



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina de Arquitetura e Projeto de Sistemas

GABARITO – AD2 1º semestre de 2019.

Nome:

Polo:

Matrícula:

Observações:

1. Prova com consulta.

LER ATENTAMENTE AS INSTRUÇÕES A SEGUIR:

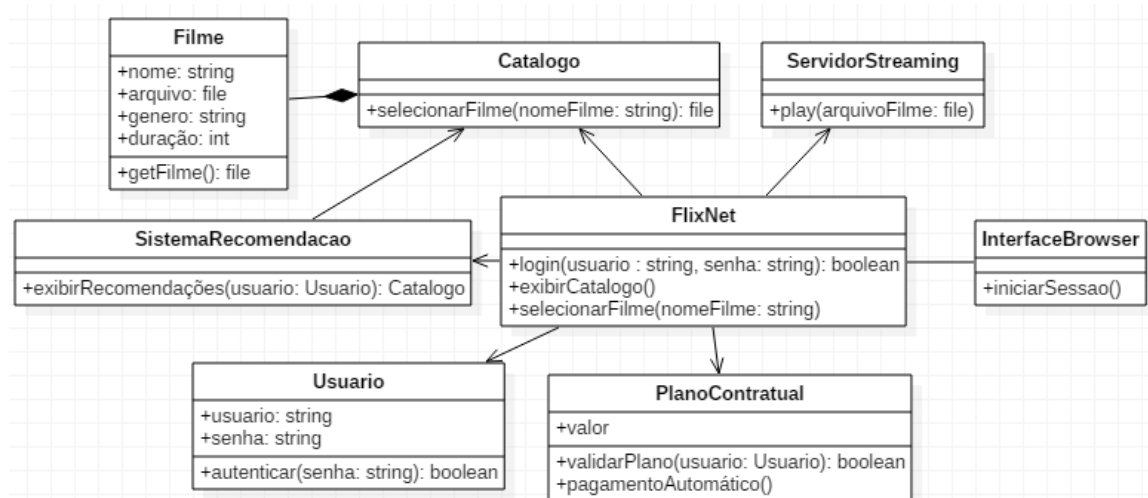
1. As respostas devem ser enviadas em um **único arquivo em formato exclusivamente .PDF, não compactado**. Além disso, o conteúdo deste arquivo deve **seguir exatamente o template das respostas**, caso exista. Caso não atenda a estes pontos, a **AD não será corrigida**. ADs enviadas no MODO RASCUNHO também **não serão corrigidas**. ADs MANUSCRITAS ou ESCANEADAS também **não serão corrigidas**.
 2. Como a avaliação à distância é individual, caso sejam constatadas semelhanças entre provas de alunos distintos, **será atribuída a nota ZERO** a TODAS as provas envolvidas. As soluções para as questões podem ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser distinta.
 3. Além disso, às questões desta AD respondidas de maneira muito semelhantes às respostas oriundas dos gabaritos já publicados de ADs e APs de períodos anteriores, **será atribuída a nota ZERO**, incluindo também cópias diretas, indiretas (semelhanças/paráfrases) ou sem sentido de tópicos dos slides das aulas. A AD é uma atividade de pesquisa (trabalho da disciplina) e deve ser elaborada como tal, não se atendo somente ao conteúdo dos slides das aulas.
 4. Por fim, a pesquisa na Internet e em livros é estimulada, devendo ser referenciada na AD, mas as respostas devem ser construídas com as palavras do próprio aluno e atender diretamente ao que pede à questão, evitando respostas prolixas ou extensas. Às respostas copiadas ou semelhantes a soluções da Internet ou de livros, e/ou que não atendem (fora do escopo) ou excedem demasiadamente ao que pede a questão, **será atribuída a nota ZERO**.
-

Questão 1 [10 pontos]

Considere a situação em que você foi contratado por uma pequena empresa que quer criar um serviço de streaming similar ao Netflix. Essa empresa contratou-o para gerar a documentação que viabilizaria a criação do sistema de streaming de filmes chamado **FlixNet**. **FlixNet** no início rodará apenas no browser e possuirá um plano contratual mensal. A empresa propõe começar pequeno, contendo apenas filmes em seu catálogo e um sistema de recomendação de filmes para recomendar possíveis títulos que o usuário deseja assistir de acordo com os seus gostos.

Para utilizar-se do sistema, o usuário precisa assinar um plano mensal que permita-o acessar a plataforma **FlixNet**. Uma vez contratado o serviço, o usuário precisa acessar o **FlixNet** pelo browser e realizar o login com os dados cadastrados. A etapa de login verifica se os dados do usuário estão corretos (nome e senha) e se o plano contratado ainda está vigente. Uma vez autenticado o login, o sistema exibe o catálogo para o usuário, exibindo filmes que sejam relevantes para o usuário de acordo com o sistema de recomendação. Para assistir um filme, o usuário precisa escolher um dos títulos disponíveis pela plataforma e selecionar a opção de “play”, resultando na exibição do filme escolhido. Ao término do filme, o sistema irá exibir uma tela de avaliação referente ao filme exibido para que o usuário dê uma nota. Essa nota é utilizada pelo sistema de recomendação para customizar o catálogo do usuário. Caso o usuário não queira, ele pode selecionar a opção para realizar a avaliação em um outro momento. Porém, a empresa responsável pela **FlixNet** deseja testar uma segurança para evitar avaliações falsas. Desta forma, filmes só podem ser avaliados pelo usuário caso o mesmo tenha assistido pelo menos alguma parte do filme. Além disso, o usuário pode adicionar filmes na categoria de favoritos. Por fim, o usuário pode cancelar o contrato com a **FlixNet** a qualquer momento.

Dado o diagrama de classes abaixo, que representa o atual esboço da **FlixNet** para o desenvolvimento do sistema de streaming, faça:



- [2.0 pontos] Modifique o diagrama de classes do enunciado para alterar o serviço original de tal forma que inclua os diferentes tipos de planos contratuais: *Mensal*, *Semestral* e *Anual*. Justifique citando o padrão GRASP utilizado para realizar essa alteração.
- [2.0 pontos] Altere o diagrama de classes do item (a) para assegurar que exista apenas uma única instancia da classe *FlixNet*. Inclua uma nota no diagrama sobre o funcionamento do método para utilizar a instancia. Justifique as alterações do diagrama citando o padrão GoF utilizado.

- c) [2.0 pontos] Suponha que objetos precisam ser capazes de referenciar um *Filme* em particular que esteja no *Catalogo*, dado seu nome. Qual classe consegue identificar o objeto *Filme* desejado através do nome? Justifique citando um padrão GRASP.
- d) [2.0 pontos] A empresa da *FlixNet* quer disponibilizar o serviço em duas plataformas: Browser e Android. Para isso, faça alterações no diagrama de classes para que a interface da aplicação (atual classe "*InterfaceBrowser*") crie os botões da interface (classe "*Botao*") para cada plataforma. Além disso, renomeie o nome da classe para fazer mais sentido ao novo contexto (i.e., "*InterfaceFlixNet*"). Justifique a alteração citando o padrão GoF utilizado.
- e) [2.0 pontos] Evolua o diagrama de classes gerado nos itens anteriores referente a *FlixNet* para permitir que o sistema suporte formas de pagamento em *Cartão de Crédito* e *PayPal*. Justifique a alteração citando os padrões GRASP utilizados. Dica: Não se esqueça dos serviços de autorização.

Gabarito:

Questão 1

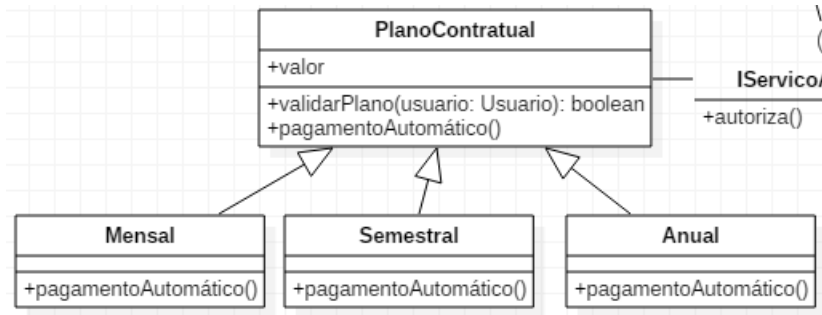
A) 2.0 pontos

0.5 pontos: Padrão *Polymorphism*

0.5 pontos: Classes *Mensal*, *Semestral*, *Anual*

0.5 pontos: Herança das classes criadas

0.5 pontos: Overload do método *pagamentoAutomático*



B) 2.0 pontos

0.5 pontos: pela criação do atributo "*instancia*" privado (0.25) e estático (0.25).

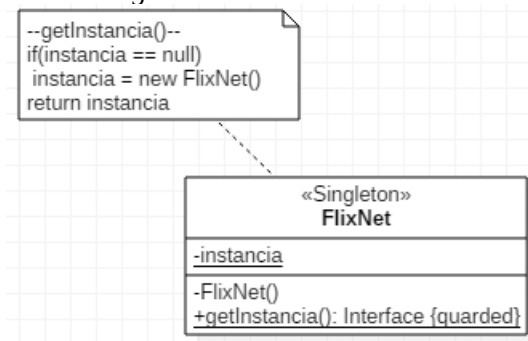
0.25 pontos: pela criação do construtor privado.

0.25 pontos: pela criação do método "*getInstancia()*" (0.2) e por ser `synchronized` (0.05).

0.5 pontos: pela adição da nota referente ao método "*getInstancia()*" (só se tiver criado o método).

0.5 pontos: padrão *Singleton*.

Padrão *Singleton*.



C) 2.0 pontos

1.0 pontos: Padrão *Information Expert*

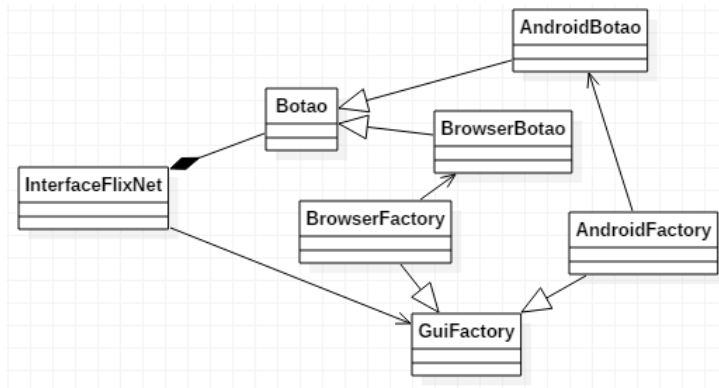
1.0 pontos: Classe *Catalogo*

Como a classe *Catalogo* agrega todos os *Filmes* que estão no catálogo, então pelo padrão *Information Expert*, a classe *Catalogo* possui as informações necessárias para cumprir essa responsabilidade.

D) 2.0 pontos

1.0 pontos: Padrão Abstract Factory
1.0 pontos: Classes

Padrão Abstract Factory



E) 2.0 pontos
1.0 pontos: Padrão Indirection
1.0 pontos: Classes e Interface

Padrão Indirection

