



Título do projeto: Herança, interfaces e polimorfismo

Período de realização: acompanhamento nas aulas de 11/04 a 26/04

Data do pull para correção no GitHub Classroom (Tarefa "LPM_Projetos_3_4_5"): 02/05

Agora que o protótipo do sistema consegue diferenciar séries e filmes e carregar dados para a memória, falta um ajuste fundamental: precisamos conseguir cadastrar novos clientes, filmes, e séries e salvar dados destas mesmas entidades.

Além disso, é necessário começar a deixar nosso sistema mais robusto, evitando que erros simples como conversão de tipos e falha em arquivos não derrubem o protótipo.

Portanto, neste momento, o *backlog* do projeto conta com as tarefas abaixo e seus testes:

- 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;
- Implementação do protótipo de sistema cobrindo os requisitos atuais (documento anterior);
- Garantir a robustez do sistema principal.

Artefatos esperados:

Até 19/04:

- Atualização do diagrama UML;
- Implementação e teste da classe PlataformaStreaming;
- Método(s) para carregamento dos dados;
- Criação de um pequeno aplicativo para chamar o carregamento dos dados;

Até 26/04:

- Classe filme e carga do catálogo de filmes;
- Primeira versão do protótipo do aplicativo;

Até 02/05:

- Cadastro de novas entidades;
- Salvar dados das entidades;
- Robustez no sistema principal.

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: 12 pontos
- Documentação de código: 4 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".