

Listas simples - Subturma B

Arquivo 1: No.java

```
// A classe que representa um único elemento da lista.
public class No {
    int dado;
    No proximo; // A "seta" que aponta para o próximo nó.

    // Construtor para criar um novo nó.
    public No(int dado) {
        this.dado = dado;
        this.proximo = null; // Quando um nó é criado, ele ainda não aponta para ninguém.
    }
}
```

Arquivo 2: ListaEncadeada.java

```
// A classe que gerencia todos os nós e a lógica da lista.
public class ListaEncadeada {
    No inicio; // O ponto de partida da nossa lista. Equivalente ao "head".

    // Método já ensinado
    public void adicionarNoInicio(int dado) {
        No novoNo = new No(dado);
        novoNo.proximo = inicio;
        inicio = novoNo;
    }

    // Método já ensinado
    public void imprimirLista() {
        System.out.print("Lista: [ ");
        No atual = inicio; // Começamos a percorrer pelo início.
        while (atual != null) {
            System.out.print(atual.dado + " → ");
            atual = atual.proximo; // Pulamos para o próximo nó.
        }
        System.out.println("NULL ]");
    }

    // ----- ATIVIDADE PRÁTICA COMEÇA AQUI ----- //

    /*
    * OBJETIVO 1: Somar todos os elementos
    * Implemente um método que percorra a lista e retorne
    */
}
```

```

    * a soma de todos os valores armazenados.
    */
    public int somarTudo() {
        // Seu código vem aqui
        // Dica: você vai precisar de uma variável acumuladora (ex: soma = 0).
        // A cada nó percorrido, some o dado do nó na variável.
        // Ao final, retorne a soma.
        return 0; // Apague esta linha e substitua pelo seu código.
    }

    /*
    * OBJETIVO 2: Verificar existência
    * Implemente um método que verifique se um número
    * existe na lista. Retorne true se encontrar, false se não.
    */
    public boolean verificarExistencia(int valor) {
        // Seu código vem aqui
        // Dica: percorra a lista com while.
        // Compare o dado de cada nó com o valor procurado.
        // Se encontrar, retorne true na hora.
        // Se terminar o laço sem encontrar, retorne false.
        return false; // Apague esta linha e substitua pelo seu código.
    }
}

```

Arquivo 3: **Principal.java** (para testar)

```

public class Principal {
    public static void main(String[] args) {
        ListaEncadeada lista = new ListaEncadeada();
        lista.adicionarNoInicio(10);
        lista.adicionarNoInicio(20);
        lista.adicionarNoInicio(30);
        lista.adicionarNoInicio(40);
        lista.imprimirLista(); // Deve imprimir: Lista: [ 40 → 30 → 20 → 10 → NULL ]

        System.out.println("--- Testando a Atividade ---");

        // Testando o Objetivo 1
        System.out.println("Somatório da lista: " + lista.somarTudo());
        // Deve imprimir: 100

        // Testando o Objetivo 2
        System.out.println("Existe o número 30? " + lista.verificarExistencia(30));
        // Deve imprimir: true
        System.out.println("Existe o número 99? " + lista.verificarExistencia(99));
        // Deve imprimir: false
    }
}

```

```
// Teste extra em lista vazia
ListaEncadeada listaVazia = new ListaEncadeada();
System.out.println("Somatório da lista vazia: " + listaVazia.somarTudo());
// Deve imprimir: 0
System.out.println("Existe o número 1 na lista vazia? " + listaVazia.verificarExistencia(1));
// Deve imprimir: false
    }
}
```