

CMP1054

Estrutura de Dados I

Java

Listas Lineares autoreferenciadas

Prof. Dr. José Olimpio Ferreira

Implementação de Listas por meio de Estruturas Auto-Referenciadas

- Cada **item** da lista contém a informação que é necessária (endereço) para alcançar o próximo **item**.
- Permite utilizar posições não contíguas de memória.
- É possível inserir e retirar **itens** sem necessidade de deslocar os **itens** seguintes da lista.
- A **lista** é constituída de **nós (células)**.
- Cada **nó** contém um **item** da lista e uma **referência** para o **nó seguinte**.
- A classe *Lista* contém uma referência para o **primeiro nó** da lista.

Conjunto de operações

- **Lista ()**: Cria uma lista vazia.
- **inserirInicio(x)**: Insere o item x antes do primeiro nó da lista.
- **retira(x)**: Retorna o item x que da lista, com uma dada chave, retirando-o da lista.
- **vazia()**: Esta função retorna true se lista vazia; senão retorna false.
- **getLista()**: Imprime os itens da lista na ordem de ocorrência.
- **pesquisa(x)**: Verifica se um item x (com uma certa chave) está na lista e se positivo retorna uma cópia do item, caso contrário retorna nullptr.

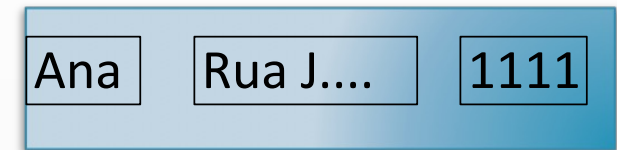
Objetos do tipo **Item**

- **Atributos:**

- **string nome**
- **string endereco**
- **string cpf**

- **Métodos:**

- **construtor**
- **setNome, setEndereco, setCpf**
- **getNome, getEndereco, getCpf**



A diagram representing an object of type **Item**. It consists of a light blue rounded rectangle containing three smaller white boxes with black borders. The first box contains the text "Ana", the second box contains "Rua J....", and the third box contains "1111".

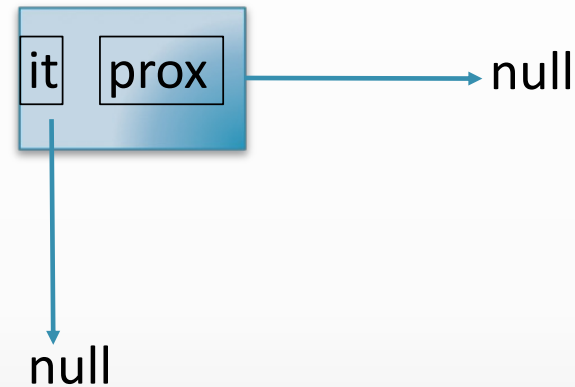
Objetos do tipo **No**

- **Atributos:**

- Item it
- No prox

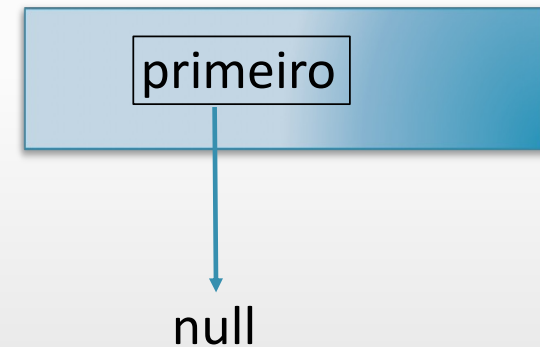
- **Métodos**

- construtor
- setItem e setProx
- getItem e getProx

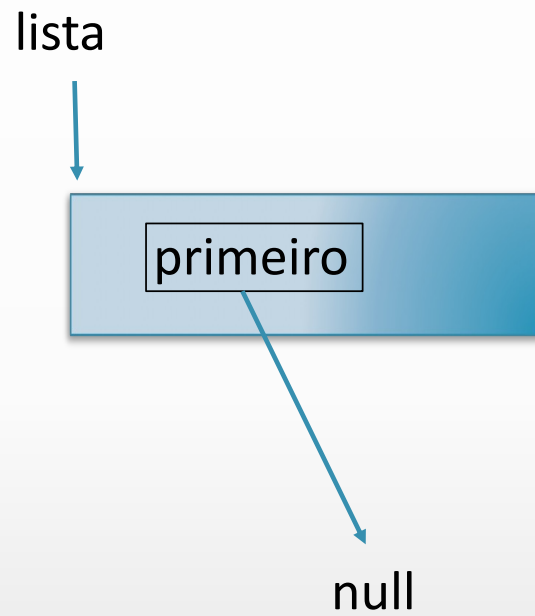


Objetos do tipo **Lista**

- **Atributos:**
 - **No primeiro**
- **Métodos:**
 - **construtor**
 - **inserelnicio**
 - **retira**
 - **pesquisa**
 - **Vazia**

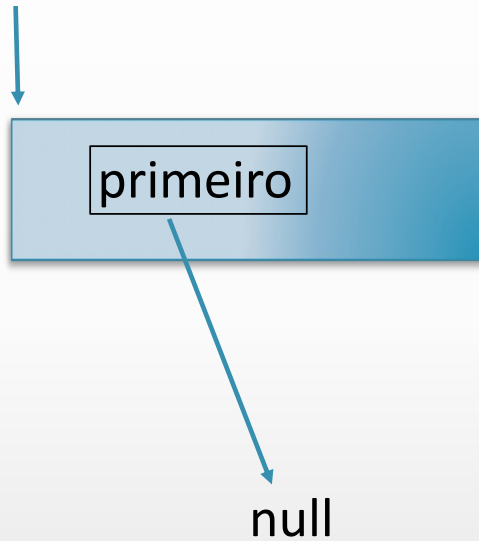


lista vazia

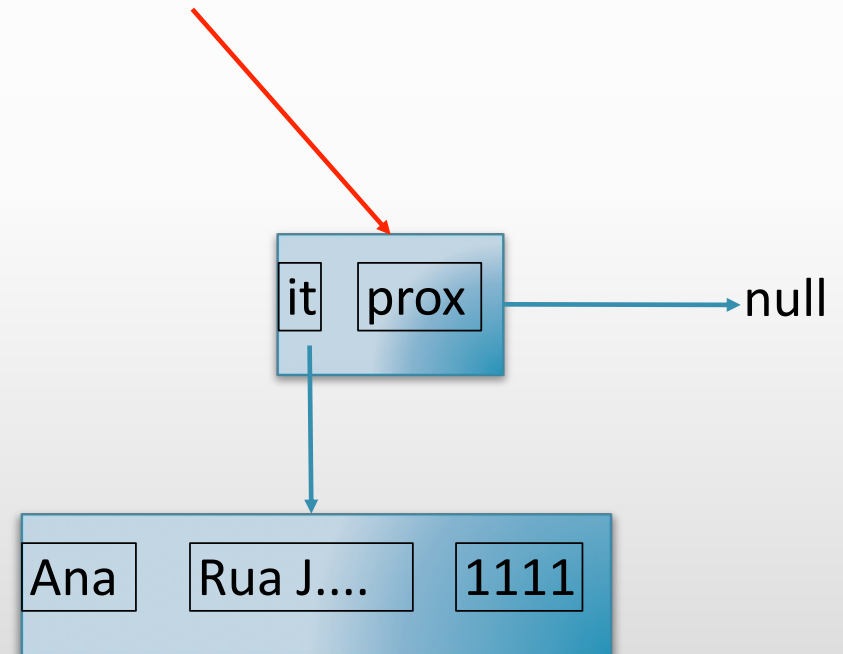


inserelnicio (Item x)

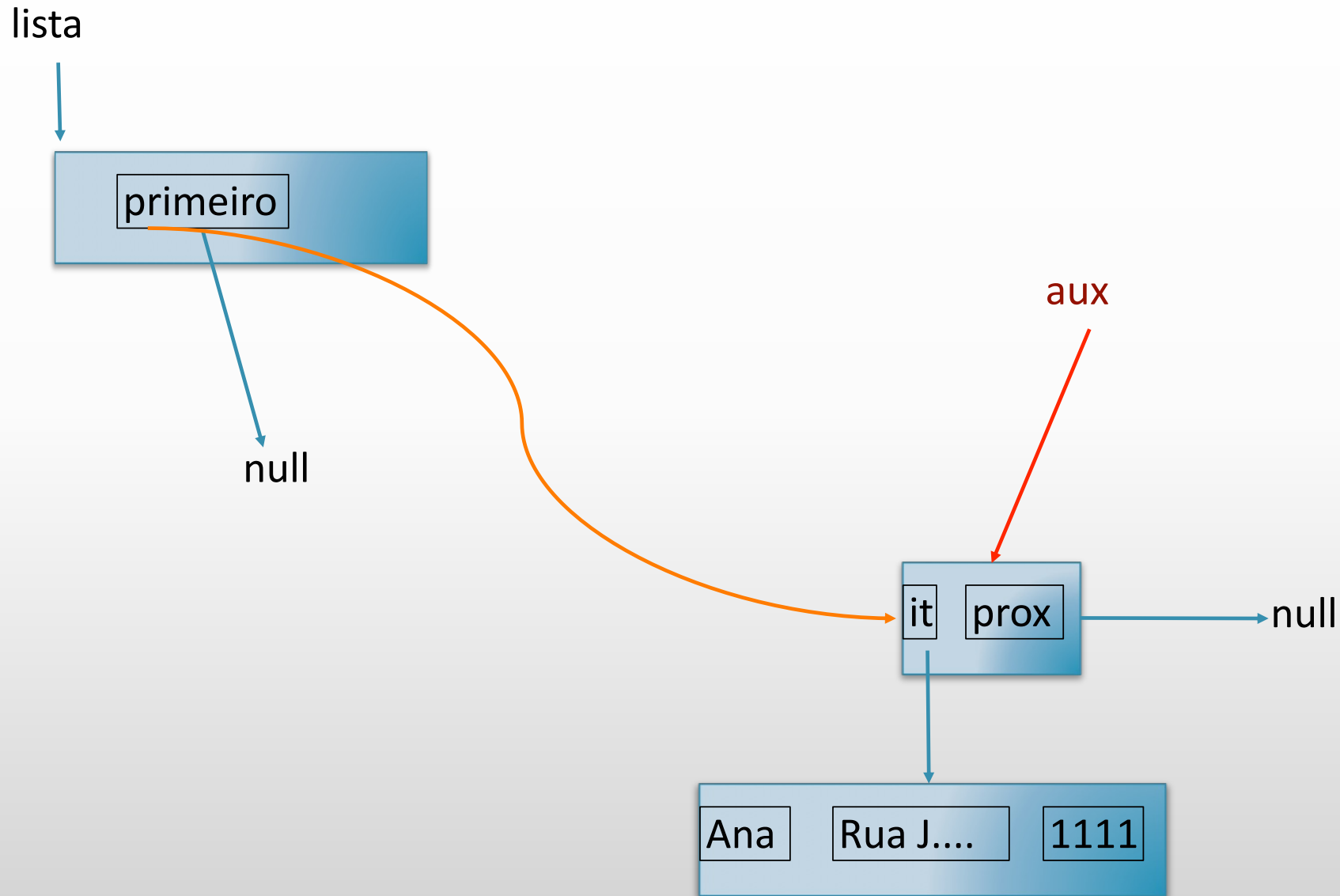
lista



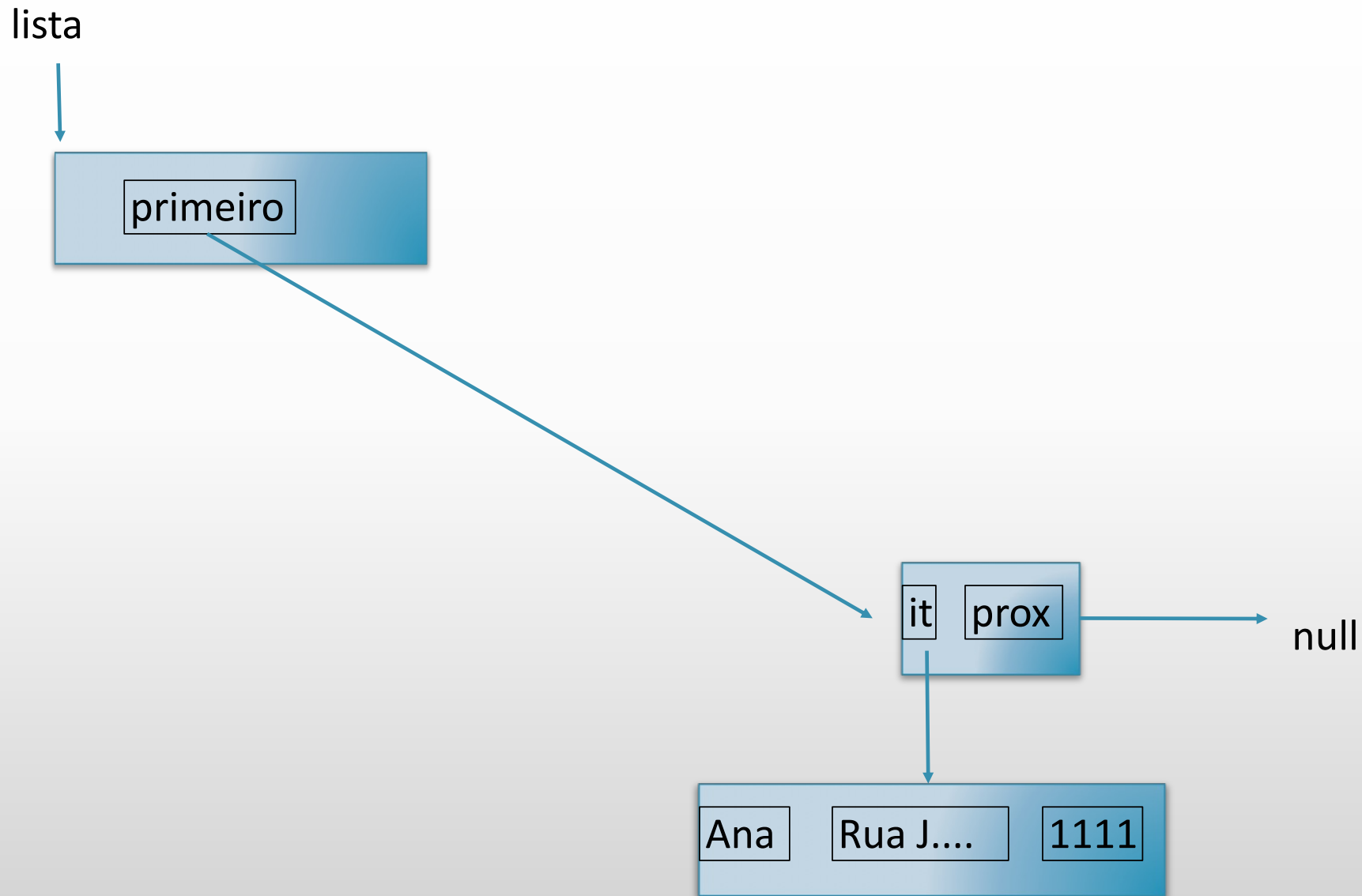
aux



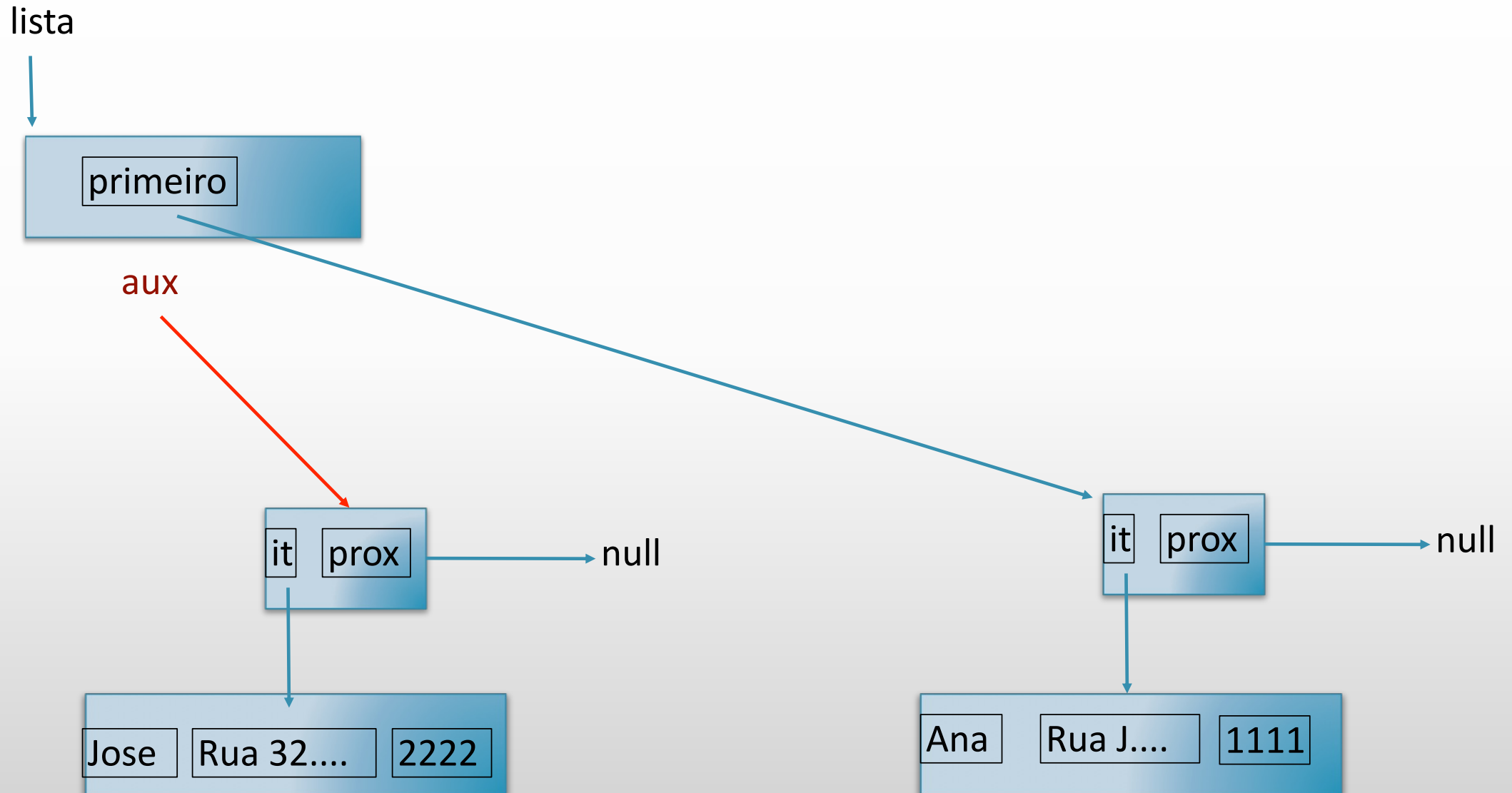
inserelnicio (Item x)



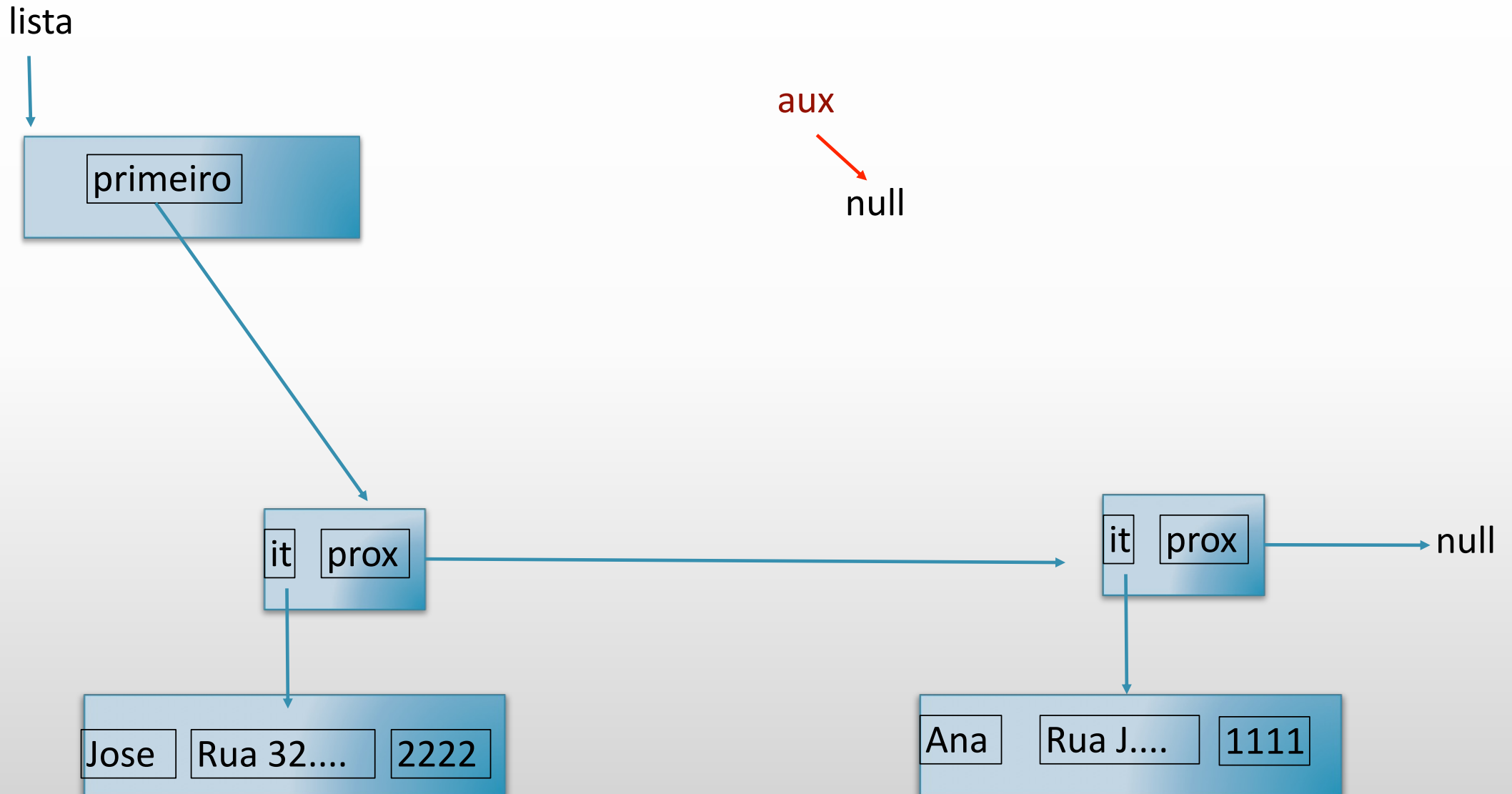
Lista com 1 item



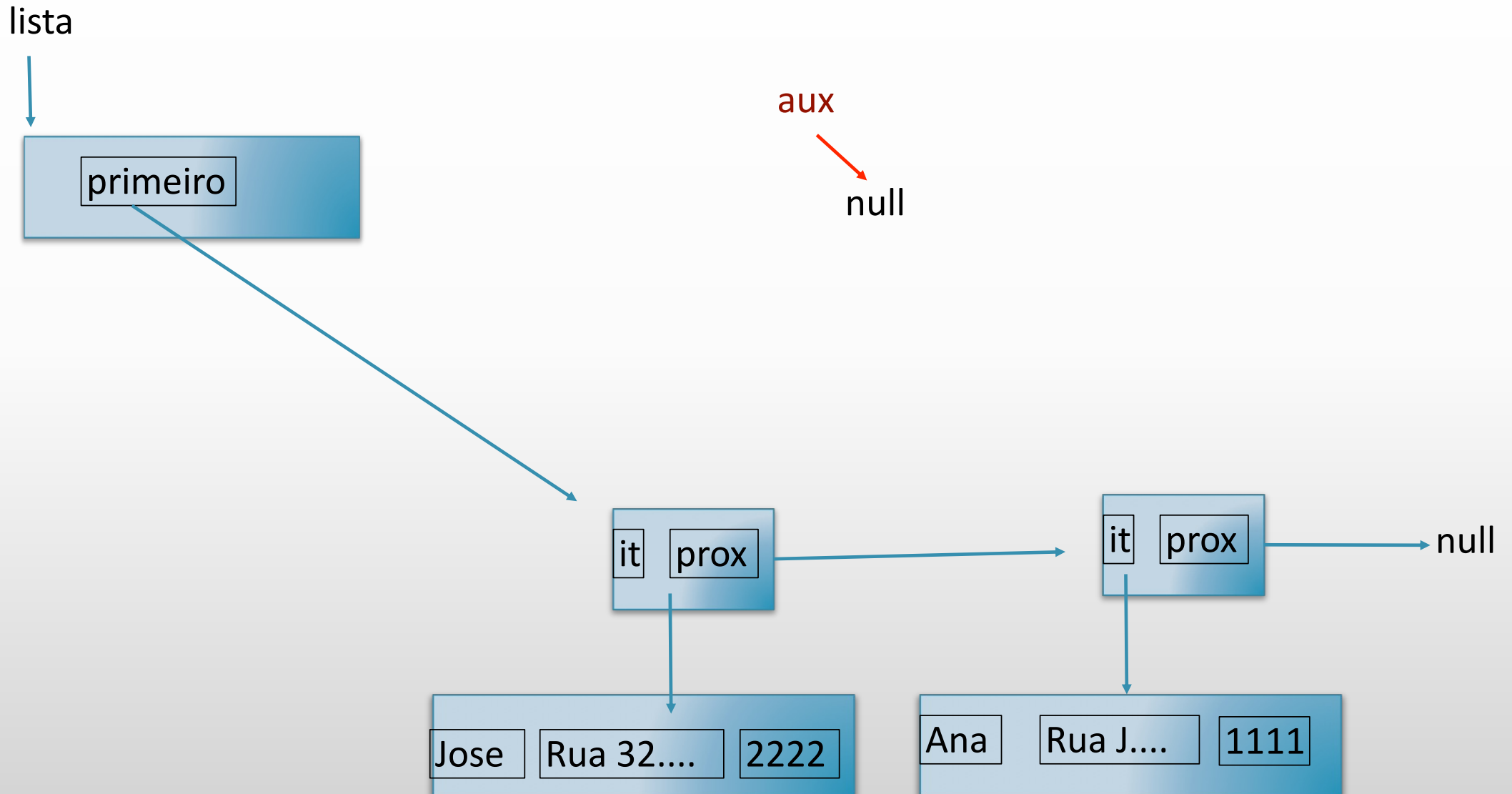
inserelnicio (Item x)



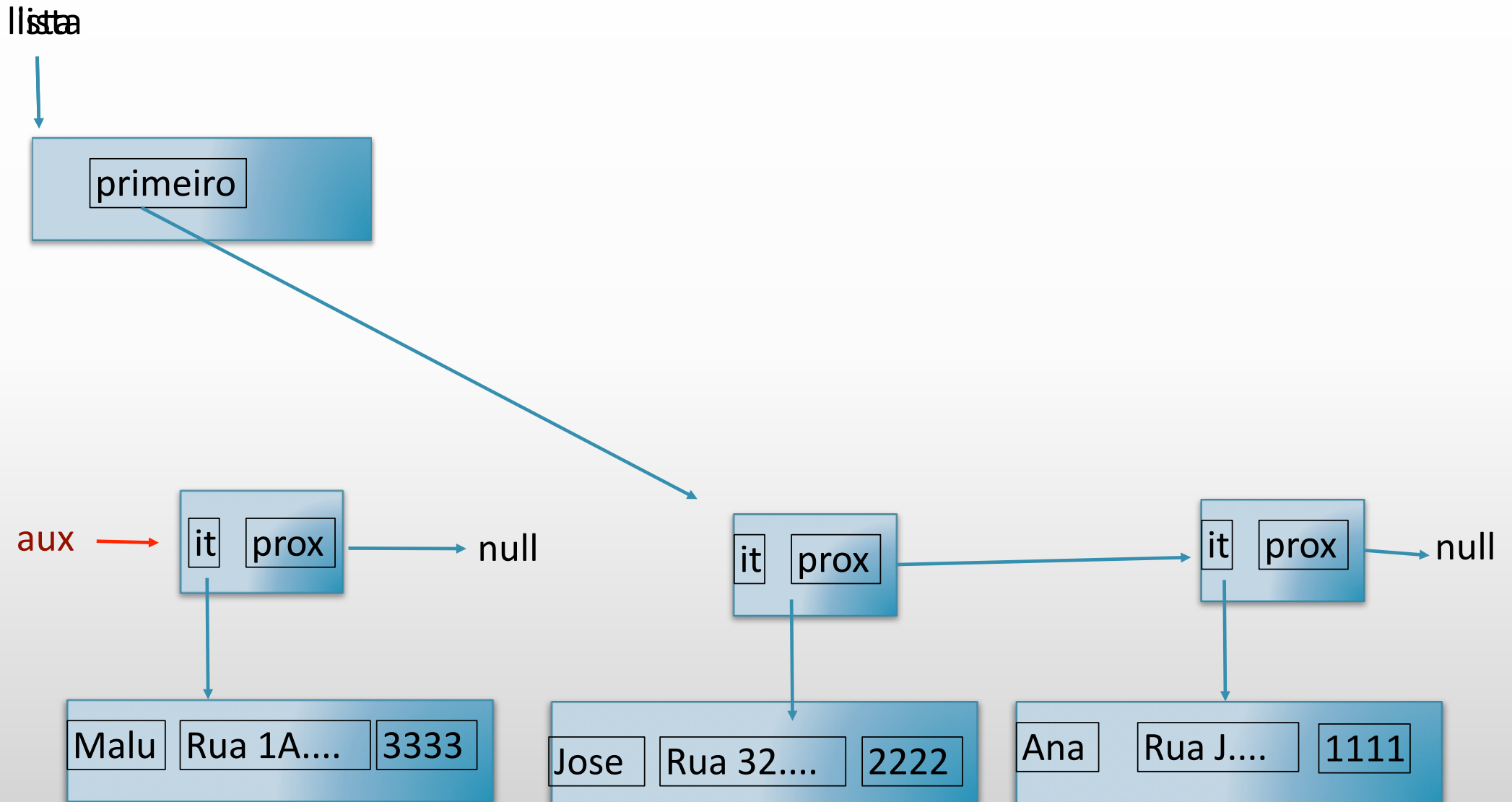
inserelnicio (Item x)



Após 2 inserções.

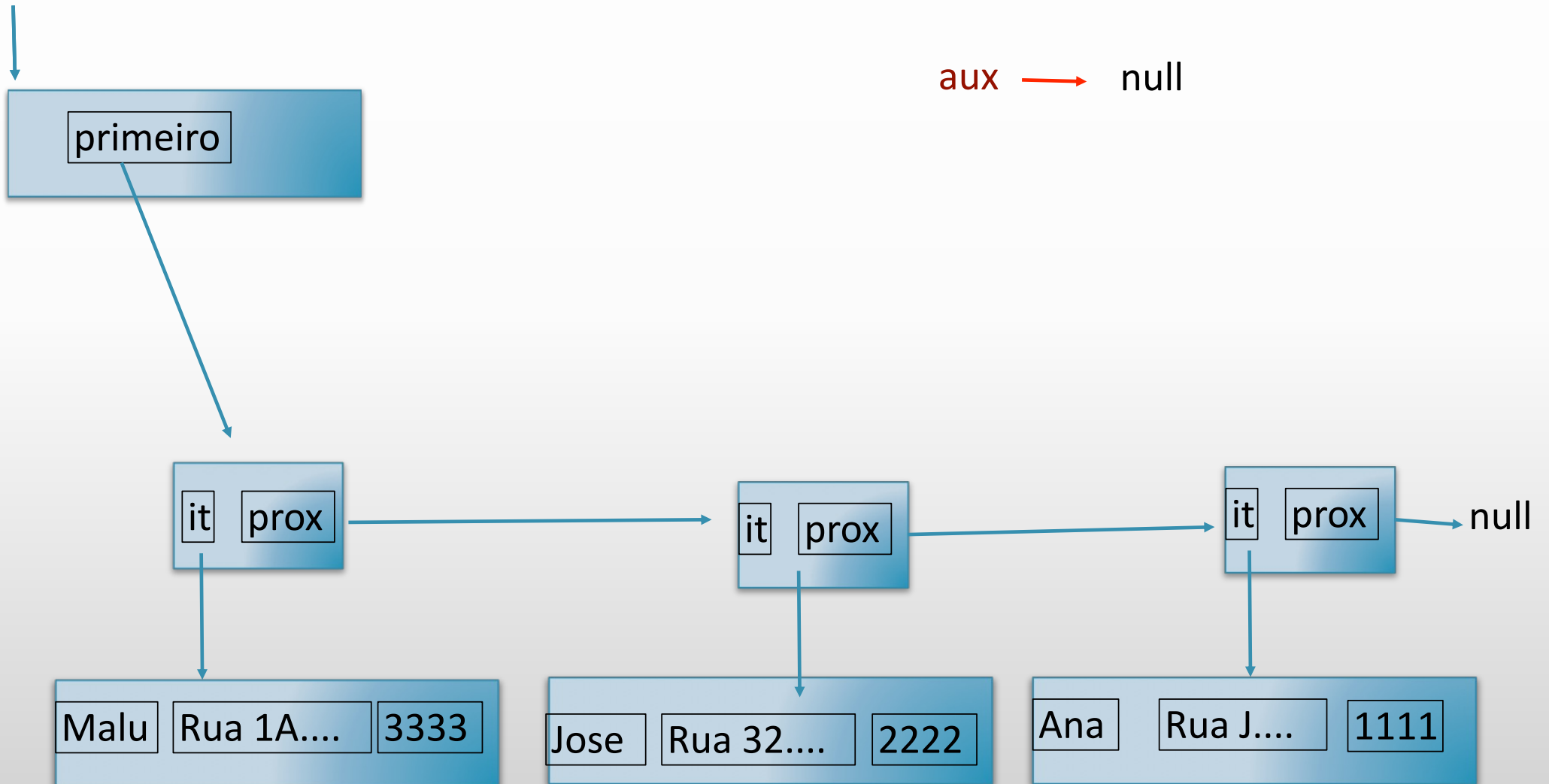


insereInicio (Item x)

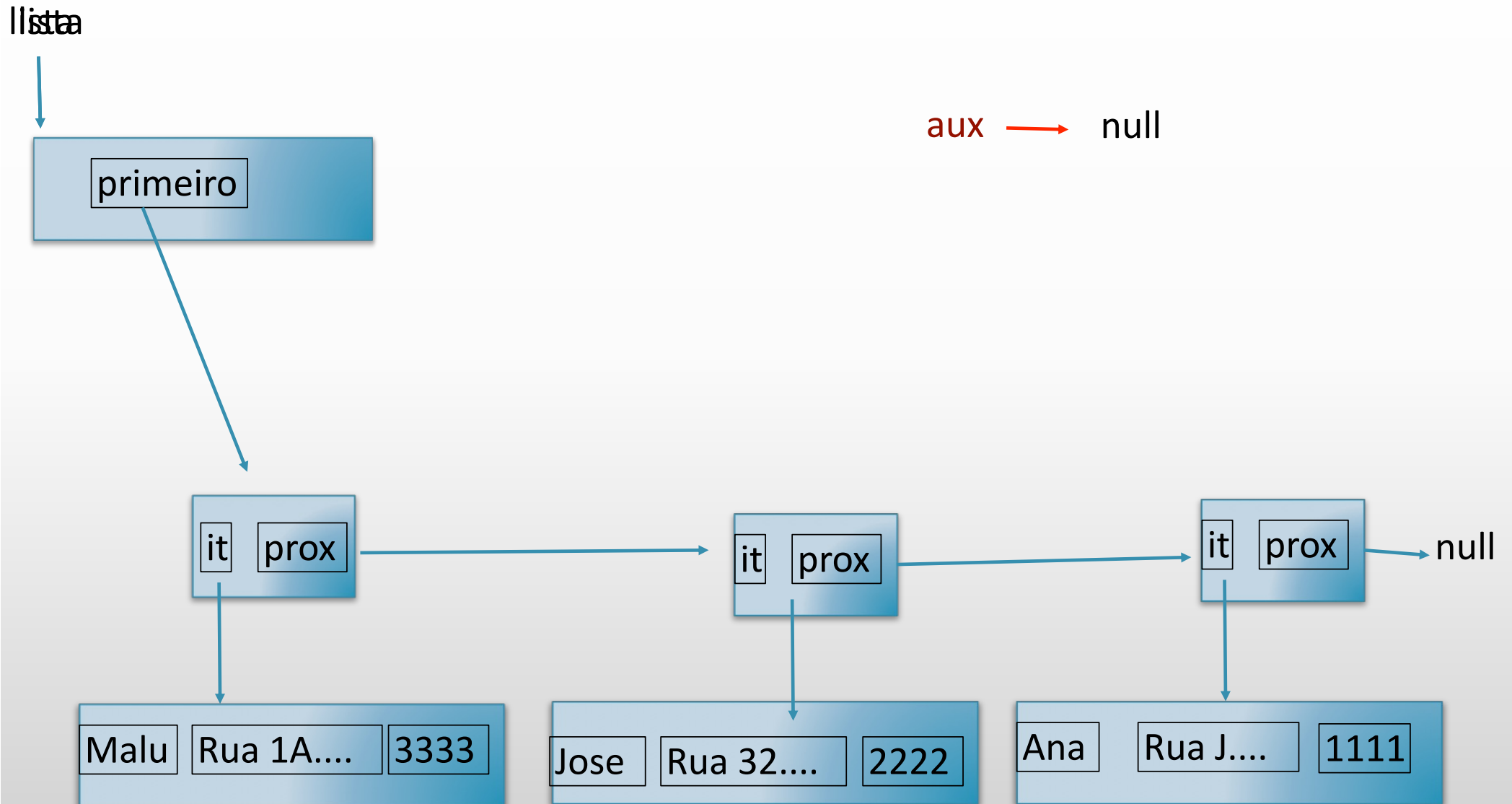


inserirInicio (Item x)

lista

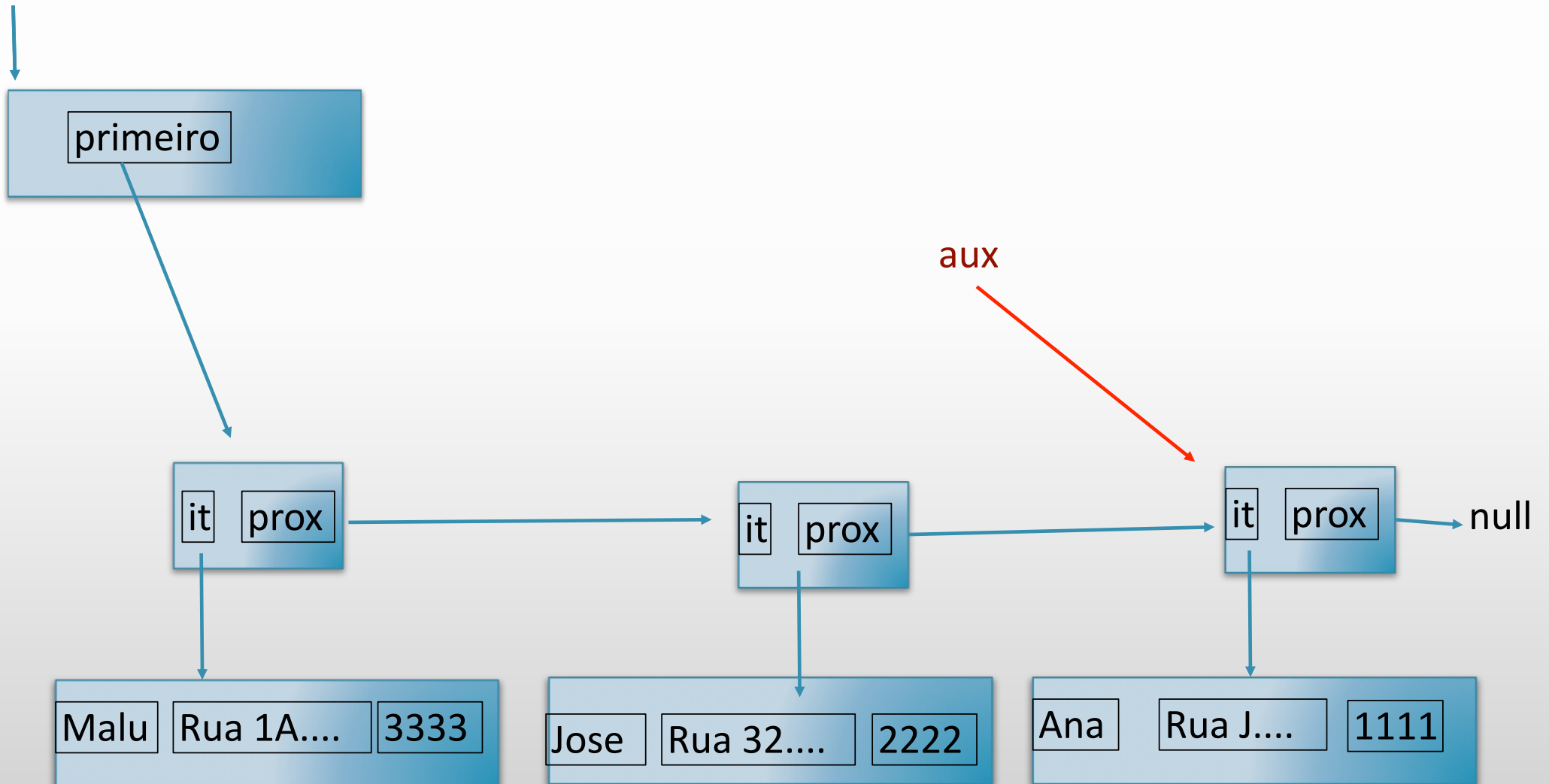


Após 3 inserções



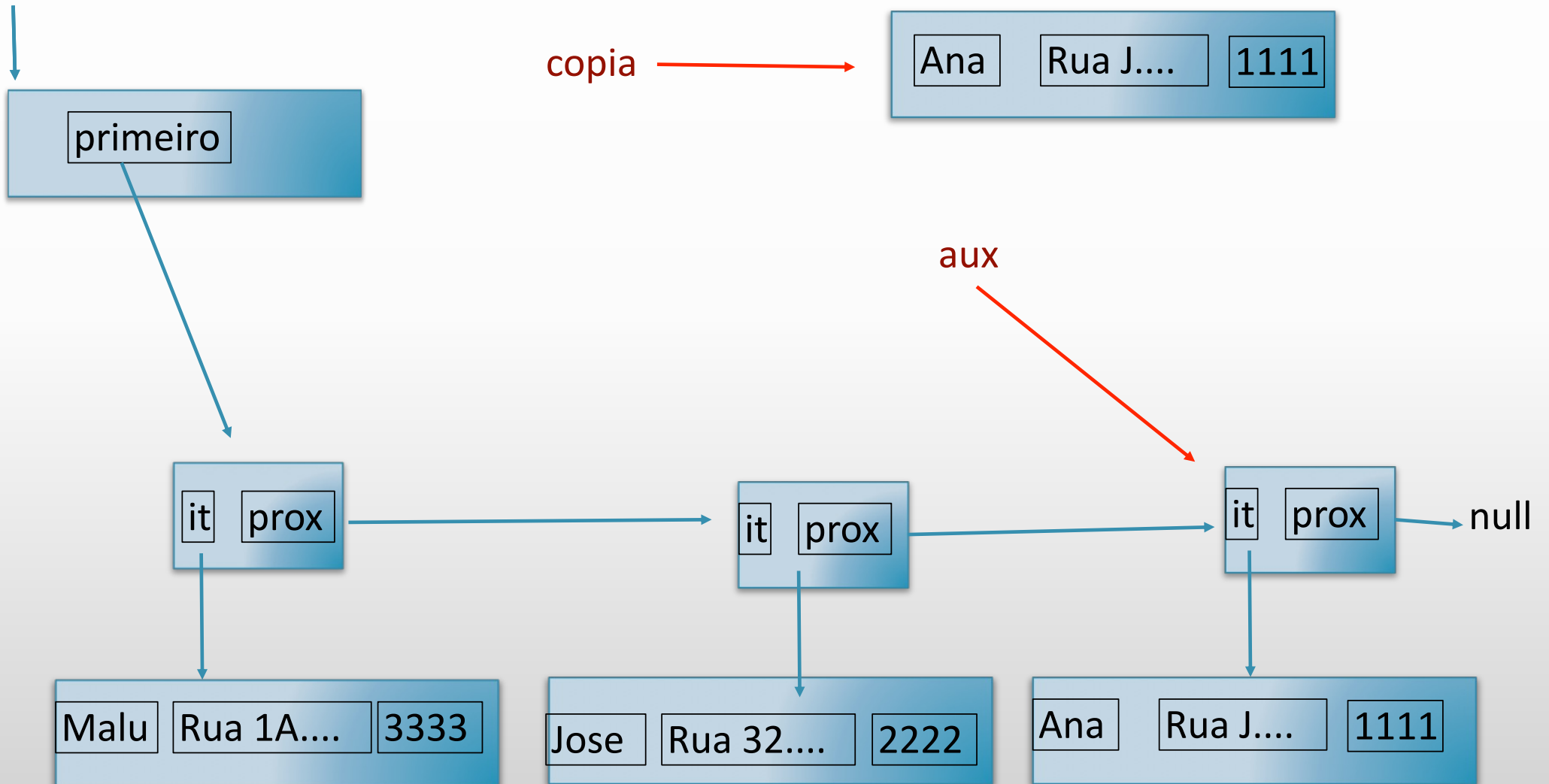
Item buscar(Item x)
chave -> 1111

lista



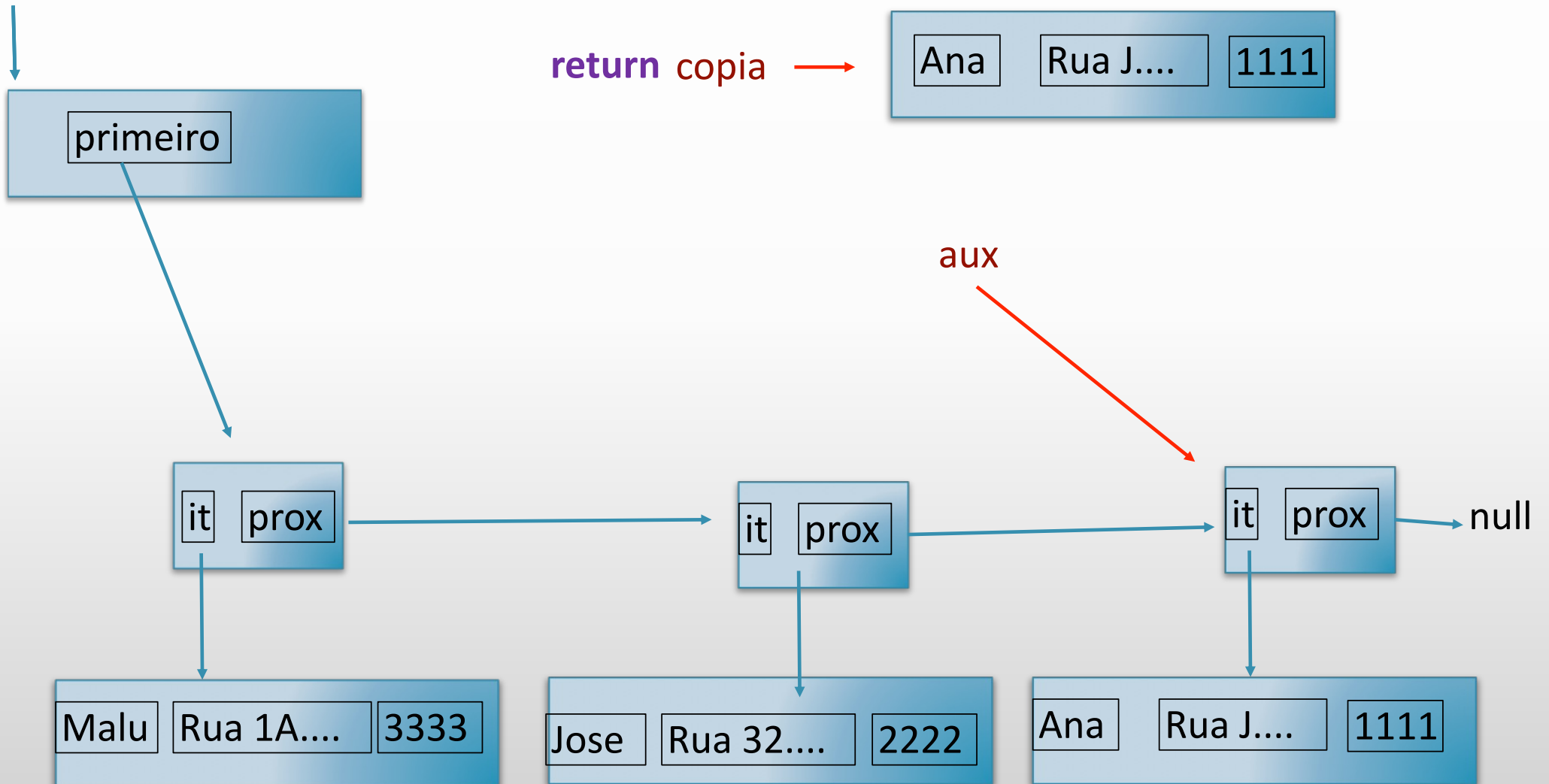
Item buscar(Item x)
chave -> 1111

lista

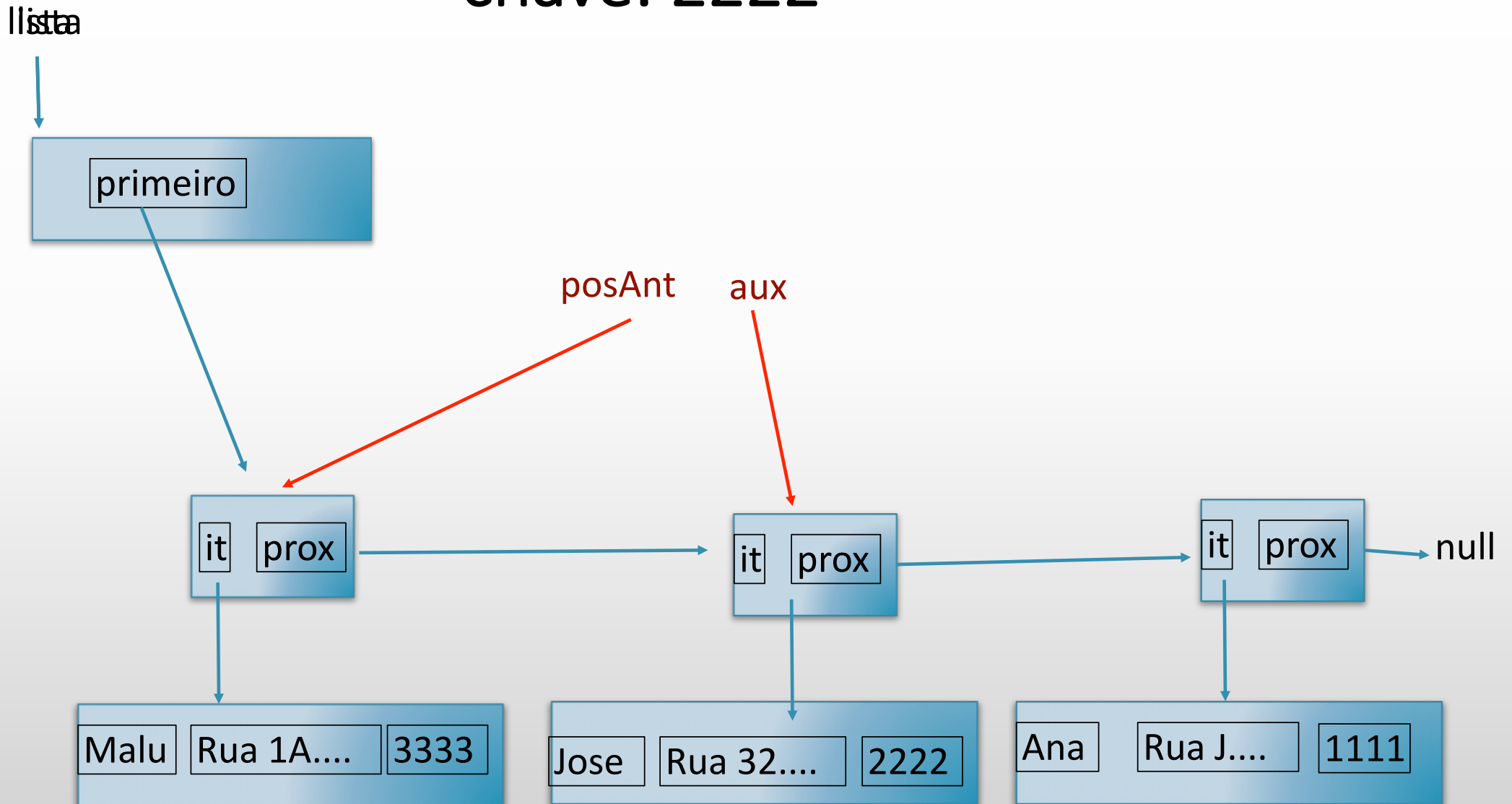


Item buscar(Item x)
chave -> 1111

lista

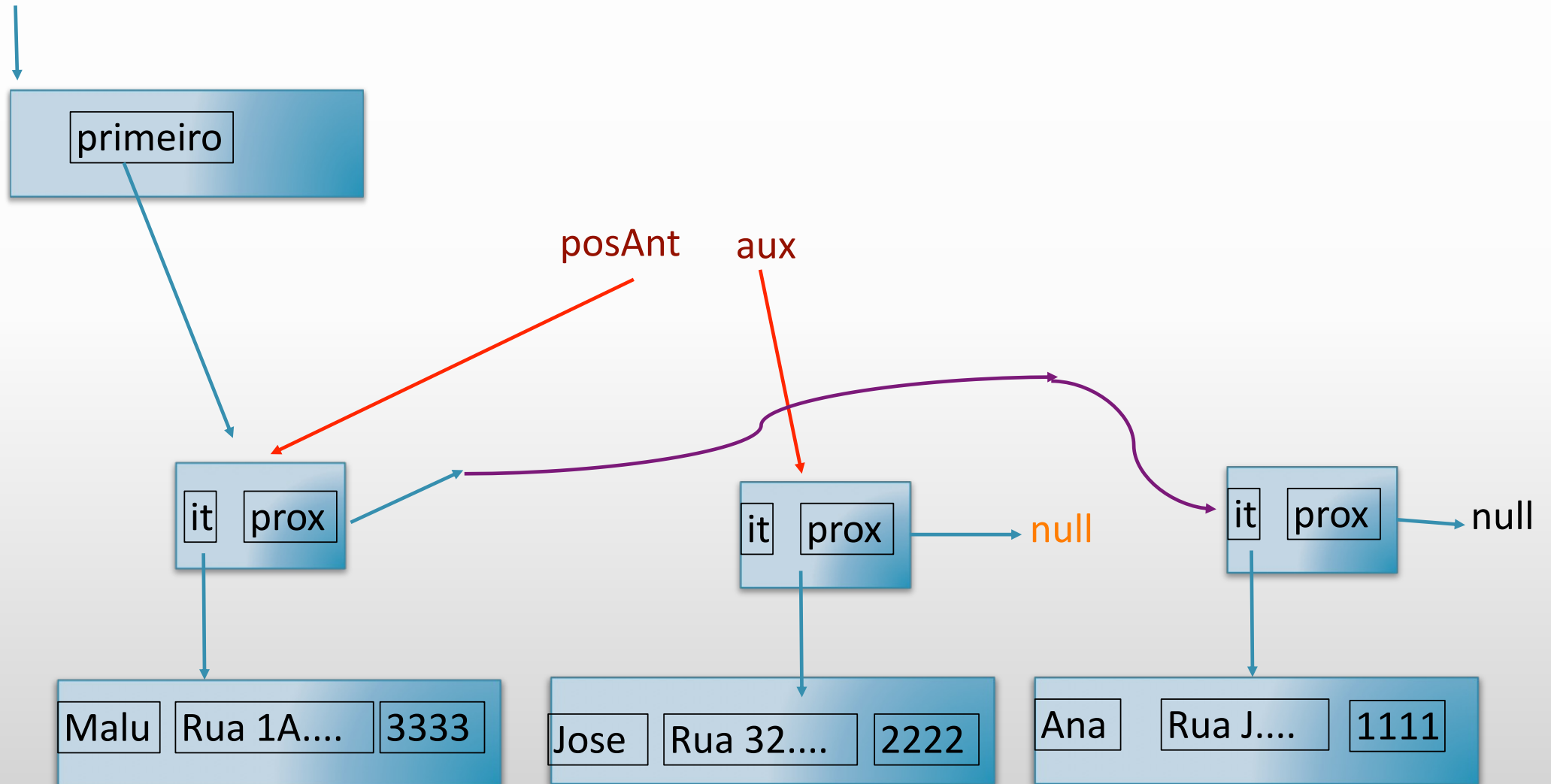


Item retira(Item x)
chave: 2222

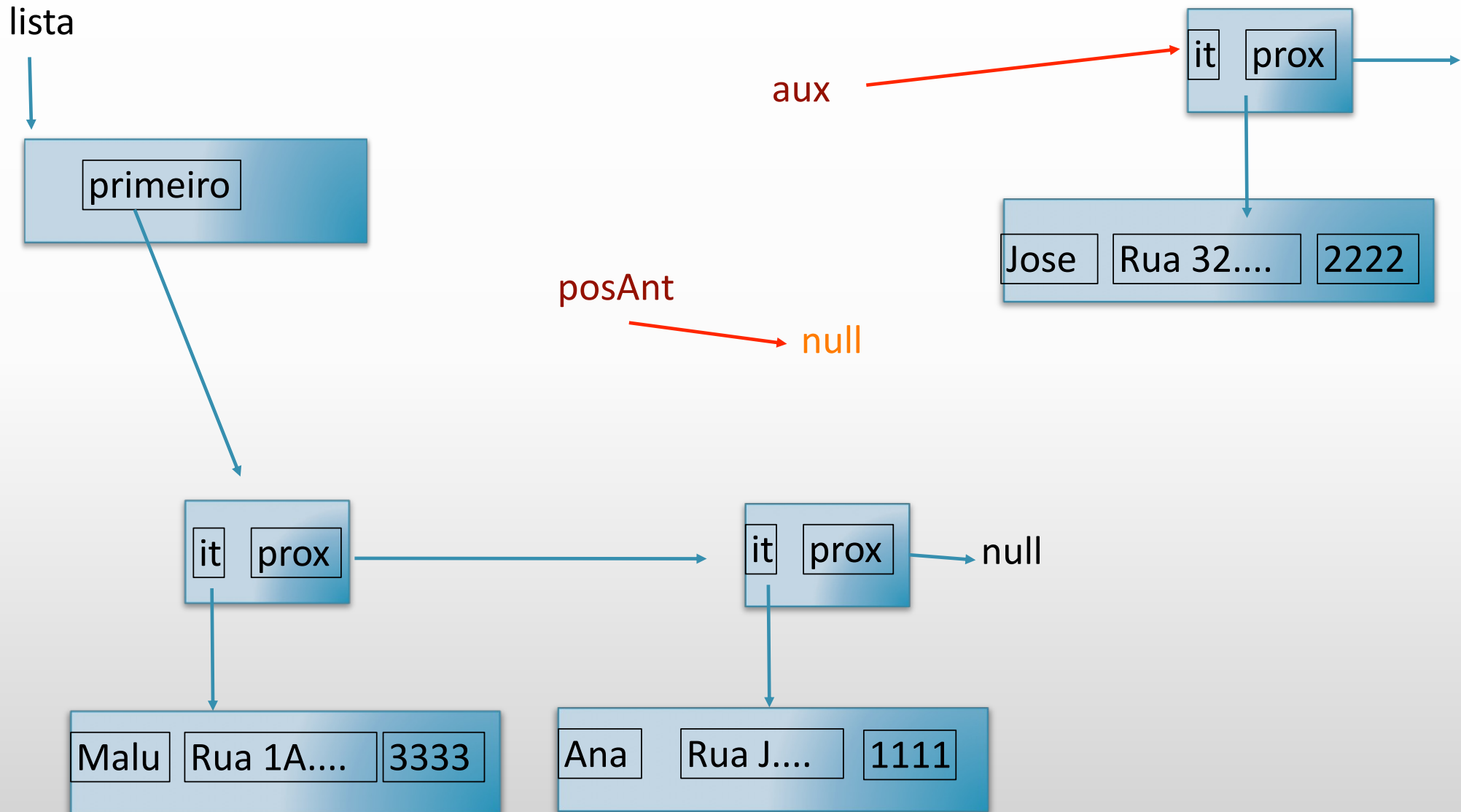


Item retira(Item x)
chave: 2222

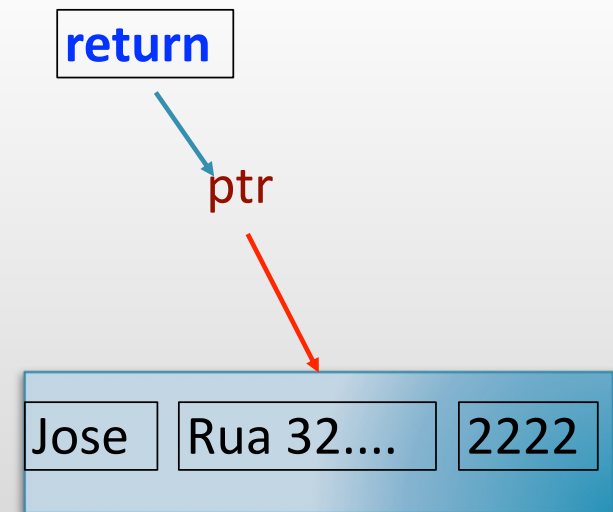
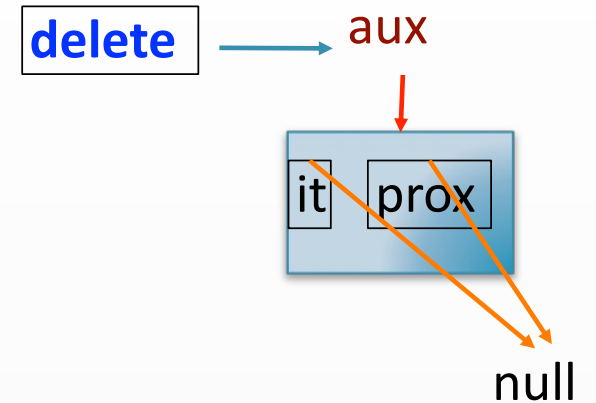
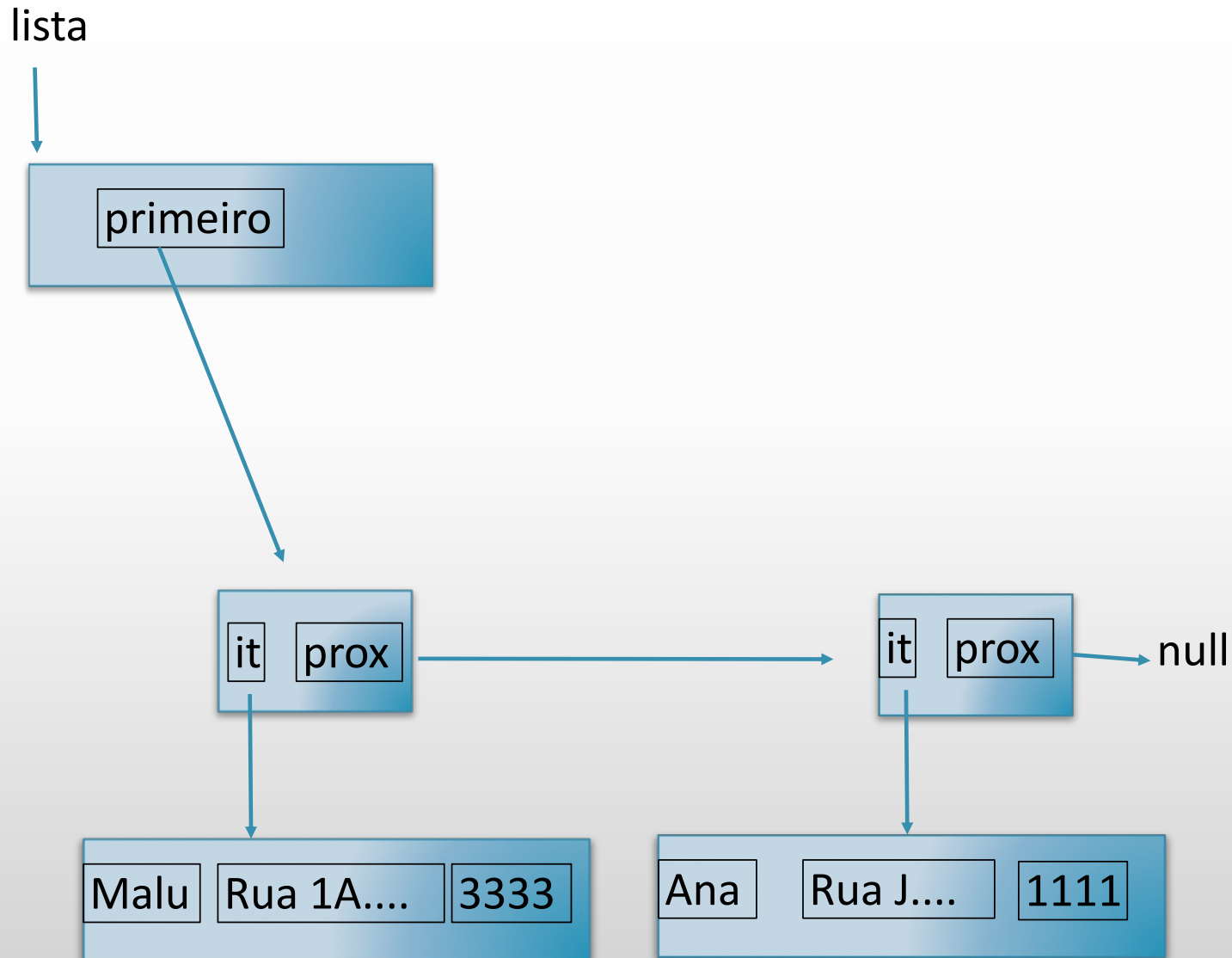
lista



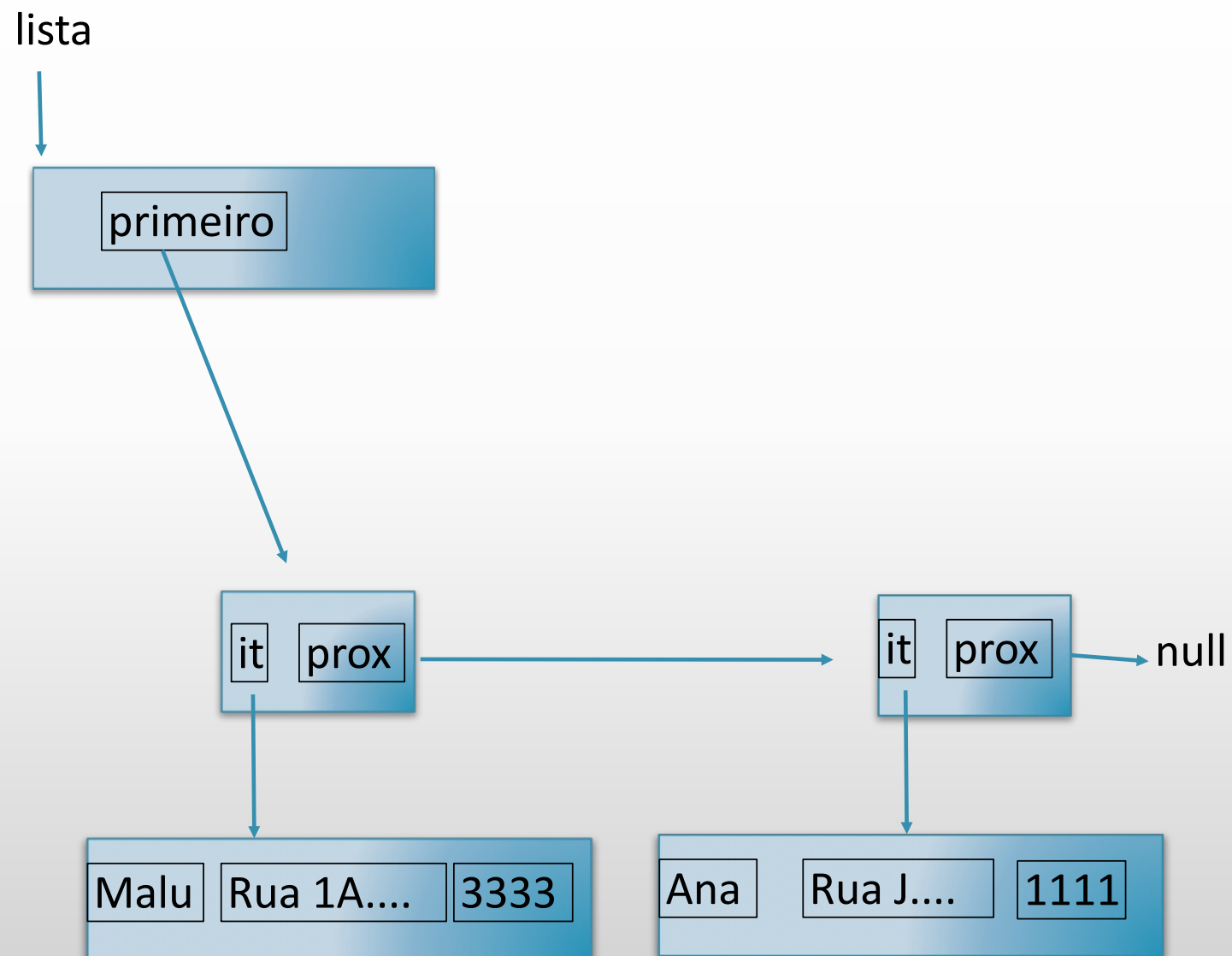
Item retira(Item x)
chave: 2222



Item Retira(Item x) chave: 2222



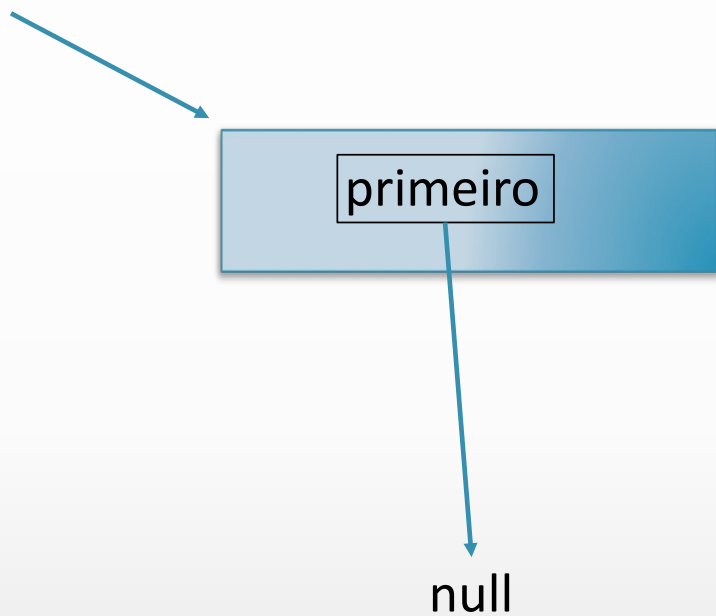
Lista com 2 itens



Após 2 retiradas

Lista vazia

lista



Implementação de Listas por meio de Estruturas Auto-Referenciadas

- Vantagens:
 - Permite inserir ou retirar itens do meio da lista a um custo constante (importante quando a lista tem de ser mantida em ordem).
 - Bom para aplicações em que não existe previsão sobre o crescimento da lista (o tamanho máximo da lista não precisa ser definido a priori).
- Desvantagem:
 - utilização de memória extra para armazenar as referências.

UML

Classe ItemCompra

- descricao: String
- qde: int
- preco: double
- + setQde(int): boolean (≥ 0)
- + setPreco(double): boolean (> 0.0)
- + setDescricao(String): void
- + getItemCompra(): String
- + getDescricao(): String

Classe Base

- leia: Scanner
- + main(): void
- + obterItemCompra(): ItemCompra

Classe Lista

Classe No

- dados: ItemCompra
- proximo: No
- + No(ItemCompra*)

- primeiro: No
- qde: int
- + inserirInicio(ItemCompra): void
- + retirar(ItemCompra): ItemCompra
- + buscar(ItemCompra): ItemCompra
- + vazia(): lógico
- + getQde(): int
- + getLista(): string