

Estrutura Abstrata de Dados

PILHA

**Implementação com ligações simples
(usando ponteiros e memória dinâmica)**

Conceitos gerais

Elementos

- Os elementos duma pilha são processados pela ordem inversa à ordem de chegada
 - o **último** elemento a **entrar** é o **primeiro** a **sair** (**LIFO** - "Last In First Out")

Operações

- Qualquer operação que se pretenda efetuar será realizada no topo da Pilha
 - a **inserção** de um (novo) elemento na pilha
 - acrescenta este elemento ao topo da pilha (torna-se o topo)
 - a **remoção** de um elemento da pilha
 - retira o elemento que se encontra no topo da pilha
 - passa o elemento que está antes, caso exista, a ser o topo de pilha
 - só é possível de realizar se a pilha não estiver vazia
 - a **pilha vazia**
 - acontece quando é removido o último elemento da pilha
 - só é possível realizar a operação de inserção na pilha

Operações básicas (únicas) sobre uma EAD PILHA

Criar uma Pilha (vazia)

```
S = criarPilha()
```

Verificar se uma Pilha está vazia

```
pilhaVazia(S)
```

Inserir um elemento numa Pilha (coloca no topo da Pilha)

```
S = push(X, S)
```

Remover um elemento de uma Pilha (o que está no topo)

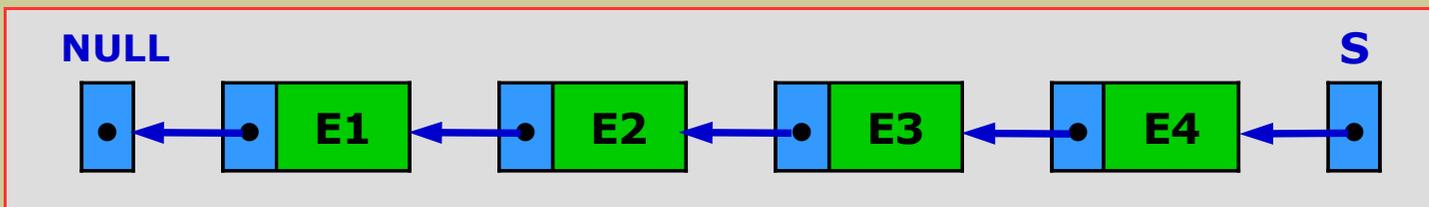
```
S = pop(S)
```

Consultar uma Pilha (devolve o elemento do topo)

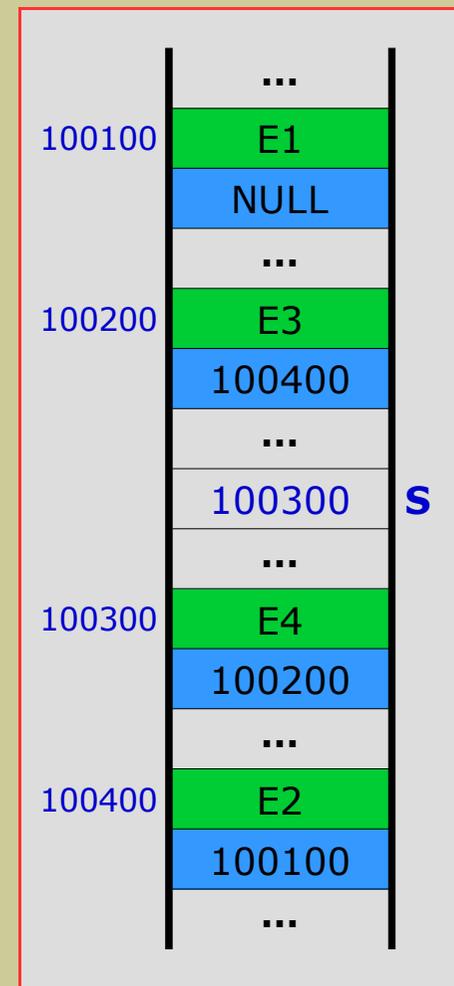
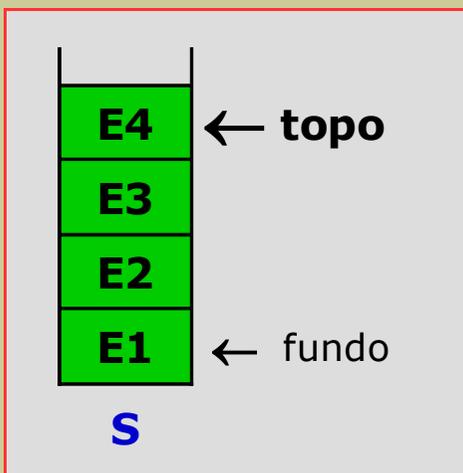
```
X = topo(S)
```

Representação gráfica

EAD Pilha



Analogia



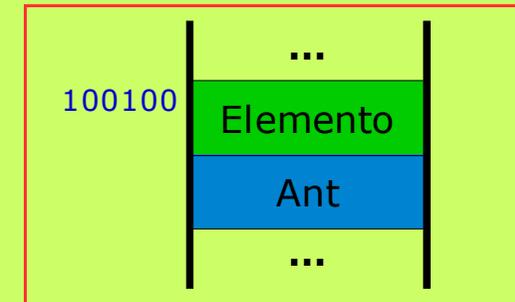
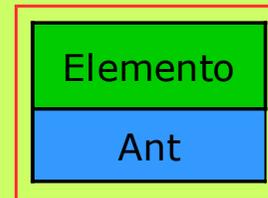
Identificador S

- É um **ponteiro** para o **topo** da **pilha**

Nodos e ligações

Definição de um nodo de uma AED PILHA (em linguagem C)

```
struct NodoPilha{  
    INFOPilha Elemento;  
    struct NodoPilha *Ant;  
};  
typedef struct NodoPilha *PNodoPilha;
```



- Cada nodo aponta para o nodo anterior da pilha
- O nodo do fundo da pilha aponta para NULL
- A memória para os nodos é
 - atribuída quando um elemento é inserido na pilha
 - libertada quando um elemento é removido da pilha.
- Pilha vazia: quando o topo da pilha é um ponteiro nulo (NULL)
- Pilha cheia: quando não há memória para alocar um novo elemento

Operações sobre a estrutura INFOPilha

Mostrar os dados de um elemento do tipo INFOPilha

```
void mostrarElementoPilha (INFOPilha X)
```

- operação que mostra/guarda os dados/informação do elemento X do tipo INFOPilha

Criar um elemento do tipo INFOPilha

```
INFOPilha criarElementoPilha ()
```

- operação que cria um elemento do tipo INFOPilha com dados vindos de fonte adequada

Comparar dois elementos do tipo INFOPilha

```
int CompararElementosPilha (INFOPilha X, INFOPilha Y)
```

- operação que compara dois elementos do tipo INFOPilha (segundo o campo que é a chave)
- devolve: **-1** ($X < Y$), **0** ($X = Y$) ou **1** ($X > Y$)

Operações básicas sobre um nodo

Criar um nodo de uma Pilha

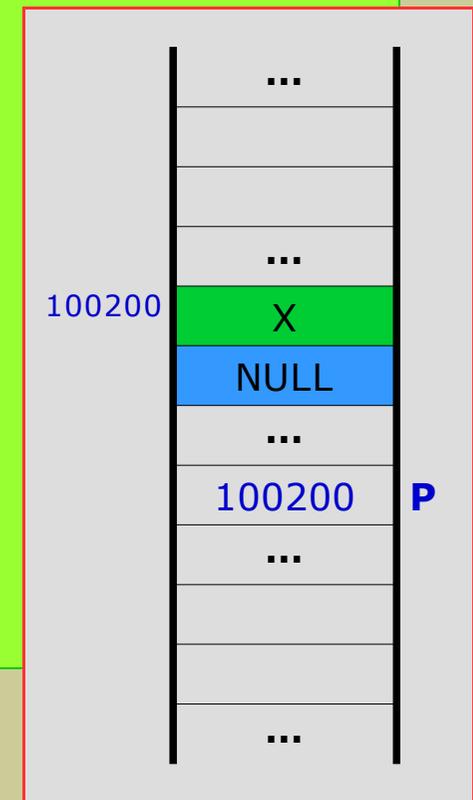
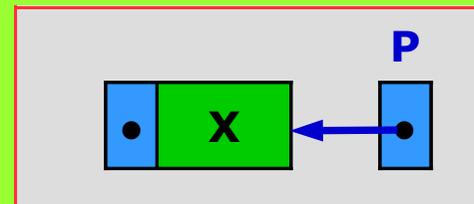
Operação: criar um novo nodo de uma pilha

Entrada: a informação propriamente dita (elemento X), que fará parte de um nodo

Saída: um ponteiro para um **nodo** com informação (elemento X) e ligação a NULL

PNodoPilha criarNodoPilha (INFOPilha X)

```
{
  PNodoPilha P;
  P = (PNodoPilha) malloc (sizeof(struct NodoPilha));
  if (P == NULL)
    return NULL;
  P→Elemento = X;
  P→Ant = NULL;
  return P;
}
```



Libertar/destruir um nodo de uma Pilha

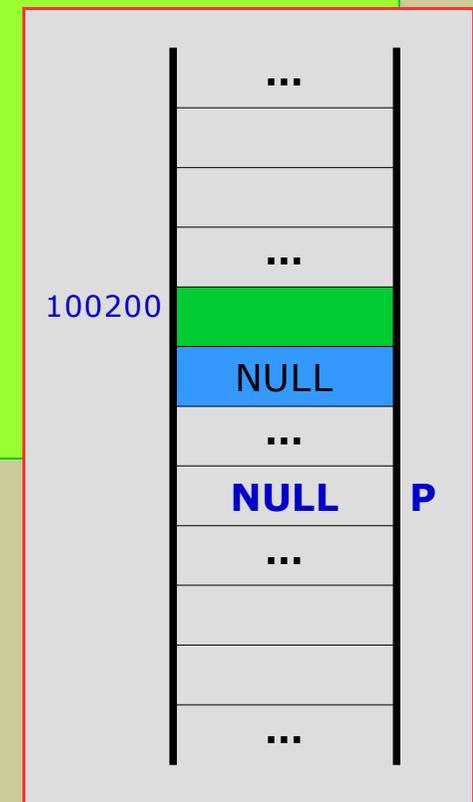
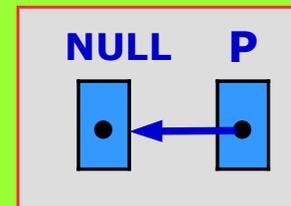
Operação: libertar todos os nodos de uma pilha

Entrada: um ponteiro para o nodo que se pretende destruir

Saída: o ponteiro a apontar para NULL

PNodoPilha libertarNodoPilha (PNodoPilha P)

```
{  
  P→Anterior = NULL;  
  free(P);  
  P = NULL;  
  return P;  
}
```



Operações básicas (únicas) sobre uma Pilha

Criar uma Pilha

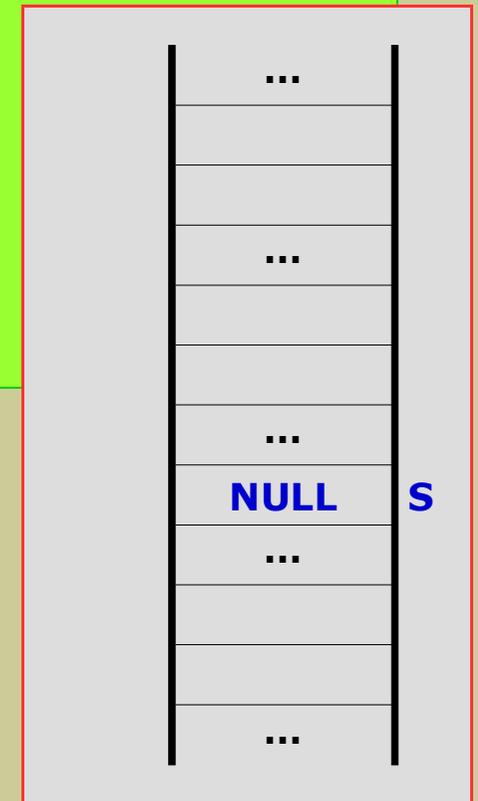
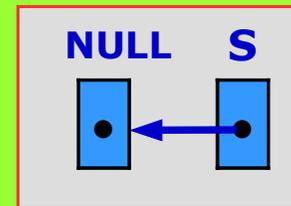
Operação: criar uma pilha vazia

Entrada: nada

Saída: devolve uma Pilha S vazia (sem elementos)

PNodePilha criarPilha ()

```
{  
  PNodePilha S;  
  S = NULL;  
  return S;  
}
```



Verificar se uma Pilha está vazia

Operação: verificar se uma pilha está vazia

Entrada: uma Pilha S

Saída: 1 (se a Pilha S está vazia) ou 0 (se a Pilha S não está vazia)

```
int pilhaVazia (PNodoPilha S)
```

```
{
```

```
  if (S == NULL)
```

```
    return 1;
```

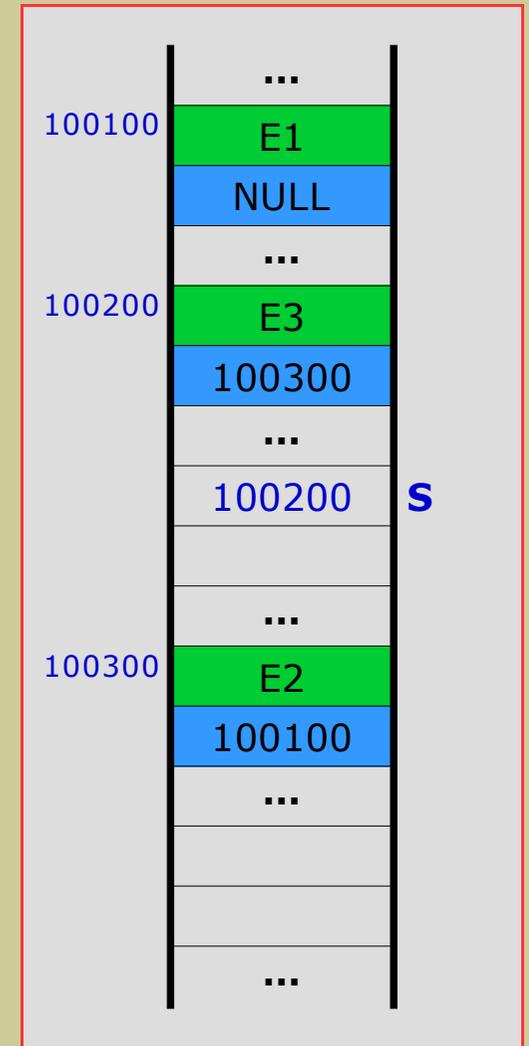
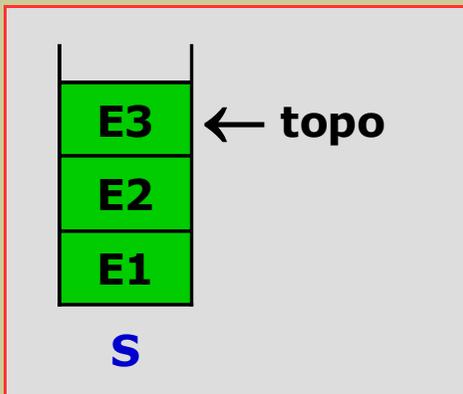
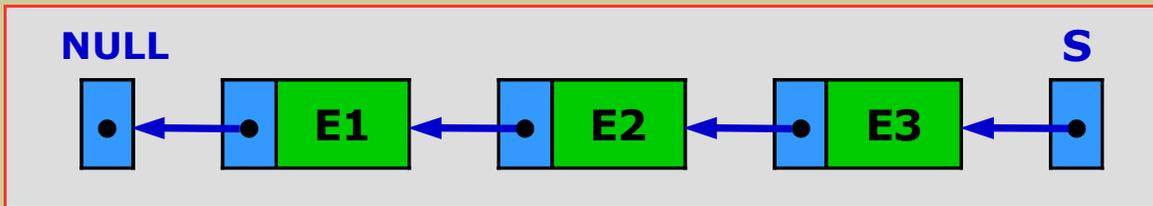
```
  else
```

```
    return 0;
```

```
}
```

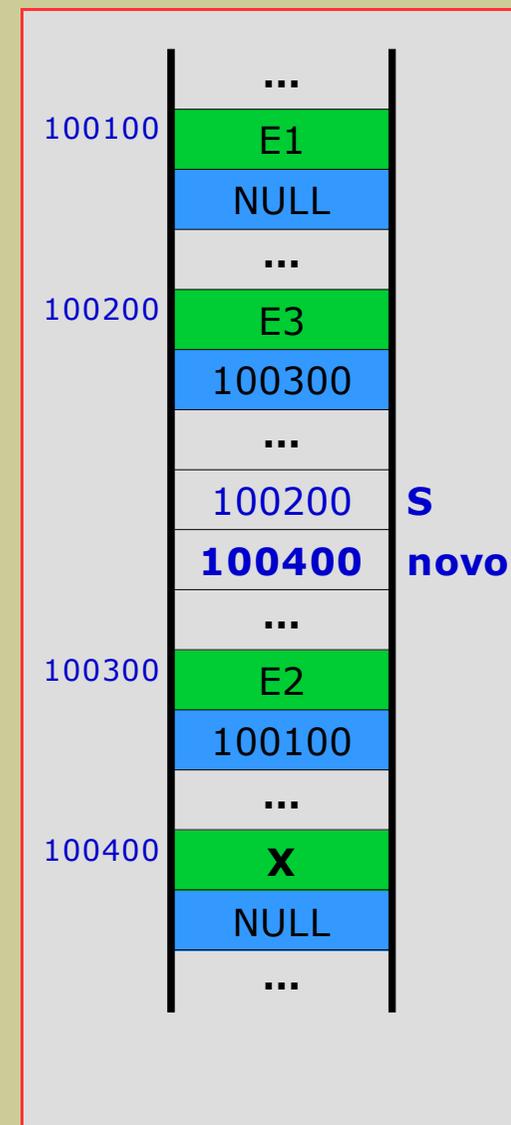
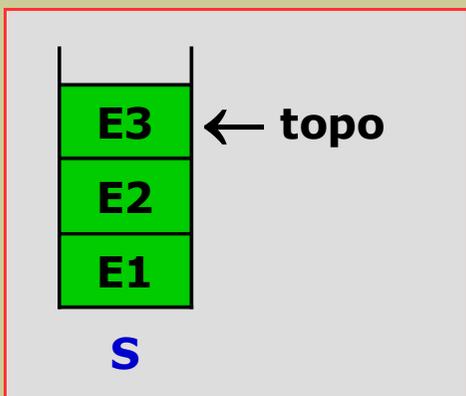
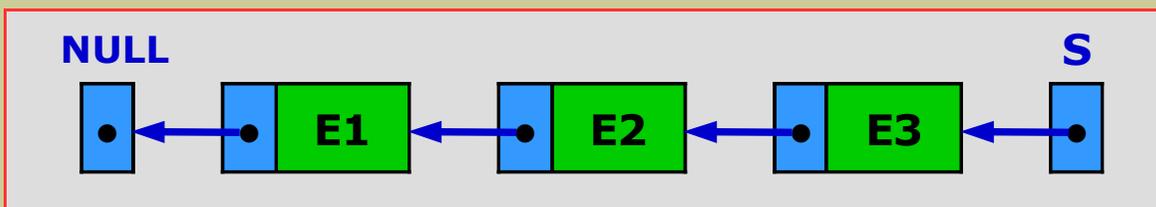
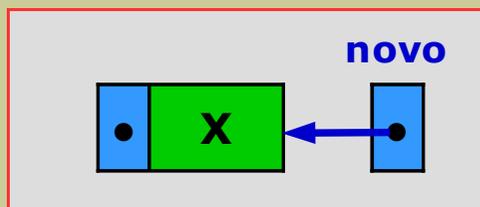
Inserir um elemento numa Pilha

- Operação **push**: $S = \text{push}(X, S)$
- Pilha S antes da operação



Inserir um elemento numa Pilha

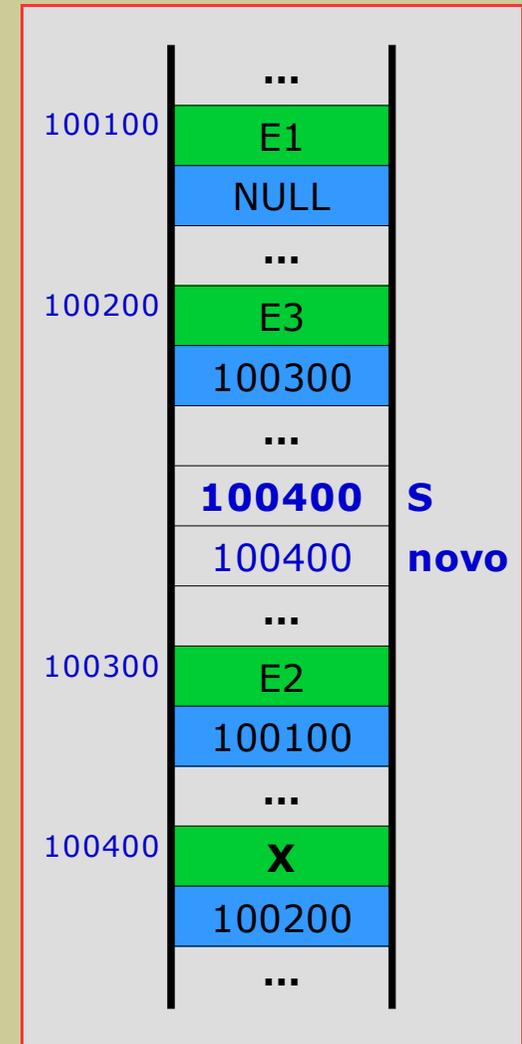
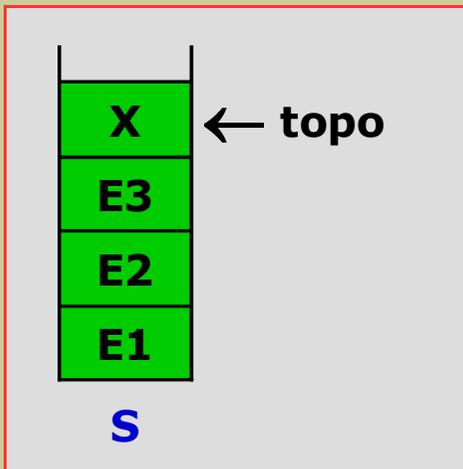
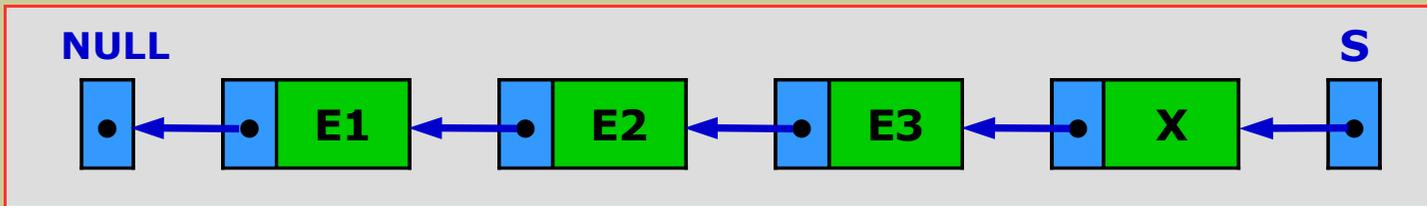
- Operação **push**: $S = \text{push}(X, S)$
 - Passo 1: criar um nodo com o novo elemento, X
 - devolve um ponteiro P para o novo nodo



Inserir um elemento numa Pilha

- Operação: $S = \text{push}(X, S)$

- Passo 2: inserir o nodo com o novo elemento (P) na Pilha S



Inserir um elemento numa Pilha

Operação: inserir um elemento no topo de uma Pilha (push)

Entrada: um elemento X a inserir e uma Pilha S

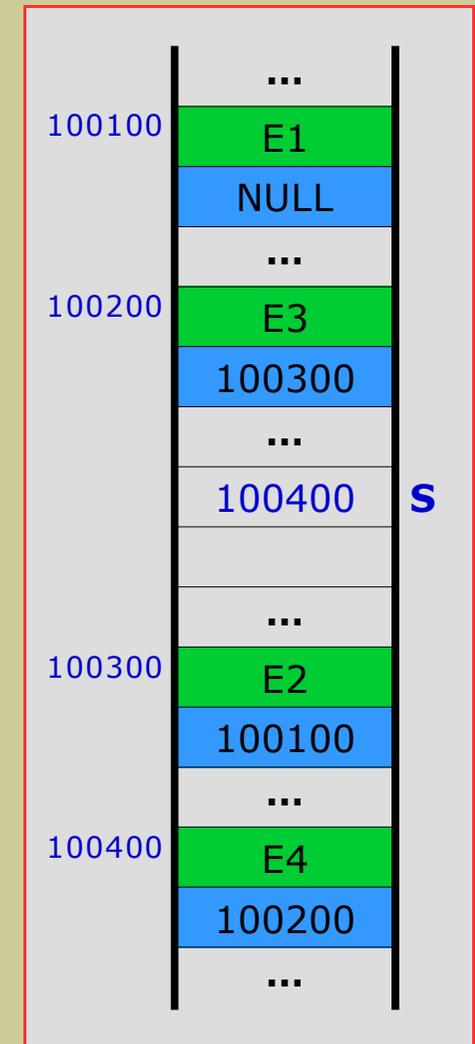
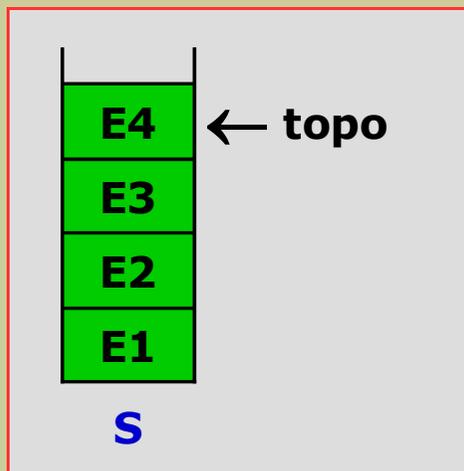
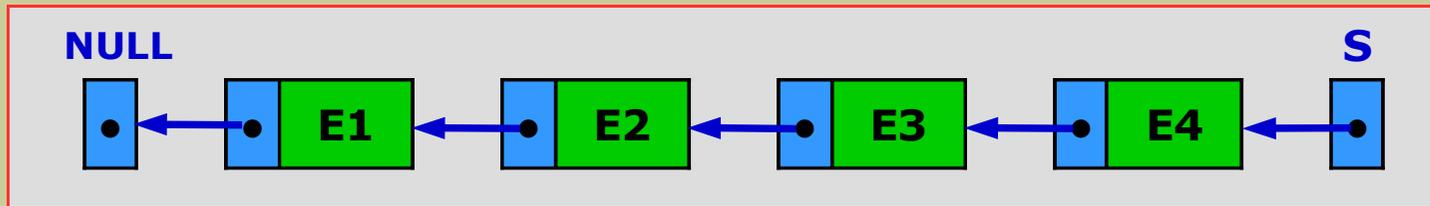
Saída: a Pilha S atualizada (com mais um elemento)

PNodoPilha push (INFOPilha X, PNodoPilha S)

```
{  
  PNodoPilha novo;  
  novo = criarNodoPilha(X);    // novo = ponteiro para o novo elemento com X  
  if (novo == NULL)  
    return S;  
  novo→Ant = S;  
  S = novo;  
  return S;  
}
```

Remover um elemento de uma Pilha

- Operação **pop**: $S = \text{pop}(S)$
- Pilha S antes da operação

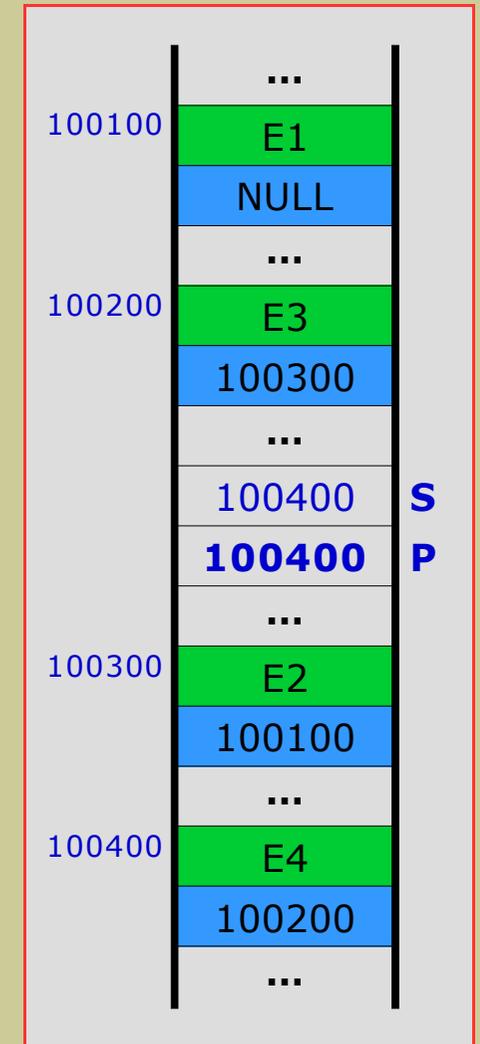
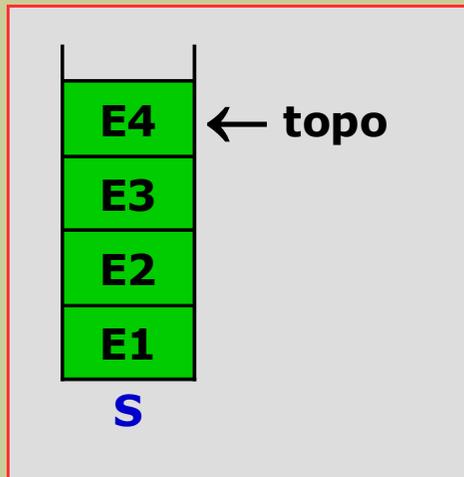
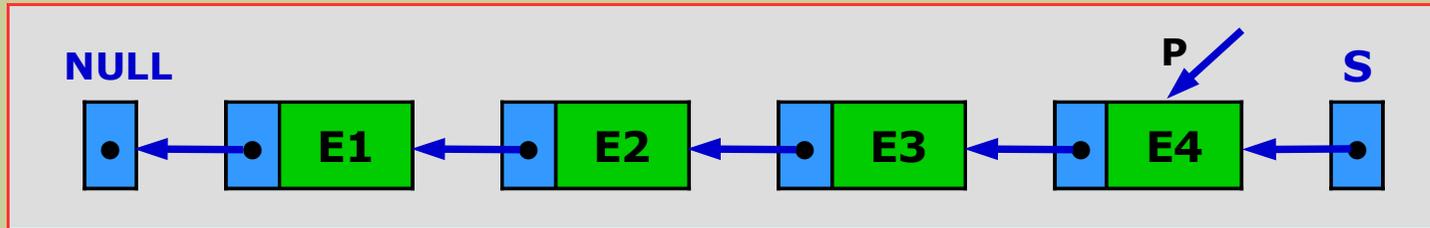


Remover um elemento de uma Pilha

- Operação **pop**: $S = \text{pop}(S)$

- Passo 1:

- marcar o nodo do topo da Pilha S com P

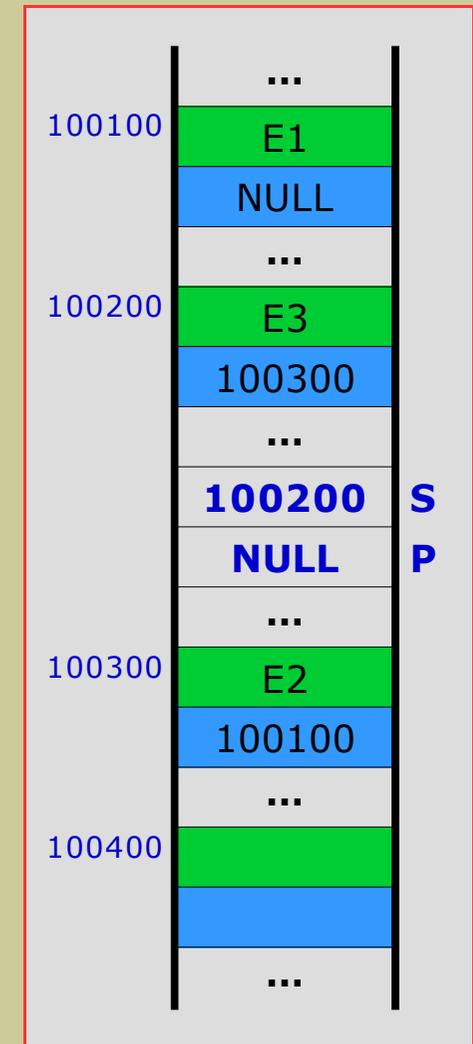
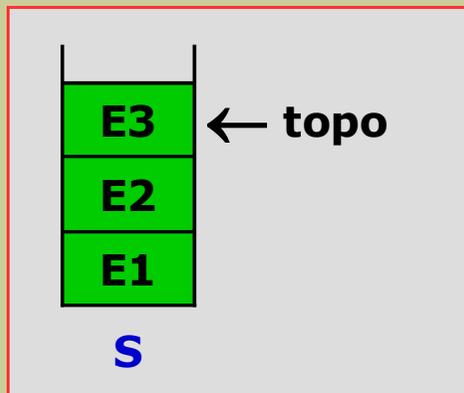
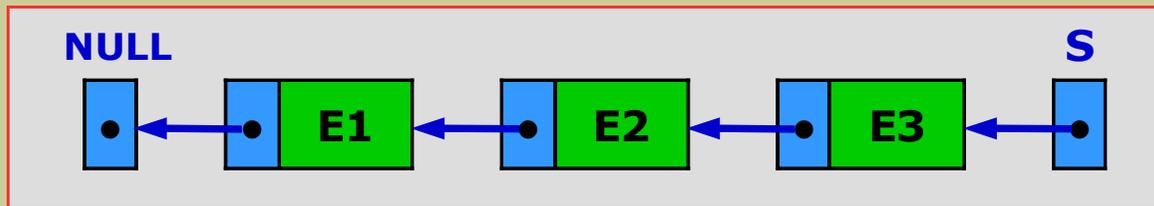


Remover um elemento de uma Pilha

- Operação **pop**: $S = \text{pop}(S)$

- Passo 2:

- remover o topo de S, que passa a ser o nodo anterior
- libertar o nodo com o topo anterior de S (P)



Remover um elemento de uma Pilha

Operação: remove o elemento do topo de uma Pilha (pop)

Entrada: uma Pilha S

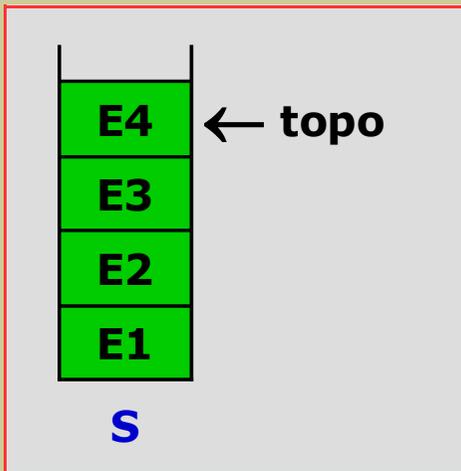
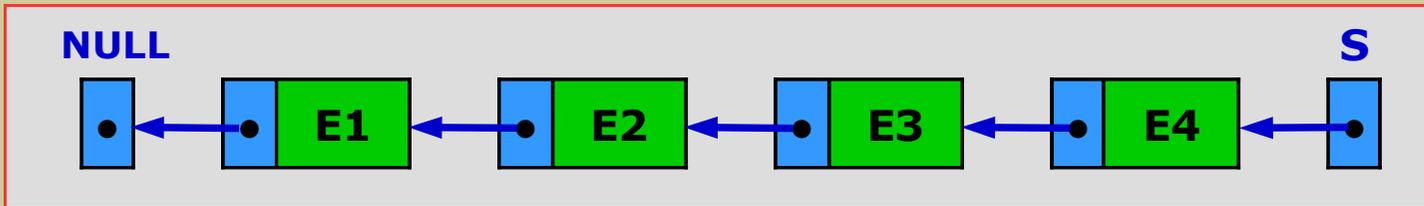
Saída: a Pilha S atualizada (sem o elemento do topo)

PNodoPilha pop (PNodoPilha S)

```
{  
  PNodoPilha P;  
  P = S;  
  S = S→Ant;  
  P = libertarNodoPilha(P);  
  return S;  
}
```

Consultar uma Pilha

- Representação gráfica



$\text{topo}(S) = \mathbf{E4}$ ($S \rightarrow \text{Elemento}$)

Consultar uma Pilha

Operação: consulta o elemento do topo de uma Pilha (topo)

Entrada: uma Pilha S

Saída: o elemento que está no topo da Pilha S (do tipo INFOPilha)

INFOPilha topo (PNodoPilha S)

```
{  
  return S→Elemento;  
}
```