Pontifícia Universidade Católica de Minas Gerais
Bacharelado em Engenharia de Software - Unidade Praça da Liberdade
Laboratório de Programação Modular
Prof. João Caram - **Semestre** 1/2023
Projeto prático 3 - Valor: 20 pontos

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

Dentre as formas de entretenimento contemporâneas, os serviços de *streaming* de conteúdo se multiplicaram e tomaram conta de boa parte do mercado nos últimos anos, como pode ser percebido em números relatados por pesquisas recentes.

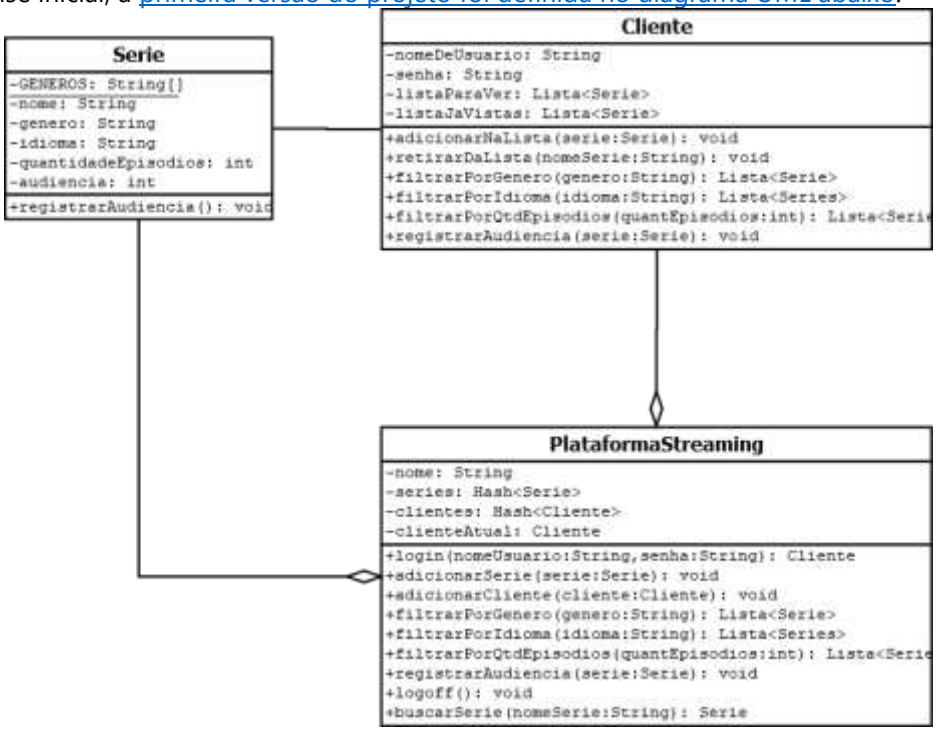
A companhia de monitoramento de mercado Hibou realizou uma pesquisa em maio de 2022, e seu resultado aponta que 71% dos brasileiros usam ou já usaram algum serviço de *streaming*.[1] Segundo a mesma pesquisa, 70% destes usuários têm como principal motivo da assinatura o fato de poder ver séries originais das plataformas, muitas vezes realizando o que ficou conhecido como *maratonar uma série*, ou seja, ver todos os seus episódios de maneira consecutiva em poucos dias.

Recentemente, a Forbes divulgou números espantosos para os serviços de streaming mais conhecidos[2]: a Netflix tinha mundialmente, em fevereiro de 2023, mais de 230 milhões de assinantes. Outras empresas apresentam números igualmente enormes: 161 milhões para a Disney+ e mais de 77 milhões para o grupo Paramount.

Considerando este contexto, seu grupo de trabalho recebeu a responsabilidade de desenvolver um sistema de software que possa ser vendido para companhias de *streaming*. O sistema se mostra abrangente e de média complexidade, de modo que a estratégia escolhida foi a de desenvolvimento por módulos que serão integrados cumulativamente. O primeiro passo é entender os requisitos mais básicos de um serviço simples de *streaming*:

- O serviço precisa manter dados de um número grande de clientes, cada um com seu login único.
- Igualmente, os dados de séries precisam ser armazenados. Uma série terá cadastrados os dados seguintes: nome, idioma e gênero. É preciso registrar quantos clientes já assistiram uma série.
- Os clientes poderão adicionar séries a duas listas em sua conta: uma contendo séries para assistir futuramente e outra para registrar e consultar suas séries já assistidas.
- O cliente pode realizar buscas em suas listas, ou na lista geral de séries do sistema. A busca pode ser por nome, gênero ou idioma.

A partir da análise inicial, a primeira versão do projeto foi definida no diagrama UML abaixo:



[1] Disponível em http://www.lehibou.com.br/wp-content/uploads/2022/07/22STR01.pdf
[2] Disponível em https://www.forbes.com/sites/marisadellatto/2023/02/16/paramount-gains-subscribers-as-disney-reports-losses-where-all-the-major-streaming-services-stand/?sh=14df9763c4ac

A tarefa do grupo para a primeira aula deste projeto, valendo 2 pontos, consiste em:

- Implementar e testar a classe Cliente;
- Implementar e testar a classe Série.

Após a implementação inicial e o teste das classes, no segundo passo do projeto será necessário um módulo que leia arquivos com dados em formato texto (csv) e crie os objetos correspondentes para utilização no sistema. Os arquivos são 3, cada um sendo formatado linha a linha como indicado:

`Espectadores`
`Nome;Login;Senha`

`Séries`
`IdSerie;Nome;DataDeLançamento`

`Audiência`
`Login;F/A;IdSerie` *(sendo "F" para lista de séries a assistir futuramente e "A" para séries já assistidas.)*

Para o protótipo do sistema, os gêneros das séries podem ser definidos aleatoriamente.

O grupo precisa, então, atualizar o projeto para contemplar o requisito de leitura dos dados. Portanto, as tarefas constantes do trabalho são, no momento:

a) implementar as classes conforme validação do diagrama inicial;
b) atualizar o diagrama para o requisito de leitura;
c) implementar os códigos para leitura e testar seu funcionamento.

Artefatos esperados:

Aula de 12/04:

- Implementação e teste de Cliente e Série;

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:

- Requisitos ainda não priorizados;

Até 02/05:

- Requisitos 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: 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".