Exercícios de Implementação: Listas Encadeadas

Foco: Criação de Nós, Travessia e Manipulação Básica de Ponteiros

---

1. Criar e Conectar Nós Manualmente

Crie uma estrutura de `Nó` (Node) que armazene um número inteiro e um ponteiro para o próximo nó. Em seguida, crie três nós manualmente com os valores 10, 20 e 30 e conecte-os para formar a lista: `10 -> 20 -> 30`. Por fim, imprima o valor do segundo nó (`20`) acessando-o a partir do nó principal (`head`).

- **Objetivo:** Entender a estrutura básica de um nó e como o ponteiro `next` funciona.
- **Saída Esperada (no console):**

```
Valor do segundo nó: 20
```

-
-

2. Percorrer e Imprimir a Lista

Crie uma função que receba o `head` (cabeçalho) de uma lista encadeada e imprima o valor de cada nó em uma única linha, até chegar ao final da lista (onde o ponteiro `next` é nulo).

- **Entrada:** Uma lista `1 -> 2 -> 3 -> 4`
- **Saída Esperada (no console):**

```
1 -> 2 -> 3 -> 4 -> NULL
```

-
-

3. Adicionar um Nó no Início da Lista

Implemente uma função que receba o `head` da lista e um valor. A função deve criar um novo nó com esse valor, fazer com que ele aponte para o antigo `head` e retornar o novo nó como o novo `head` da lista.

- **Entrada:** Lista `[5, 10, 15]`, valor `2`
- **Saída:** Lista `[2, 5, 10, 15]`

4. Adicionar um Nó no Fim da Lista

Crie uma função que receba o `head` da lista e um valor. A função deve percorrer a lista até encontrar o último nó e, então, adicionar o novo nó após ele.

- **Entrada:** Lista `[10, 20]`, valor `30`
- **Saída:** Lista `[10, 20, 30]`

5. Contar o Número de Nós

Implemente uma função que receba o head de uma lista e retorne a quantidade total de nós presentes nela.

- **Entrada:** Lista [7, 7, 7, 7, 7]
- **Saída:** 5

6. Encontrar um Elemento na Lista

Crie uma função que receba o head de uma lista e um valor. A função deve percorrer a lista e retornar true se encontrar um nó com o valor especificado, e false caso contrário.

- **Entrada:** Lista [1, 2, 6, 3, 4], valor 6
- **Saída:** true
- **Entrada:** Lista [1, 2, 6, 3, 4], valor 5
- **Saída:** false

7. Encontrar o Último Elemento da Lista

Implemente uma função que receba o head de uma lista e retorne o valor do último nó. Se a lista estiver vazia, deve retornar um indicativo de erro (como null ou um valor especial).

- **Entrada:** Lista [1, 2, 3, 4, 5]
- **Saída:** 5

8. Remover o Primeiro Nó (Head)

Crie uma função que receba o head de uma lista, remova o primeiro nó e retorne o segundo nó como o novo head.

- **Entrada:** Lista [100, 200, 300]
- **Saída:** A função deve retornar o head da nova lista [200, 300]

9. Somar todos os Valores da Lista

Implemente uma função que percorra toda a lista encadeada e retorne a soma dos valores de todos os seus nós.

- **Entrada:** Lista [5, 10, 15, 20]
- **Saída:** 50

10. Concatenar Duas Listas

Crie uma função que receba o head de duas listas encadeadas (lista1 e lista2). A função deve conectar o final da lista1 com o início da lista2. Retorne o head da lista1.

- **Entrada:** lista1 = [1, 2], lista2 = [3, 4, 5]
- **Saída:** A lista1 agora deve ser [1, 2, 3, 4, 5]