



**Certified Tech
Developer**

The Ultimate Degree

Back End I

Durante esta aula, sugerimos que faça os exercícios das aulas anteriores que não completou para recuperar o atraso ou, caso os tenha feito, deixamos-lhe os seguintes desafios para continuar a colocar em prática o que aprendeu.

Aula integradora: Proxy pattern

Objetivo

Realizar o diagrama **UML** e **programar em Java**, implementando o **Proxy pattern** de acordo com o seguinte enunciado.

Considere que:

- Exercício individual
- Nível de complexidade intermediário: 🔥🔥

Enunciado

Agora nosso aplicativo de streaming vai incorporar a transmissão de **séries**. Nesse caso, precisamos desenvolver uma funcionalidade em particular: como cliente, queremos que o aplicativo nos permita fazer um pedido com o nome da série e nos envie o link de reprodução. O único limite imposto ao nosso pedido é que o número de séries não pode ser superior a 5 por semana. Para saber quantas já foram vistas, será feito um contador. Propõe-se a criação de uma estrutura de classes semelhante ao exercício anterior, com as seguintes classes:

- Uma Interface **ISerie** que define um método **getSerie** que recebe o **nome**, devolvendo uma **String**.
- Uma classe **Serie** que implementa a interface **ISerie** e devolve uma **String** que representa o link, que simplesmente é o prefixo **"www."** seguido pelo **nome** da série.
- Uma classe **SerieProxy** com um atributo **qtdViews** que conta o número de vezes que uma série foi solicitada e pede a **Serie** um link com **getSerie** e verifica o número de vezes que uma série foi reproduzida e se o valor máximo for excedido, ela lança uma exceção do tipo **SerieNaoHabilitadaException**, criada para esse fim.

No método principal (main), faça solicitações de séries com o nome. Mostrar o link da série caso esteja habilitado. Se não estiver, exiba: *"Excede o número de reproduções permitidas"*. **Bons estudos!**