



**Título do projeto:** Polimorfismo paramétrico, coleções e tratamento de erros.

**Período de realização:** acompanhamento nas aulas de 02/05 a 23/05

**Data do pull para correção no GitHub Classroom (Tarefa "LPM\_Projetos\_3\_4\_5"):** 30/05

O projeto originalmente chamado de "Minhas Séries" já acumula algumas semanas de iterações e, assim, deve apresentar características que realizem ações já pedidas pelo cliente. Relembrando o *backlog*, temos:

- Implementação das classes Cliente, Serie e PlataformaStreaming como originalmente planejadas;
- Atualização do diagrama original para contemplar requisito de carga de dados;
- Implementação dos métodos para carga de dados de séries, clientes e audiência em funcionamento;
- Atualização do diagrama para abrigar o novo tipo de mídia: Filme;
- Implementação da lógica de carga de dados do catálogo de filmes;
- Implementação do cadastro de clientes, filmes e séries;
- Salvar dados de clientes, filmes e séries;
- Refatoração / reescrita dos filtros de listas de mídias assistidas e futuras.
- Implementação do protótipo de sistema cobrindo os requisitos atuais (documento anterior);

Com tantas tarefas realizadas e em andamento, é a hora de garantir que esteja tudo organizado e documentado.

A novidade em requisitos funcionais do sistema é que, a partir de agora, a empresa de *streaming* quer permitir que os clientes deixem uma avaliação entre 1 e 5 estrelas para séries e filmes que tenha visto. Esta avaliação não é obrigatória, ou seja, o cliente pode terminar de ver a mídia e preferir não avaliar.

#### **Atividades valendo 2 pontos na aula da primeira semana do projeto (02/05):**

- Atualizar o diagrama, o código e os testes para incluir o requisito de avaliação de mídias descrito acima;
- Revisar a documentação atual do código e gerar o *javadoc* correspondente (**veja o exemplo do professor**);
- Garantir, na documentação do repositório, que:
  - Esteja preenchida corretamente a documentação da página principal;
  - Os diagramas estejam numerados de acordo com a regra e entregues em formato jpg ou png;
  - As instruções para o uso do sistema estejam preenchidas conforme pedido;
  - As tarefas do *backlog* acima estejam documentadas como pendentes ou terminadas (**veja o exemplo**);

#### **Artefatos esperados:**

##### **Até 09/05:**

- Funcionamento do sistema de avaliação de séries e filmes;

##### **Até 16/05:**

- Ainda não priorizados

##### **Até 23/05:**

- Ainda não priorizados

#### **Instruções e observações:**

- O projeto deve estar hospedado na tarefa correspondente do GitHub Classroom;
- A execução do projeto segue, em linhas gerais, a metodologia ágil Scrum para desenvolvimento. Assim, haverá acompanhamento semanal da evolução do projeto;
- Os requisitos do projeto cobrem conteúdos que serão ministrados ao longo das semanas das matérias de *Programação Modular* e *Laboratório de Programação Modular*;

#### **Critérios de pontuação:**

- Aderência às classes do diagrama: 2 pontos
- Requisitos de corretamente implementados: 14 pontos
- Documentação de código: 2 pontos
- Implementação na aula inicial: 2 pontos
- Atraso no cronograma de artefatos: desconto entre 1 e 2 pontos por semana atrasada

**A nota final** se dará pela soma acima, multiplicada por um peso entre 0 e 1 relativo ao acompanhamento semanal do projeto. Lembre-se: não é só a entrega do produto finalizado que importa, é todo o processo de sua construção

e as entregas parciais para o "cliente".