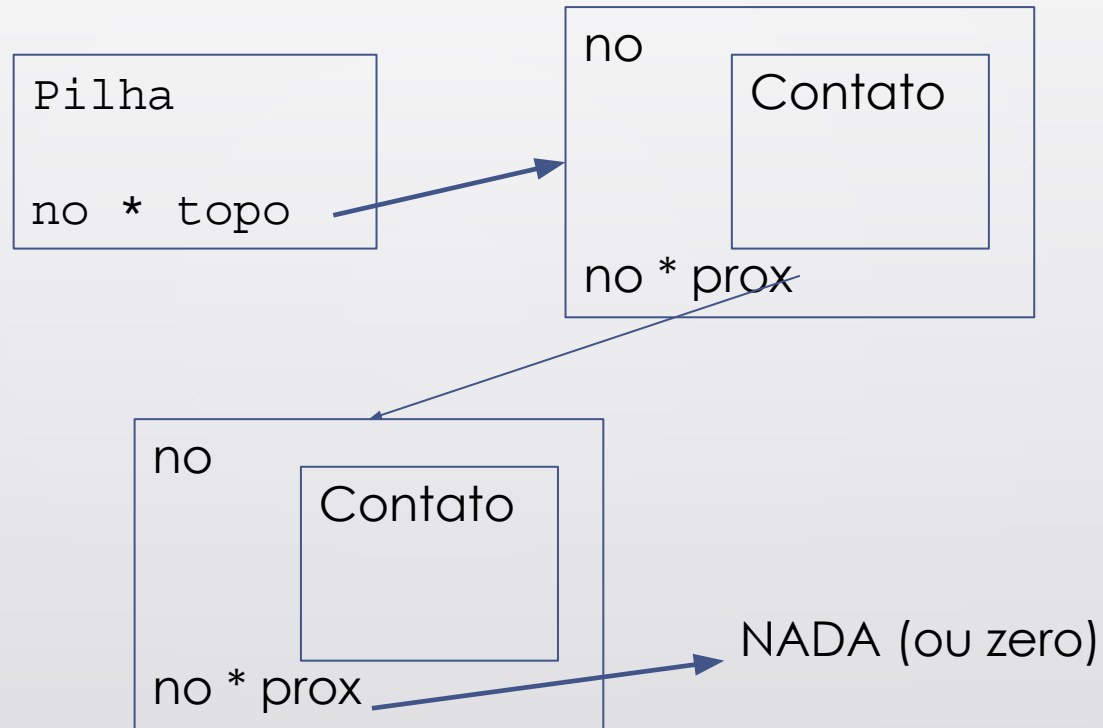
PILHA com 1 elemento

Exemplo para Pilha com header

```
typedef struct{  
    char nome[100];  
    char email[150];  
    char fone[15];  
} contato;
```

```
typedef struct{  
    contato c;  
    struct no* prox;  
} no;
```

```
typedef struct{  
    no *topo;  
} pilha;
```



PILHA com 2 elementos



Inserir na Pilha (Empilhar - push)

- Alocar Memória
 - Verificar se alocação ocorreu
- inserir/copiar dados no novo nó
- Novo aponta para topo antigo
- Topo aponta para novo



Remoção da Pilha (Desempilhar - pop)

- Verificar se não está vazia
- Armazenar dado do topo
- Topo aponta para próximo do topo;
- Liberar memória do topo removido



Dica da Função remover

A função remover normalmente precisa “devolver” 2 informações:

1. conteúdo removido
 - a. normalmente através de um ponteiro passado como parâmetro
 - b. semelhante à função `scanf ("%d", &inteiro)`
2. status, indicar se a função foi realizada com sucesso
 - a. normalmente através de um inteiro
 - b. (1 para indicar sucesso; 0 para indicar falha)

Possível assinatura:

- `int remover (Pilha *p, Conteudo *c);`
- `int pop (Pilha *p, Conteudo *c);`



Dica para Função Inserir (push)

- Se as informações a serem inseridas são strings (vetor de char), você **não pode usar a atribuição com o operador =**
- Neste caso para copiar as informações do tipo string, você deve usar **strcpy**
- Possível assinatura para função Inserir
 - `int inserir(Pilha *p, Conteudo C);`
 - `int push(Pilha *p, Conteudo C);`
- nos casos acima, o int deve indicar se a operação foi realizada corretamente; (1 para indicar sucesso; 0 para indicar falha)



Atividade

- Criar uma pilha encadeada em C, que realize os passos abaixo:
 - 1- inserir 3 contatos;
 - 2- remover um Contato e mostrar na tela os dados
 - 3 - remover um Contato e mostrar na tela os dados
 - 4 - empilhar (inserir) um novo contato
 - 5 - remover um Contato e mostrar na tela os dados
 - 6 - remover um Contato e mostrar na tela os dados
 - 7 – tentar remover contato (deve dar erro)