



Fundação CECIERJ - Vice Presidência de Educação Superior a
Distância **Curso de Tecnologia em Sistemas de
Computação Disciplina: Engenharia de Software**
Gabarito AD2 2º semestre de 2014.

Atenção: Como a AD é individual, caso seja constatado que provas de alunos distintos são cópias uma das outras, independentemente de qualquer motivo, a todas será atribuída nota ZERO. As soluções para as questões podem sim, ser buscadas por grupos de alunos, mas a redação final das respostas para as questões da prova tem que ser individual!

1. Modelagem em UML: Considerando o problema da loja de venda de móveis abaixo, apresente (7,0):
 - a. a descrição do caso de uso do requisito **4** (1,0 ponto)

Nome: Emitir pedido de compra

Ator: Almoхарife

Trigger: Móvel não existe no estoque e ator seleciona opção emitir pedido de compra

Fluxo Principal:

1 - Sistema apresenta interface de listagem de móveis com as seguintes opções:

- Nome
- Fornecedor
- Comprar
- Voltar

2- Ator selecionar a opção comprar **[A01]**

3- Sistema apresenta interface de emissão de pedido de compra com as seguintes opções:

- Móvel (Somente leitura)
- Fornecedor (Somente leitura)
- Preço última compra (Somente leitura)
- Quantidade
- Confirmar

- Cancelar

4- Ator seleciona a opção Confirmar

5- Sistema emite o pedido de compra e exibe a mensagem “Pedido de compra realizado com sucesso”

Fluxo A01: Ator seleciona a opção Voltar:

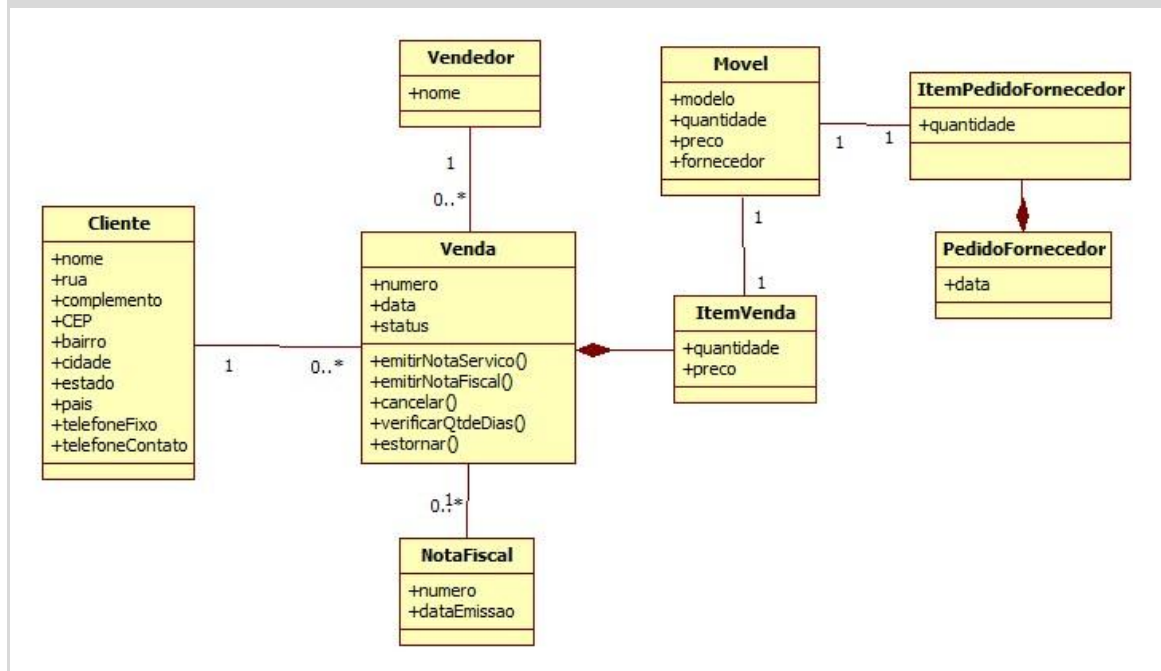
A01.1- Sistema retorna para o passo 1 do fluxo principal

Fluxo A02: Ator seleciona a opção Cancelar

A02.2- Sistema retorna para o passo 1 do fluxo principal

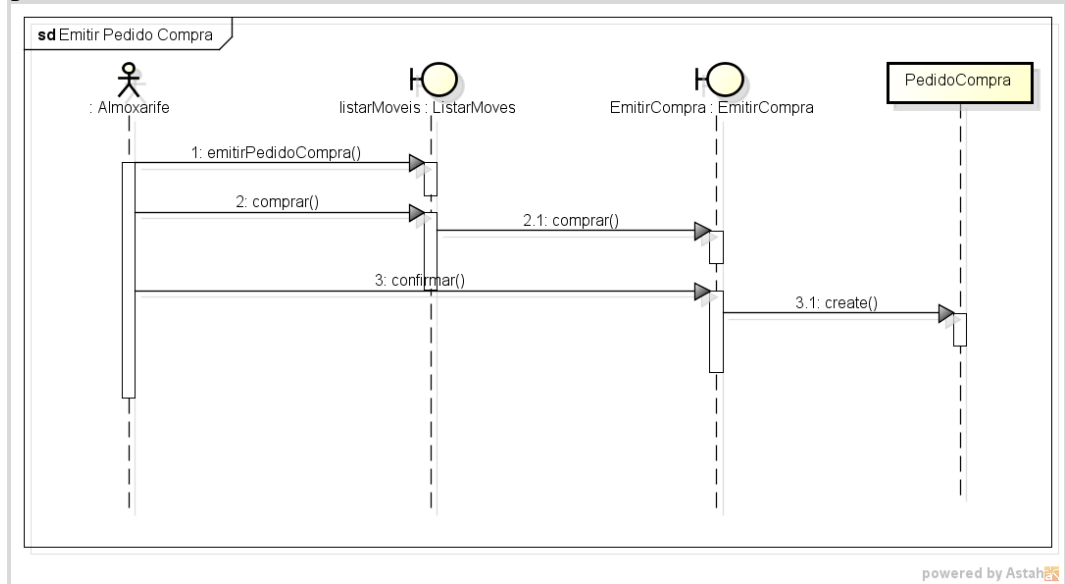
b. o diagrama de classes para o sistema (1,5 pontos)

Existem diversas possibilidades de construção de diagramas de classes que podem atender aos requisitos especificados. A seguir, segue o exemplo de uma dessas possibilidades.



c. o diagrama de sequência para o caso de uso descrito em a) (1,0 ponto)

A seguir, segue o exemplo de uma das possibilidades de diagrama de sequência para o caso de uso descrito



- d. Indique que técnicas de inspeção podem ser usadas para revisar estes diagramas, descrevendo os critérios que você usou para escolher e os tipos de defeito que podem ser encontrados. Dê um exemplo aplicando a técnica nos diagramas que você criou (1,0 ponto)

Poderiam ser utilizadas OORT's (Técnicas de Leitura Orientada a objetos). Pois elas apoiam a inspeção de projetos orientados a objetos. Essas técnicas podem revelar defeitos de omissão e ambiguidade, por exemplo.

Um exemplo de aplicação seria verificar se todos os métodos utilizados no diagrama de sequência estão presentes no diagrama de classes (OORT's – Leitura Horizontal v3.0 – Leitura 1 – Diagramas de Sequência x Classes).

Outras técnicas de leitura como ad-hoc (leitura de acordo com a experiência do desenvolvedor) ou um checklist preparado para modelos de projeto também poderiam ser utilizadas.

- e. Apresente o conjunto de casos de testes para realizar o teste do caso de uso que você descreveu em a) e modelou em c), indicando qual abordagem foi utilizada para gerar esta informação (1,5 pontos)

Apesar de não termos acesso ao código fonte do sistema em questão, nós temos acesso a descrição dos casos de uso. Por este motivo, podemos utilizar algum critério da técnica funcional.

Por exemplo, poderia ser definido um critério onde todos os fluxos do caso de uso fossem executados. Incluindo necessariamente as sequências alternativas de ações (fluxos alternativos). A partir de alguma abordagem específica (particionamento por equivalência, análise valor-limite ou grafo de causa-efeito) poderia ser aplicada para

identificar os casos de teste válidos e inválidos que deveriam ser usados para testar o UC.

- f. Utilize as métricas NOC, DIT e CBO para identificar as classes que poderiam ser mais propensas à falha no seu projeto. Indique as classes e o motivo desta indicação (1,0 ponto)

	NOC	DIT	CBO
Cliente	0	0	1
Vendedor	0	0	1
Venda	0	0	4
NotaFiscal	0	0	1
Movel	0	0	2
ItemVenda	0	0	2
ItemPedidoFornecedor	0	0	2
PedidoFornecedor	0	0	1

A classe Venda é mais indicada para ser testada, pois possui maior CBO. Portanto qualquer problema nessa classe implica em possível efeito colateral em outros 5 pontos do sistema.

Nenhuma das classes possui valores para as métricas NOC e DIT, pois não há herança no diagrama de classes.

- 1) O cliente telefona para a loja e informa que deseja comprar um móvel. O vendedor pergunta sobre a quantidade desejada, informando o preço unitário do móvel desejado ao cliente;
- 2) Se o cliente confirmar a compra, o funcionário verifica se ele já tem cadastro na loja. Caso não tenha cadastro, o funcionário pergunta o nome completo do cliente, seu endereço completo (rua, complemento, CEP, bairro, cidade, estado e país), telefone fixo e telefone de contato;
- 3) Cadastrado o cliente e confirmada a compra, o vendedor emite a nota de serviço, que indica os dados do cliente, o nome do vendedor, a data da compra, o mobiliário desejado (com suas respectivas quantidades e preços). A nota é encaminhada ao almoxarifado da loja;
- 4) Ao receber uma nota de serviço, o almoxarife verifica a existência de estoque para os produtos desejados. Se não existir estoque para algum móvel, o almoxarife emite um pedido de compra para o fornecedor do móvel, comprando sempre duas unidades além do que seria necessário para atender ao pedido do cliente;
- 5) O cliente pode cancelar a compra até dois dias depois de realizada. Se uma compra for cancelada, os produtos voltam para o estoque e os pedidos de compra que tiverem sido emitidos para estes produtos são cancelados junto ao fornecedor;
- 6) Quando os produtos estiverem disponíveis, o almoxarife emite a nota fiscal e encaminha os produtos para entrega ao cliente. A nota fiscal possui os mesmos dados da nota de serviço, além de indicar seu número e a data em que foi emitida.

2. Explique a diferença entre as linguagens de programação interpretadas e as linguagens de programação compiladas. Para que tipos de projetos devem ser consideradas. (valor 1,0 ponto)

Nas linguagens compiladas, o código-fonte passa por uma etapa de compilação, na qual é transformado em código-objeto (código compreensível pelo computador). Em seguida, diferentes unidades de compilação passam por um processo de linkagem, em que referências entre elas são resolvidas para transformar o conjunto em um programa executável.

Nas linguagens interpretadas, o código-fonte é pré-compilado e transformado em uma linguagem intermediária, que é posteriormente executada por um interpretador, ou seja, uma máquina de execução independente da arquitetura do computador que está sendo usado para rodar o programa.

3. Explique o que é uma inversão de comando na codificação de um projeto de software (valor: 1,0 ponto; máximo: 5 linhas).

A inversão de comando ocorre quando uma rotina recebe um parâmetro de controle, como um flag, por exemplo, que determina como a rotina deve agir. Dizemos que é uma inversão de comando porque um agente externo, que chama a rotina, determina como a rotina deve atuar, quebrando sua independência. A inversão de comando geralmente dificulta o entendimento da rotina e do sistema.

4. Ao discutirmos os princípios de projeto de software, vimos um ciclo onde requisitos geram a demanda por mais software, que entra em operação, provoca mudanças no ambiente de trabalho, mudanças estas que acabam gerando novos requisitos. Explique porque este ciclo ocorre e como normalmente é encerrado. (Valor: 1,0 ponto; máximo: 10 linhas)

Um software normalmente é construído de acordo com os requisitos de seus futuros usuários. No entanto, quando o software entra em operação, ele altera o ambiente de trabalho, mudando a forma com que seus usuários trabalham. Desta forma, o software geralmente precisa ser adaptado para a nova realidade, da qual ele participou na formação. Esta é uma das razões pela qual um software usado está em constante alteração, sendo ajustado a medida que os processos que ele suporta evoluem.