



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 2018.

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**.
-

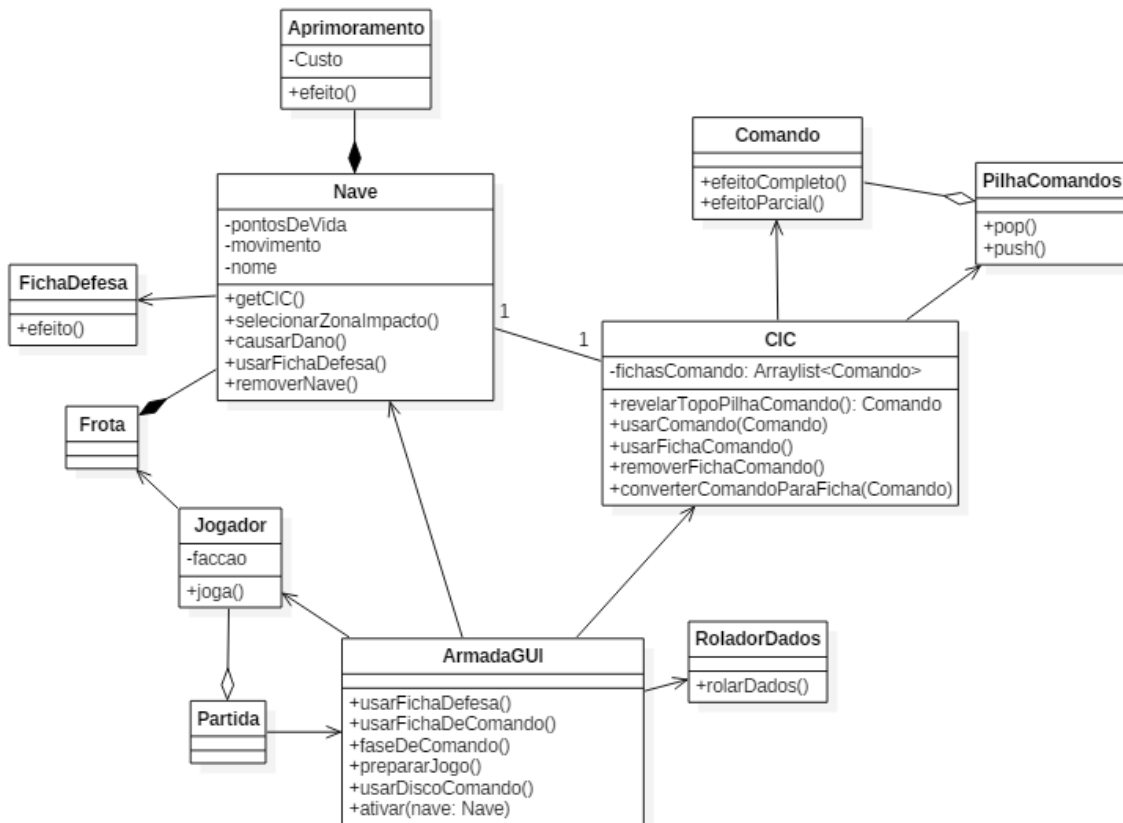
Questão 1 [10 pontos]

Considere a situação em que você foi contratado pela empresa *Fantasy Flight Games* para gerar a documentação que viabilizaria a adaptação do jogo **Star Wars: Armada** para as plataformas Windows e MacOS. O jogo **Star Wars: Armada** é um jogo de tabuleiro baseado em turnos que simulam batalhas espaciais entre dois jogadores. Cada jogador assume o papel de almirante da frota, servindo com a Marinha Imperial ou a Aliança Rebelde. É o trabalho do jogador emitir os comandos táticos que decidirão o curso da batalha e derrotar as naves do jogador adversário. A qualquer momento o jogador poderá desistir da partida, renunciando o jogo. Caso o jogador saia do jogo sem antes desistir, então automaticamente contará que o jogador desistiu da partida ao fechar o jogo. No momento, a *Fantasy Flight Games* não quer que seja possível salvar o jogo para continuar posteriormente.

Cada jogador precisa escolher a facção em que deseja jogar e montar a sua frota de naves para combater o adversário. Cada nave da frota pode ter aprimoramentos diferentes que melhoram o seu desempenho durante o jogo. Esses aprimoramentos podem ser de quatro tipos: Comandante, Oficial, Defensivo e Armamento. Todos os aprimoramentos têm um custo associado que aumenta o custo total da nave para mantê-la na frota e um efeito de acordo com o tipo do aprimoramento.

Para mais informações sobre o jogo, acesse: [http://starwars-armada.wikia.com/wiki/Star Wars: Armada Wiki](http://starwars-armada.wikia.com/wiki/Star_Wars:_Armada_Wiki).

Dado o diagrama de classes abaixo que representa o atual esboço da *Fantasy Flight Games* para o desenvolvimento do jogo, faça:



- a) [2.0 pontos] Altere o diagrama de classes do enunciado para alterar o jogo original de tal forma que inclua os diferentes tipos de aprimoramentos nas naves: *Comandante*, *Defensivo*, *Oficial* e *Armamento*. Justifique citando o padrão GRASP utilizado para realizar essa alteração.
- b) [2.0 pontos] Altere o diagrama de classes do item (a) para assegurar que exista apenas uma única instancia da classe *ArmadaGUI*. 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 uma *Nave* em particular, dado seu nome. Qual classe consegue identificar o objeto *Nave* através do nome? Justifique citando um padrão GRASP.
- d) [2.0 pontos] Altere o diagrama de classes para criar uma classe alternativa de *Jogador* para suportar um usuário do tipo Inteligência Artificial e, além disso, permita que a *Inteligência Artificial* utilize diferentes estratégias de acordo com o valor do seu atributo privado dificuldade que representa o nível de dificuldade selecionado para aquela partida. As estratégias são heurísticas que a IA utiliza para planejar seus movimentos. Justifique citando o padrão de comportamento utilizado para realizar a alteração. Além disso, utilize como exemplos o *MinMax* (*MinMax*), *MinMax* com *Alpha-Beta-Pruning* (*AlphaBetaPruning*) e *MinMax* com *Alpha-Beta-Pruning* usando a heurística *Null-Move* (*NullMove*).
- e) [2.0 pontos] Com base no diagrama de classes do enunciado, qual classe é responsável pela criação de objetos do tipo *Aprimoramento*? Justifique citando o padrão GRASP utilizado e elabore um diagrama de sequência que demonstra essa relação de responsabilidade.

Gabarito:

Questão 1

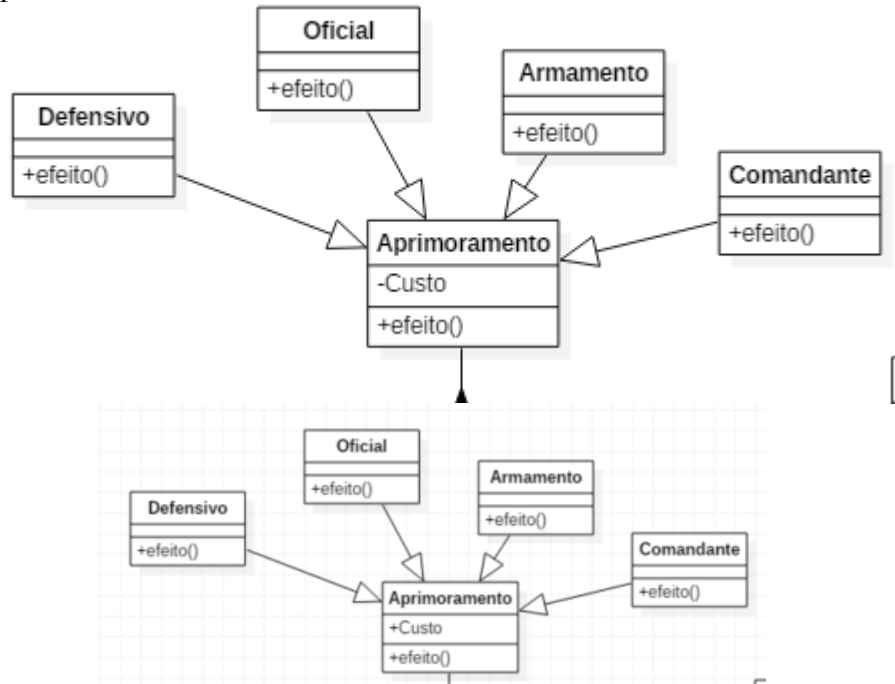
a) 2.0 pontos

0.5 pontos: Padrão Polymorphism

0.5 pontos: Classes Defensivo, Oficial, Armamento, Comandante

0.5 pontos: Herança das classes criadas

0.5 pontos: Overload do método Efeito



b) 2.0 pontos

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

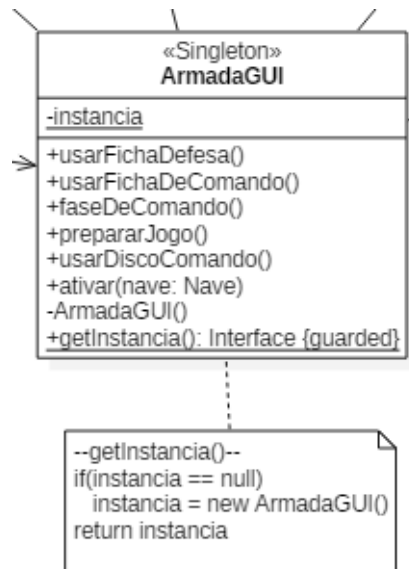
0.25 pontos: pela criação do construtor privado.

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 pelo padrão *Information Expert*

1.0 pontos pela responsabilidade do Frota (apenas se acertou o padrão).

Resposta: Como a classe *Frota* agrega todas as Naves, então pelo padrão *Information Expert*, a classe *Frota* possui as informações necessárias para cumprir essa responsabilidade.

d) 2.0 pontos

0.5 por ter criado a classe que representa os possíveis algoritmos de cálculo (*AlgoritmoBusca*).

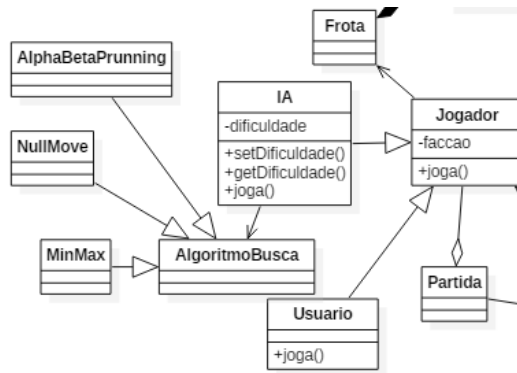
0.5 pela herança dos algoritmos (MinMax, AlphaBetaPruning, NullMove) a classe *AlgoritmoBusca*.

0.3 pontos por ter criado a classe IA e Usuário, diferenciando os jogadores.

0.2 pontos por ter o atributo privado dificuldade na classe IA.

0.5 pontos pelo padrão *Strategy* (apenas se o diagrama for coerente ao padrão).

Padrão *Strategy*.



e) 2.0 pontos

0.75 pontos por ser a classe Nave.

0.75 pontos diagrama de sequencia.

0.5 pontos padrão Creator (apenas se o diagrama for coerente ao padrão).

