



BUCKET LIST

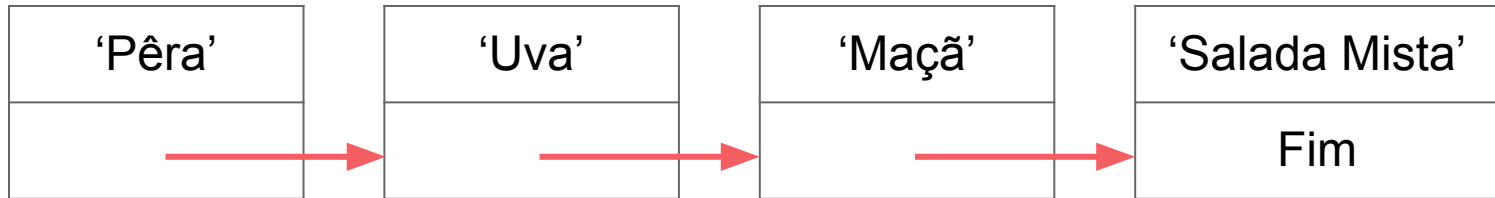
- ☒ Bucket of Fried Chicken
- ☒ Bucket of Shrimp
- ☐ Bucket of Tartar Sauce
- ☐ Bucket of Chili
- ☐ Bucket of Popcorn
- ☐ Bucket of Cholesterol Medicine

Aula 5 - Listas

Implementação Dinâmica

Estruturas de Dados 2018/1
Prof. Diego Furtado Silva

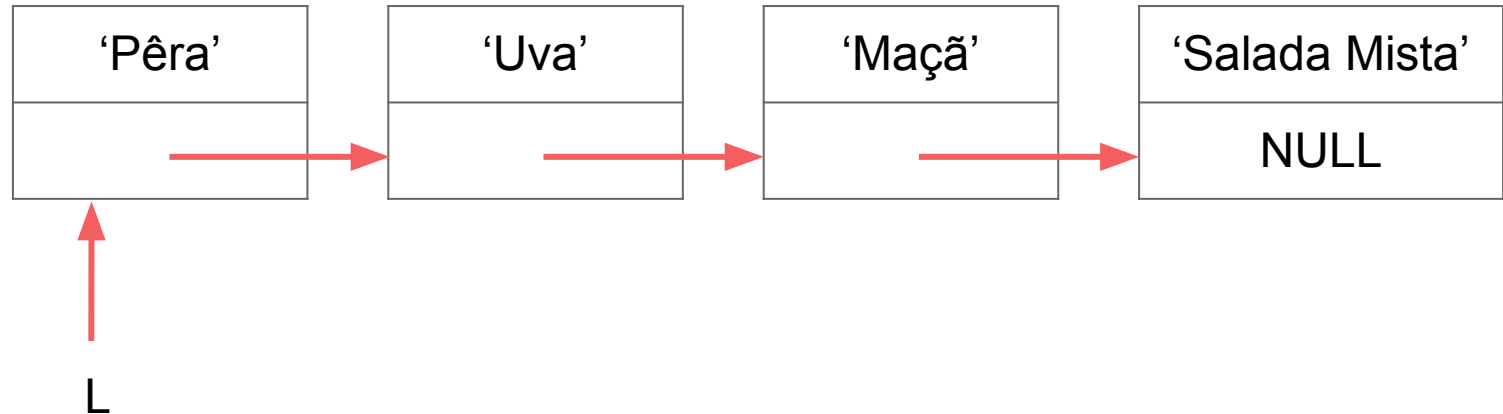
TAD Listas com listas ligadas



TAD Listas com listas ligadas

Vamos começar pela representação da lista

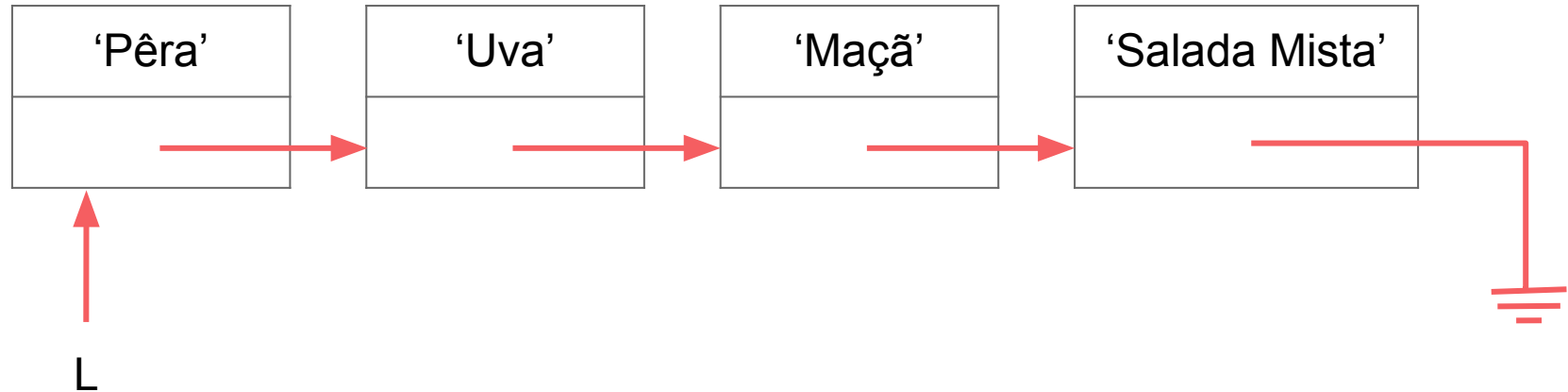
- Forma comum: manter um ponteiro “primeiro”



TAD Listas com listas ligadas

Vamos começar pela representação da lista

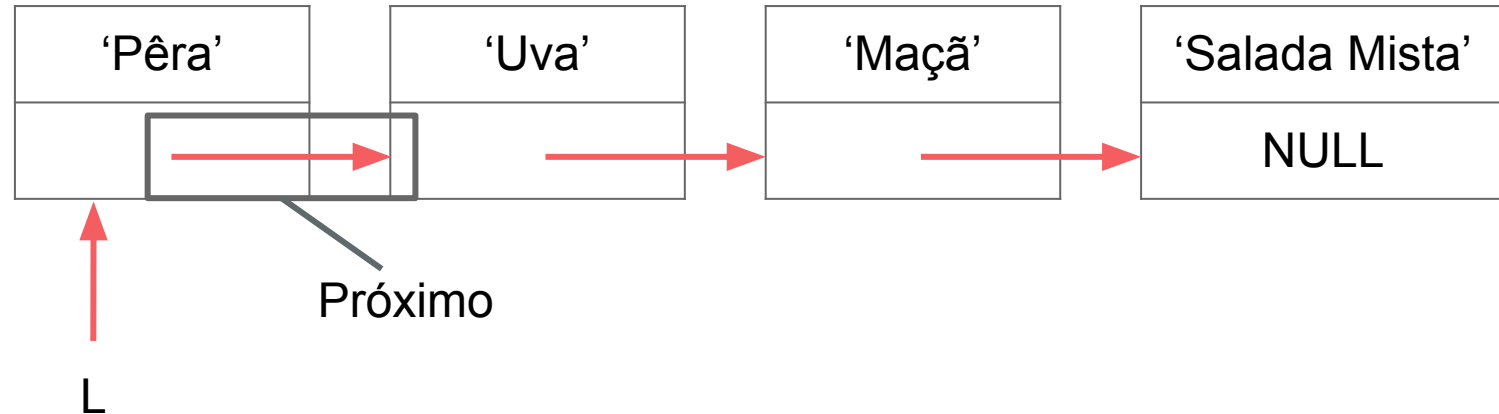
- Há outras representações para o fim da lista



TAD Listas com listas ligadas

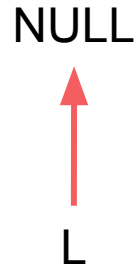
Vamos começar pela representação da lista

- Cada elemento deve apontar para o próximo



TAD Listas com listas ligadas

Lista vazia: apontando para nulo



Então, como se inicia uma lista vazia?

TAD Listas com listas ligadas

Na implementação com arranjo

- Posição é um valor inteiro entre 0 e “Ultimo”

Na implementação dinâmica

- Uma posição é definida por um ponteiro, que aponta cada nó/elemento da lista

TAD Listas

Vamos definir os tipos?

- tipo lista
- tipo item (nó)
- tipo apontador
- tipo chave
- etc

PINK PANTHER'S TO DO LIST:

-TO DO

-TO DO

-TO DO, TO DO, TO DO,
TO DO, TO DOOOO...



TAD Listas – Criar lista

Pré-condição:

- Nenhuma

Pós-condição

- Estrutura de dados é iniciada

TAD Listas – Criar lista

Pré-condição:

- Nenhuma

Pós-condição

- Estrutura de dados é iniciada

Antes

$L \rightarrow ?$

Depois

$L \rightarrow \text{NULL}$

TAD Listas – Inserir (primeira posição)

Pré-condição:

- Existe **memória disponível**

Pós-condição

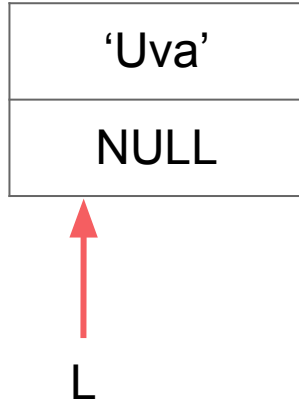
- Elemento é inserido na primeira posição

TAD Listas – Inserir (primeira posição)

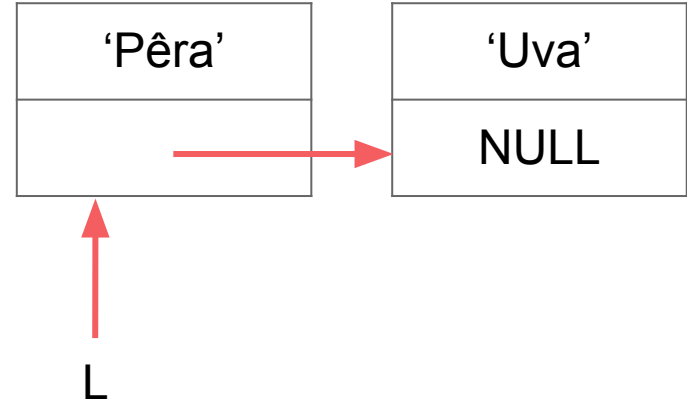
- Não requer alocação prévia, com tamanho especificado
- Entretanto, **memória *heap* não é ilimitada**
 - Verificar se há memória ao usar ***malloc***
 - Retorno da função *malloc* é **NULL** quando não há memória suficiente para alocar

TAD Listas – Inserir (primeira posição)

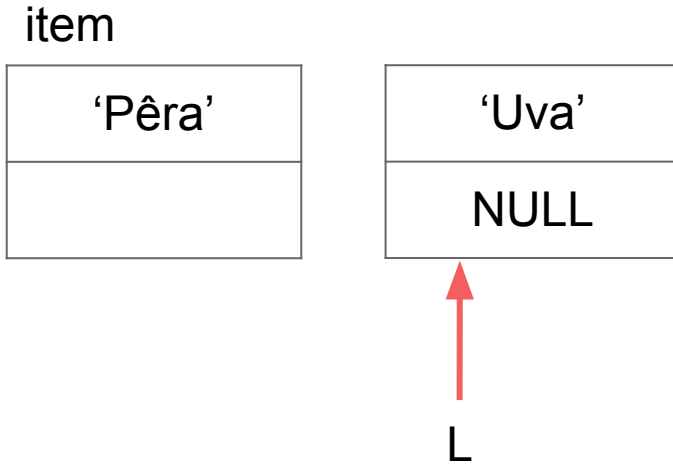
Antes



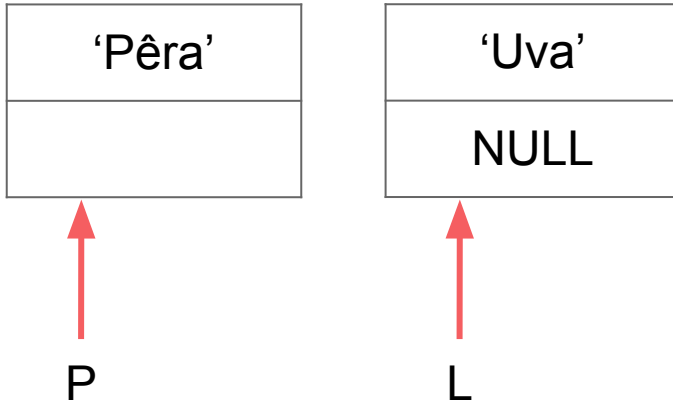
Depois



TAD Listas – Inserir (primeira posição)

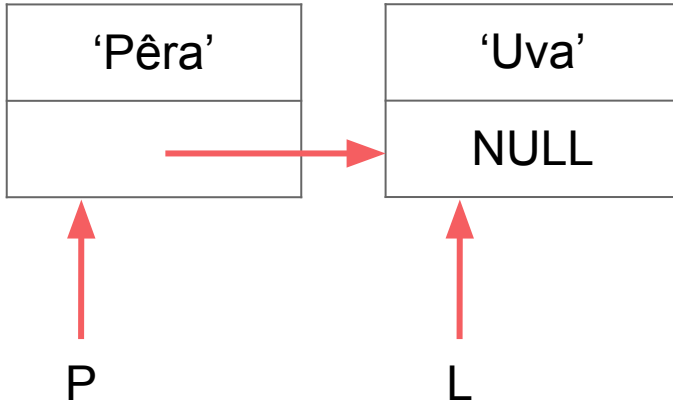


TAD Listas – Inserir (primeira posição)



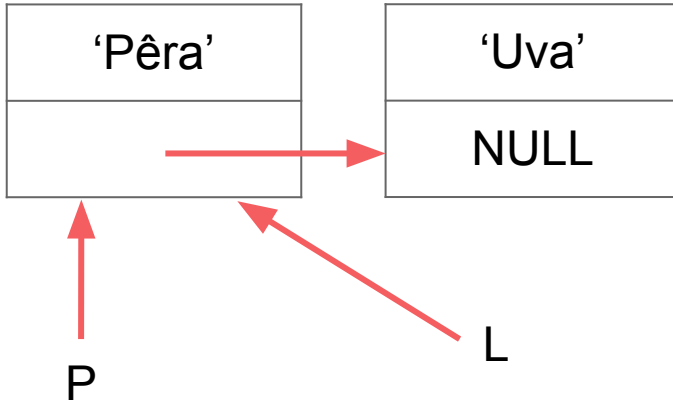
$P \rightarrow \text{item} = \text{item};$

TAD Listas - Inserir (primeira posição)



```
P->item = item;  
P->prox = *L;
```

TAD Listas - Inserir (primeira posição)



```
P->item = item;  
P->prox = *L;  
*L = P;
```

TAD Listas – Inserir (primeira posição)

- E se a lista estiver vazia? Funciona?

TAD Listas – Pesquisar elemento

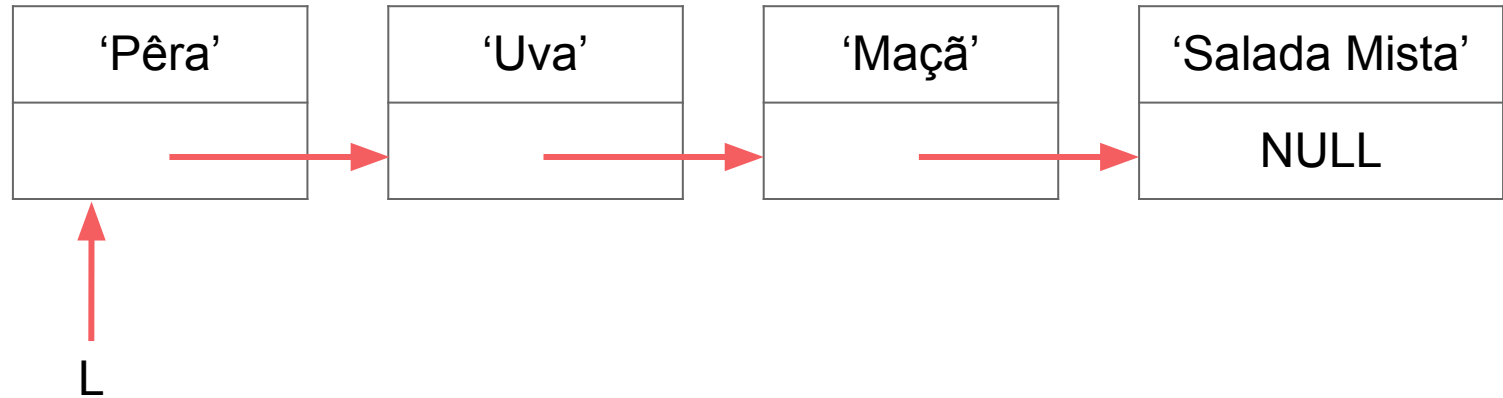
Pré-condição:

- Nenhuma

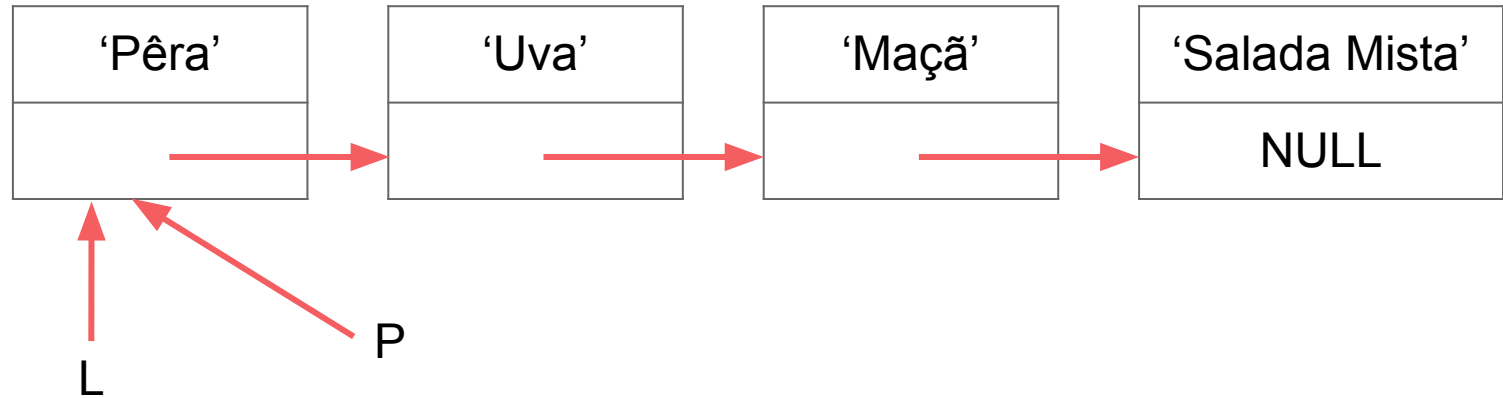
Pós-condição

- Retorna a posição do elemento na lista, ou um valor especial caso não exista

TAD Listas - Pesquisar elemento

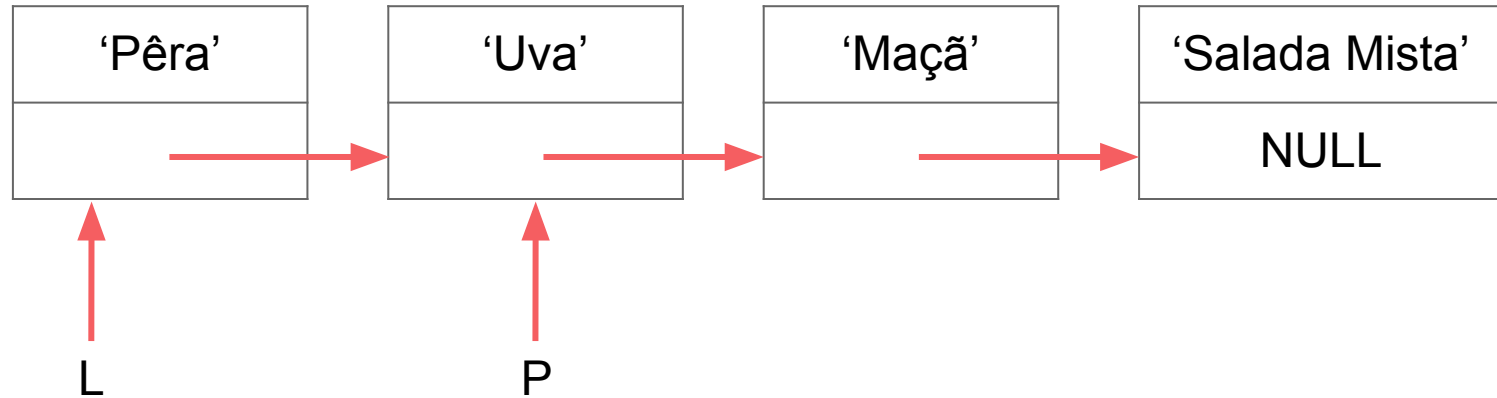


TAD Listas - Pesquisar elemento



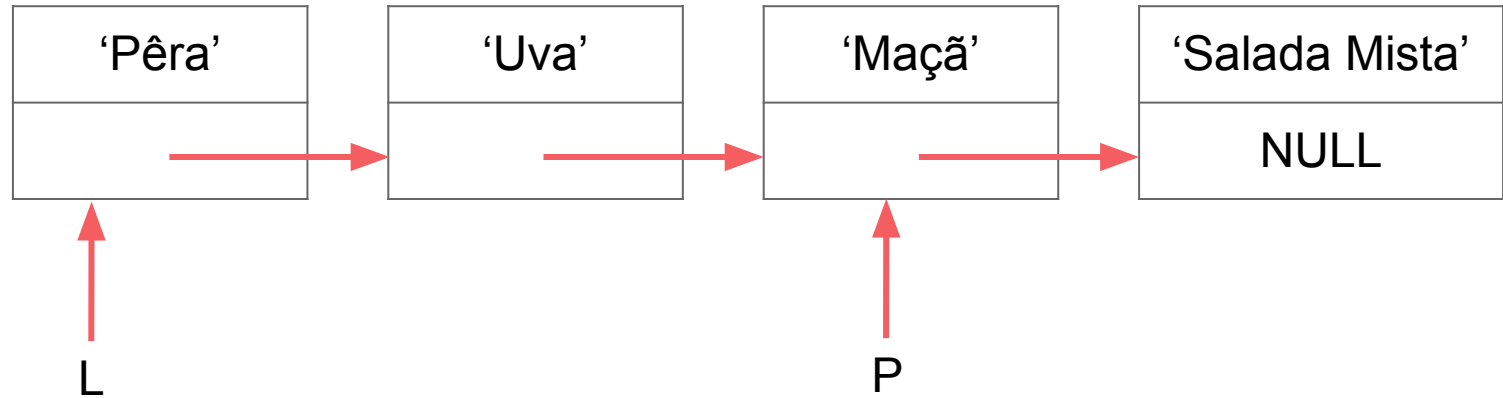
TAD Listas - Pesquisar elemento

$P = P \rightarrow \text{prox};$



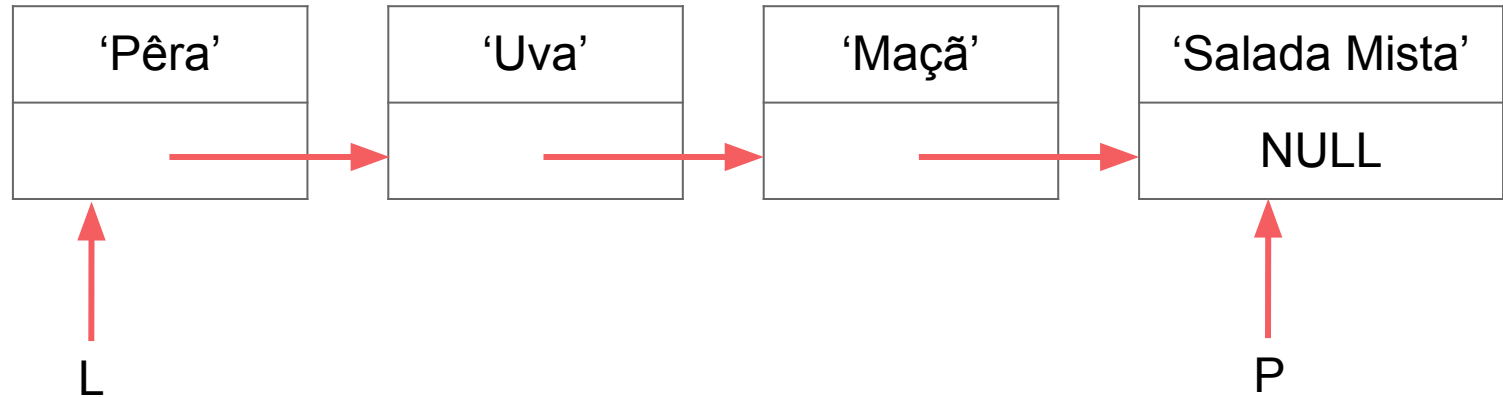
TAD Listas - Pesquisar elemento

$P = P \rightarrow \text{prox};$



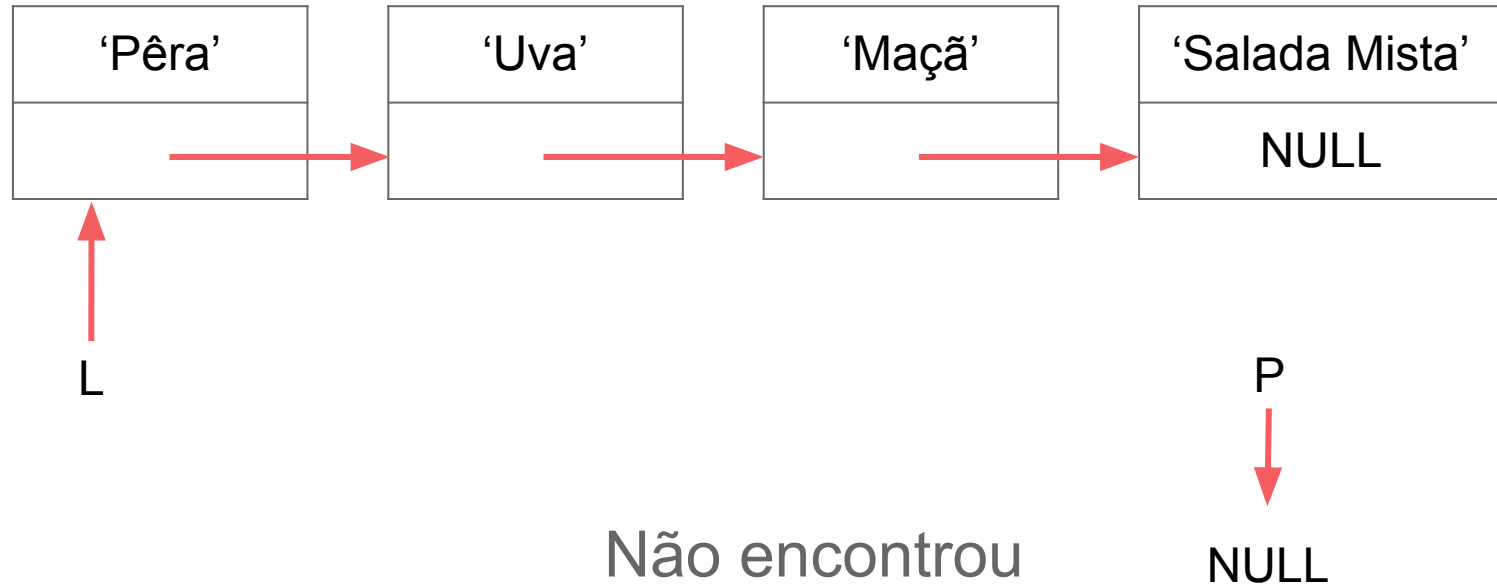
TAD Listas - Pesquisar elemento

$P = P \rightarrow \text{prox};$



TAD Listas - Pesquisar elemento

$P = P \rightarrow \text{prox};$



TAD Listas – Pesquisar elemento

O que muda se a lista for ordenada?

TAD Listas – Remover elemento por posição

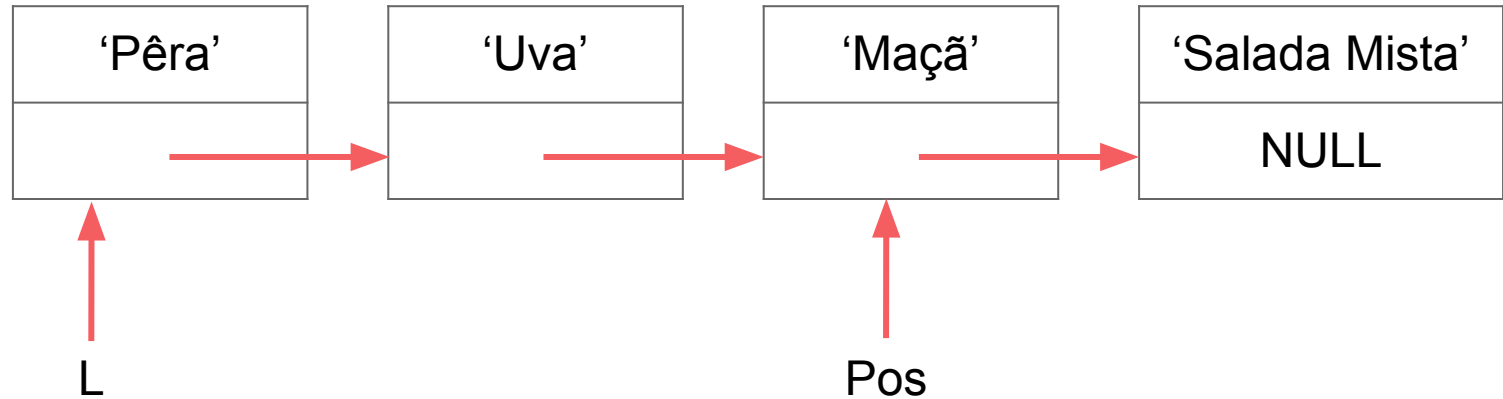
Pré-condição:

- Posição válida é fornecida

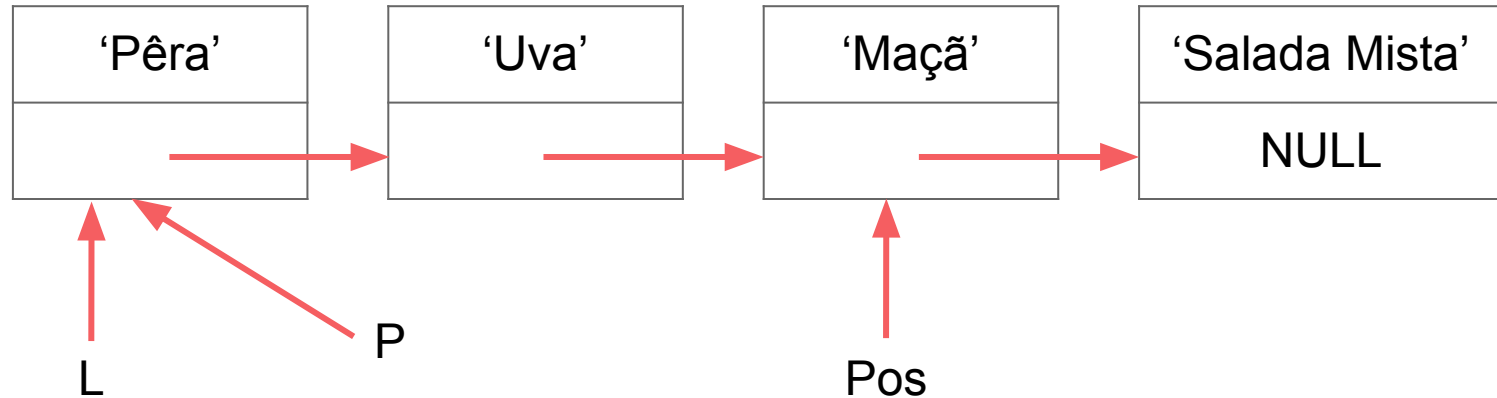
Pós-condição

- O elemento daquela posição é removido da lista

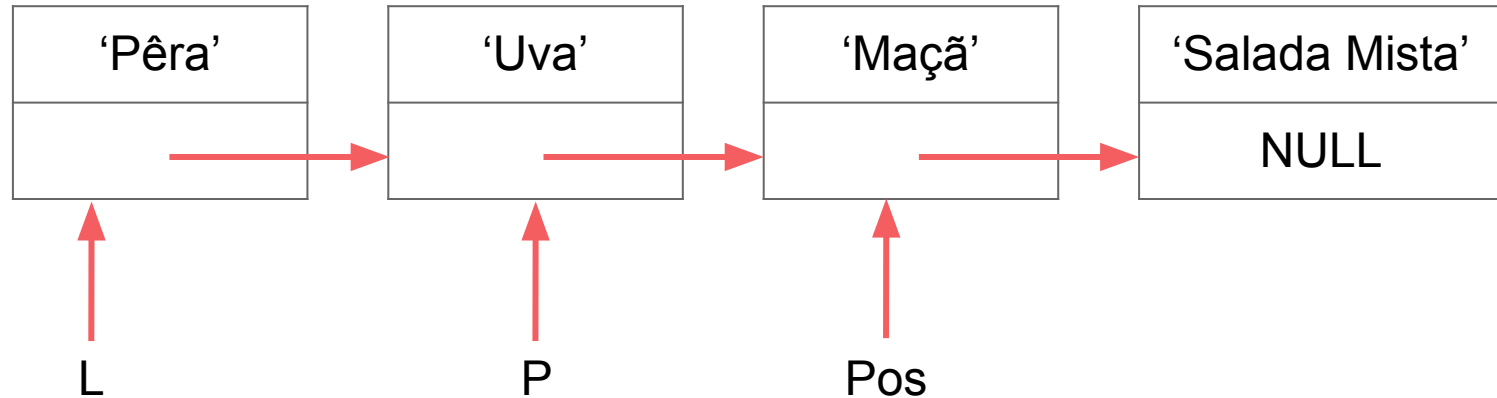
TAD Listas - Remover elemento por posição



TAD Listas - Remover elemento por posição

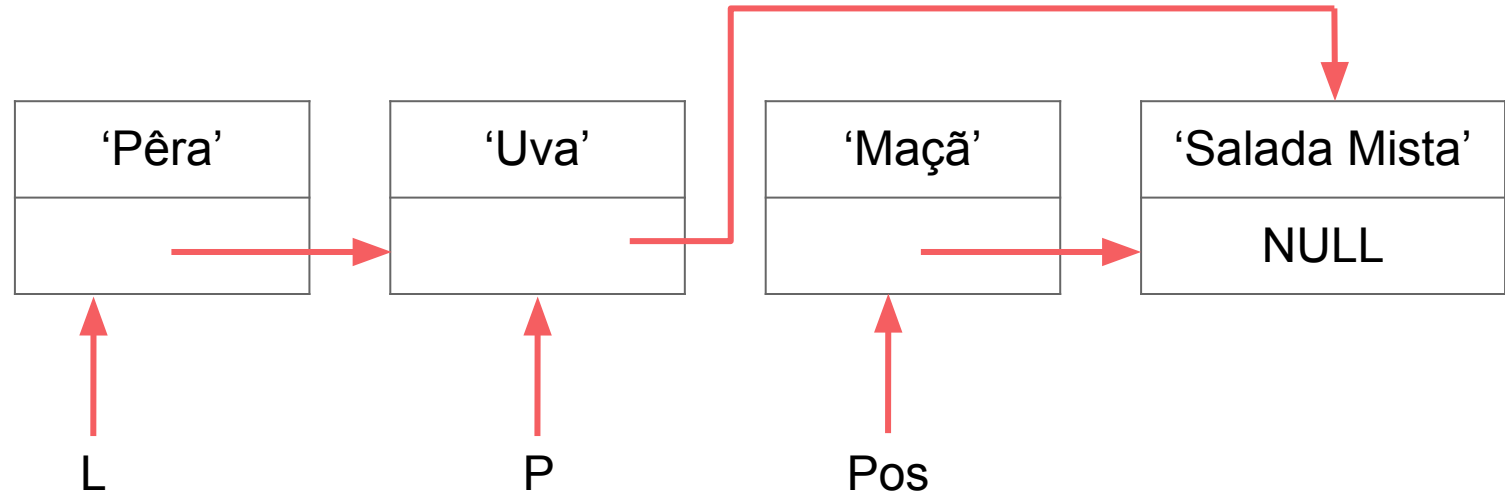


TAD Listas - Remover elemento por posição



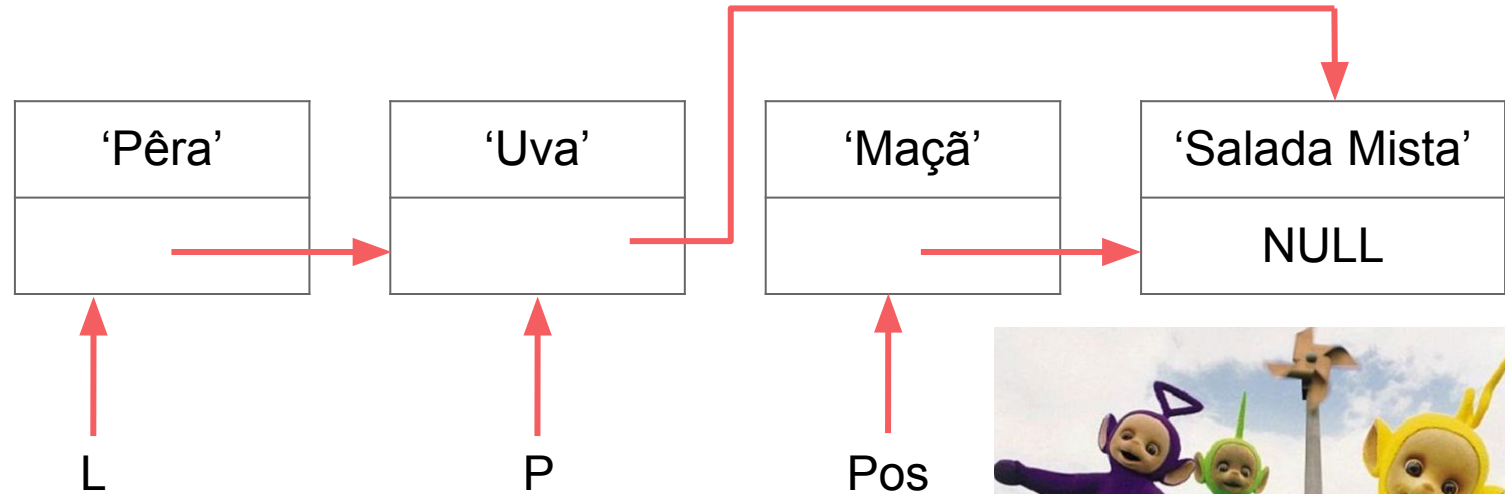
$P \rightarrow \text{prox} == \text{Pos}$? Então **pare**

TAD Listas - Remover elemento por posição

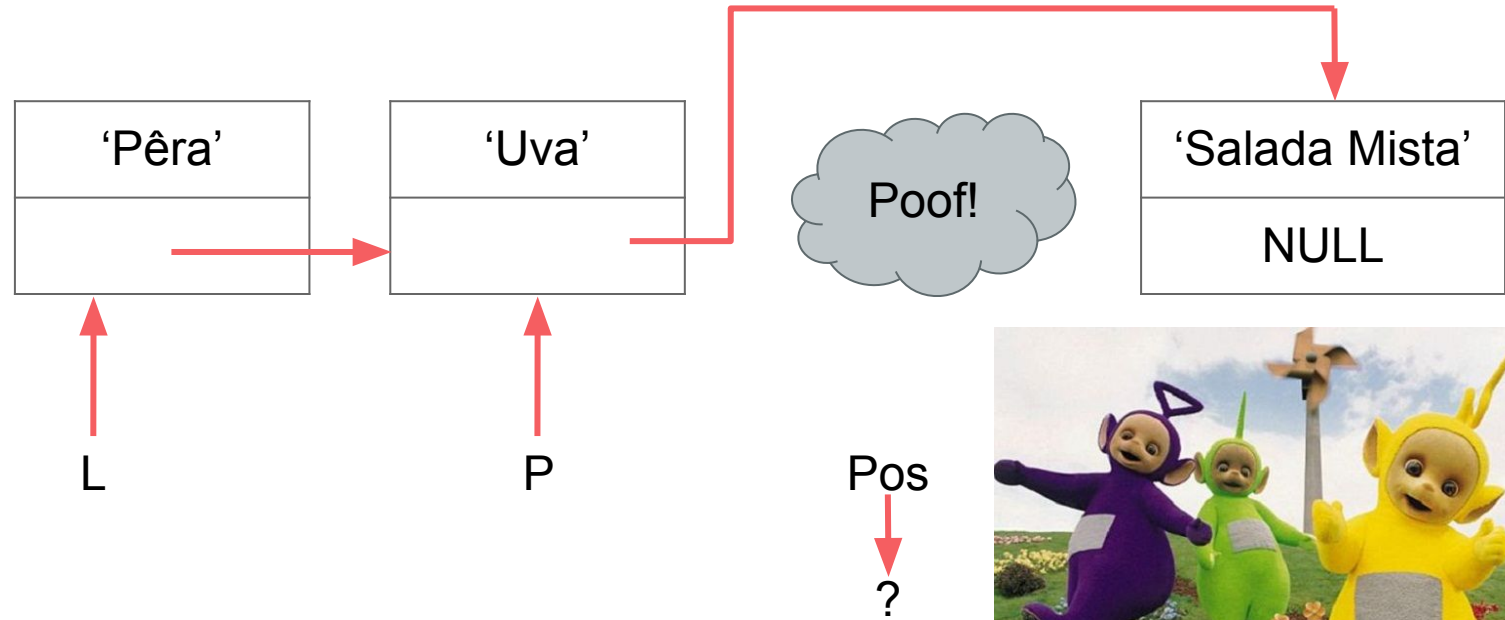


$P \rightarrow \text{prox} = \text{Pos} \rightarrow \text{prox}$

TAD Listas - Remover elemento por posição



TAD Listas - Remover elemento por posição



TAD Listas – Remover elemento por posição

Casos especiais

- Fim da lista
- Início da lista
- Lista com único elemento

Vamos fazer no quadro

Anotem, tirem foto, memorizem (cof cof), etc

TAD Listas – Exercícios

Pensar nas pré- e pós-condições e implementar outras funções básicas da lista:

- Contar número de elementos
- Retornar um elemento (por posição ou por chave)
- Verificar se a lista está vazia
- Verificar se a lista está cheia

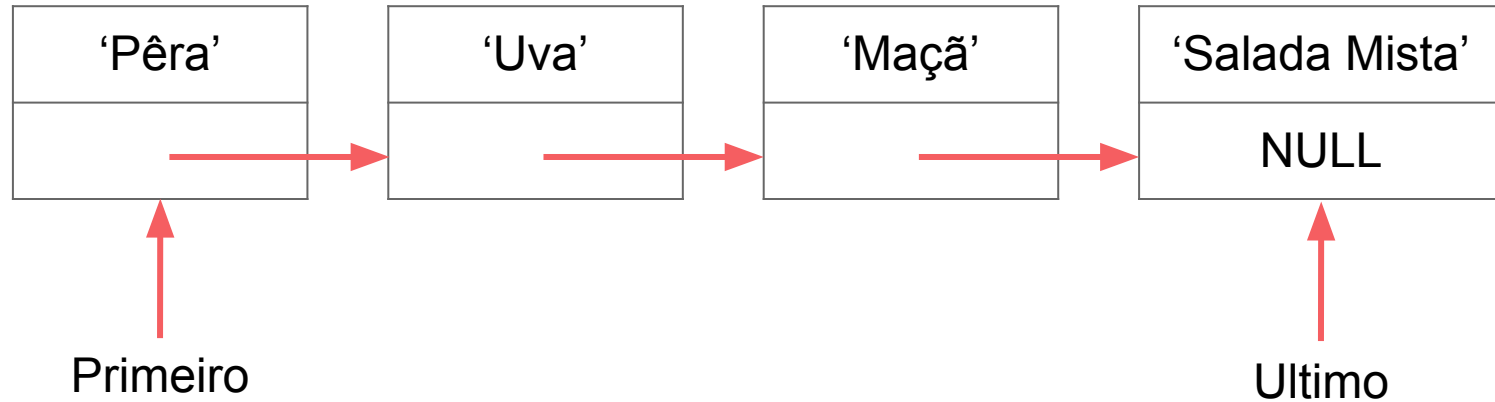
p.s. esperar pela próxima aula pode ajudar na implementação

TAD Listas – Inserir – Alternativas

Inserir na última posição

- Inserir sempre na primeira posição deixa a lista invertida
- Percorrer a lista toda vez para inserir no final é custoso
- Podemos mudar levemente nossa estrutura

TAD Listas - Inserir - Alternativas

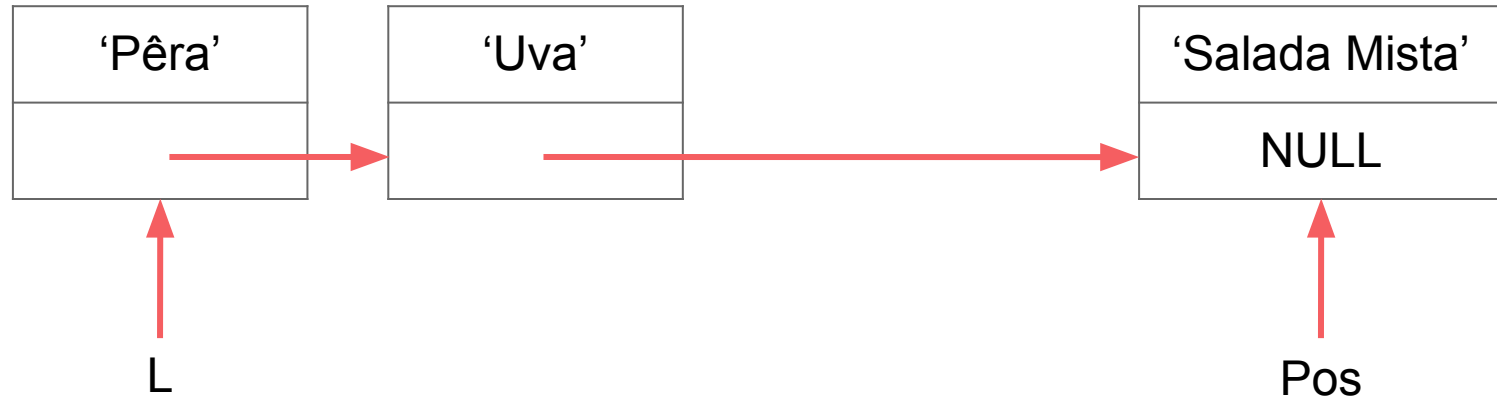


TAD Listas – Inserir – Alternativas

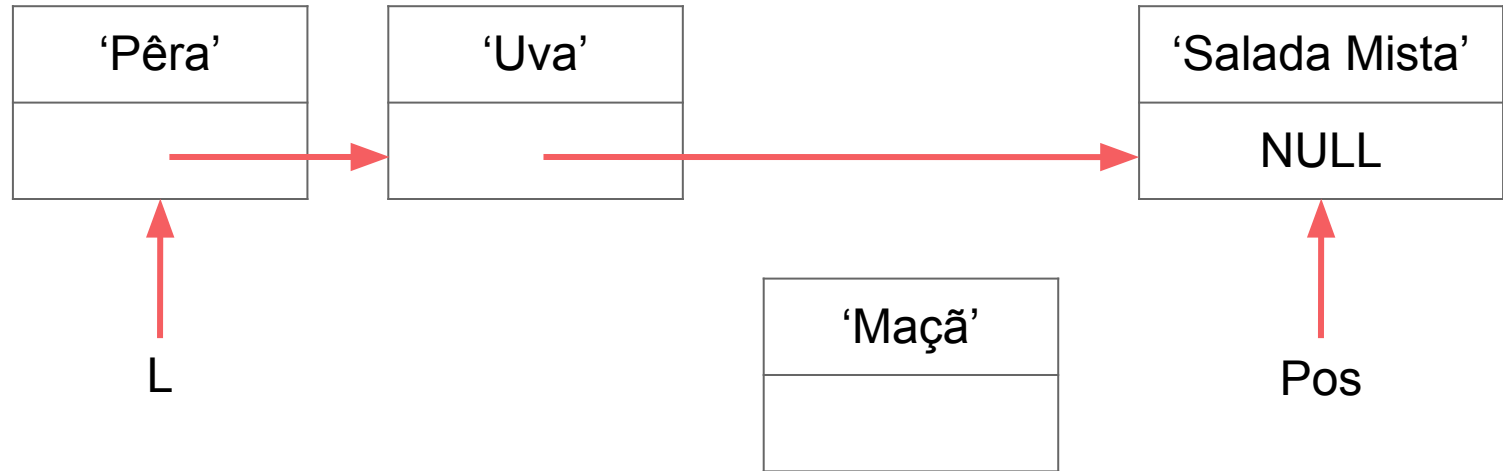
Inserir em qualquer posição

- A partir de uma posição válida
- Precisamos “mexer uns ponteiros”

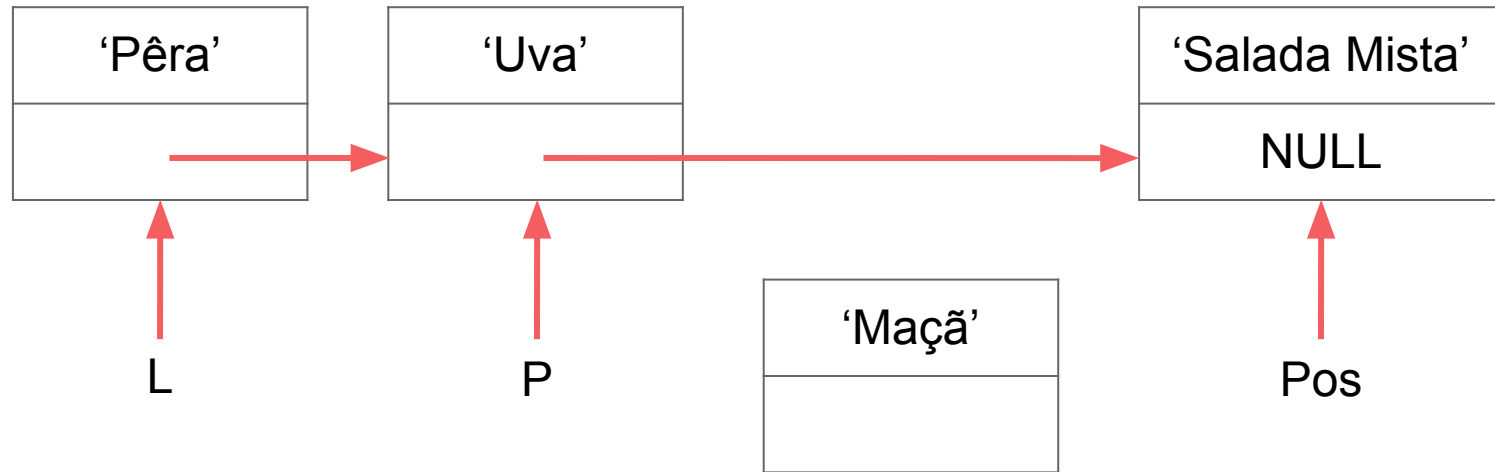
TAD Listas - Inserir - Alternativas



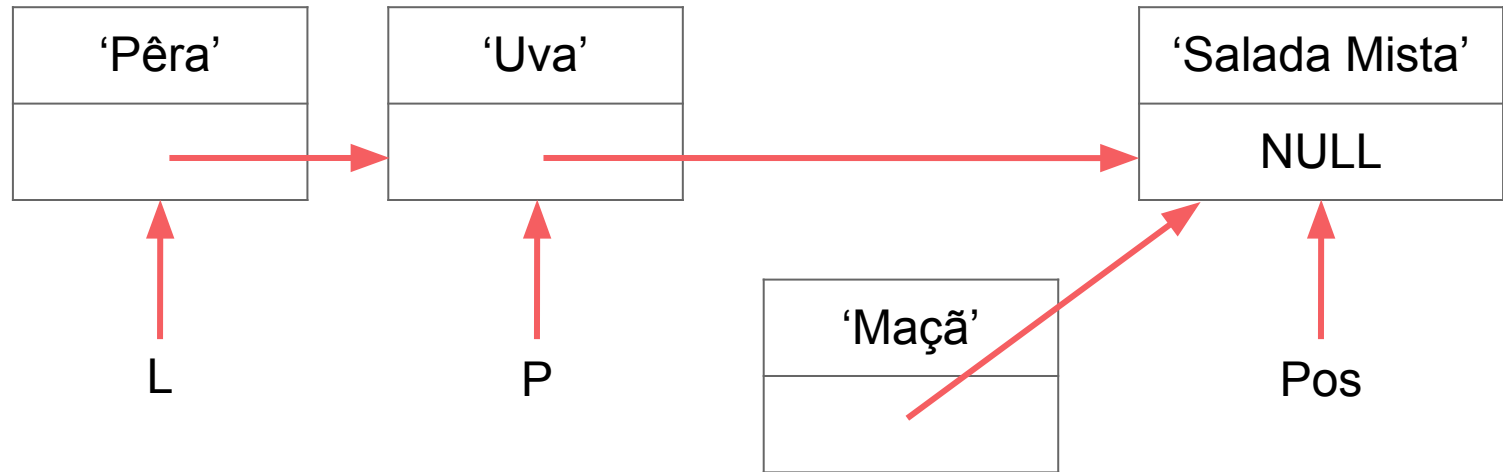
TAD Listas - Inserir - Alternativas



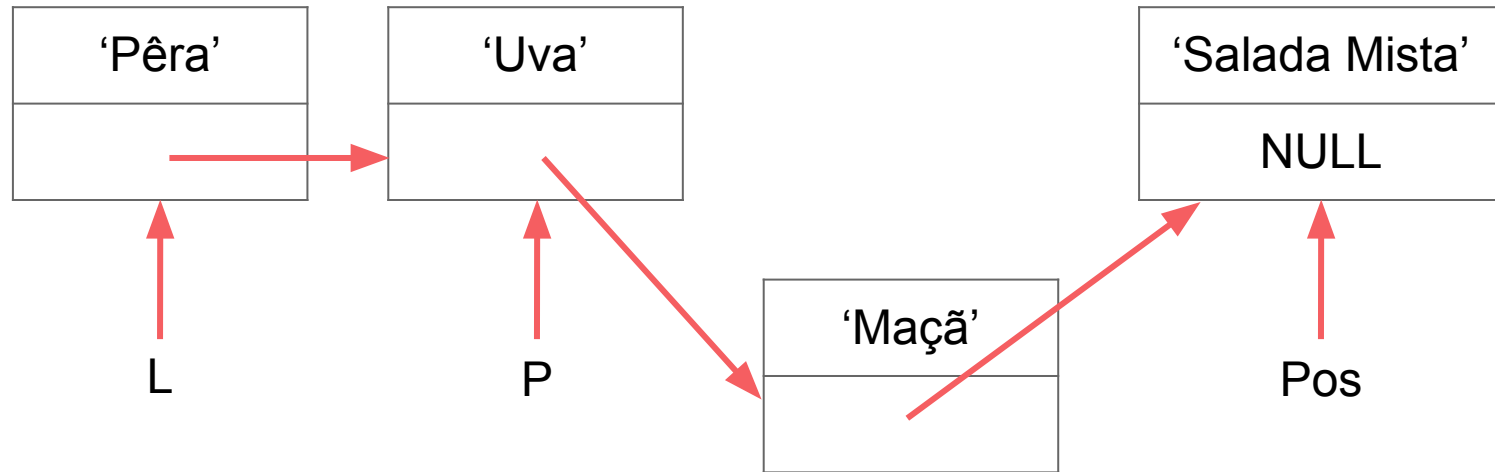
TAD Listas - Inserir - Alternativas



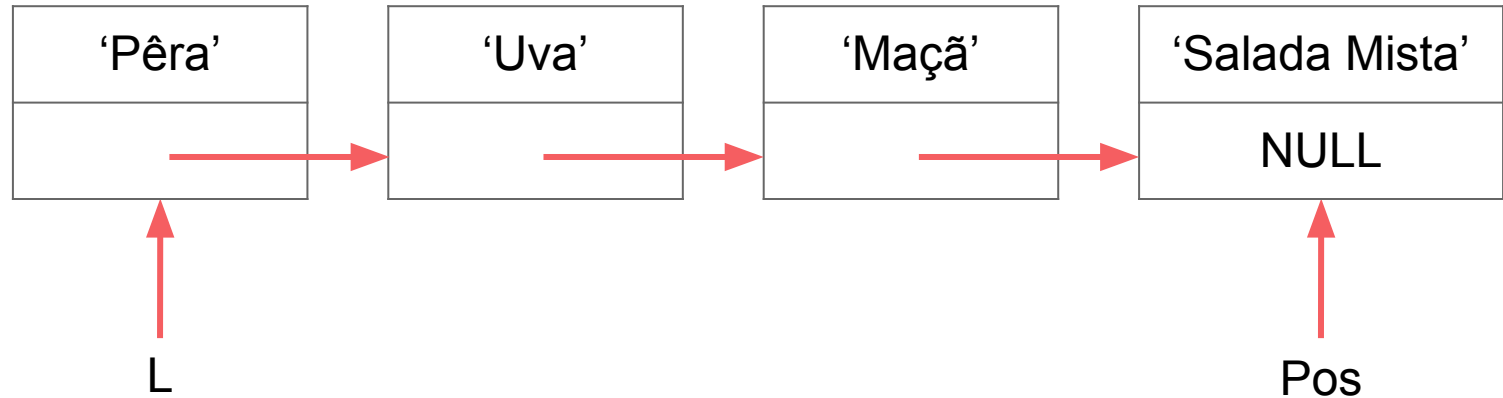
TAD Listas - Inserir - Alternativas



TAD Listas - Inserir - Alternativas



TAD Listas - Inserir - Alternativas



TAD Listas – + Exercícios

Implementar a estrutura com ponteiros de primeiro e último e a inserção na última posição

p.s. esperar pela próxima aula pode ajudar na implementação

