

## **Exercício de Revisão de Análise e Projeto de Sistemas - Programação Orientada a Objetos: Sistema para Empresa de Streaming de Filmes**

### **Documento de Visão: Sistema StreamFlix**

#### **1. Introdução**

O Sistema StreamFlix pretende fornecer uma plataforma de streaming de vídeo robusta e intuitiva para usuários, permitindo-lhes acessar uma ampla variedade de filmes e séries de forma conveniente. Este documento delinea os objetivos gerais do sistema, seus objetivos específicos, o contexto em que estará inserido e uma lista de requisitos para guiar o desenvolvimento.

#### **2. Objetivo Geral**

O objetivo geral do Sistema StreamFlix é proporcionar aos usuários uma experiência de streaming de vídeo agradável e conveniente, oferecendo uma ampla gama de filmes e séries para assistir.

#### **3. Objetivos Específicos do Software**

- Facilitar a busca e navegação por filmes e séries.
- Permitir que os usuários adicionem filmes e séries à sua lista de favoritos.
- Oferecer recomendações personalizadas com base nas preferências de visualização dos usuários.
- Garantir a segurança e privacidade dos dados dos usuários.
- Proporcionar uma interface de usuário intuitiva e amigável.

#### **4. Contexto**

O Sistema StreamFlix será utilizado por uma variedade de usuários, incluindo entusiastas de filmes e séries, famílias e indivíduos que desejam acessar conteúdo de entretenimento de alta qualidade. O sistema estará inserido no mercado de entretenimento digital, competindo com outras plataformas de streaming populares.

O Sistema StreamFlix visa proporcionar uma experiência de streaming de vídeo agradável e conveniente para os usuários, oferecendo uma ampla variedade de filmes e séries para assistir. Ao atender aos objetivos e requisitos delineados neste documento, espera-se criar um sistema que atenda às necessidades e expectativas dos usuários, contribuindo para o sucesso e a satisfação dos clientes.

## 5. Requisitos

### 5.1 Lista de Requisitos Escrito da Forma Tradicional:

1. ~~O sistema deverá permitir que os usuários se cadastrem e façam login.~~
2. ~~O sistema deverá permitir que os usuários pesquisem filmes e séries por título, diretor, gênero ou atores.~~
3. ~~O sistema deverá permitir que os usuários adicionem filmes e séries à sua lista de favoritos.~~
4. ~~O sistema deverá permitir que os usuários removam filmes e séries de sua lista de favoritos.~~
5. ~~O sistema deverá exibir detalhes completos de filmes e séries, incluindo título, diretor, ano de lançamento, gênero, duração (para filmes) e número de temporadas (para séries).~~
6. ~~O sistema deverá oferecer uma interface de usuário intuitiva e responsiva.~~
7. ~~O sistema deverá garantir a segurança dos dados dos usuários, incluindo informações pessoais e histórico de visualização.~~
8. ~~O sistema deverá fornecer suporte técnico e assistência aos usuários, se necessário.~~

### 5.2 Lista de User Stories:

1. Como usuário, quero me cadastrar no sistema para poder acessar o conteúdo disponível.
2. Como usuário, quero poder fazer login na minha conta para acessar minha lista de favoritos e continuar assistindo de onde parei.
3. Como usuário, quero poder pesquisar filmes e séries por título, diretor, gênero ou atores para encontrar conteúdo de meu interesse.
4. Como usuário, quero poder adicionar filmes e séries à minha lista de favoritos para acessá-los facilmente mais tarde.
5. Como usuário, quero poder remover filmes e séries da minha lista de favoritos se não me interessarem mais.
6. Como usuário, quero ver detalhes completos de filmes e séries, incluindo título, diretor, ano de lançamento, gênero, duração (para filmes) e número de temporadas (para séries).
7. Como usuário, quero uma interface de usuário intuitiva e responsiva para facilitar a navegação e a seleção de conteúdo.
8. Como usuário, quero ter certeza de que minhas informações pessoais e histórico de visualização estão seguros e protegidos.
9. Como usuário, quero poder entrar em contato com o suporte técnico se tiver problemas ou dúvidas sobre o sistema.

Esse exercício deve permitir que possa ser revisado e praticado conceitos importantes de programação orientada a objetos, como classes, objetos, encapsulamento, herança, polimorfismo e métodos de acesso. Também incentiva a prática de boas práticas de programação, como comentar o código e utilizar um design orientado a objetos.

A empresa de streaming "StreamFlix" está expandindo seus serviços e deseja um sistema mais robusto para gerenciar seus filmes, séries e usuários. Você foi contratado como desenvolvedor para criar esse sistema utilizando programação orientada a objetos. Abaixo estão os requisitos e funcionalidades que devem ser implementados:

Classes:

### 1. Filme:

**- Atributos:**

- Título
- Diretor
- Ano de Lançamento
- Gênero
- Duração (em minutos)
- Lista de Atores Principais

**- Métodos:**

- Construtor para inicializar os atributos
- Métodos de acesso para todos os atributos
- Método para retornar uma representação textual do filme (toString)

### 2. Série:

**- Atributos:**

- Título
- Diretor
- Ano de Lançamento
- Gênero
- Número de Temporadas
- Lista de Atores Principais

**- Métodos:**

- Construtor para inicializar os atributos
- Métodos de acesso para todos os atributos
- Método para retornar uma representação textual da série (toString)

### 3. Usuário:

**- Atributos:**

- Nome
- Email
- Senha
- Lista de Filmes/Séries Favoritos

**- Métodos:**

- Construtor para inicializar os atributos
- Métodos de acesso para todos os atributos
- Método para adicionar um filme/série à lista de favoritos do usuário
- Método para remover um filme/série da lista de favoritos do usuário
- Método para listar todos os filmes/séries favoritos do usuário

**Funcionalidades:**

1. Criar pelo menos 3 objetos do tipo Filme e 3 objetos do tipo Série para popular o sistema.
2. Criar pelo menos 2 objetos do tipo Usuário.
3. Adicionar filmes/séries à lista de favoritos de um usuário.
4. Remover filmes/séries da lista de favoritos de um usuário.
5. Exibir todos os filmes/séries favoritos de um usuário específico.
6. Permitir que um usuário novo seja registrado no sistema.

**Requisitos adicionais:**

- Utilize encapsulamento onde apropriado.
- Demonstre o uso de herança, se aplicável (por exemplo, uma classe “Video” que é pai de “Filme” e “Série”).
- Demonstre o uso de polimorfismo, se aplicável (por exemplo, um método para imprimir detalhes que funciona para ambas as classes “Filme” e “Série”).
- Comente o código de forma adequada para explicar o que cada parte faz.