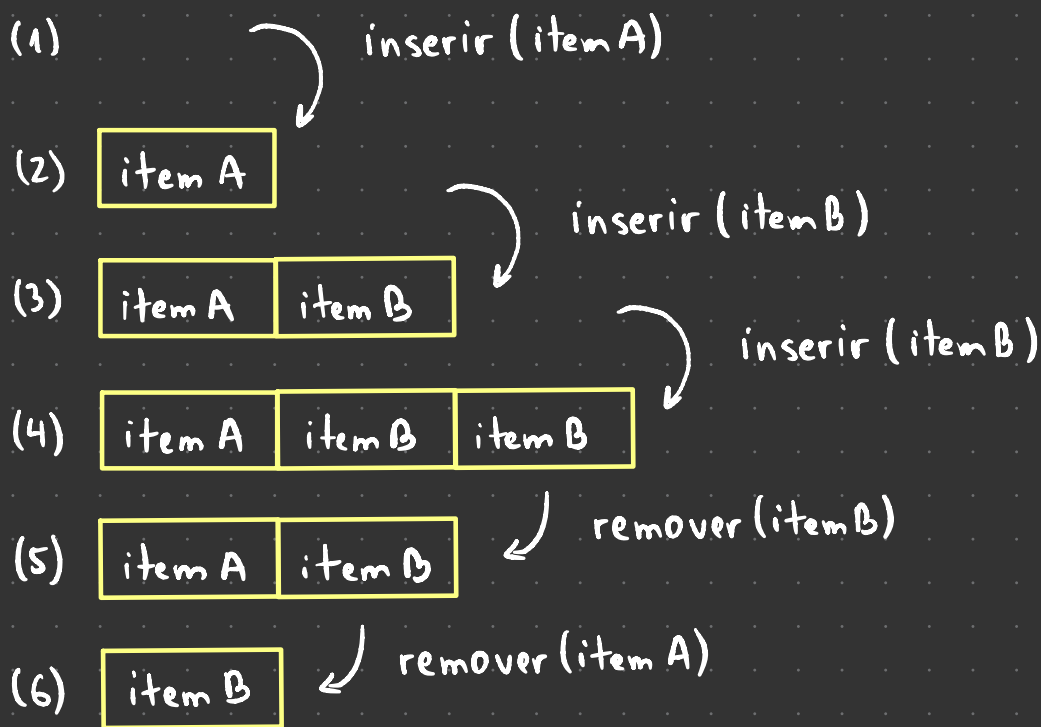


# LISTAS

- ↳ Extremamente similares às listas do dia a dia (supermercado, farmácia)
- ↳ Sua política NÃO é tão rigorosa quanto das pilhas e filas.

↳ Exemplo (supermercado):



\* OBS: cada item é chamada de **NÓ**

↳ Regras (políticas):

1. Só é possível inserir no final
2. Pode-se remover **qualquer nó** da lista
3. Não é possível remover um nó de uma lista vazia

### ↳ Operações:

- (a) inserir() ou insert() / add() / push()
- (b) remover() ou remove() / delete() / pop()
- (c) primeiro() ou first()
- (d) ultimo() ou last()
- (e) estaVazia() ou isEmpty()
- (e) tamanho() ou size()

### ↳ Em C:

```
int inserir(lista *l, int dado);  
int remover(lista *l, int dado);  
int primeiro(lista l);  
int ultimo(lista l);  
int estaVazia(lista l);  
int tamanho(lista l);
```

### ↳ Funções adicionais:

- (a) criar() ou create()
- (b) destruir() ou destroy()
- (c) tamanhoBytes() ou sizeBytes()
- (d) mostrar() ou show()
- (e) substituir() ou replace()
- (f) contar() ou count()
- (g) removerTodos() ou remove All()

### ↳ Em C:

```
int criar(lista *l);  
int destruir(lista *l);  
size_t tamanhoBytes(lista l);  
void mostrar(lista l);  
int substituir(lista *l, int d1, int d2);  
int contar(lista l, int dado);  
int removerTodos(lista *l, int dado);
```

## Os Tipos de lista:

### (1) Linear simplesmente encadeada:

primeiro: [100]

ultimo: [150]



dado prox



dado prox

### (2) Linear duplamente encadeada:

primeiro: [100]

ultimo: [150]



ant dado prox



ant dado prox

### (3) Circular simplesmente encadeada:

primeiro: [100]

ultimo: [150]



dado prox



dado prox

### (4) Circular duplamente encadeada:

primeiro: [100]

ultimo: [150]



ant dado prox



ant dado prox