

# Funções do TAD Pilha

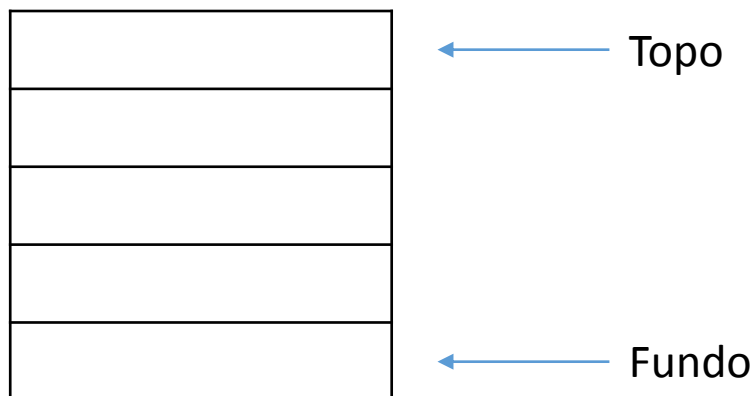
Prof. D.Sc. Saulo Ribeiro

# Funções do TAD Lista

- Pilha inicPilha( void ) : cria e retorna uma pilha vazia
- void adicPilha( Pilha p, TipoP elem): adiciona(empilha) um elemento no topo da pilha
- void elimPilha( Pilha p): elimina(desempilha) o elemento que está no topo da pilha
- TipoP infoPilha(Pilha p): retorna o elemento que está no topo da pilha
- int vaziaPilha(Pilha p) : retorna verdadeiro se a pilha está vazia e falso caso contrário
- void destruirPilha(Pilha p): destrói a pilha p, desalocando toda memória ocupada

# Notação

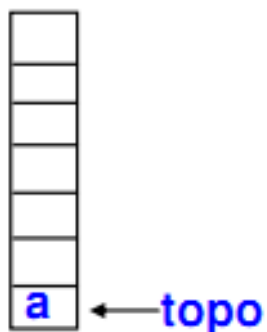
- Uma pilha é uma sequencia de zero ou mais elementos do mesmo tipo.
- A pilha só pode crescer(empilhar) e decrescer(desempilhar) por uma de suas extremidades(topo da pilha).



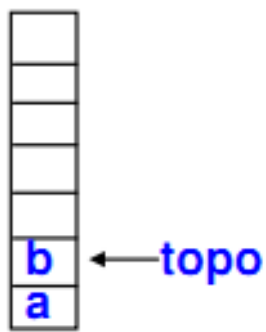
# Notação

- novo elemento é inserido no topo e acesso é apenas ao topo
  - o primeiro que sai é o último que entrou (LIFO – *last in, first out*)
- operações básicas:
  - empilhar (*push*) um novo elemento, inserindo-o no topo
  - desempilhar (*pop*) um elemento, removendo-o do topo

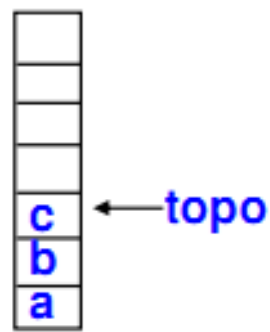
push (a)



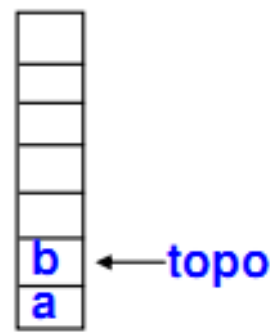
push (b)



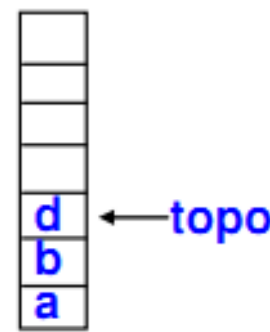
push (c)



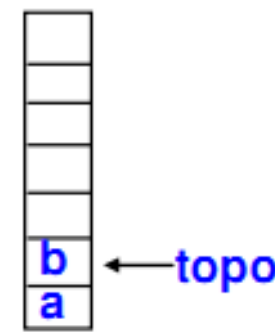
pop ()  
retorna c



push (d)



pop ()  
retorna d



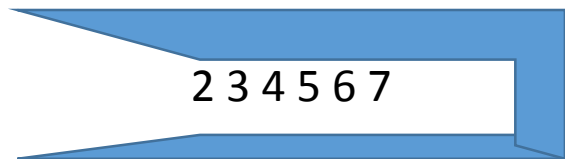
# Notação



- Pilha vazia: .



# Notação

- **Pilha cheia (com elementos):** Pilha com 6 elementos. No topo está o elemento 2
- 
- E no fundo o elemento 7

# Notação

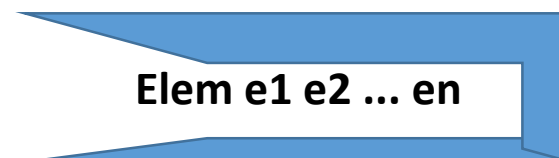
- Pilha inicPilha( void ) : cria e retorna uma pilha vazia
- Pos: inicPilha = { }



# Notação

- void adicPilha( Pilha p, TipoP elem): adiciona(empilha/push) um elemento no topo da pilha

- Pos: inicPilha =





# Notação

- void elimPilha( Pilha p): elimina(desempilha/pop) o elemento que está no topo da pilha

- Pre: p =

e1 e2 ... en

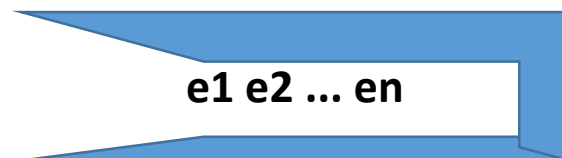
- Pos: p =

e2 ... en

# Notação

- TipoP infoPilha(Pilha p): retorna o elemento que está no topo da pilha

- Pre:  $n > 0$



- Pos: infoPilha = e1

# Notação

- `int vaziaPilha(Pilha p)` : retorna verdadeiro se a pilha está vazia e falso caso contrário
- Pos: `vaziaPilha = (p = { })`

# Notação



- void destruirPilha(Pilha p): destrói a pilha p, desalocando toda memória ocupada
- Pos: p foi destruído (desalocado)

# Implementação

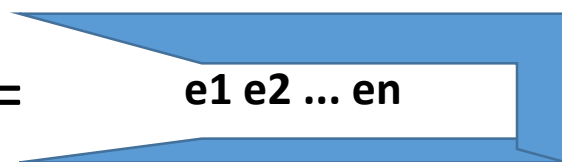


- Vetor
- Lista
  - Basta sempre por o iterador no no primeiro elemento da lista e fazer as operações: insLista, elimLista, infoLista.

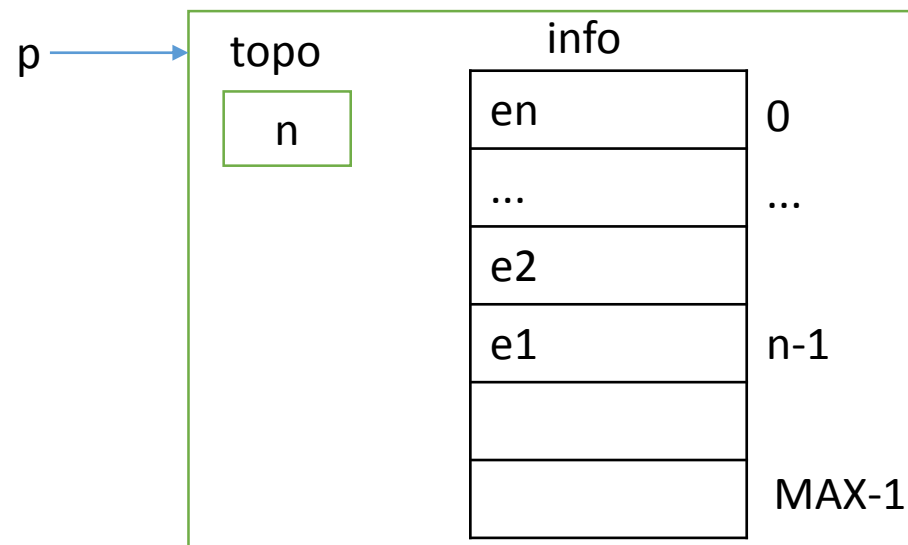
# Implementação: Vetor

- Um vetor é uma boa forma de representar uma pilha se o número máximo de elementos é conhecido.

- A pilha  $p =$

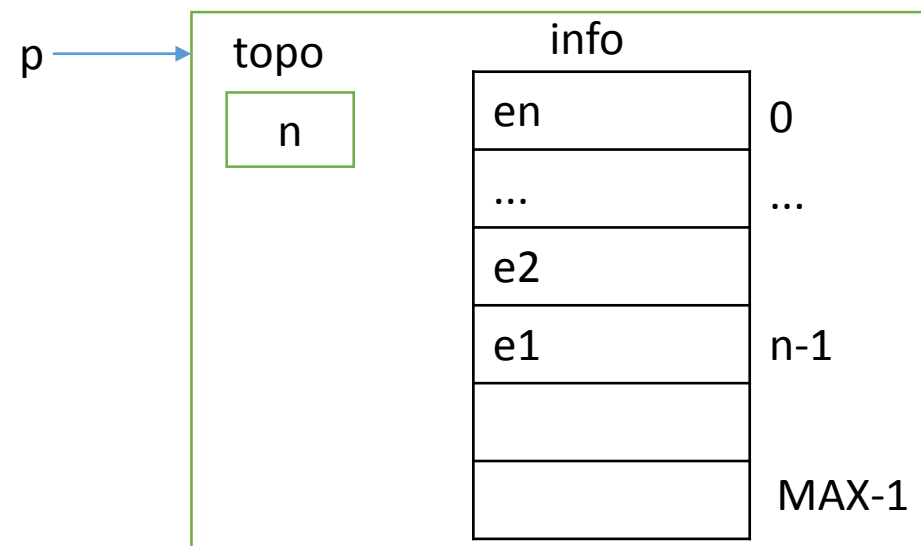


- Se representa com a estrutura



# Implementação: Vetor

```
• typedef struct{  
    int topo;  
    Tipo P info[MAX];  
} Tpilha, *Pilha;
```



# Funções do TAD Lista

- Pilha inicPilha( void ) : cria e retorna uma pilha vazia
- void adicPilha( Pilha p, TipoP elem): adiciona(empilha) um elemento no topo da pilha
- void elimPilha( Pilha p): elimina(desempilha) o elemento que está no topo da pilha
- TipoP infoPilha(Pilha p): retorna o elemento que está no topo da pilha
- int vaziaPilha(Pilha p) : retorna verdadeiro se a pilha está vazia e falso caso contrário
- void destruirPilha(Pilha p): destrói a pilha p, desalocando toda memória ocupada