Nome:_____ **Nº:**_____

— Teste —
Desenvolvimento de Sistemas Software

MiEI 2019/20

08/01/2018

Duração máxima: 2h00 — **Versão A**

Leia o teste com atenção e responda nas folhas do teste!

Grupo I

Considere que se pretende desenvolver uma aplicação para uma loja de jogos, a DSSGamer. A aplicação deverá suportar, quer o registo de jogos, por programadores, quer a sua compra ou aluguer por jogadores (desde que sejam adequados à sua faixa etária). Para efeito de validação da faixa etária, para cada jogador é registado o seu nome, morada e data de nascimento, bem como meios de pagamento.

Por cada compra ou aluguer, a loja recebe uma comissão (uma percentagem sobre o valor da transação). O valor das comissões é fixado pelo gestor da loja e varia em função de se tratar de uma compra ou de um aluguer. Os alugueres têm uma duração que varia de jogo para jogo. Um utilizador pode emprestar jogos comprados, mas apenas a utilizadores que constem da sua lista de amigos. Quando o jogo é emprestado, não pode ser jogado até ser devolvido. Jogos alugados não podem ser emprestados.

Os programadores também se devem registar na plataforma, sendo guardado o seu nome, morada e detalhes bancários. Quando registam um jogo na plataforma, devem indicar se pretendem que o jogo seja vendido e/ou alugado e os respectivos preços. Deverá ainda indicar o limite de idade para o jogo. Os valores possíveis estão pré-definidos na plataforma. Pode ainda optar por disponibilizar o jogo gratuitamente. Nesse caso, a taxa a pagar é um valor fixo, também definido pelo gestor da loja.

Quando um programador submete um novo jogo na loja, este fica a aguardar validação. O jogo pode então ser aprovado, caso em que fica disponível na loja, ou rejeitado. Neste último caso o jogo fica no estado rejeitado até que o programador submeta uma nova versão, ou decida retirá-lo, caso em que é eliminado da plataforma. Os jogos disponíveis podem ser transacionados (comprados/alugados). Por norma, os jogos são transacionados ao seu preço normal, no entanto, podem também ser colocados em promoção. Por decisão da loja, as promoções duram no máximo uma semana. Se após esse tempo o jogo não tiver sido recolocado a preço normal, tal é feito automaticamente. Os jogos podem ainda ser denunciados. Nesse caso, ficam suspensos, não podendo ser transacionados, até serem novamente aprovados. Enquanto o jogo estiver suspenso, o programador pode submeter uma nova versão para aprovação, ou remover o jogo. Aliás, a plataforma permite que os jogos disponíveis sejam removidos a qualquer momento.

Rascunho:

Nome:_____ **Nº:**_____

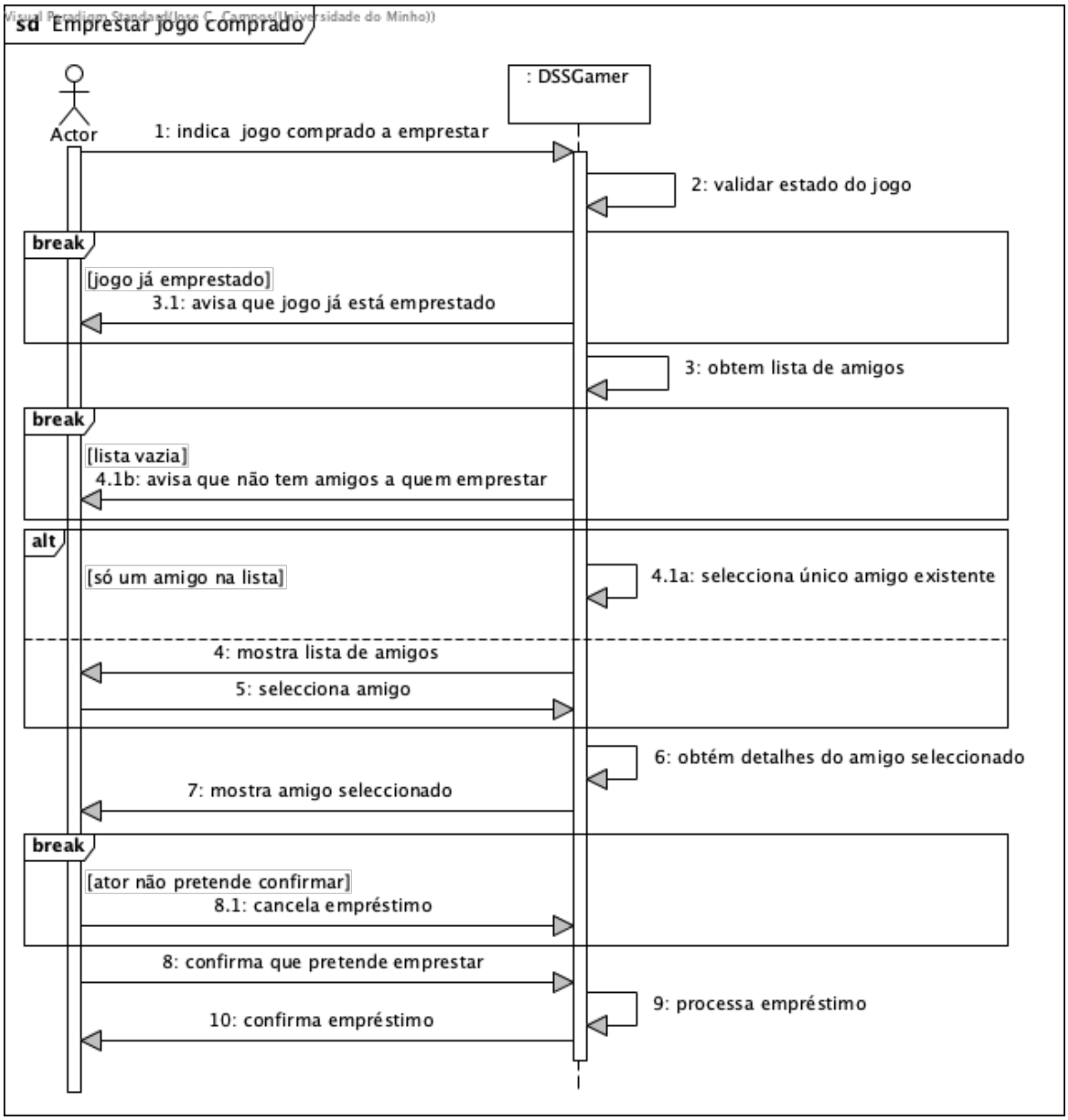
1. Tendo em conta a descrição acima, desenvolva o **Modelo de Domínio** que, na sua opinião, permite capturar a informação relevante para se perceber o contexto do problema. (4 valores)

Resposta:

Rascunho:

Nome:_____ **Nº:**_____

2. **Especifique**, utilizando o formato textual, **o Use Case "Emprestar jogo comprado"**, apresentado no Diagrama de Sequência de Sistema da figura abaixo. (3 valores)



Rascunho:

Nome:_____ **Nº:**_____

Resposta:

Rascunho:

3. **Desenhe agora o Diagrama de Máquina de Estado** que permite descrever o ciclo de vida de um jogo na loja, tal como descrito no último parágrafo da página 1. (3 valores)

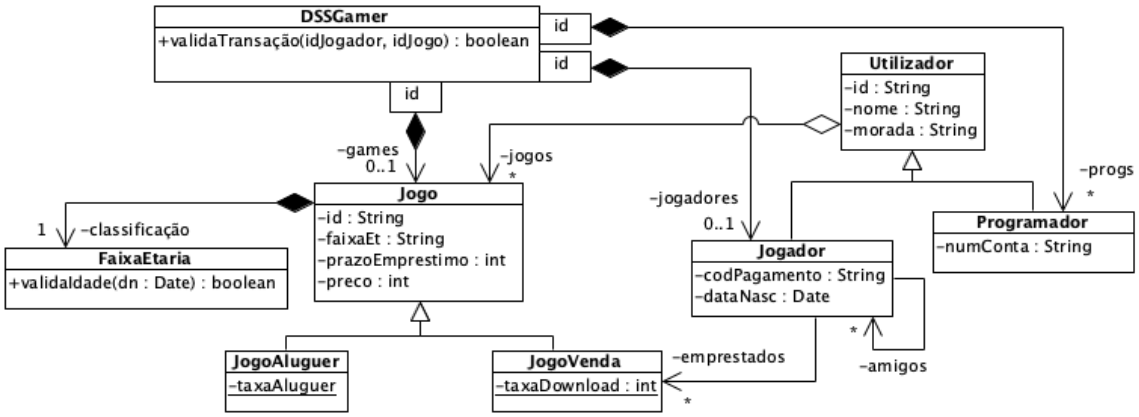
Resposta:

Rascunho:

Nome:_____ **Nº:**_____

Grupo II

Considere o seguinte excerto de uma proposta de Diagrama de Classes para o sistema:



4. Sabendo que se pretende **implementar** a arquitectura acima recorrendo a uma Base de Dados, e a uma estratégia de uma tabela por classe, para assegurar a **persistência** da informação, **reformule o Diagrama de Classes**, indicando ainda as tabelas utilizadas. (3 valores) **Justifique**, de forma breve, as alterações que efectuou na arquitectura. (1 valor).

Resposta:

Rascunho:

Nome:_____ **Nº:**_____

Resposta (cont.):

Rascunho:

5. Considerando novamente o diagrama de classes apresentado acima, desenhe o **Diagrama de Sequência** para a operação

   `+validaTransação(idJogador:String, idJogo:String): Boolean`

   da classe `DSSGamer`, que verifica se um dado jogador tem idade para jogar um jogo, dados os identificadores de ambos. (2 valores)

Resposta:

Rascunho:

Nome:_____ **Nº:**_____

6. Responda às seguintes questões, preenchendo os círculos para assinalar **todas** as respostas correctas$^{(2\ \text{valores})}$:

   (a) A UML é...

   ◯ um processo de desenvolvimento.
   ◯ definida numa norma OMG.
   ◯ pensada especificamente para a programação em Java.

   (b) Um Modelo de Domínio...

   ◯ é um tipo de diagrama UML.
   ◯ pode ser descrito com um Diagrama de Classes.
   ◯ caracteriza as entidades de um dado problema.

   (c) A comunicação entre objectos pode ser modelada com...

   ◯ Diagramas de Sequência
   ◯ Diagramas de Use Case
   ◯ Diagramas de Componentes

   (d) a expressão OCL:

   ```
   context Jogador
   inv: emprestados->size()>0 implies not amigos->isEmpty()
   ```

   relativa ao diagrama de classes apresentado na página 7 do teste, expressa o invariante:

   ◯ quem tem jogos emprestados, pode ter amigos.
   ◯ só quem tem amigos pode ter jogos emprestados.
   ◯ quem tem jogos emprestados, não pode ter amigos.

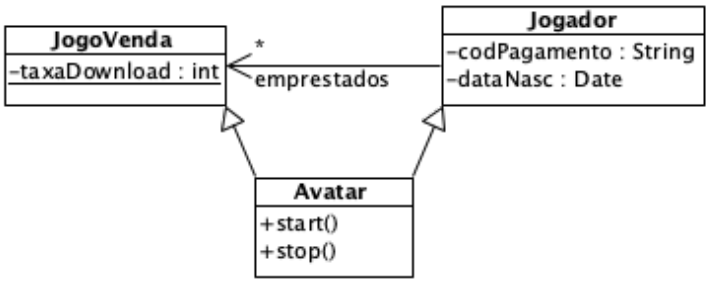
   (e) Os DAOs...

   ◯ encapsulam o acesso à Base de Dados.
   ◯ implementam persistência de dados.
   ◯ criam o problema do ORM.

Rascunho:

Nome:_____ **Nº:**_____

7. Considere agora que se pretende implementar a entidade *Avatar*, que permitirá aos utilizadores automatizar algumas tarefas em jogos, deixando um Avatar responsável pela sua execução (por exemplo, proceder a colheitas quando estas estiverem prontas). Os Avatares deverão ser vistos, quer como jogadores, quer como jogos que podem ser comprados, tendo sido desenvolvida a seguinte solução:



Proponha uma solução arquitetural que possa ser implementada em Java (2 valores)

Resposta:

Rascunho: