

PILHA

PILHA

- ◉ Pilha é uma estrutura de dados e um tipo abstrato de dados
- ◉ Seu comportamento se baseia no armazenamento de tipo Last In First Out (LIFO), ou seja, estruturas onde o Último elemento a Entrar é o Primeiro a Sair.

PILHA

- ◉ Conceitualmente, a pilha possui duas operações básicas:
 - POP: retira um elemento do topo da pilha; e
 - PUSH: insere um elemento no topo da pilha.
- ◉ Para termos de implementação ainda temos funções de inicialização de pilhas, verificação se está vazia, acessa topo, ...

PILHA

- ◉ Em termos de implementação, a pilha pode ser encadeada, estática, genérica ou não genérica.
- ◉ O mais importante é compreender que a pilha é uma lista com regras mais restritas de acesso e operações mais simples.

PILHA

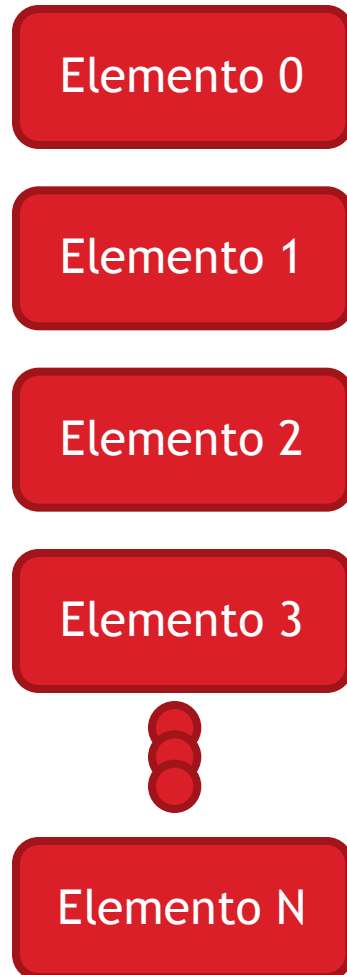
◉ A pilha é muito empregada na Computação:

- arquitetura
- sistemas operacionais: empilhamento de chamadas de função;
- linguagens formais e autômatos;
- compiladores;
- etc...

TAD PILHA

- Mas como especificamos um Tipo Abstrato de Dado Pilha?

TAD PILHA



Como era de se imaginar, em uma pilha os elementos são empilhados

TAD PILHA

Elemento 0

Elemento 1

Elemento 2

Elemento 3

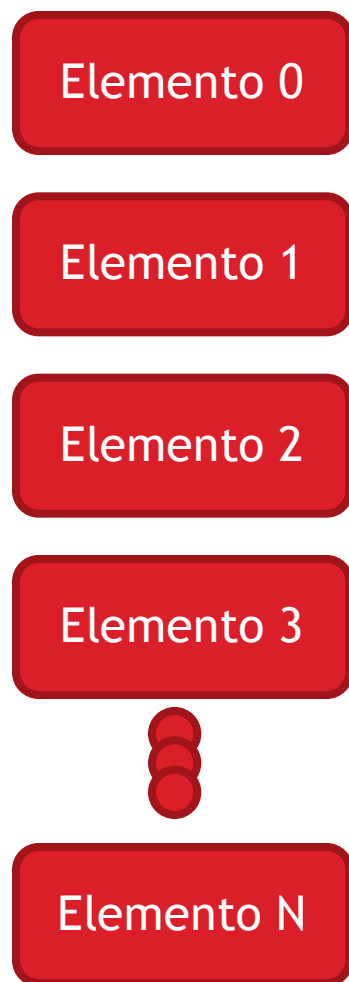


Elemento N

O primeiro elemento, denominamos de topo.

O topo possui função importante para uma Pilha, pois todas as operações utilizam apenas o topo

TAD PILHA



Como somente se acessa o topo, não se precisa utilizar estruturas de dados muito complexas para implementá-la.

Uma lista encadeada simples já resolveria muito bem para a implementação de uma pilha.

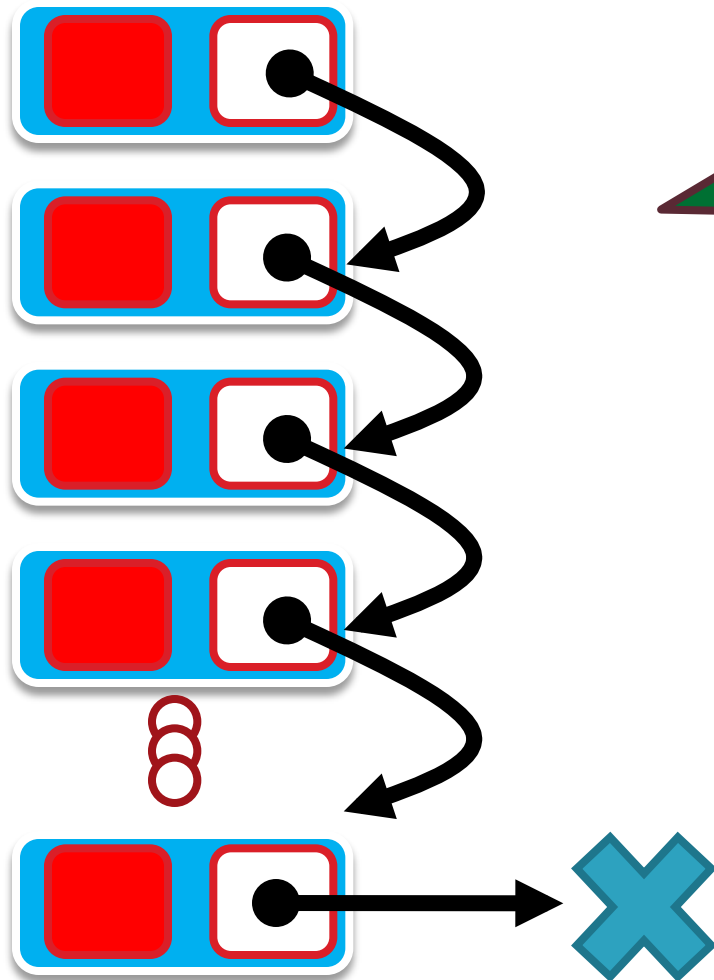
Mas é indicado trabalhar com estruturas genéricas, para evitar reimplementações (retrabalhos)

TAD PILHA



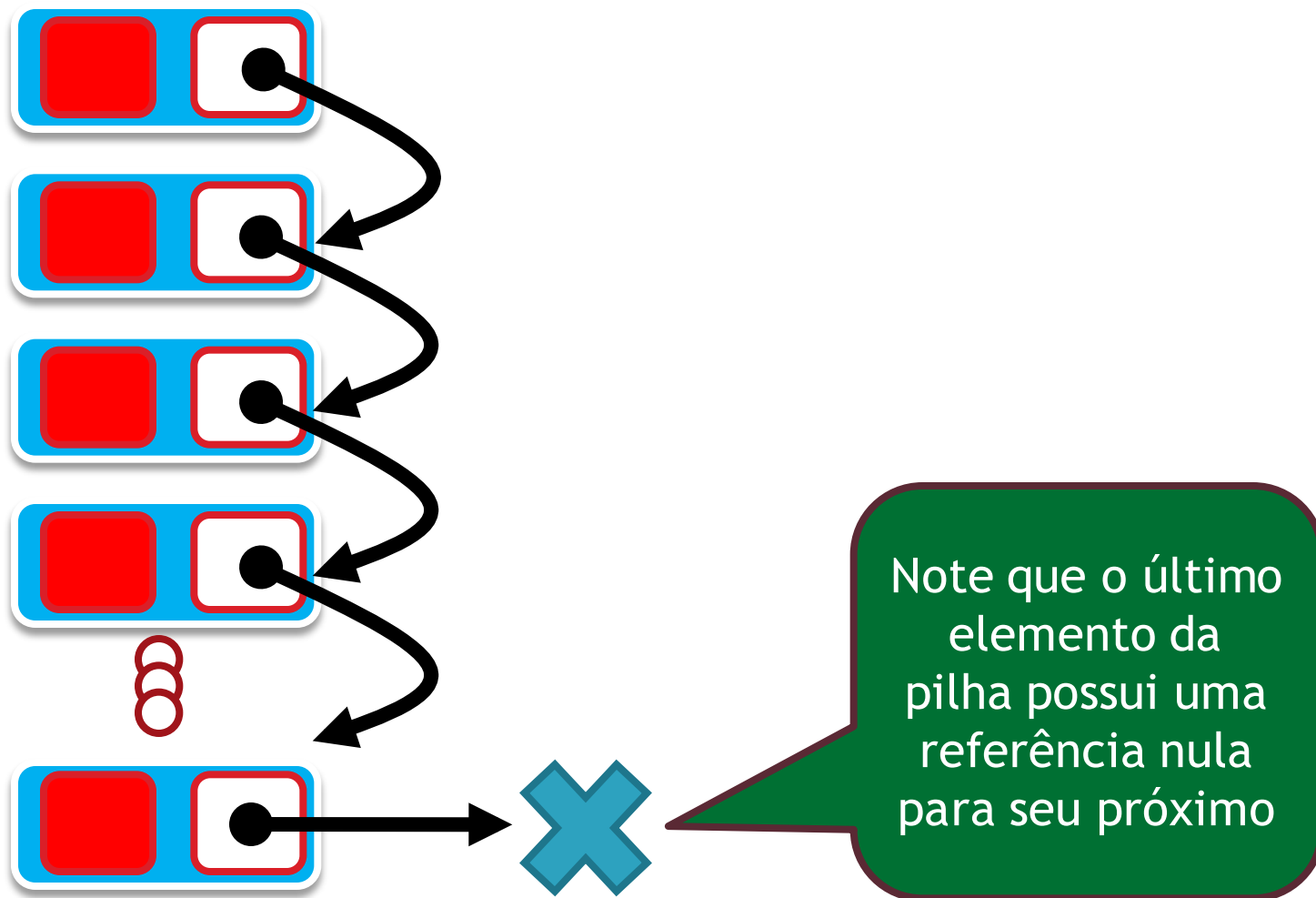
Portanto, iremos trabalhar com uma pilha implementada através da estrutura que classificamos como Lista Encadeada Dinâmica Genérica.

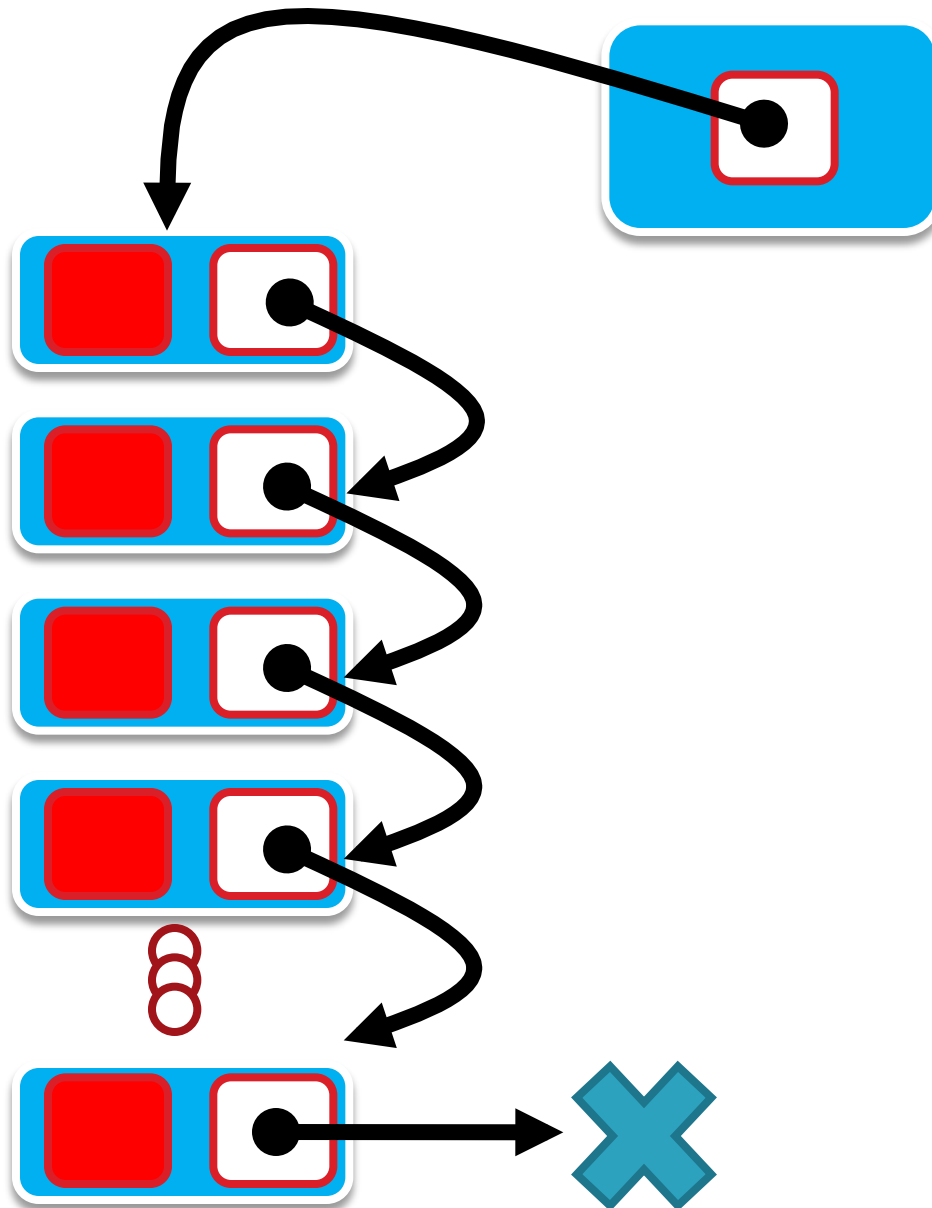
TAD PILHA



Toda estrutura de encadeamento simples deve acomodar os dados utilizando referências para os próximos elementos.

TAD PILHA





A Pilha deve ser representada por uma estrutura que contenha o Topo, pois a partir do mesmo, consegue-se chegar a qualquer elemento

Vamos ver como
ficam cada uma das
estruturas

TPILHA

TPilha

Referência
do topo

A estrutura que
representa a Pilha
deve ter a
referência do topo

TPILHA

TPilha

TElementoP
* topo;

A referência é um
ponteiro para o
elemento da Pilha

TPILHA

TPilha

TElementoP
* topo;



```
template<typename TIPO>
struct TPilha{
    TElementoP<TIPO> * topo;
};
```



TPILHA

Cada elemento de Pilha, como uma lista encadeada simples, possui um registro para dado e a referência para o próximo elemento.

TElementoP

DADO

**REFERÊNCIA
PARA
PRÓXIMO**

TPILHA

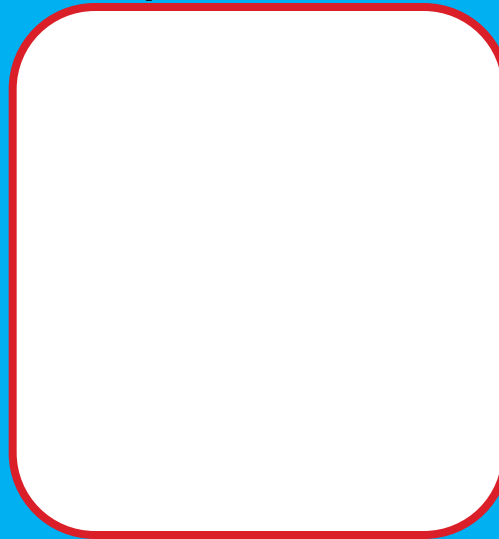
O dado é do tipo genérico.
A referência do próximo
elemento é um ponteiro para
TElementoP.

TElementoP

TIPO dado;



TElementoP
* proximo;



TPILHA

TElementoP



```
template<typename TIPO>
struct TElementoP{
    TIPO dado;
    TElementoP<TIPO> *proximo;
};
```

Como funcionam as
principais operações?

PUSH

- ⦿ A operação PUSH adiciona um item à pilha.
- ⦿ No momento da inserção deve-se:
 1. criar um novo elemento;
 2. acomodar o dado no novo elemento;
 3. fazer o elemento ter como próximo a referência do topo; e
 4. referência do topo deve ser alterada para o endereço de memória do novo elemento.
- ⦿ PUSH é equivalente à operação “insere início”

PUSH (INSERIR)

TPilha



A pilha está vazia!
Logo, a referência
para seu topo é
NULL

PUSH (INSERIR)

TPilha



Primeiro cria-se o
elemento

PUSH (INSERIR)

TPilha

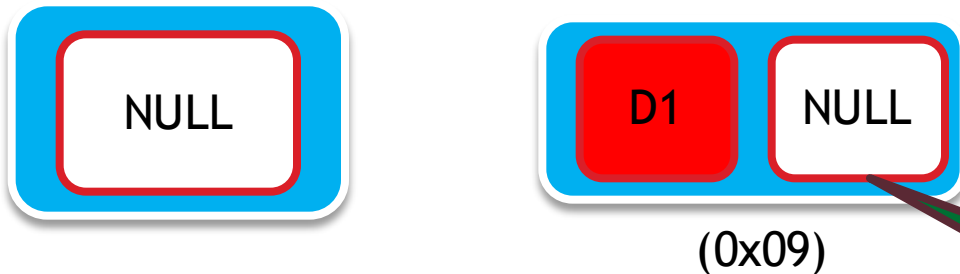


(0x09)

Depois acomoda-se o
dato

PUSH (INSERIR)

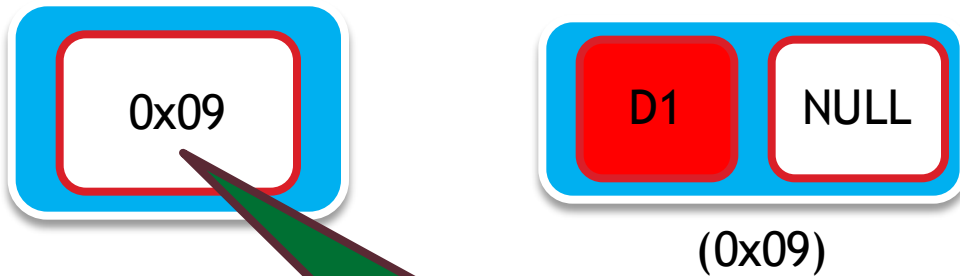
TPilha



A referência do próximo no novo elemento deve ser o mesmo endereço que está gravado como Topo da Pilha

PUSH (INSERIR)

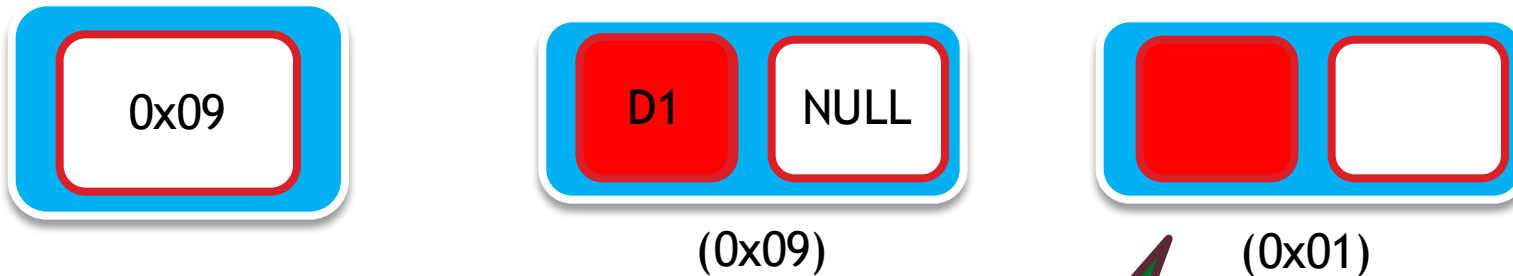
TPilha



Depois, o Topo deve ser atualizado para o endereço do novo elemento

PUSH (INSERIR)

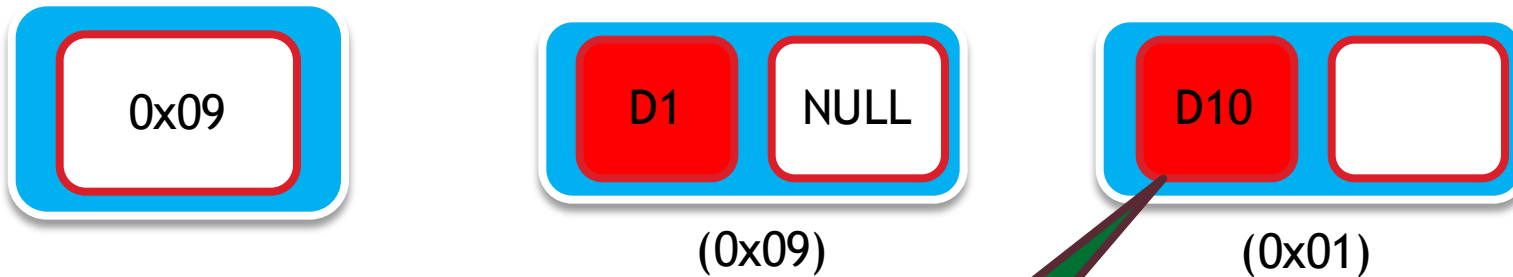
TPilha



Para acomodar um novo dado na pilha, primeiro cria-se o elemento

PUSH (INSERIR)

TPilha



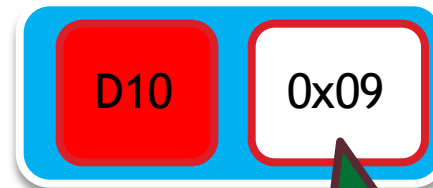
Depois acomoda-se o
dato

PUSH (INSERIR)

TPilha



(0x09)

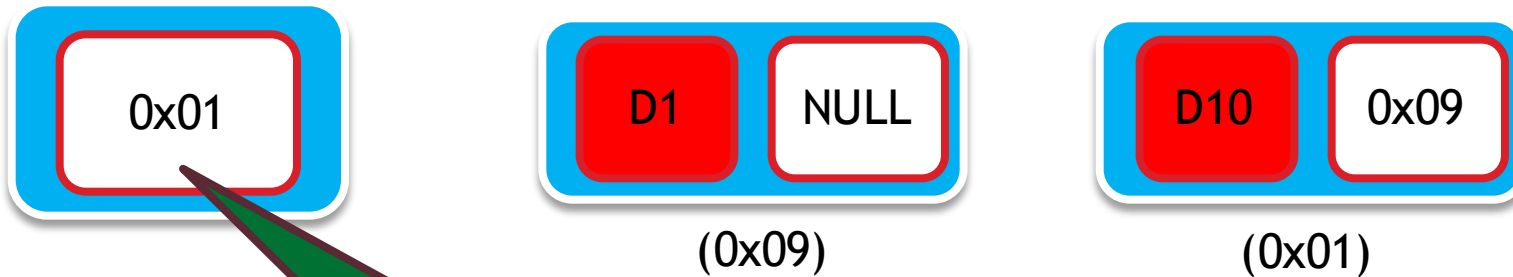


(0x01)

A referência do próximo no novo elemento deve ser o mesmo endereço que está gravado como Topo da Pilha

PUSH (INSERIR)

TPilha



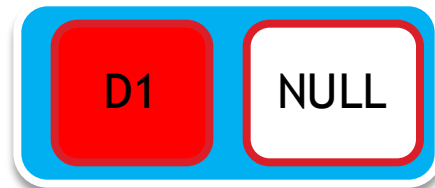
Depois, o Topo deve ser atualizado para o endereço do novo elemento

POP

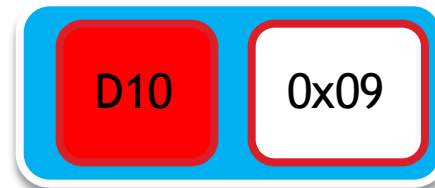
- ⦿ A operação POP retira um item da pilha.
- ⦿ Note que esta operação, deve retornar o dado do elemento recém retirado.
- ⦿ Para retirar deve-se:
 1. realizar uma cópia do dado do topo;
 2. gravar a referência do topo atual da lista;
 3. gravar no topo da pilha a referência do endereço armazenado em próximo do topo atual;
 4. liberar memória alocada pelo endereço do antigo topo; e
 5. retornar dado do elemento recém removido.
- ⦿ POP é equivalente à operação “remove início”

POP (RETIRADA)

TPilha



(0x09)



(0x01)

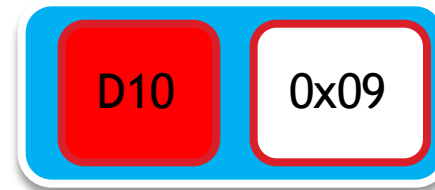
Vamos realizar a
operação de POP

POP (RETIRADA)

TPilha



(0x09)



(0x01)



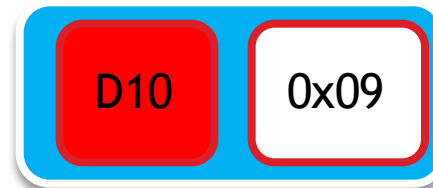
Inicialmente deve se
realizar uma cópia
do dado do
elemento que está
no topo

POP (RETIRADA)

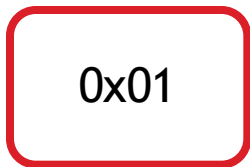
TPilha



(0x09)



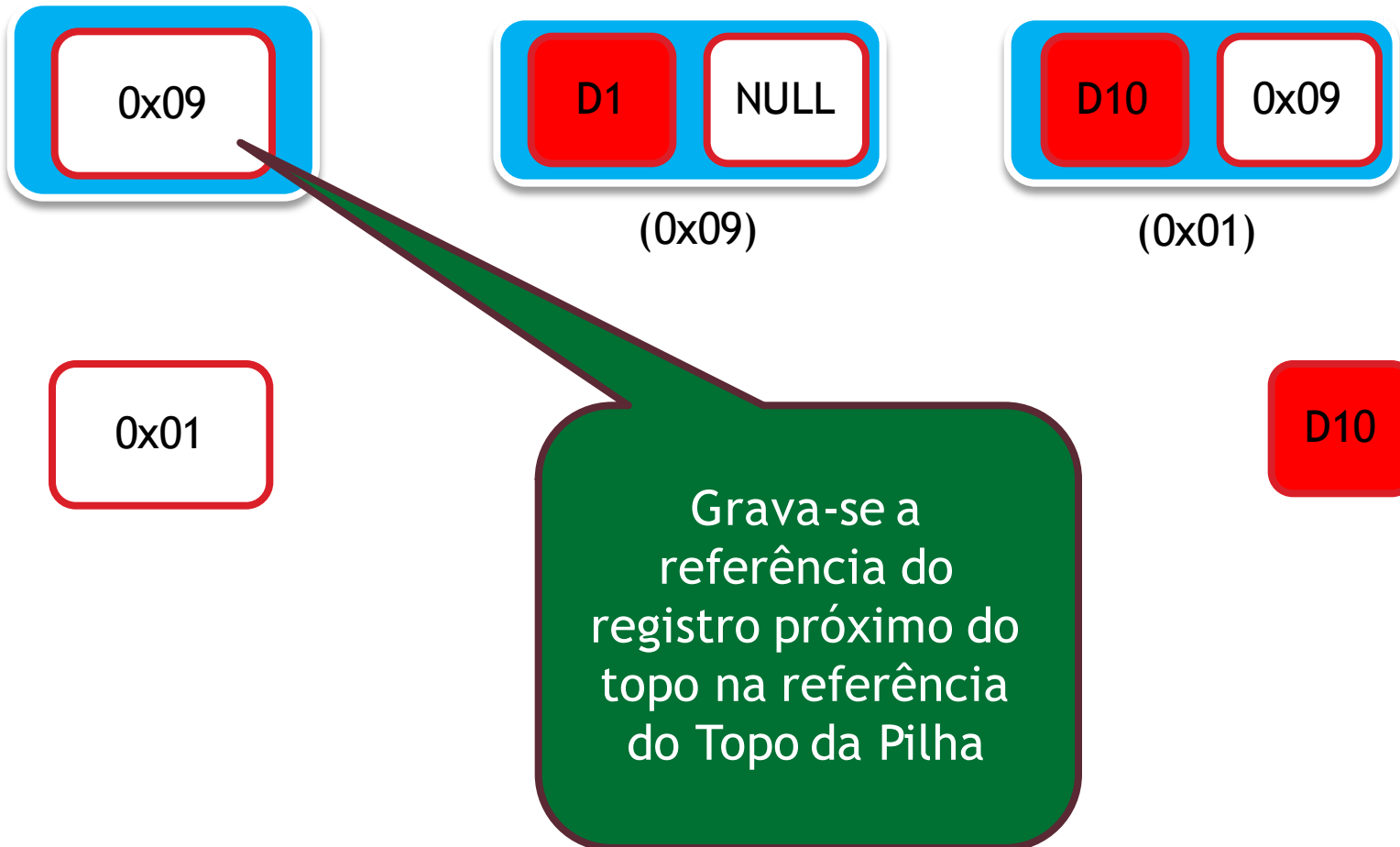
(0x01)



Depois, deve-se
gravar a referência
de memória do topo
atual

POP (RETIRADA)

TPilha

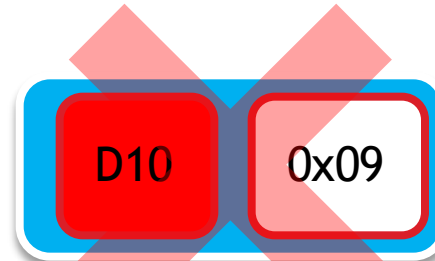


POP (RETIRADA)

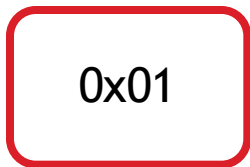
TPilha



(0x09)



(0x01)



Agora solicita-se
liberação da
memória do
endereço do antigo
topo



POP (RETIRADA)

TPilha



(0x09)



Ao final, deve se
retornar o dado do
elemento recém
apagado

POP (RETIRADA)

TPilha



(0x09)

Agora vamos realizar novamente a operação de POP.

Para isto realizamos uma cópia do dado gravado no topo

A red rounded rectangle containing the text 'D1'.

POP (RETIRADA)

TPilha



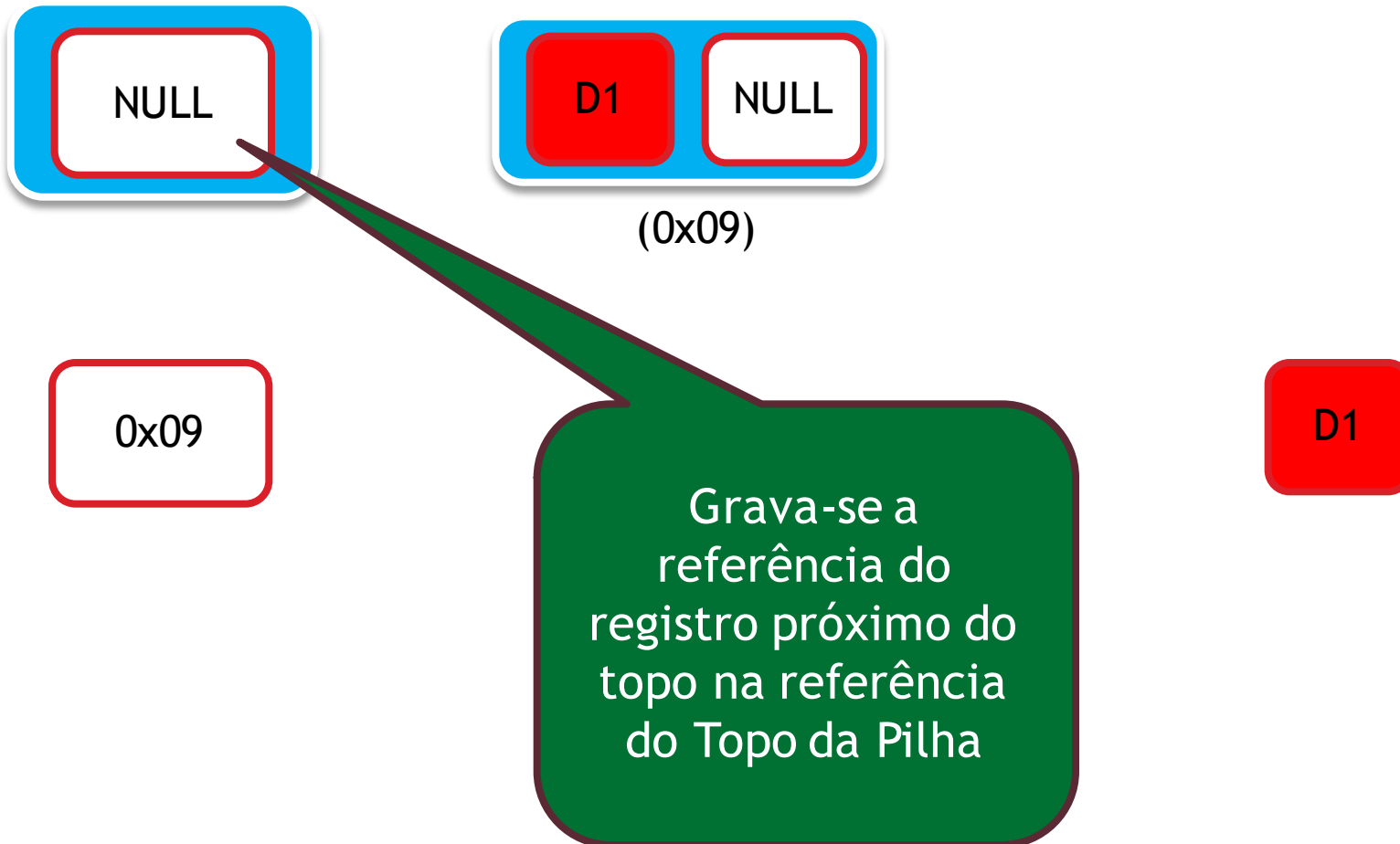
(0x09)



Depois, deve-se
gravar a referência
de memória do topo
atual

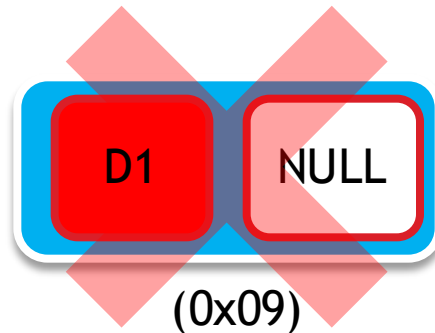
POP (RETIRADA)

TPilha



POP (RETIRADA)

TPilha



Agora solicita-se
liberação da
memória do
endereço do antigo
topo



POP (RETIRADA)

TPilha

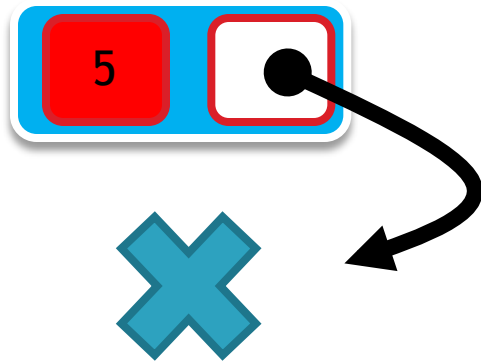


D1

Ao final, deve se
retornar o dado do
elemento recém
apagado

Recursividade e Pilha

FATORIAL DE 5



Note que o último elemento da pilha possui uma referência nula para seu próximo

FATORIAL

