

AVALIAÇÃO 01 - LISTA DE EXERCÍCIOS

Disciplina de Backend - Professor Ramon Venson - SATC 2024

Exercício 01

Entrega: 11 de Março

Implemente um programa em java que seja capaz de gerar aleatoriamente e retornar no console o seguinte modelo de mensagem:

```
Cássio Ramos é um futebolista brasileiro de 32 anos que atua como goleiro. Atualmente defende o Corinthians.
```

A mensagem de texto deverá conter os seguintes atributos aleatórios:

- **Nome** e **sobrenome** aleatórios
- Idade (entre 17 e 40 anos)
- **Posição** (aleatória)
- **Clube** (aleatório)

Exercício 02

Entrega: 18 de Março

Utilizando como base o exercício anterior, crie uma classe chamada `JogadorGerador` contendo um método capaz de realizar as requisições HTTP apenas uma vez e construir múltiplos objetos do tipo `Jogador`.

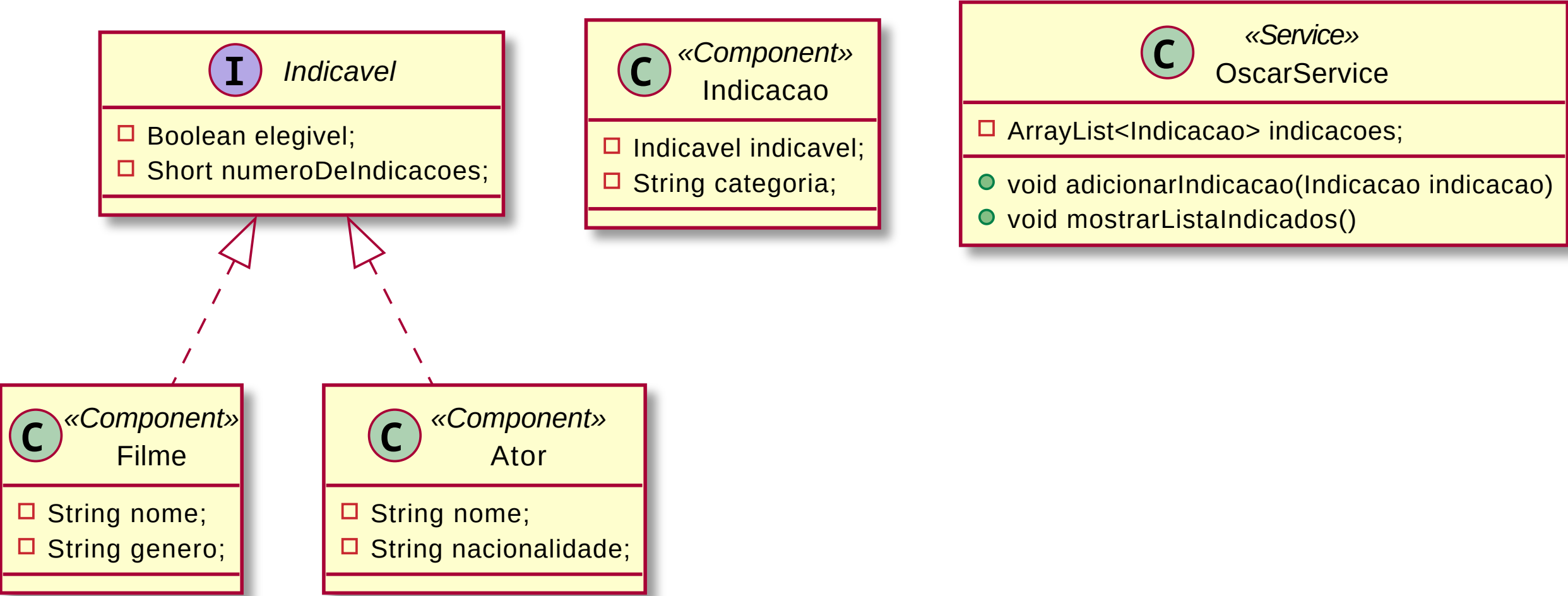
Crie também uma classe chamada `Jogador` contendo métodos que permitam que cada dado do jogador seja acessado individualmente através de métodos (`getNome`, `getSobrenome`, `getPosicao`, `getIdade`, `getClube`) e um método que gere a mensagem do exercício anterior (`getDescricao`).

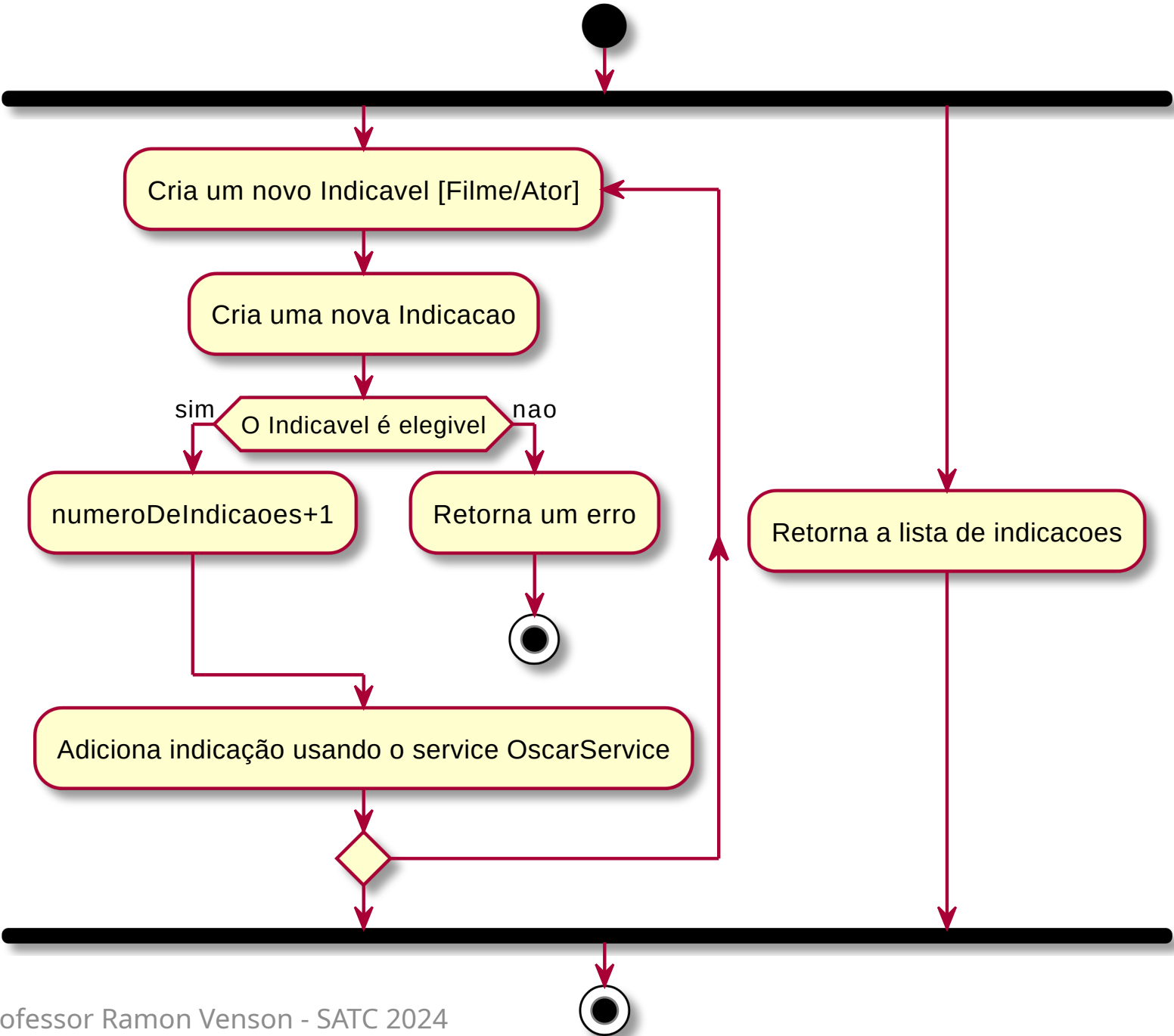
Exercício 3

Entrega: 25 de Março

Crie um novo projeto usando o Spring Framework que implemente um **service** chamado `OscarService` que seja responsável por adicionar filmes e atores à uma lista de indicados (array do tipo `Indicacao`). O service também deverá retornar a lista com todos os indicados e suas categorias;

Um ator ou filme não poderá ser indicado um atributo `elegivel` for falso. Para cada nova indicação, um atributo `numeroDeIndicacoes` deve ser incrementado. Ambos os atributos devem ser implementados por meio de uma interface chamada `Indicavel`.





Todos os atributos devem ter sua visibilidade `private`. Crie getters e setters para os atributos que necessitem de acesso/modificação.

O projeto deverá seguir a estrutura dos diagramas UML, porém alterações que incluam funcionalidades ou alterem o nome das classes são permitidas desde que mantenham o formato da implementação.

O projeto deverá conter pelo menos:

- 1x Service
- 1x Interface
- 3x Components

BONUS: Implemente também a persistência da lista de indicados em um arquivo de texto;

Exercício 04

Crie um web service capaz de prover através de requisição GET uma mensagem como a do exercício 01 / exercício 02. Novas mensagens/jogadores devem ser geradas à cada requisição. O web service deverá carregar uma lista de **nomes**, **sobrenomes**, **posições** e **clubes** apenas durante sua inicialização.

Bônus: Implemente uma rota extra para gerar e apresentar jogadores diretamente no formato JSON.