

# PILHAS

PUC MINAS

ALGORITMOS E ESTRUTURAS DE DADOS II

# PILHA

- Estrutura de dados em que todas as **inserções, retiradas e acessos** são feitos em **somente uma extremidade**;
  - **topo da pilha.**
- Regra básica:
  - o **último a entrar**;
    - é o **primeiro a sair** / ser atendido;
  - *last-in first-out (LIFO)*.
- Dados “ordenados” do “mais recentemente acessado para o menos recentemente”.

# PILHA – MODELO INTUITIVO

- Pilha de pratos em uma prateleira:
  - é conveniente retirar ou adicionar pratos somente na extremidade superior da pilha.
- Os itens são colocados um sobre os outros;
  - o item inserido mais recentemente está no **topo**;
  - e o inserido menos recentemente no **fundo** da pilha.

# PILHA – EXEMPLOS

- *Undo* de editores diversos;
- Histórico de navegação;
- Chamadas de funções;
- Análise de expressões matemáticas;
- Inversões.

# PILHA – EXEMPLOS

- Ideal para **processamento de estruturas aninhadas** de profundidade imprevisível;
- ordem de remoção garante que as **estruturas mais internas serão processadas antes das mais externas.**

# PILHA – EXEMPLOS

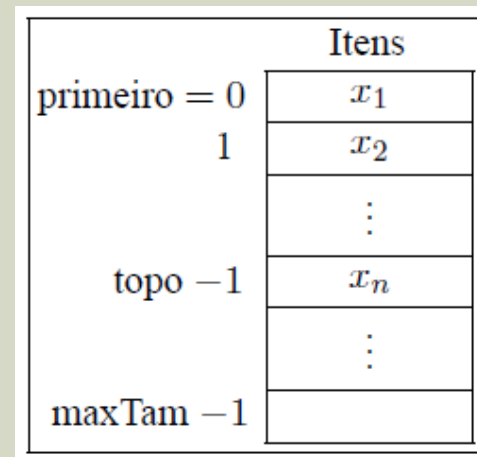
- Aplicação em estruturas aninhadas:
  - Quando é necessário analisar um conjunto de dados e armazenar uma **lista de tarefas a serem realizadas posteriormente**.
  - Controle de **sequência de chamadas de subprogramas**.
  - **Recursividade**.

# PRINCIPAIS OPERAÇÕES


- Criar uma pilha vazia;
- Empilhar um novo item (*push*);
- Desempilhar um item (*pop*);
- Verificar se a pilha está vazia;
- Consultar o topo (*top*).

# PILHAS E VETORES

- Itens da pilha são armazenados em **posições contíguas de memória**:
  - cria-se um **vetor com M posições vazias**;
  - **variável de controle**:
    - **topo**;
    - **próxima posição vazia**.



Direção de inserção na pilha da figura ao lado





# PILHAS E VETORES – OPERAÇÕES

## ■ *Push:*

- relativamente simples:
  - verificar se a pilha não está cheia;
  - realizar operação no “topo” da pilha;
  - atualizar “topo”.

# PILHAS E VETORES – OPERAÇÕES

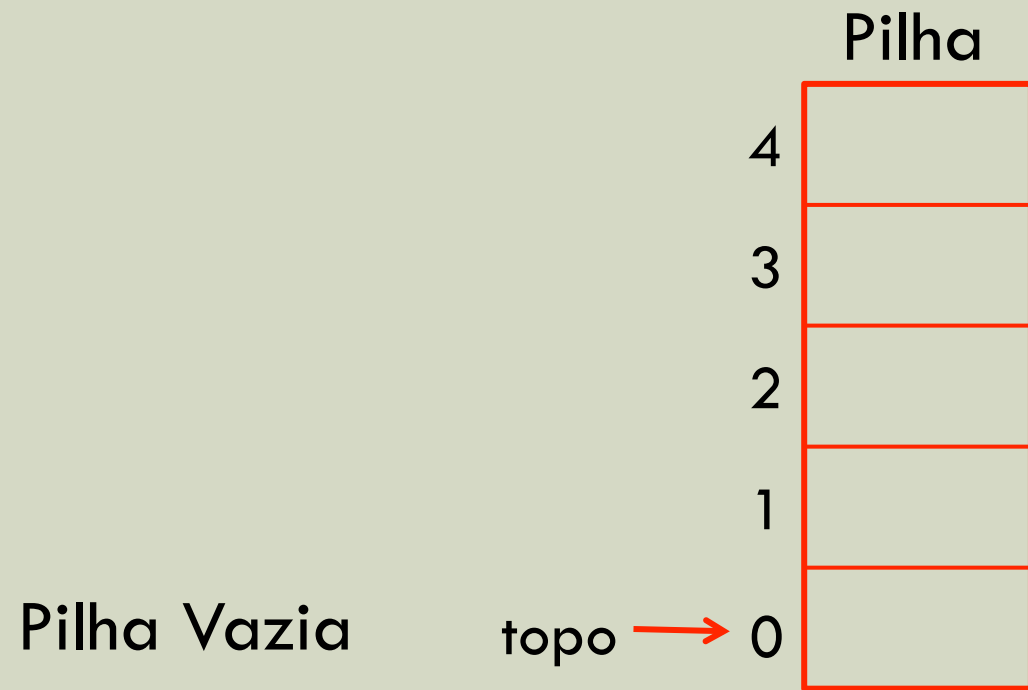
## ■ *Pop:*

- relativamente simples:
  - verificar se a pilha não está vazia;
  - atualizar “topo”;
  - realizar operação no “topo” da pilha.

# PILHAS E VETORES – OPERAÇÕES

- **Verificar se a pilha está vazia:**
  - “topo” == 0
- ***Top:***
  - mais simples ainda.

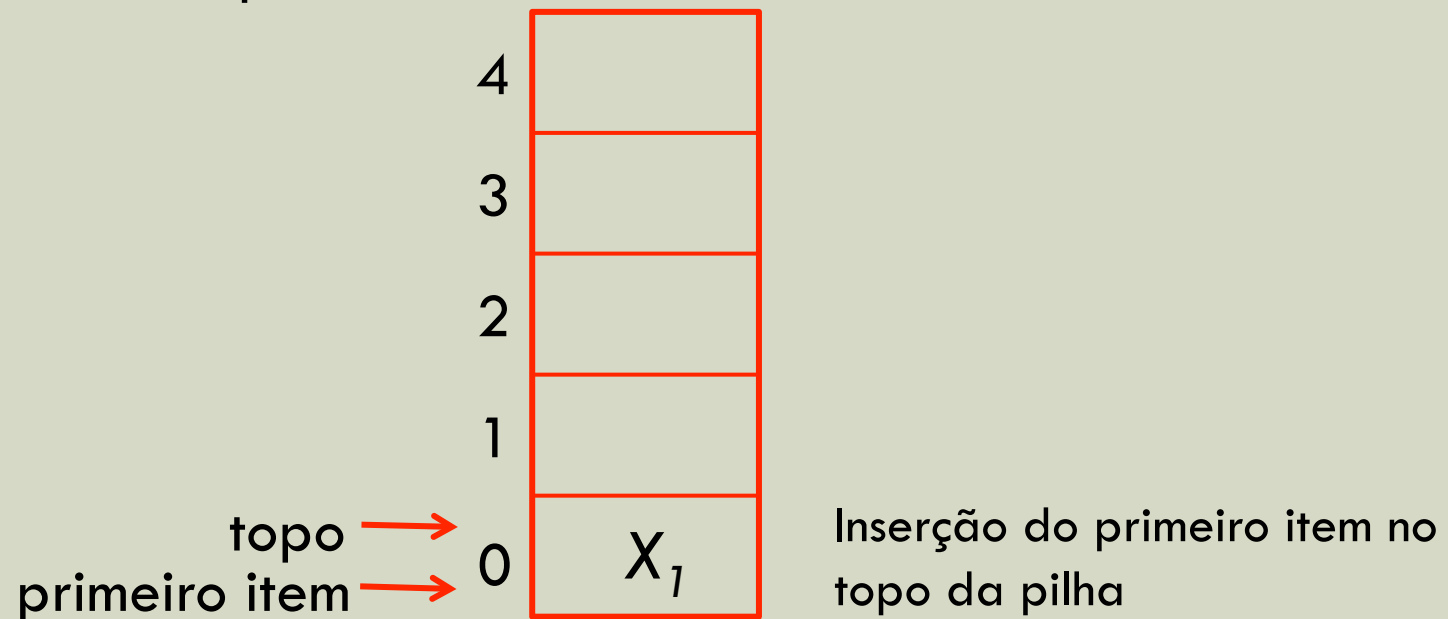
# PILHAS E VETORES



# PILHAS E VETORES

Inserção de itens na pilha

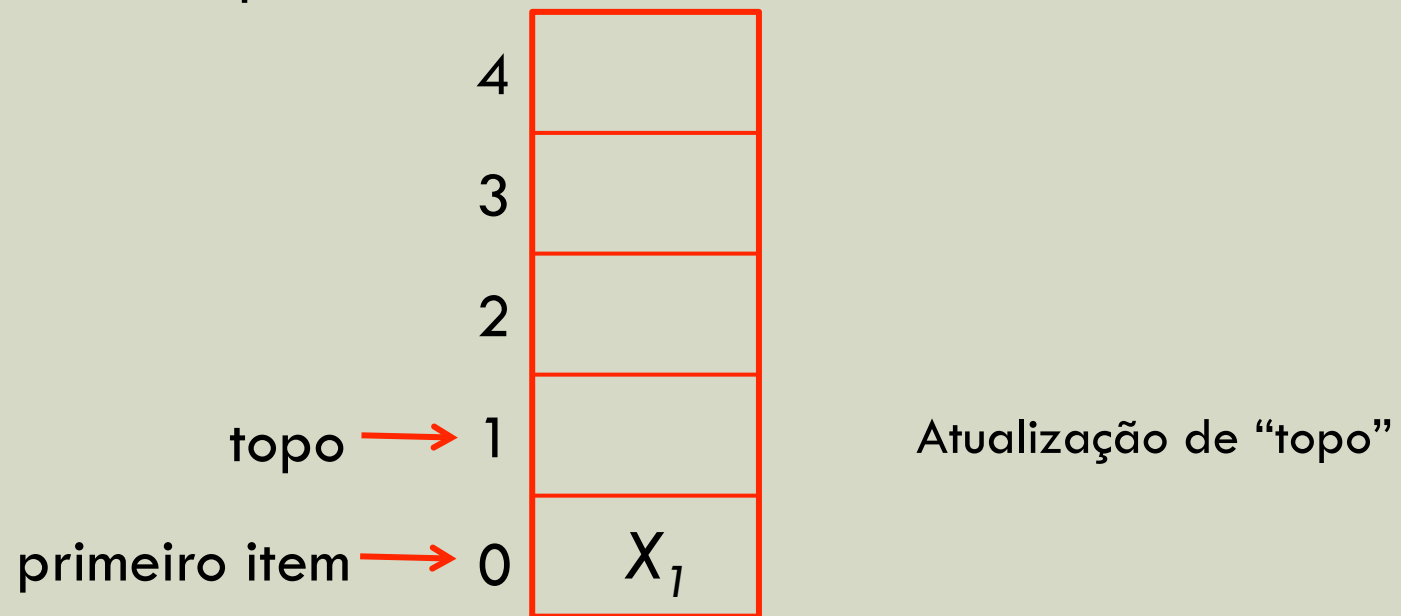
Pilha



# PILHAS E VETORES

Inserção de itens na pilha

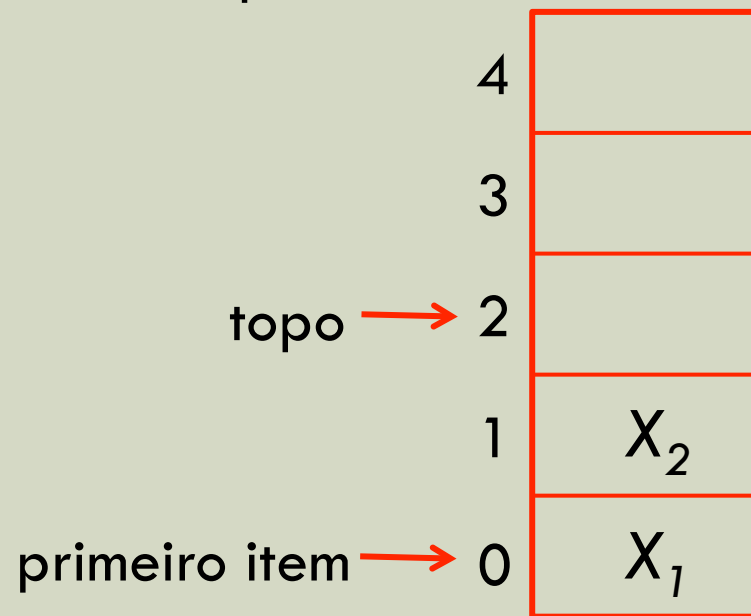
Pilha



# PILHAS E VETORES

Inserção de itens na pilha

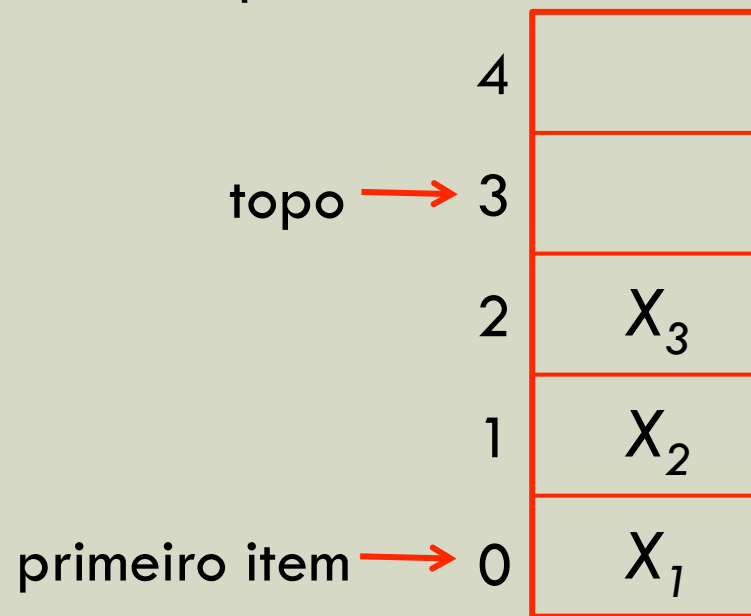
Pilha



# PILHAS E VETORES

Inserção de itens na pilha

Pilha





# PILHAS E VETORES

Inserção de itens na pilha

Pilha

topo → 4

3

$X_4$

2

$X_3$

1

$X_2$

primeiro item → 0

0

$X_1$

# PILHAS E VETORES

Inserção de itens na pilha

topo

Pilha

primeiro item

4

$X_5$

3

$X_4$

2

$X_3$

1

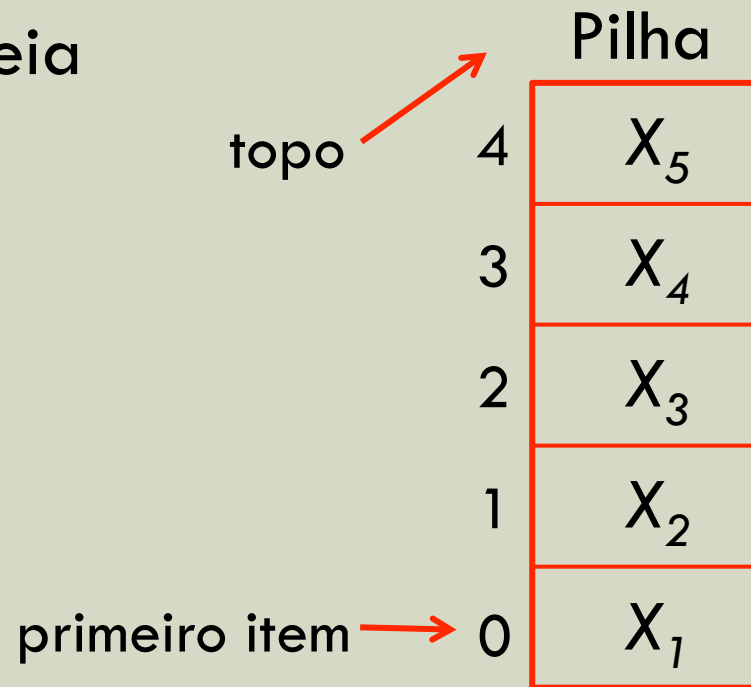
$X_2$

0

$X_1$

# PILHAS E VETORES

Pilha Cheia



# PILHAS E VETORES

Remoção de itens da pilha

Pilha

topo → 4

$X_5$

Atualização de “topo”

3

$X_4$

2

$X_3$

1

$X_2$

primeiro item → 0

$X_1$

# PILHAS E VETORES

Remoção de itens da pilha

topo → 4

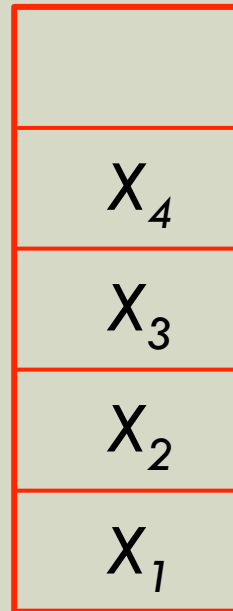
3

2

1

primeiro item → 0

Pilha

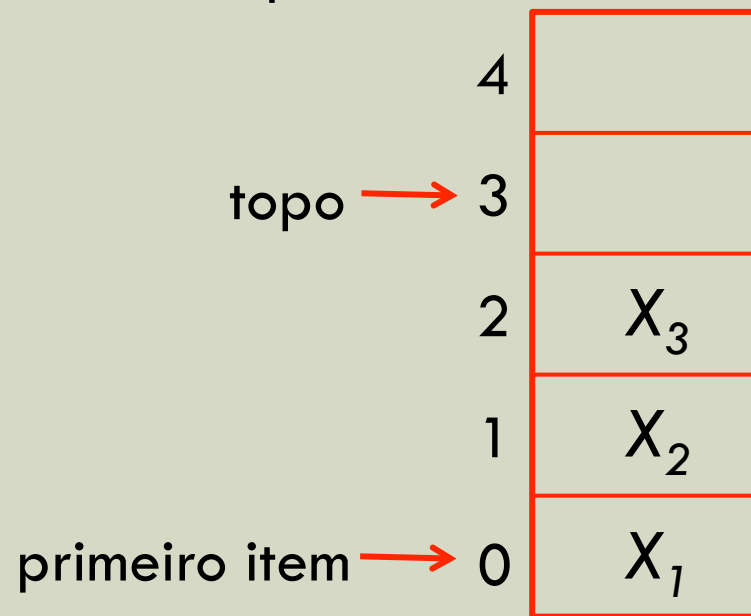


Remoção do item do  
“topo” da pilha

# PILHAS E VETORES

Remoção de itens da pilha

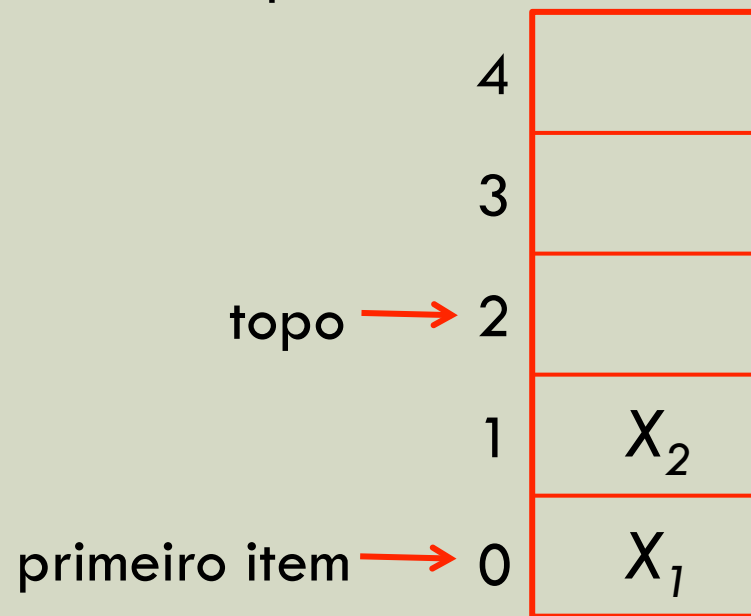
Pilha



# PILHAS E VETORES

Remoção de itens da pilha

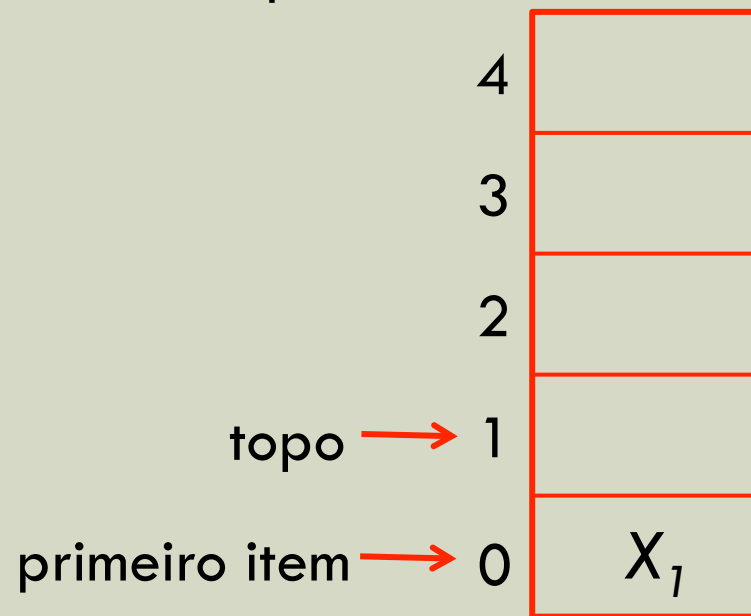
Pilha



# PILHAS E VETORES

Remoção de itens da pilha

Pilha

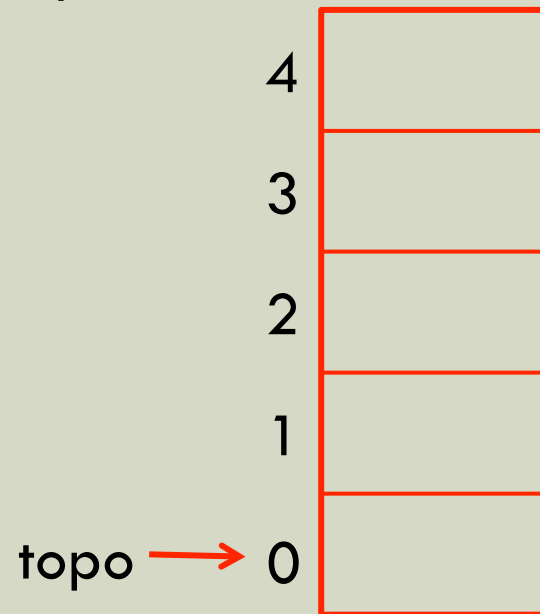




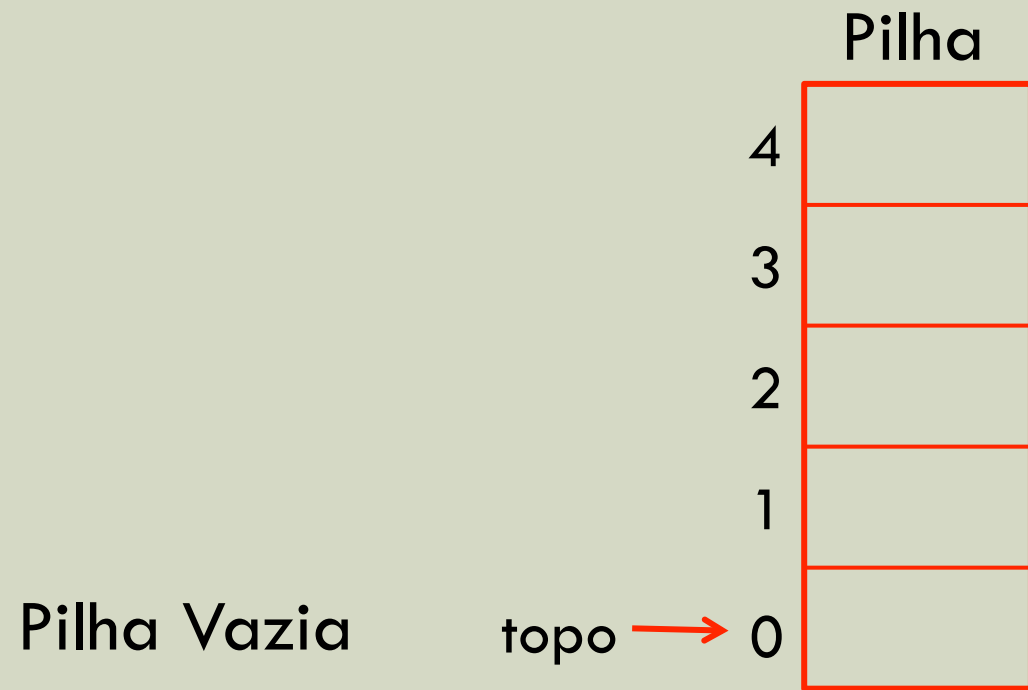
# PILHAS E VETORES

Remoção de itens da pilha

Pilha



# PILHAS E VETORES



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

- **Classe Pilha:**
  - **Apontadores de controle:**
    - **fundo e topo.**
  - **Célula sentinela, nó-cabeça ou nodo-cabeça:**
    - **elemento do fundo da pilha;**
    - **serve apenas para controle;**
      - para facilitar a implementação das operações *push* e *pop* quando a pilha está vazia.

# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

- Pilha implementada por meio de células.
- Cada **célula** contém:
  - um item da pilha;
  - uma referência para outra célula.
- **Classe Pilha** contém:
  - referência para o fundo da pilha;
    - célula sentinela;
  - referência para o topo da pilha.

# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS – OPERAÇÕES

- **Criar uma pilha vazia:**
  - criação da célula sentinela;
  - “fundo” e “topo” referenciam a mesma célula:
    - célula sentinela.

# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS – OPERAÇÕES

## ■ *Push:*

- **criar uma célula nova, ligá-la à célula referenciada por “topo” e colocar nela o novo item;**
- **atualizar “topo”.**
- **Encadeamento com a célula “topo”:**
  - **atenção à diferença do encadeamento!**

# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS – OPERAÇÕES

## ■ ***Pop:***

- **verificar se a pilha está vazia;**
- **retornar item do “topo”;**
- **atualizar “topo”.**

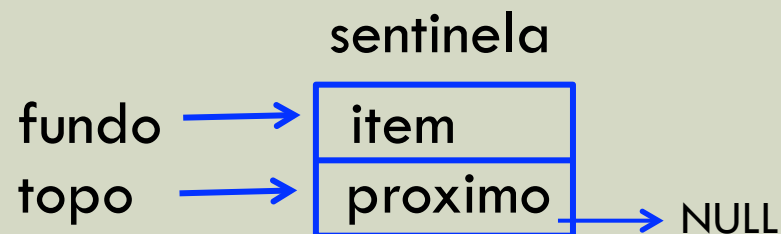
## PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS – OPERAÇÕES

- **Verificar se a pilha está vazia:**
  - **“fundo” == “topo”**
- **Consultar topo:**
  - **apenas retorna o item referenciado por “topo”.**



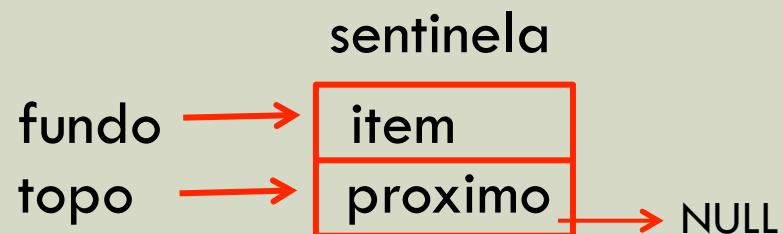
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

Pilha vazia: “fundo” e “topo” referenciam a célula sentinela.



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

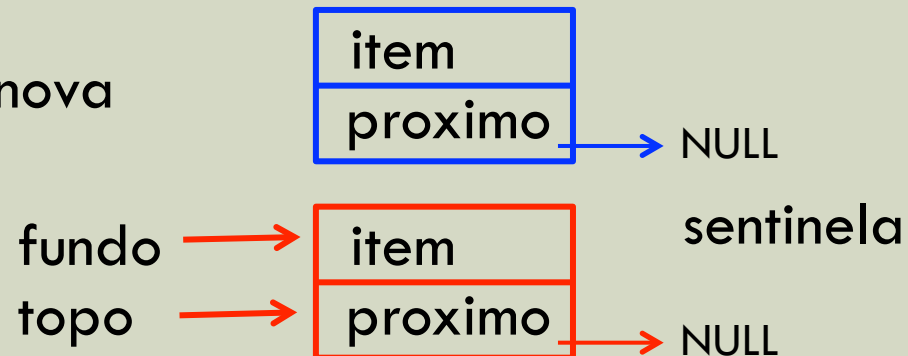
*Push*



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*

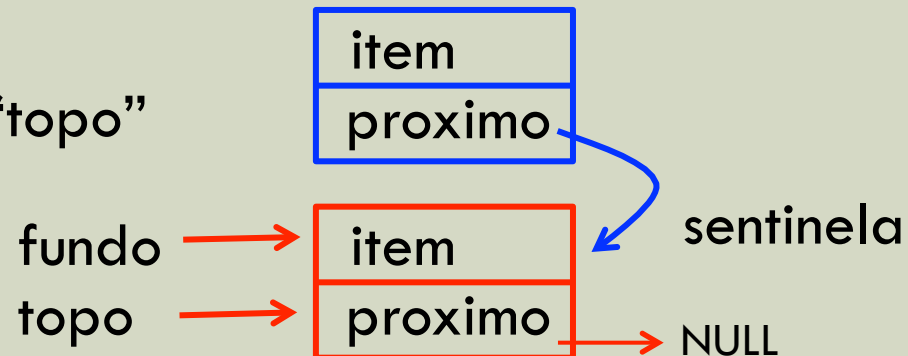
Criar uma célula nova



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*

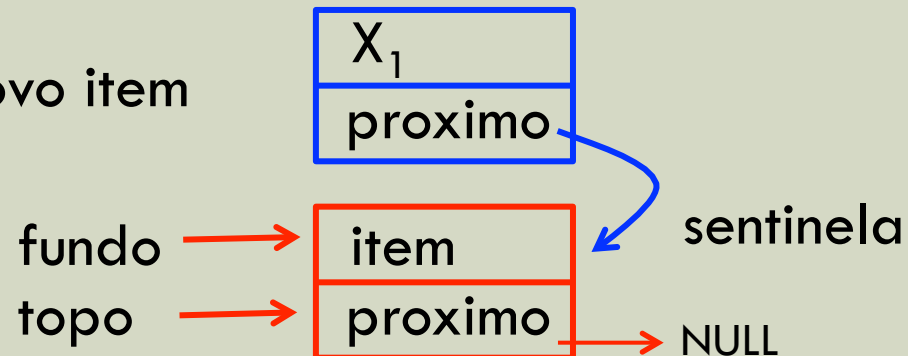
Ligá-la à célula  
referenciada por “topo”



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*

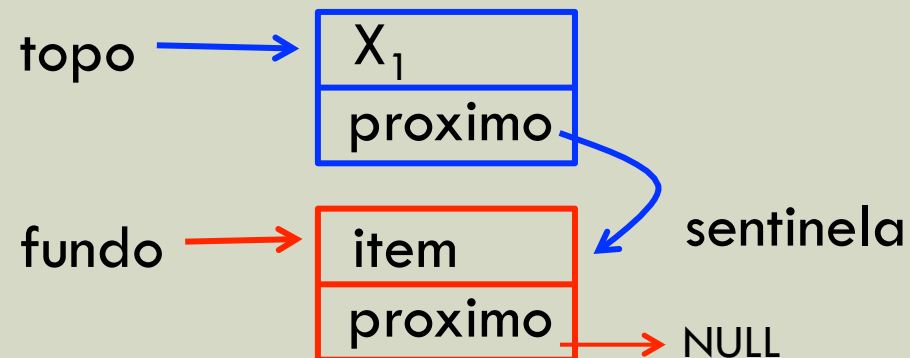
Colocar nela o novo item



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*

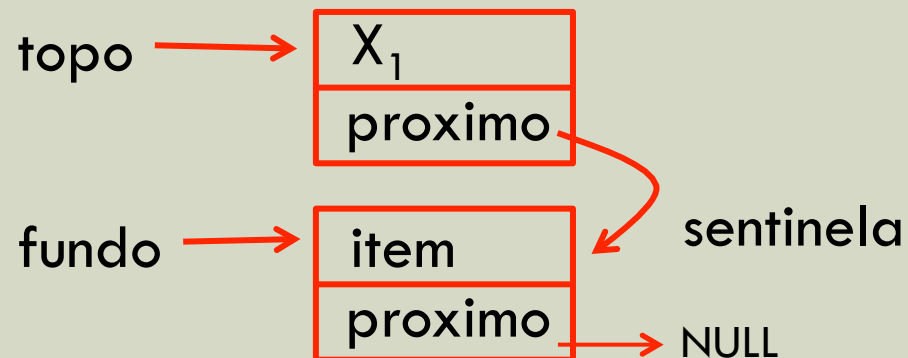
Atualizar “topo”



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*

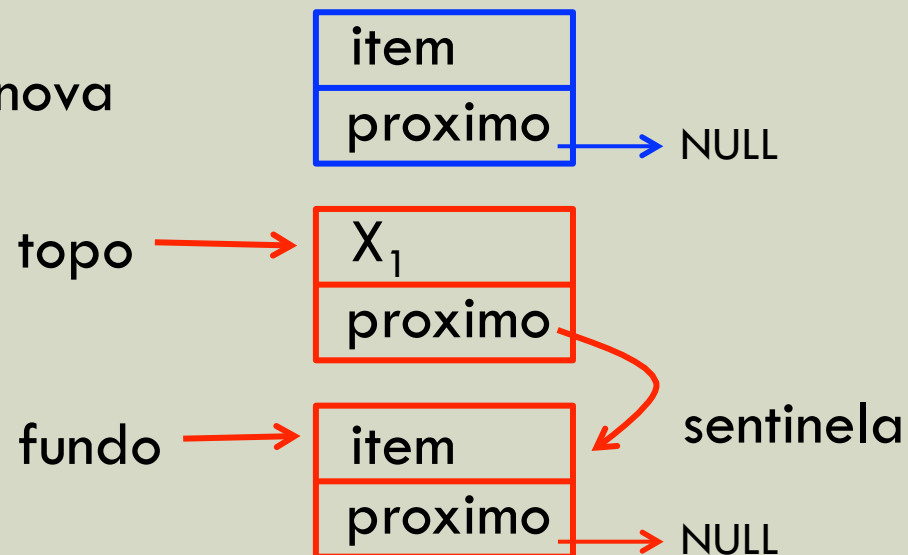
Pilha com apenas um elemento:



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*

Criar uma célula nova

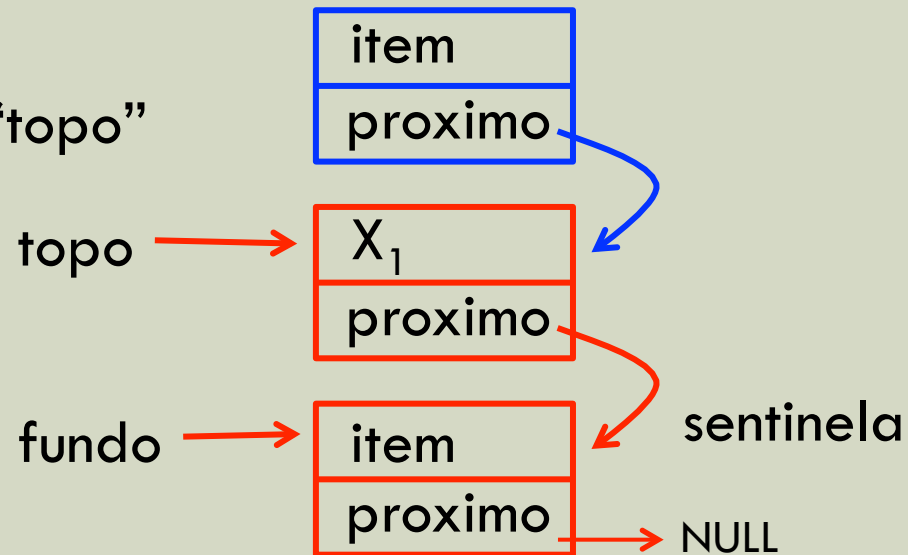




# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*

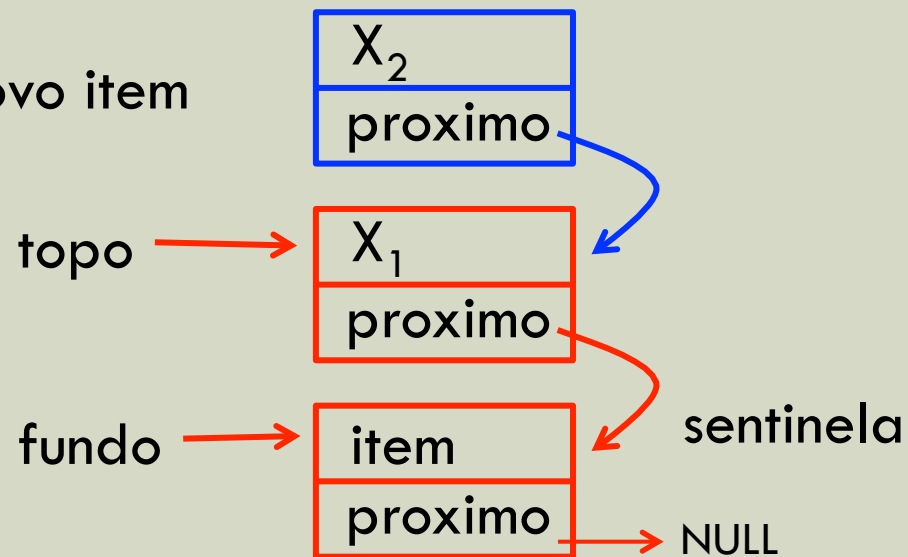
Ligá-la à célula  
referenciada por “topo”



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

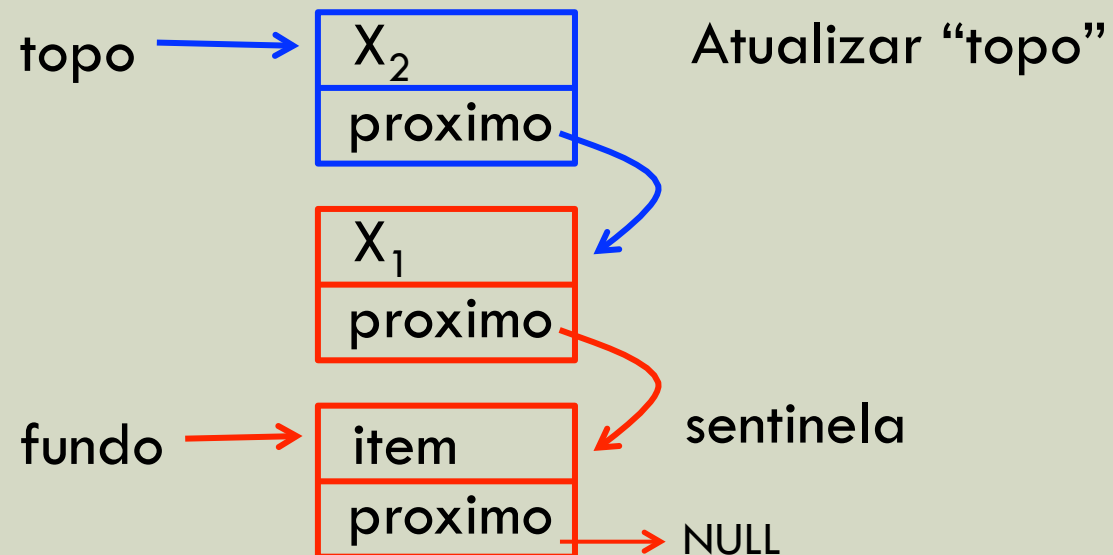
*Push*

Colocar nela o novo item



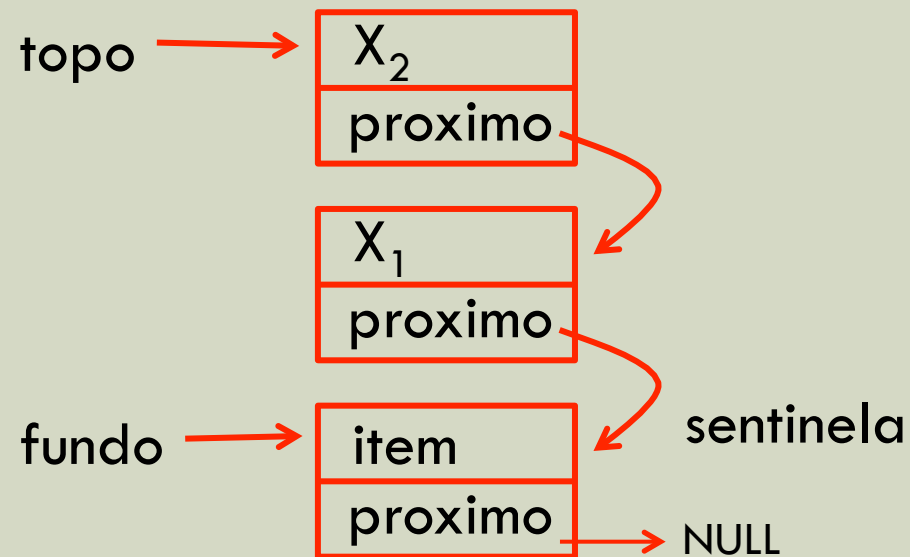
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*



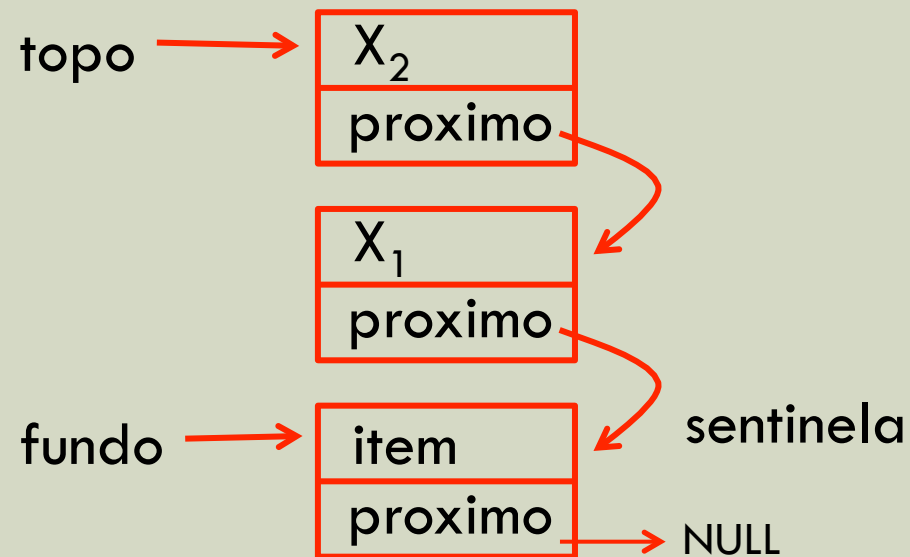
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Push*



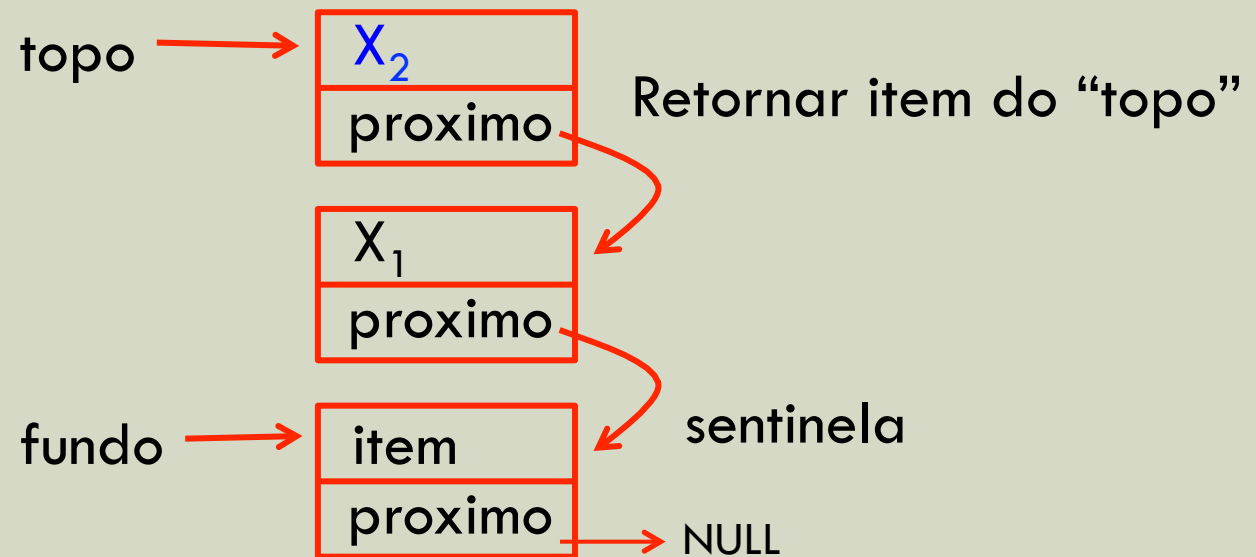
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*



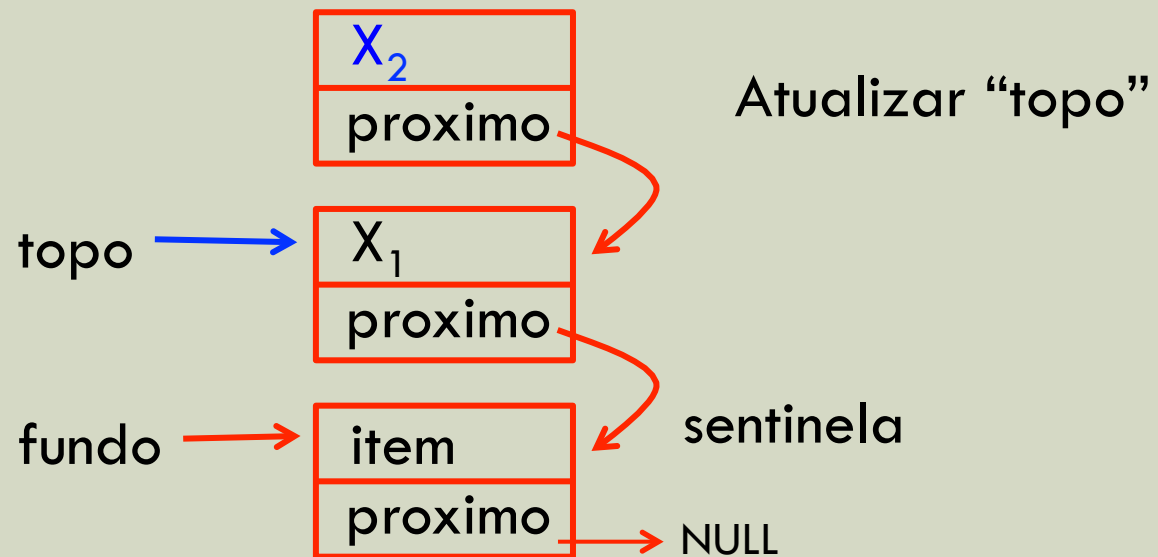
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*



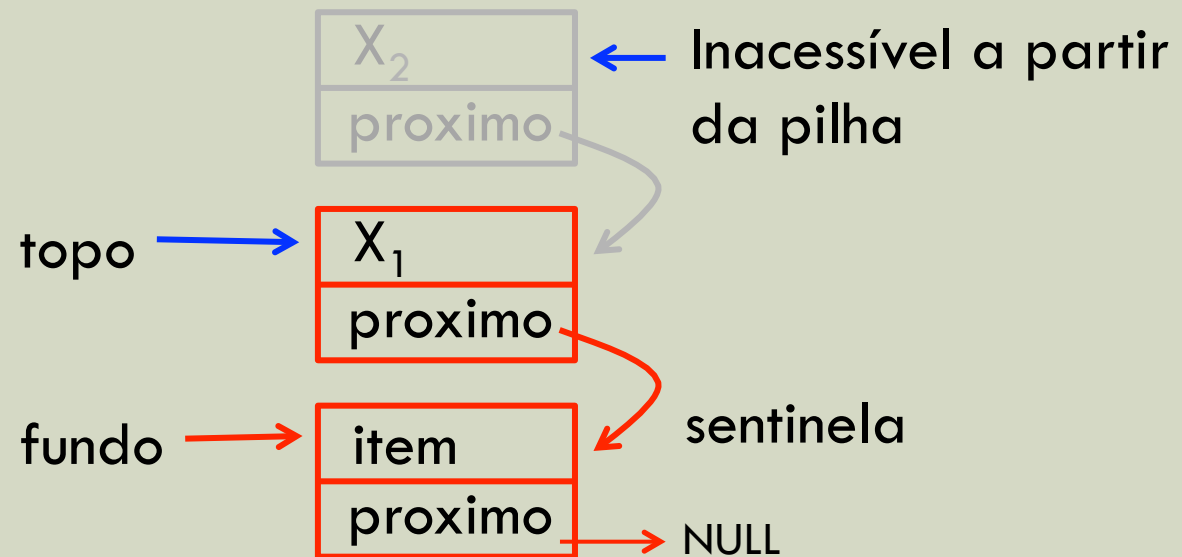
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

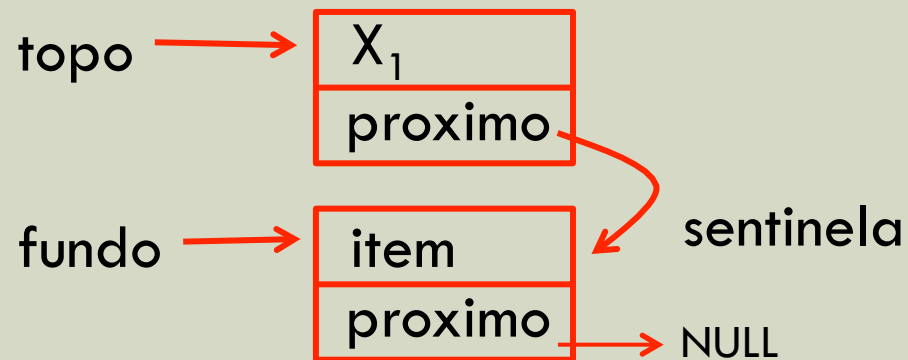
*Pop*





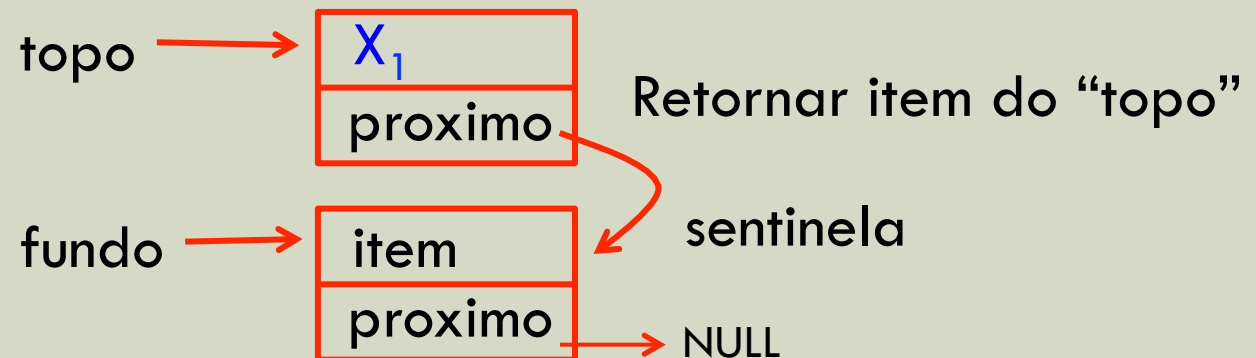
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*



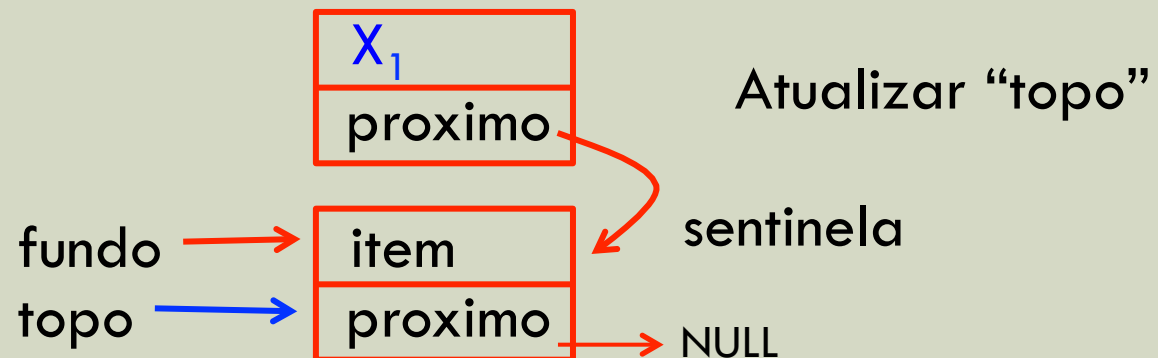
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*



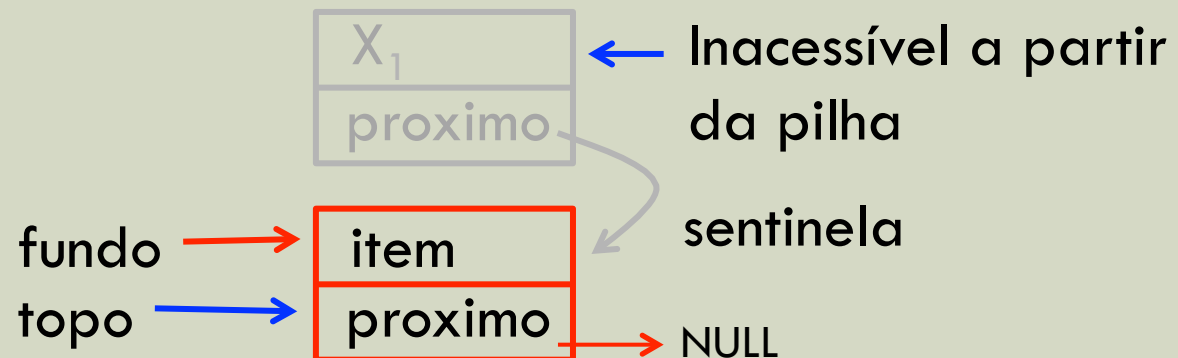
# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*



# PILHAS IMPLEMENTADAS POR MEIO DE ESTRUTURAS AUTO-REFERENCIADAS

*Pop*

Pilha vazia: “fundo” e “topo” referenciam a célula sentinela.

