

# Descrição da situação problema

---

Minha tarefa foi criar um programa Java que permite ao usuário gerenciar uma biblioteca de filmes. O programa deve ser capaz de adicionar, remover, listar os filmes presentes na biblioteca

## Códigos Utilizados

---

Aqui estão os códigos Java utilizados para implementar a solução:

- **Main.java**

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) throws Exception {
        Scanner scanner = new Scanner(System.in);
        Vetor bibliotecaFilmes = new Vetor(1);

        String exemploFilme = "Matrix (Dir: Wachowskis, Gênero: Ficção Científica)";
        bibliotecaFilmes.adiciona(exemploFilme);

        int escolha;
        do {
            System.out.println("\033[36m\n--- Biblioteca de Filmes - Menu ---\033[0m");
            System.out.println("1. \033[1mAdicionar Filme\033[0m");
            System.out.println("2. \033[1mRemover Filme\033[0m");
            System.out.println("3. \033[1mListar Filmes\033[0m");
            System.out.println("4. Sair");
            System.out.print("Escolha uma opção: ");

            escolha = scanner.nextInt();
            scanner.nextLine();

            switch (escolha) {
                case 1 -> {
                    System.out.print("Digite o nome do filme: ");
                    String nomeFilme = scanner.nextLine();
                    System.out.print("Digite o diretor do filme: ");
                    String diretorFilme = scanner.nextLine();
                    System.out.print("Digite o gênero do filme: ");
                    String generoFilme = scanner.nextLine();
                    String filmeInfo = nomeFilme + " (Diretor: " + diretorFilme + ", G
                    try {
                        bibliotecaFilmes.adiciona(filmeInfo);
                        System.out.println("\033[32mFilme adicionado com sucesso!\033[
                    } catch (Exception e) {
                        System.out.println("\033[31mErro ao adicionar o filme: " + e.g
```

```

    }
}
case 2 -> {
    if (bibliotecaFilmes.getTamanho() == 0) {
        System.out.println("\033[31mA biblioteca de filmes está vazia.
    } else {
        System.out.println("Biblioteca de Filmes antes da remoção:");
        listarFilmes(bibliotecaFilmes);

        System.out.print("Digite a posição do filme a ser removido: ")
        int posicaoRemover = scanner.nextInt();
        scanner.nextLine();
        try {
            String filmeRemovido = (String) bibliotecaFilmes.busca1(po
            bibliotecaFilmes.remove(posicaoRemover);
            System.out.println("\033[32mFilme removido com sucesso - Í
        } catch (Exception e) {
            System.out.println("\033[31mErro ao remover o filme: " + e
        }
    }
}
case 3 -> listarFilmes(bibliotecaFilmes);
case 4 -> System.out.println("\033[33mSaindo do programa...\033[0m");
default ->
    System.out.println("\033[31m\nOpção inválida. Por favor, escol

}
} while (escolha != 4);
}

private static void listarFilmes(Vetor bibliotecaFilmes) throws Exception {
    if (bibliotecaFilmes.getTamanho() == 0) {
        System.out.println("\033[31m\nA biblioteca de filmes está vazia.\033[0m");
    } else {
        System.out.println("\033[36m\n--- Biblioteca de Filmes - Lista ---\033[0m"
        for (int i = 0; i < bibliotecaFilmes.getTamanho(); i++) {
            System.out.println("\033[1mÍndice " + i + ":\033[0m " + bibliotecaFilm
        }
    }
}
}
}
}

```

## • Vetor.java

```

public class Vetor {
    private Object[] elementos;
    private int tamanho;

    public Vetor(int capacidade) {
        this.elementos = new Object[capacidade];
        this.tamanho = 0;
    }
}

```

```

public void adiciona(Object elemento) throws Exception {
    this.aumentaCapacidade();
    this.elementos[this.tamanho] = elemento;
    this.tamanho++;
}

public int getTamanho() {
    return this.tamanho;
}

@Override
public String toString() {
    StringBuilder s = new StringBuilder();
    s.append("[");

    for (int i = 0; i < this.tamanho - 1; i++) {
        s.append(this.elementos[i]);
        s.append(", ");
    }

    if (this.tamanho > 0) {
        s.append(this.elementos[this.tamanho - 1]);
    }

    s.append("]");

    return s.toString();
}

public Object busca1(int posicao) throws Exception {
    if (posicao >= 0 && posicao < tamanho) return elementos[posicao];
    else throw new Exception("Posição Inválida!");
}

public int busca2(Object buscado) {
    for (int i = 0; i < tamanho; i++) {
        elementos[i] = elementos[i].toString().toLowerCase();
        if (elementos[i].equals(buscado)) return i;
    }
    return -1;
}

public boolean adicionaInicio(int posicao, Object elemento) throws Exception {
    if (!(posicao >= 0 && posicao < tamanho)) throw new Exception("Posição Inválida");
    for (int i = this.tamanho - 1; i >= posicao; i--) this.elementos[i + 1] = this.elementos[i];
    this.elementos[posicao] = elemento;
    this.tamanho++;

    return true;
}

public void aumentaCapacidade() {

```

```
    if (this.tamanho != this.elementos.length) return;

    Object[] elementosNovos = new Object[this.elementos.length * 2];
    for (int i = 0; i < this.elementos.length; i++) elementosNovos[i] = this.eleme
    this.elementos = elementosNovos;
}

public void remove(int posicao) throws Exception {
    if (!(posicao >= 0 && posicao < tamanho)) throw new Exception("Posição Inválid
    for (int i = posicao; i < this.tamanho - 1; i++) this.elementos[i] = this.elem
    this.tamanho--;
}
}
```