



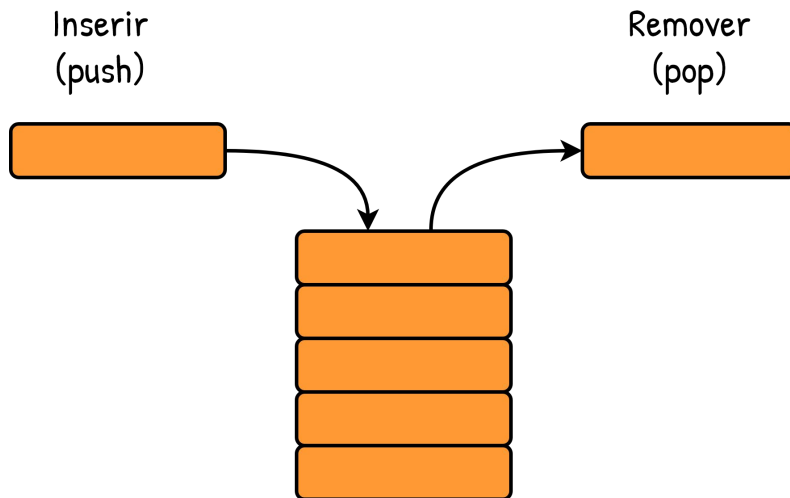
Ciência da Computação
Algoritmos e Estrutura de Dados 1

Pilha

Definição

Lista linear em que as operações de **inserção** e **remoção** ocorrem na **mesma extremidade**

LIFO: Last In First Out



Aplicações

- Avaliação de Expressões
 - $10 * (5 + 2)$
- Casamento de delimitadores
 - $(2 * (5 - 2)) * (20 + 6)$
- Conversão de expressões
 - infix $5 + 2$
 - prefix $+ 5 2$
 - postfix $5 2 +$
- Gerenciamento de Memória
- Backtracking

TAD Pilha

Dados

```
typedef int TipoElemento;  
typedef struct pilha Pilha;
```

Operações

```
Pilha* pilha_criar();  
void pilha_destruir(Pilha** endereco);
```

—————→ CRIAR E DESTRUIR A PILHA

```
bool pilha_empilhar(Pilha* p, TipoElemento elemento);  
bool pilha_desempilhar(Pilha* p, TipoElemento* saida);
```

—————→ INSERIR E REMOVER UM ELEMENTO NA PILHA

```
bool pilha_topo(Pilha* p, TipoElemento* saida);
```

—————→ ACESSA E DEVOLVE O ELEMENTO QUE ESTÁ NO
TOPO DA PILHA

```
bool pilha_vazia(Pilha* p);
```

—————→ VERIFICA SE A PILHA ESTÁ VAZIA

```
bool pilha_toString(Pilha* p, char* saida);
```

—————→ DEVOLVE UMA STRING QUE REPRESENTA A PILHA

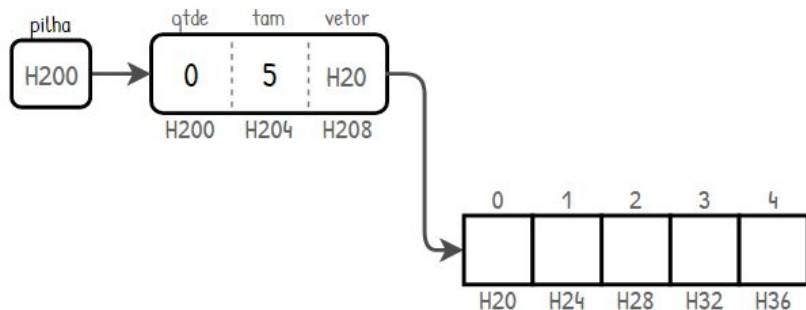
TAD Pilha

```
typedef int TipoElemento;  
/  
* DADOS  
*****/  
typedef struct pilha Pilha;  
  
/  
* PROTÓTIPOS  
*****/  
Pilha* pilha_criar();  
void pilha_destruir(Pilha** endereco);  
bool pilha_empilhar(Pilha* p, TipoElemento elemento);  
bool pilha_desempilhar(Pilha* p, TipoElemento* saida);  
bool pilha_topo(Pilha* p, TipoElemento* saida);  
bool pilha_vazia(Pilha* p);  
void pilha_toString(Pilha* p, char* saida);
```

Organização dos dados na memória

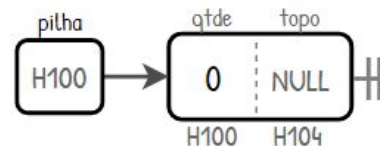
Contígua

```
struct pilha {  
    int qtde;  
    int tam;  
    TipoElemento* vetor;  
};
```



Encadeada

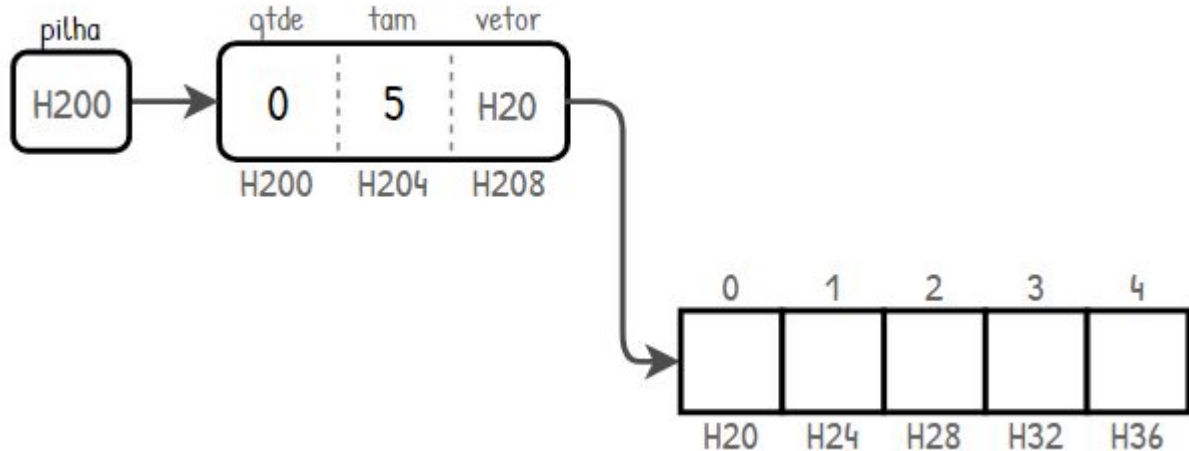
```
typedef struct no{  
    TipoElemento dado;  
    struct no *prox;  
}No;  
  
struct pilha{  
    int qtde;  
    No *topo;  
};
```



Organização Contígua

Especificação da struct

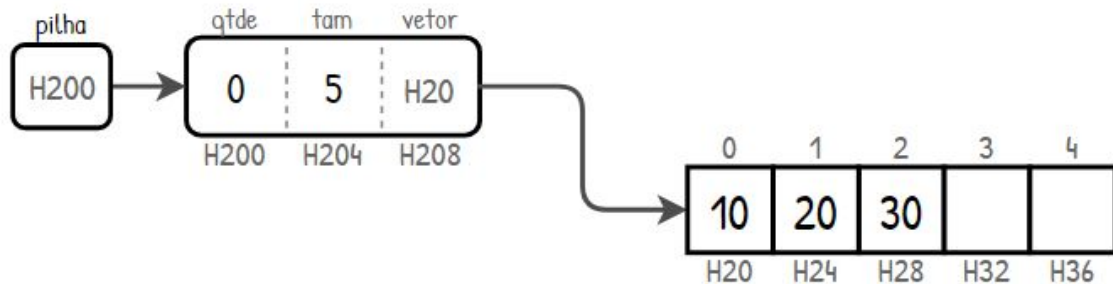
```
struct pilha {  
    int qtde;  
    int tam;  
    TipoElemento* vetor;  
};
```



Organização Contígua

```
#include "pilha.h"
/*****
 * DADOS
 *****/
struct _pilha {
    TipoElemento* vetor;
    int tamVetor;
    int qtdeElementos;
};

/*****
 * IMPLEMENTAÇÃO
 *****/
Pilha* pilha_criar();
void pilha_destruir(Pilha** endereco);
bool pilha_empilhar(Pilha* p, TipoElemento elemento);
bool pilha_desempilhar(Pilha* p, TipoElemento* saida);
bool pilha_topo(Pilha* p, TipoElemento* saida);
bool pilha_vazia(Pilha* p);
void pilha_toString(Pilha* p, char* saida);
```

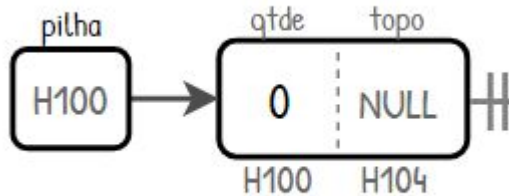


Organização Encadeada

Especificação da struct

```
typedef struct no{
    TipoElemento dado;
    struct no    *prox;
}No;

struct pilha{
    int qtde;
    No *topo;
};
```



Organização Encadeada

```
#include "pilha.h"
/*****
 * DADOS
 *****/
typedef struct no{
    TipoElemento dado;
    struct no    *prox;
}No;

struct pilha{
    int qtde;
    No *topo;
};
/*****
 * IMPLEMENTAÇÃO
 *****/
Pilha* pilha_criar();
void pilha_destruir(Pilha** endereco);
bool pilha_empilhar(Pilha* p, TipoElemento elemento);
bool pilha_desempilhar(Pilha* p, TipoElemento* saida);
bool pilha_topo(Pilha* p, TipoElemento* saida);
bool pilha_vazia(Pilha* p);
void pilha_toString(Pilha* p, char* saida);
```

