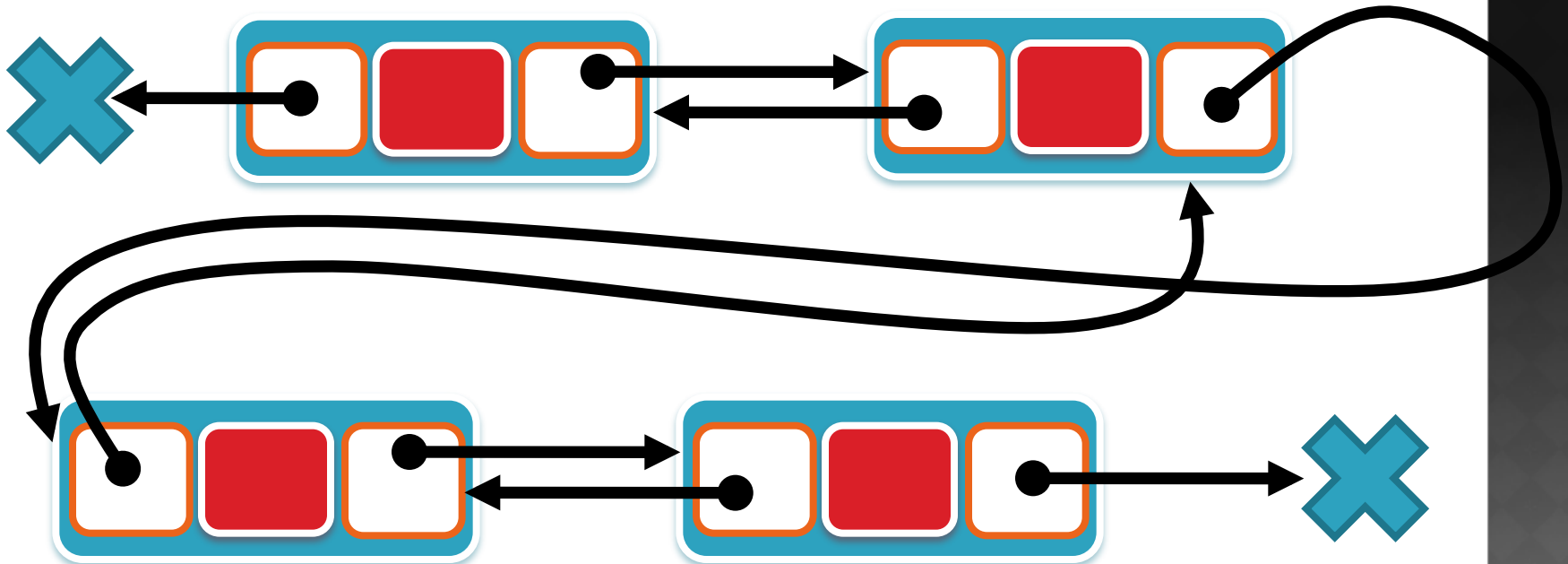


LISTA DUPLAMENTE ENCADEADA

LISTA DUPLAMENTE ENCADEADA

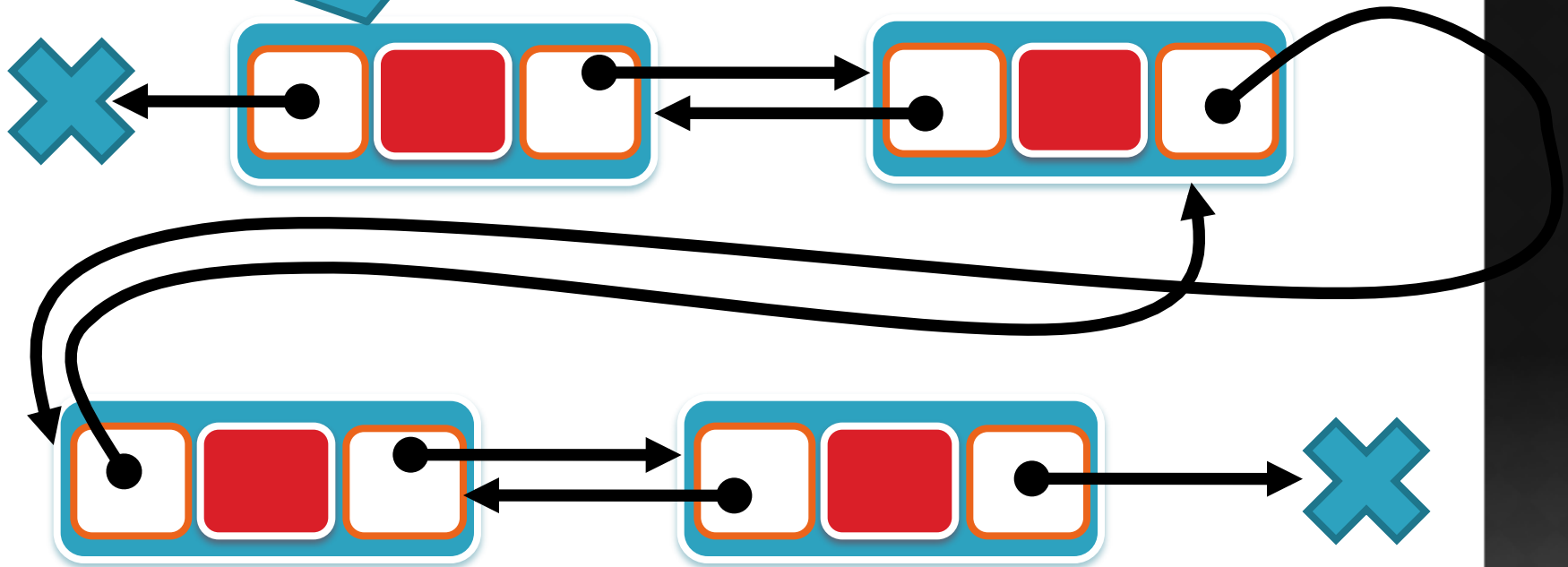
- Uma lista duplamente encadeada é uma lista onde cada elemento possui a referência dos nós imediatamente interior e próximo.



Cada elemento possui três campos: um vermelho e dois branco.

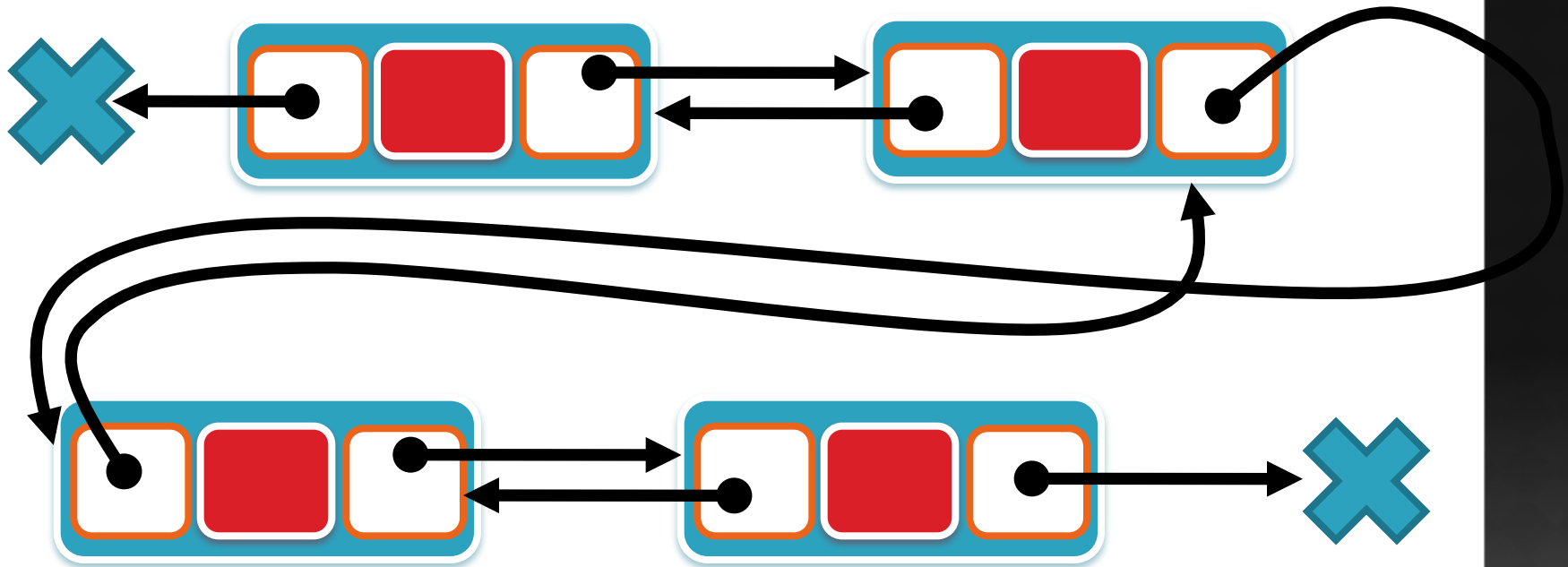
Neste exemplo, o vermelho simboliza o dado armazenado e o branco representa a referência para o anterior (esquerda) e próximo (direita) elemento .

ADA



Note que, a partir de qualquer elemento, consegue-se localizar todos os outros, devido ao encadeamento realizado através das referências: anterior e próximo.

ENCADDEADA



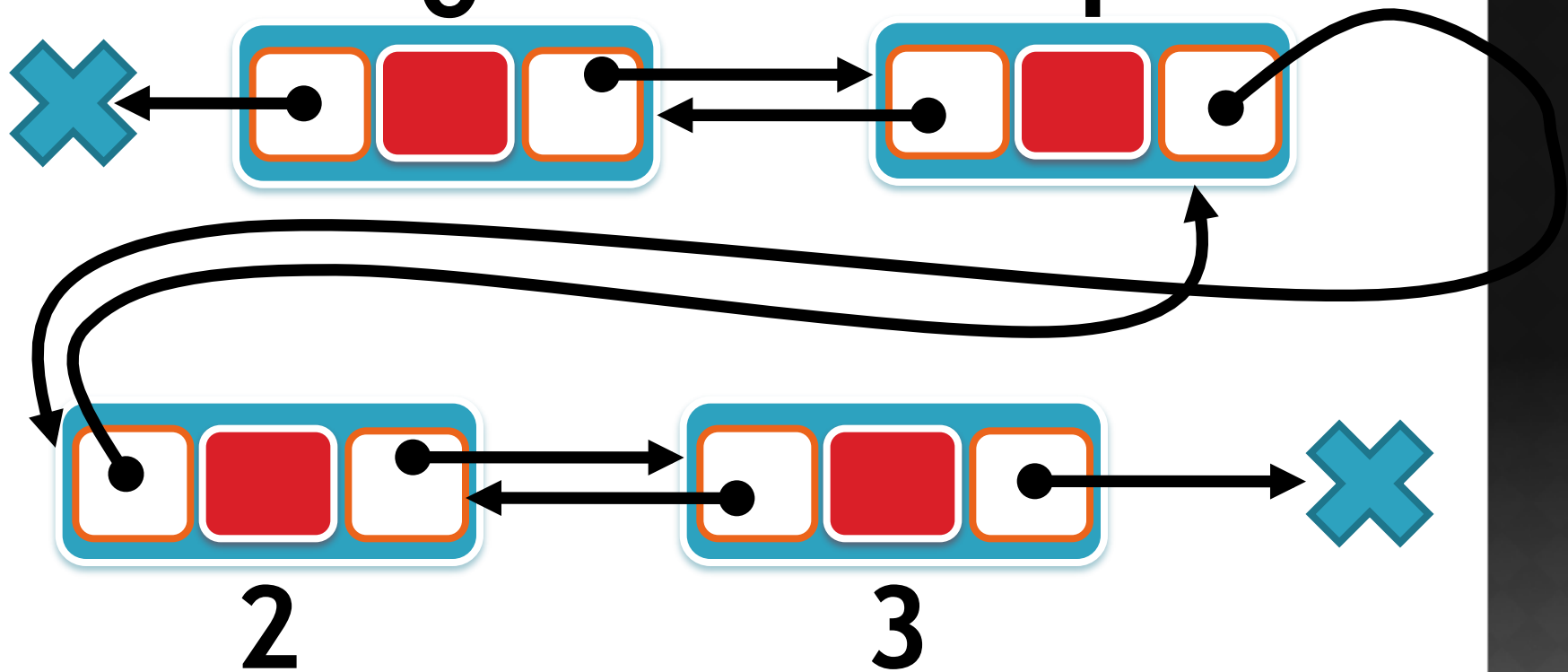
LISTA DU

Continuaremos utilizando as
posições a partir do zero

ADA

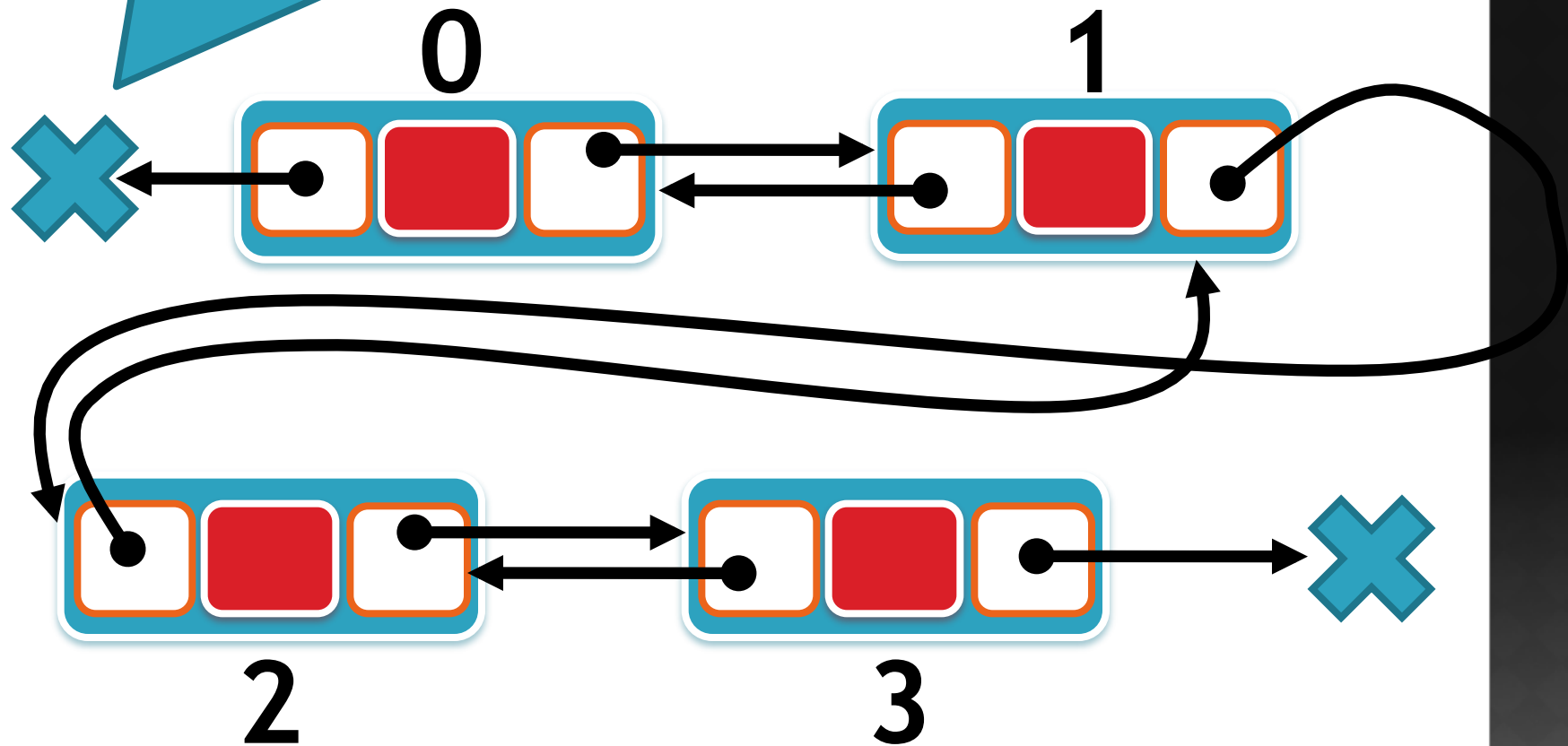
0

1



O primeiro e último elementos possuem, respectivamente, referências para anterior e próximo com o valor NULL

ENCADEADA



Em relação a lista com encadeamento simples...

Quais vantagens?

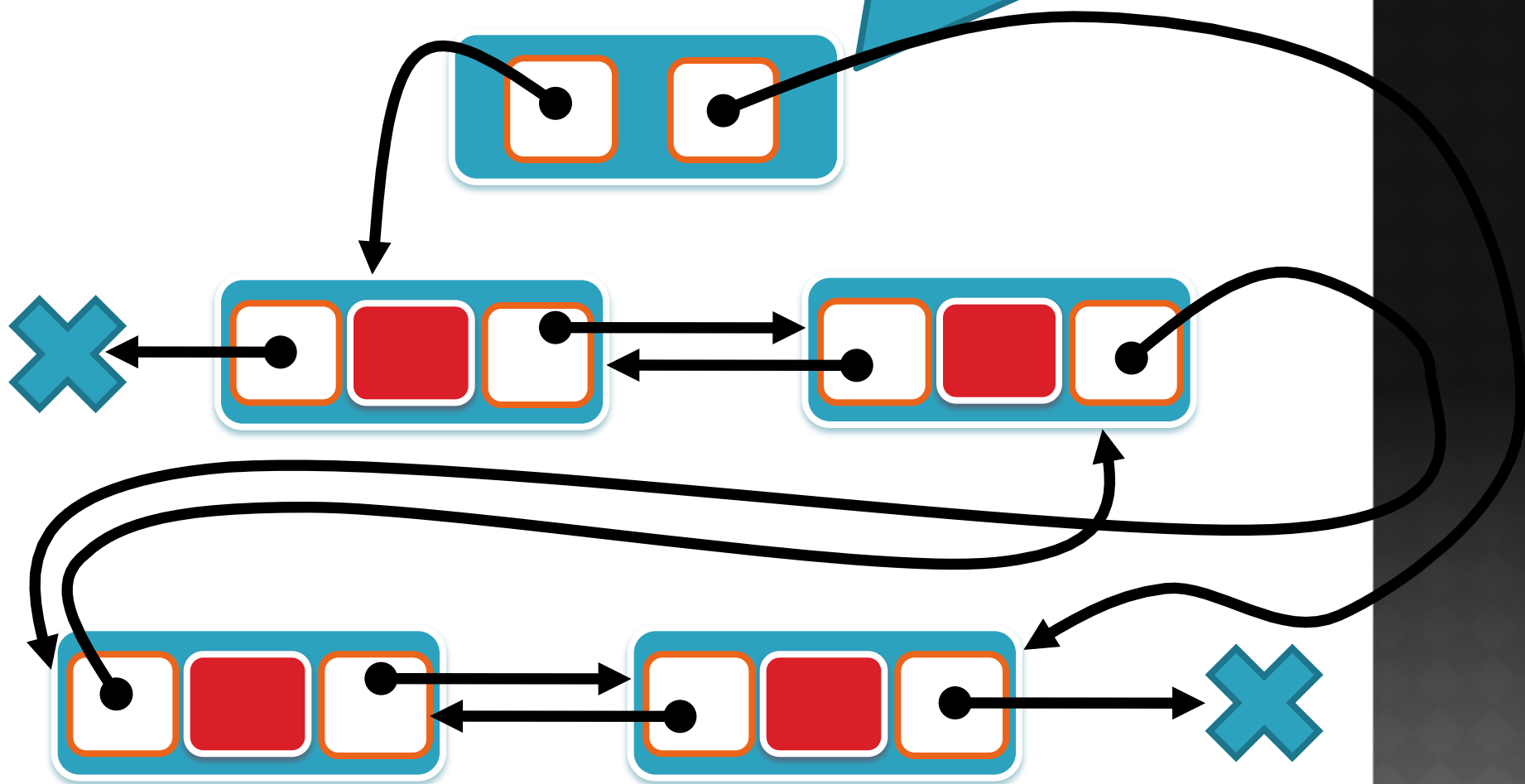
- pode se percorrer toda a lista a partir de qualquer elemento

Quais desvantagens?

- Mais trabalhosa de codificar que a Lista Encadeada

LISTA DUPLAMENTE ENCADEADA

Diferentemente de uma lista com encadeamento simples, em uma lista duplamente encadeada, gravamos a referência de dois elementos: o primeiro e o último.



Vamos evoluir conceitualmente
cada uma das estruturas para
depois pensarmos na codificação
da TAD (C++)

TELEMENTO

Temos três campos: um para o dado (vermelho) e outros dois para a referência do anterior e próximo elemento (branco)

REFERÊNCIA
ANTERIOR

DADO

REFERÊNCIA
PRÓXIMO

TLISTA

O TLista terá dois registros, que conterá a referência do primeiro e último elementos da nossa lista duplamente encadeada

REFERÊNCIA
PRIMEIRO

REFERÊNCIA
ÚLTIMO

CODIFICAÇÃO DO TAD

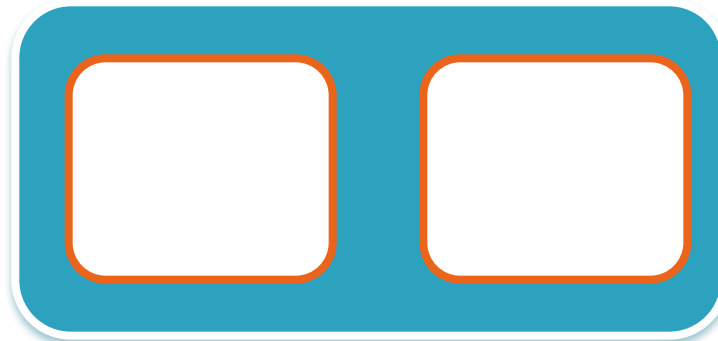
- ◉ O exemplo que será utilizado a seguir contextualizará uma lista duplamente encadeada para armazenamento de músicas
- ◉ Logo estaremos criando nossa *playlist* de músicas
- ◉ **QUAIS SERÃO OS DADOS?**
- ◉ **COMO SERÁ REALIZADA A REFERÊNCIA?**

(próximo...)

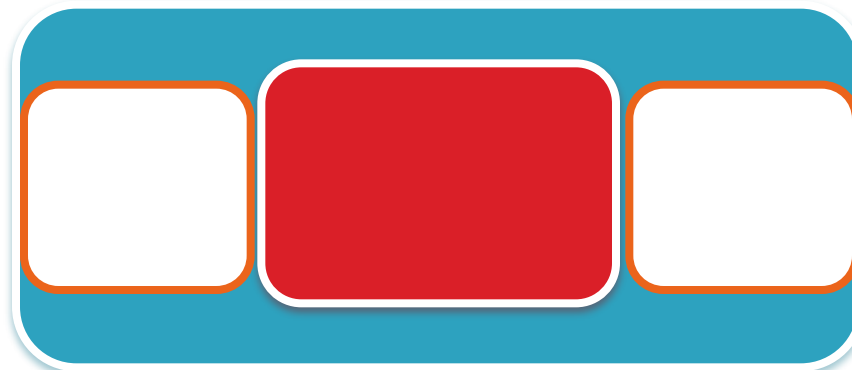
CODIFICAÇÃO DO TAD

◉ Vamos definir:

TListaDE (PlayList)



TElementoDE (elemento)



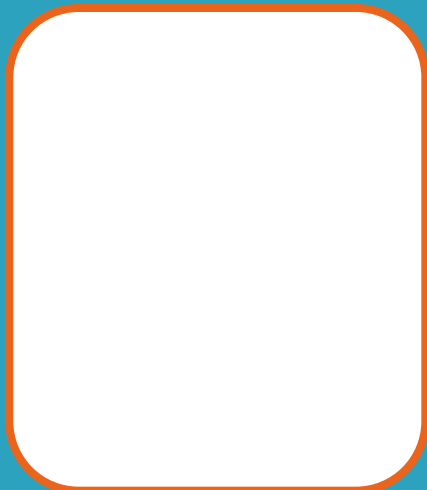
CODIFICAÇÃO DO TAD

- ⦿ Tendo definido os dados e o que se tomará como referência, vamos definir como ficarão as estruturas
- ⦿ Iniciaremos com a estrutura do elemento

CODIFICAÇÃO DO TELEMENTO

TElementoDE

anterior

A large, empty white rectangular box with rounded corners and a thick orange border, positioned below the 'anterior' label.

musica

A large, solid red rectangular box with rounded corners and a thick orange border, positioned below the 'musica' label.

proximo

A large, empty white rectangular box with rounded corners and a thick orange border, positioned below the 'proximo' label.

COD

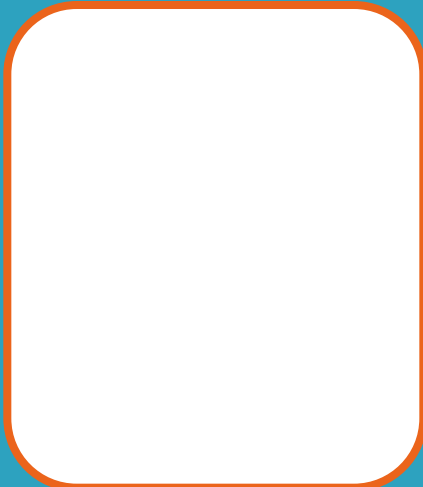
- Os registros serão:
- anterior: ponteiro para TElementoDE; e
 - musica: do tipo TMusica; e
 - proximo: ponteiro para TElementoDE.

TElementoDE

TElementoDE *
anterior;

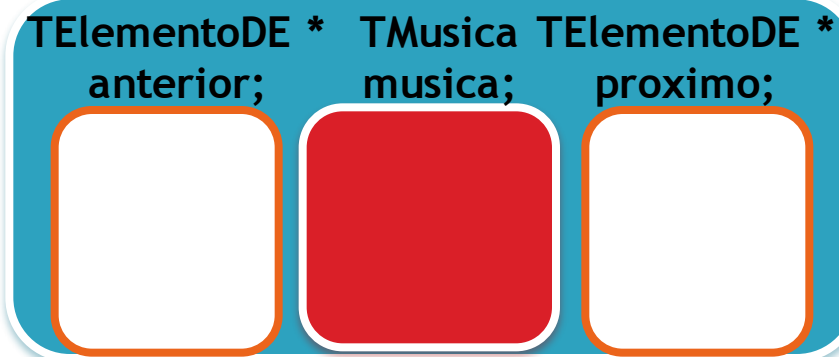
TMusica
musica;

TElementoDE *
proximo;



CODIFICAÇÃO DO TELEMENTO

TElementoDE



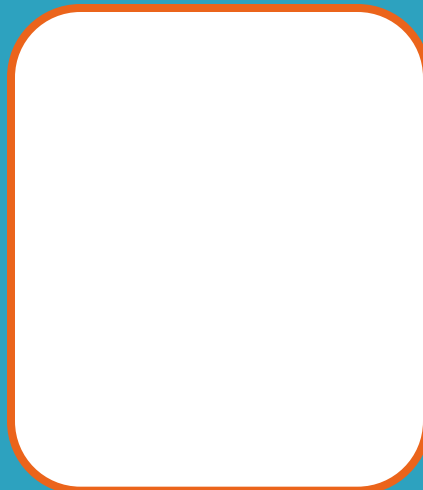
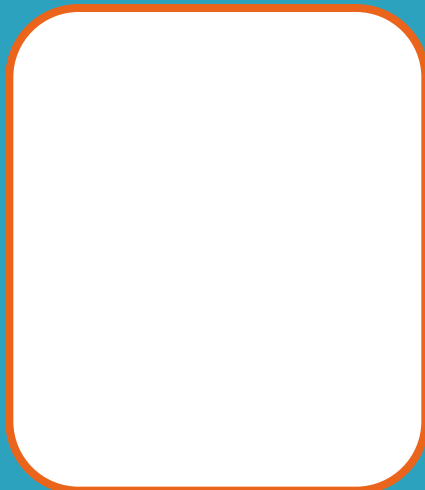
```
struct TElementoDE{  
    TMusica musica;  
    TElementoDE * anterior;  
    TElementoDE * proximo;  
};
```

CODIFICAÇÃO DO TLISTA

TListaDE

inicio

fim

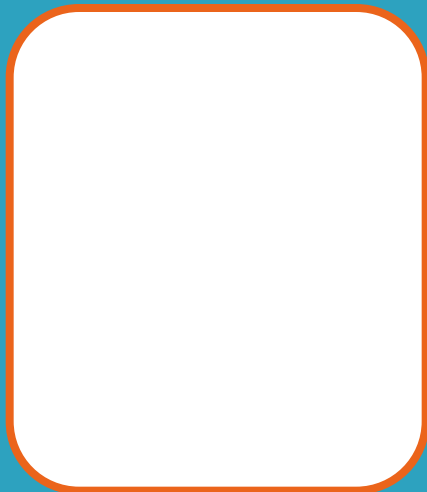



CODIFICAÇÃO DO T

Registros armazenam endereços do início e fim da lista

TListaDE

TElementoDE * TElementoDE *
início; fim;

A large, empty, rounded rectangular box with an orange border, intended for the 'início' field.A large, empty, rounded rectangular box with an orange border, intended for the 'fim' field.

CODIFICAÇÃO DO TLISTA

TListaDE

TElementoDE * TElementoDE *
início; fim;

```
struct TListaDE{  
    TElementoDE * início;  
    TElementoDE * fim;  
};
```



FUNCIONAMENTO DA TAD

- ◉ Depois de discutirmos e definirmos as estruturas relacionadas a uma lista duplamente encadeada, a seguir veremos seu funcionamento através de uma sequência de ilustrações

FUNCIONAMENTO DA TAD

Esta é nossa lista encadeada. Ela está vazia.

Logo, não possui primeiro endereço de elemento para armazenar, nem último.

Devido a isto, armazena-se NULL no registro de inicio e fim

Lista

NULL

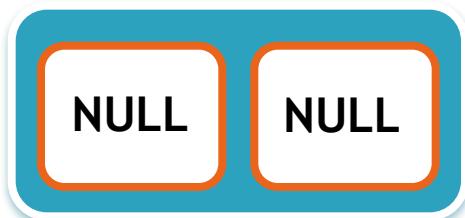
NULL

FUNCIONAMENTO DA TAD

TMusica



TListaDE



A música X é criada e deseja inserí-la na lista

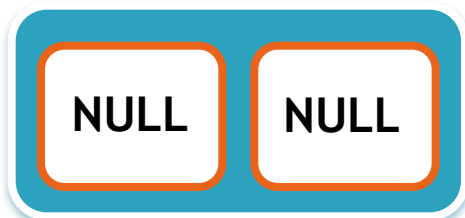
FUNCIONAMENTO DA TAD

TElementoDE



0x08

TListaDE



Para isto, precisa-se utilizar um elemento para armazená-la. Logo criamos um TElementoDE, atribuímos a música ao mesmo

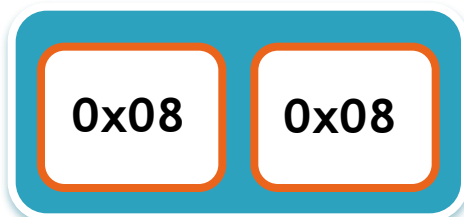
FUNCIONAMENTO DA TAD

TElementoDE



0x08

TListaDE

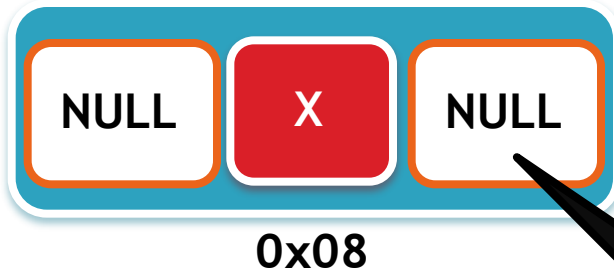


Como a lista está vazia, ao inserir um novo elemento, este torna-se o primeiro e o último elemento.

Consequentemente, os registros inicio e fim de TListaDE armazenam o endereço do elemento utilizado para a música X

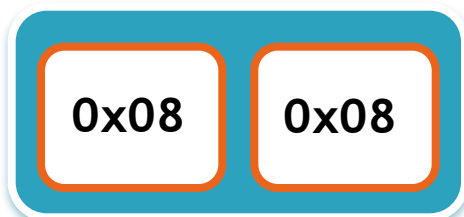
FUNCIONAMENTO DA TAD

TElementoDE



Como este elemento é o primeiro e o último, seu ponteiro para anterior e próximo deve armazenar um endereço de memória nulo

TListaDE



FUNCIONAMENTO DA TAD

TElementoDE

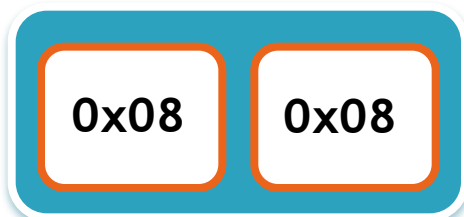


0x08

TMusica



TListaDE



Deseja-se agora inserir a música Q

FUNCIONAMENTO DA TAD

TElementoDE

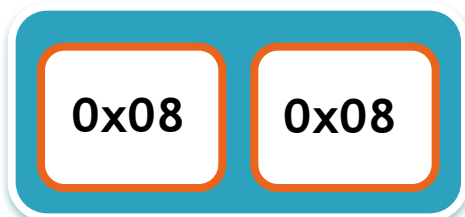


0x08

TMusica



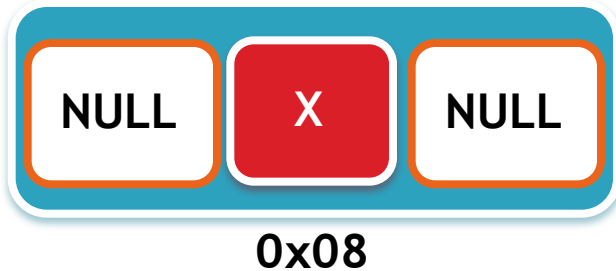
TListaDE



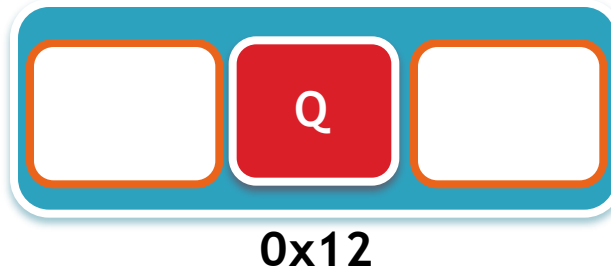
Deseja-se agora inserir a música Q

FUNCIONAMENTO DA TAD

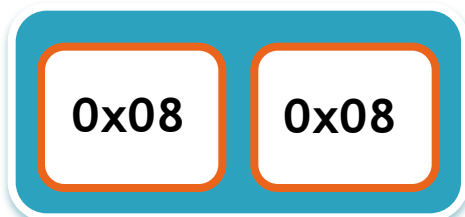
TElementoDE



TElementoDE



TListaDE



Envolvemos a música Q através de um novo elemento alocado, cujo o endereço é 0x12

FUNCIONAMENTO DA TAD

TElementoDE



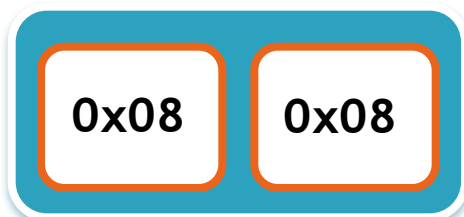
0x08

TElementoDE



0x12

TListaDE



O registro próximo de 0x08 deve ser atualizado com o endereço de memória do novo elemento.

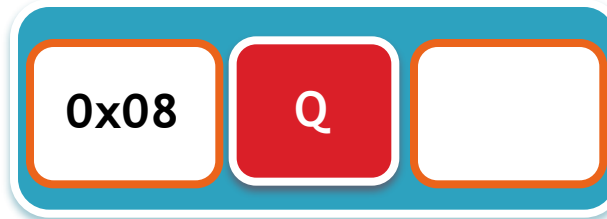
FUNCIONAMENTO DA TAD

TElementoDE



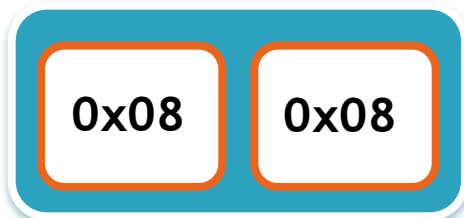
0x08

TElementoDE



0x12

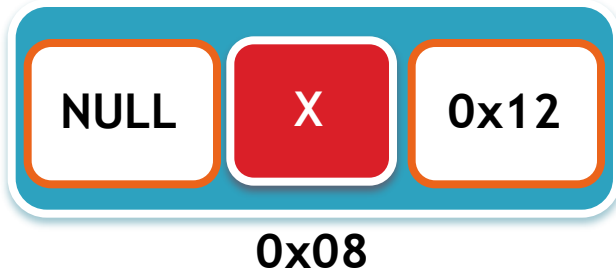
TListaDE



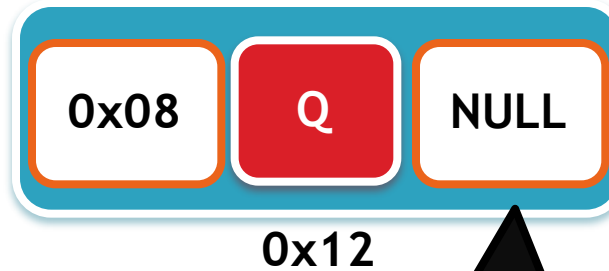
o registro anterior de 0x12 deve
ser atualizado com o valor 0x08

FUNCIONAMENTO DA TAD

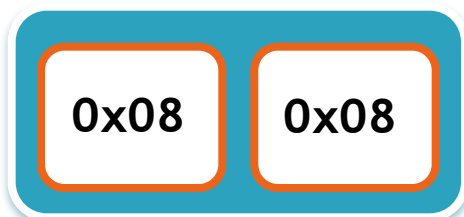
TElementoDE



TElementoDE



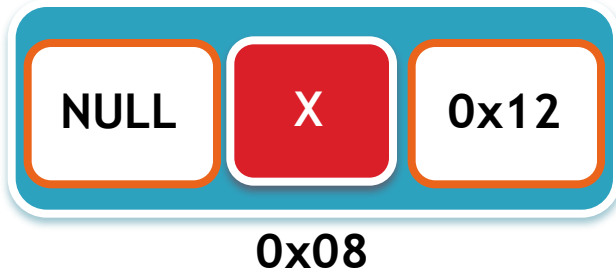
TListaDE



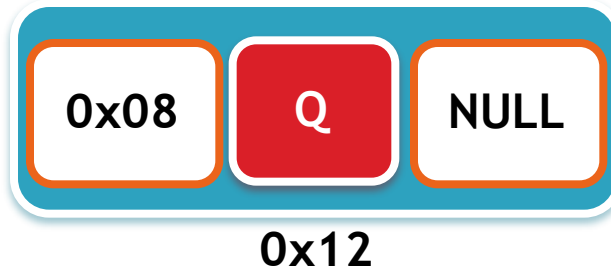
como este passa a ser p último elemento, seu ponteiro para próximo deve receber o valor NULL...

FUNCIONAMENTO DA TAD

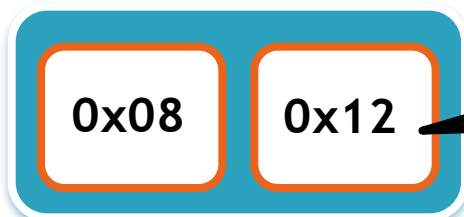
TElementoDE



TElementoDE



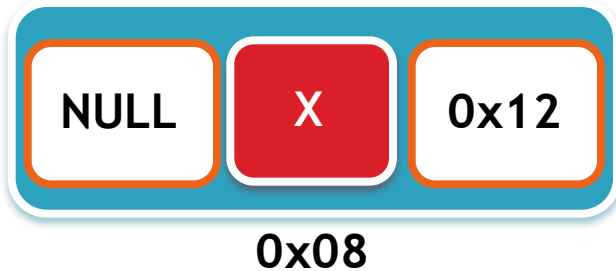
TListaDE



... e deve ser atualizado o último endereço da lista para 0x12 (novo último elemento).

FUNCIONAMENTO DA TAD

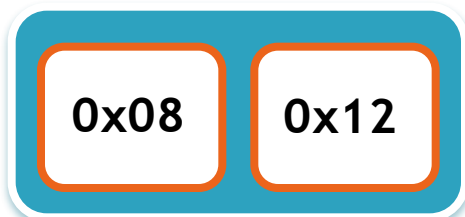
TElementoDE



TElementoDE



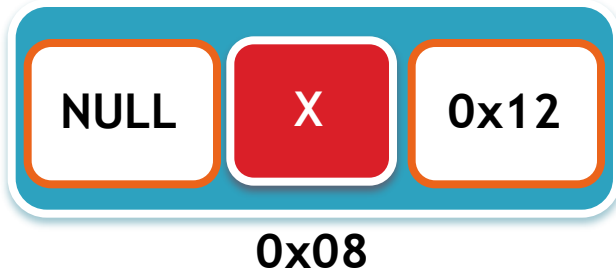
TListaDE



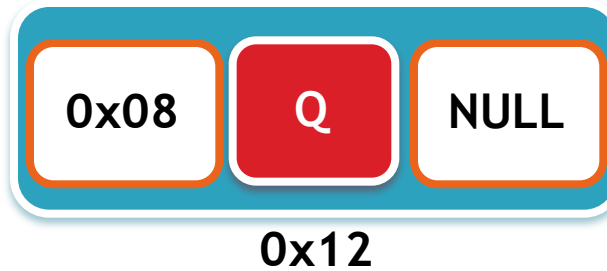
Vamos inserir mais uma música
(música A), ao final da lista

FUNCIONAMENTO DA TAD

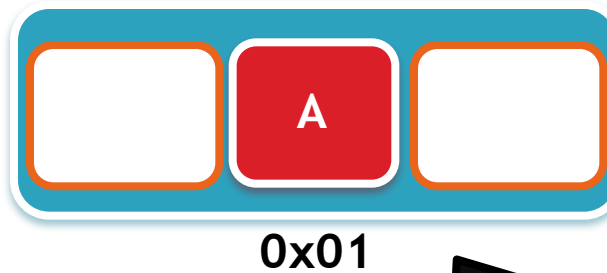
TElementoDE



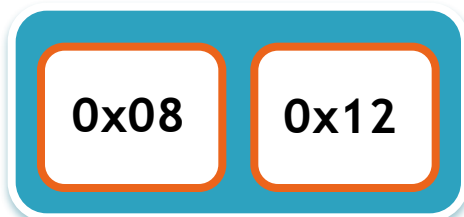
TElementoDE



TElementoDE



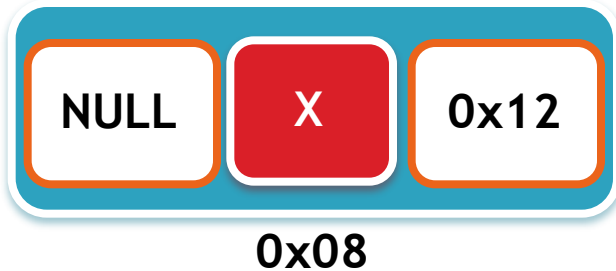
TListaDE



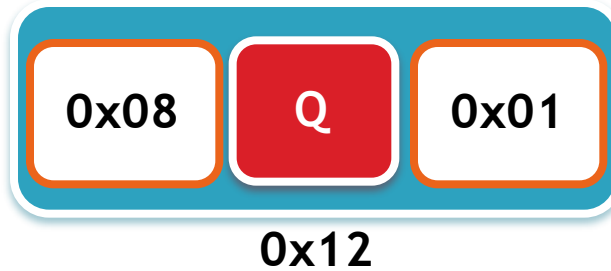
Deve se criar um novo elemento e inserir a nova música

FUNCIONAMENTO DA TAD

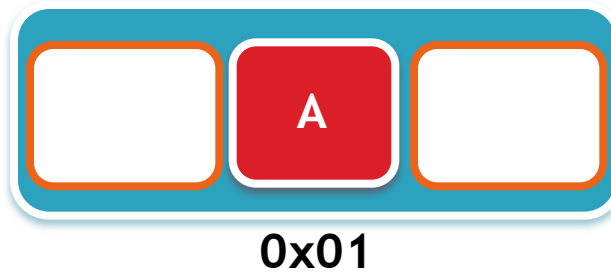
TElementoDE



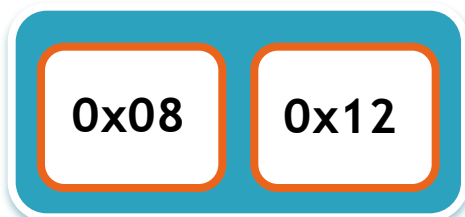
TElementoDE



TElementoDE



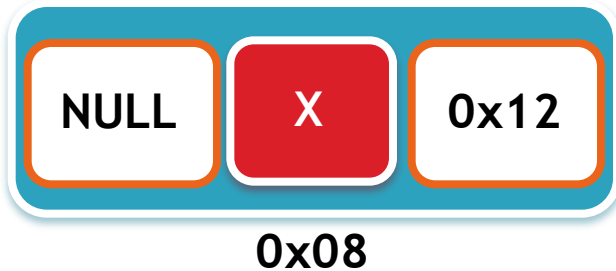
TListaDE



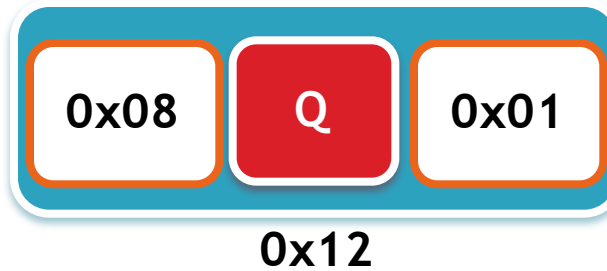
O elemento 0x12 deve atualizar o endereço do próximo elemento para 0x01 (novo elemento)

FUNCIONAMENTO DA TAD

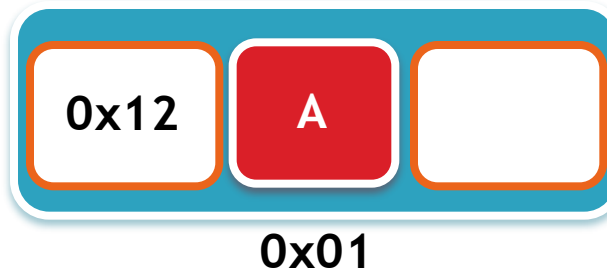
TElementoDE



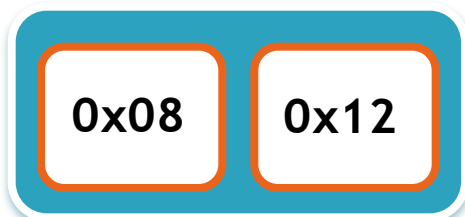
TElementoDE



TElementoDE



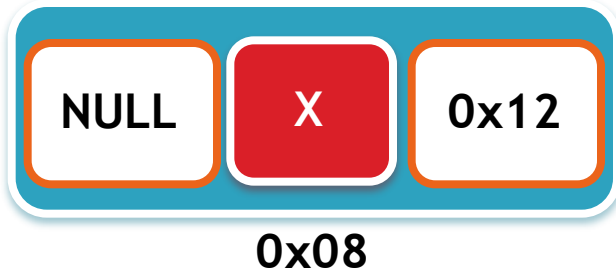
TListaDE



O anterior de 0x01 deve ser atualizado para 0x12

FUNCIONAMENTO DA TAD

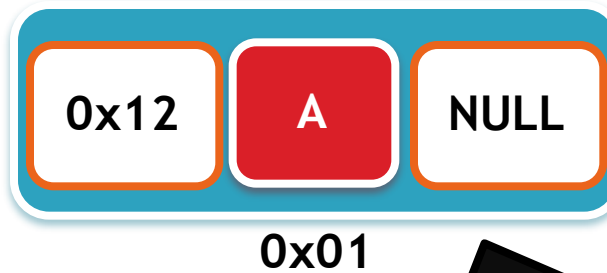
TElementoDE



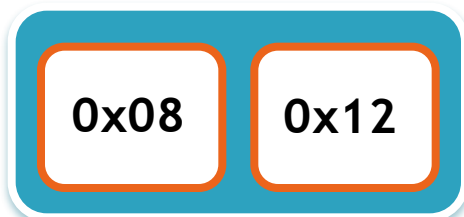
TElementoDE



TElementoDE



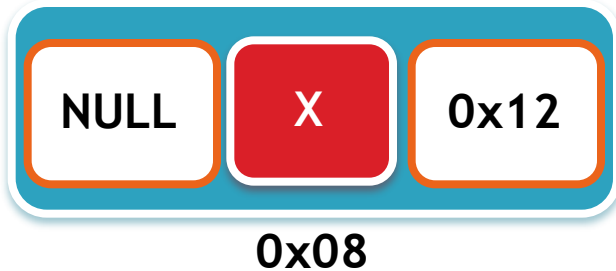
TListaDE



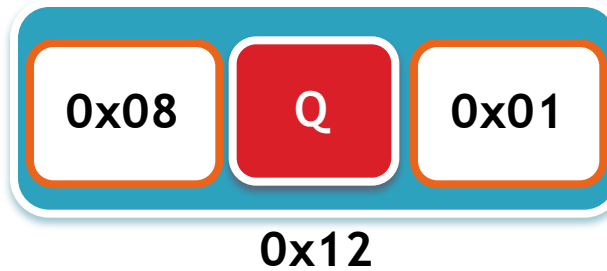
Como 0x01 é o último elemento, deve-se atualizar seu registro próximo para NULL...

FUNCIONAMENTO DA TAD

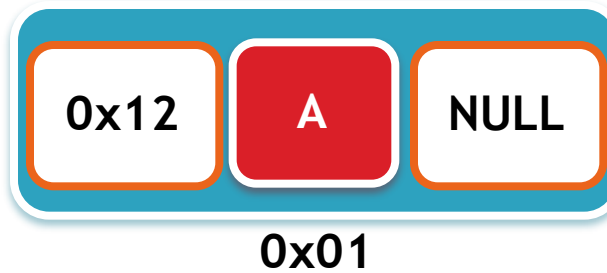
TElementoDE



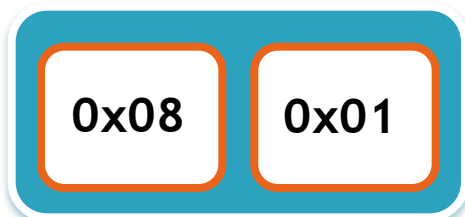
TElementoDE



TElementoDE



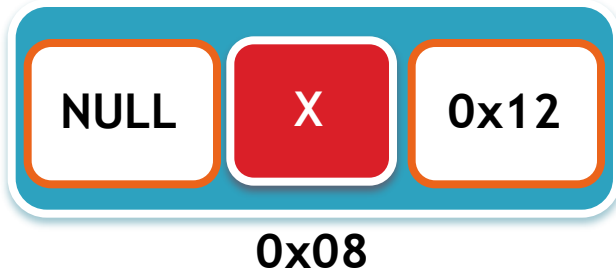
TListaDE



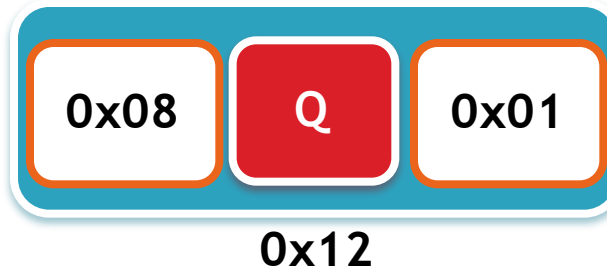
... e o registro do último elemento da lista

FUNCIONAMENTO DA TAD

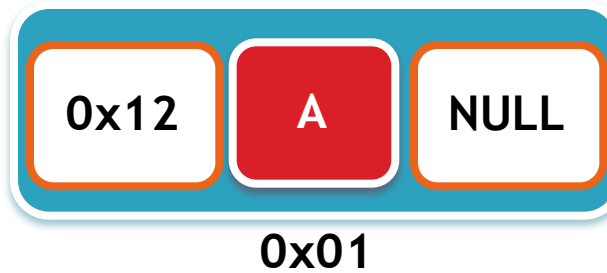
TElementoDE



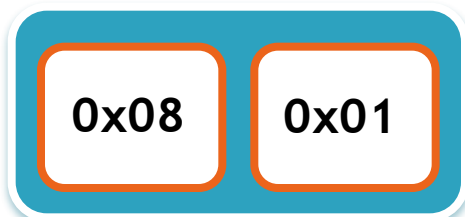
TElementoDE



TElementoDE



TListaDE



Vamos adicionar a música “\$” na posição 2 (lembrando que o primeiro elemento possui posição 0).

FUNCIONAMENTO DA TAD

TElementoDE



0x08

TElementoDE



0x12

TElementoDE



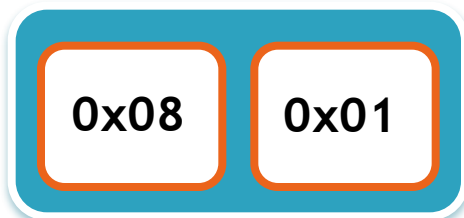
0x33

TElementoDE



0x01

TListaDE



Devemos acomodar a nova música
na elemento

FUNCIONAMENTO DA TAD

TElementoDE



0x08

TElementoDE



0x12

TElementoDE



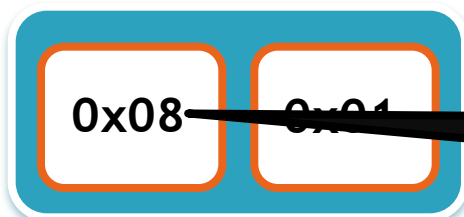
0x33

TElementoDE



0x01

TListaDE



Deve-se percorrer a lista a partir do primeiro elemento, localizando o elemento com posição anterior a que se deseja adicionar

FUNCIONAMENTO DA TAD

TElementoDE



0x08

TElementoDE



0x12

TElementoDE



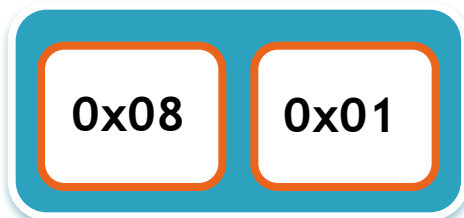
0x33

TElementoDE



0x

TListaDE



Seguindo a sequência:
0x08 → 0x12

0x12 é o elemento da posição
anterior a que desejamos inserir o
novo elemento

O novo elemento deve atualizar
seus registros para:

- anterior = 0x12 (endereço de
posição imediatamente menor)
- proximo = registro próximo do
elemento 0x12

A TAD

mentoDE

NU

Q

0x01

0x

0x12

TElementoDE

TElementoDE

0x12

\$

0x01

0x33

0x12

A

NULL

0x01

TListaDE

0x08

0x01

O valor do registro próximo do elemento 0x01 deverá conter o endereço do novo elemento

A TAD

mentoDE

NU

Q

0x33

0x08

0x12

TElementoDE

TElementoDE

0x12

\$

0x01

0x12

A

NULL

0x33

0x01

TListaDE

0x08

0x01

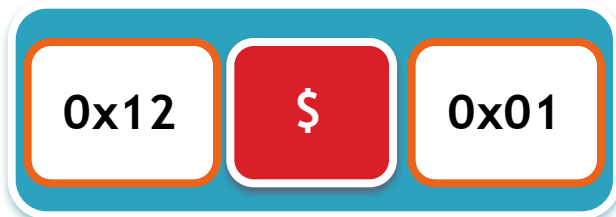
FUNCIONAMENTO

TElementoDE



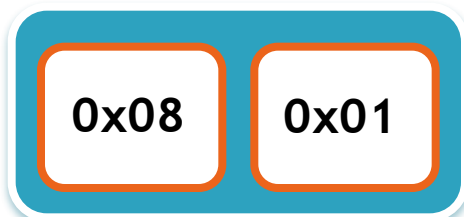
0x08

TElementoDE



0x33

TListaDE



o valor do registro anterior do elemento 0x01 deverá receber o endereço do novo elemento

TElementoDE



0x01

FUNCIONAMENTO DA TAD

TElementoDE



0x08

TElementoDE



0x12

TElementoDE



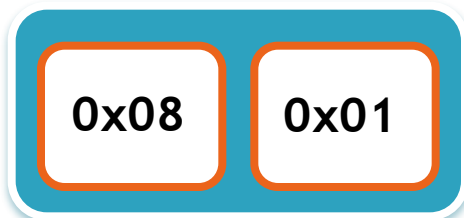
0x33

TElementoDE



0x01

TLista



Como o novo elemento não é o primeiro nem o último elemento, não são alterados os valores da lista!