



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina de Arquitetura e Projeto de Sistemas II**  
**Gabarito – AD1 2º semestre de 2009.**

**Nome –**

---

Observações:

1. Prova com consulta.

Atenção: 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. ALÉM DISSO, às questões desta AD respondidas de maneira muito semelhantes às respostas oriundas dos gabaritos já publicados de ADs de períodos anteriores, será atribuída a nota ZERO, incluindo também cópias diretas e sem sentido de tópicos dos slides das aulas.

---

Questão 1 [2 pontos]

Cite quais são as disciplinas do Processo Unificado e explique três delas. Além disso, responda: qual a relação entre as disciplinas e as fases do Processo Unificado?

***Resposta:***

O Processo Unificado (PU) descreve atividades de trabalho (e.g., redigir um caso de uso) dentro de disciplinas, que representam um conjunto de atividades (e artefatos relacionados) em uma área alvo (e.g., atividades dentro da análise de requisitos). No PU, um artefato é o termo usado para qualquer produto de trabalho: código, gráficos para a *web*, esquemas de banco de dados, documentos em texto, diagramas, modelos etc. Existem várias disciplinas, dentre elas: (i) *Modelagem de Negócio*: o artefato Modelo do Domínio, para visualizar conceitos importantes no domínio de aplicação; (ii) *Requisitos*: os artefatos Modelo de Casos de Uso e Especificação Suplementar, para captar requisitos funcionais e não funcionais; e (iii) *Projeto*: o artefato Modelo de Projeto para projetar o objetos de software.

Para explicar a relação entre as disciplinas e as fases do PU, recordamos que o PU organiza o trabalho e as iterações em quatro fases principais (i.e., concepção, elaboração,

construção e transição). Durante uma iteração, o trabalho prossegue na maioria ou em todas as disciplinas. Contudo, o esforço relativo no decorrer destas disciplinas muda ao longo do tempo. As iterações iniciais naturalmente tendem a dar uma ênfase maior aos requisitos e ao projeto, enquanto que as últimas disciplinas dão a esses itens uma ênfase menor, à medida que os requisitos e o projeto central se estabilizam por meio de um processo de realimentação e adaptação.

## Questão 2 [4 pontos]

A partir do Diagrama de Casos de Uso apresentado na Figura 1, correspondente ao Sistema de Pedido de Vendas, responda:

- [1 ponto] Escolha um dos caso de uso apresentados e realize a sua descrição, conforme o *template* da Tabela 1;
- [1 ponto] A partir do caso de uso escolhido, construa o(s) Diagrama(s) de Seqüência(s) correspondente(s);
- [1 ponto] Construa o Modelo de Classes conceitual (artefato resultante da fase *Análise Orientada a Objetos*), mantendo-o coerente com as entidades utilizadas no Diagrama de Seqüência (b) e com as funcionalidades apresentadas no Diagrama de Casos de Uso (a);
- [1 ponto] Considerando que o Diagrama de Interação é uma generalização de dois tipos de diagrama especializados da UML, o diagrama de seqüência e o diagrama de colaboração (ou de comunicação), explique o conceito de cada um deles e discuta os pontos fortes e fracos de cada um destes diagramas.

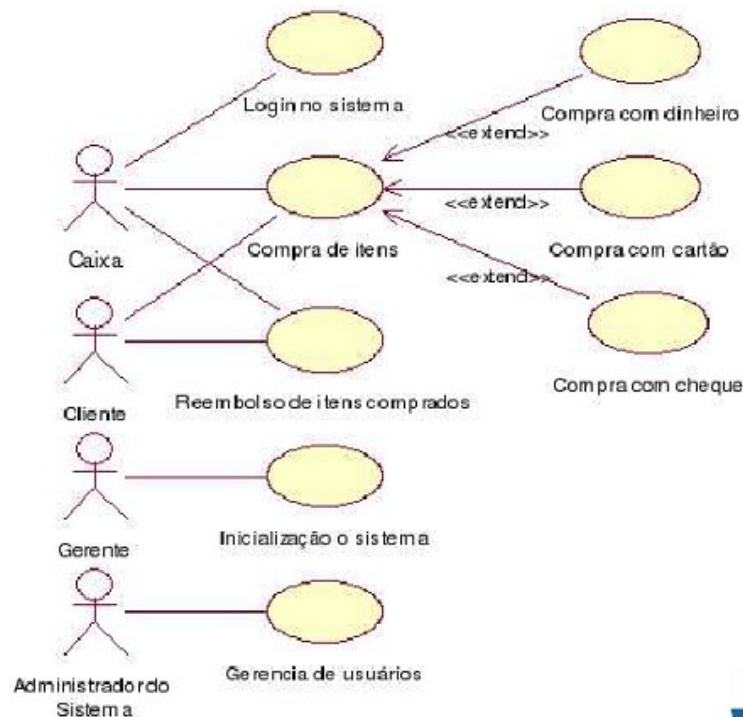


Figura 1 – Diagrama de Casos de Uso do Sistema de Pedido de Vendas

**Tabela 1 – Template para Descrição de Casos de Uso**

<b>Nome:</b>	<definir o nome do caso de uso>
<b>Objetivo:</b>	<descrever o objetivo do caso de uso>
<b>Atores:</b>	<descrever os atores que interagem com o caso de uso>
<b>Pré-condições:</b>	<descrever as pré-condições a serem atendidas para que o caso de uso possa ser executado>
<b>Trigger:</b>	<definir que evento dispara a execução desse caso de uso>
<b>Fluxo Principal:</b>	<descrever o fluxo principal do caso de uso>
<b>Fluxo Alternativo:</b>	<descrever os fluxos alternativos do caso de uso, indicando que evento dispara cada um deles. Cada fluxo deve ser nomeado A1, A2 etc.>
<b>Extensões:</b>	<definir que extensões podem ser executadas>
<b>Pós-condições:</b>	<definir que produto ou resultado concreto o ator principal obterá ao final da execução do fluxo básico>
<b>Regras de negócio:</b>	<listar as regras de negócios que devem ser respeitadas na execução do caso de uso. Cada regra deve ser nomeada RN1, RN2 etc., e ser referenciada em algum fluxo do caso de uso (básico ou alternativo)>

### **Resposta:**

#### a) Descrição do Caso de Uso escolhido *Compra Itens*<sup>1</sup>:

<b>Nome:</b>	Compra Itens.
<b>Objetivo:</b>	O sistema deve permitir ao cliente efetuar suas compras através de um pedido de venda criado por um caixa, de maneira que todos os produtos sejam registrados pelo caixa para que ocorra a sua saída do estabelecimento comercial. Ao final deste processo, sistema deve permitir o pagamento do pedido pelo cliente.
<b>Atores:</b>	Cliente e Caixa.
<b>Pré-condições:</b>	Os produtos pertencentes ao pedido do cliente devem estar cadastrados previamente no sistema.
<b>Trigger:</b>	O caixa cria um novo pedido de venda para o cliente.
<b>Fluxo Principal:</b>	<ol style="list-style-type: none"> <li>1. O caixa aciona a opção “criar um novo pedido de venda” para um determinado cliente.</li> <li>2. O sistema aciona o módulo “comprar itens”.</li> <li>3. Para cada novo item a ser adicionado no pedido do cliente <ol style="list-style-type: none"> <li>3.1. O caixa registra o item de pedido através da consulta ao produto em questão e do fornecimento da informação relativa à quantidade de itens comprada pelo cliente. [A1]</li> <li>3.2. O sistema contabiliza o item de pedido e o imprime na nota fiscal. [A2]</li> </ol> </li> <li>4. O caixa aciona a opção “terminar pedido de venda”.</li> <li>5. O sistema encerra o pedido de venda e sumariza o valor devido pelo</li> </ol>

<sup>1</sup> O aluno poderá ter escolhido também qualquer um dos demais casos de uso: *Login no sistema*, *Reembolso de itens comprados*, *Inicialização do sistema* ou *Gerência de usuário*, bem como o detalhamento dos casos de uso *Compra com dinheiro*, *Compra com cartão* ou *Compra com cheque*.

cliente.

6. O caixa registra o pagamento do pedido de acordo com a forma de pagamento requerida pelo cliente e permitida pelo sistema. [A3]
7. O sistema encerra o pedido.

**Fluxo Alternativo:**

[A1] Erro no processamento do código de barras.

1. O sistema exibe a mensagem de alerta “Erro no processamento do item de pedido”
2. O caixa efetua a operação novamente (passo 3).
3. Caso o produto não esteja cadastrado no sistema, o caixa deve acionar o gerente.

[A2] Erro na impressão da nota fiscal.

1. O sistema exibe a mensagem de alerta “Erro na impressão da nota fiscal”
2. O caixa verifica o dispositivo e continua a efetuar a operação (passo 3).

[A3] Erro no processamento da forma de pagamento.

1. O sistema exibe a mensagem de alerta “Forma de pagamento indisponível ou não permitida”
2. O caixa informa ao cliente e repete a operação (passo 6).
4. Caso haja o problema se repita, o caixa deve acionar o gerente.

**Extensões:**

Compra com dinheiro, Compra com cartão e Compra com cheque.

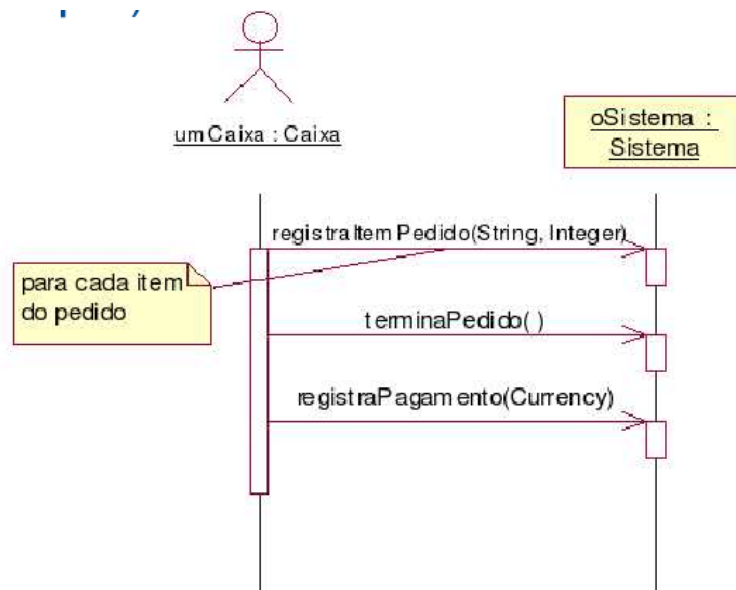
**Pós-condições:**

A venda é finalizada com sucesso.

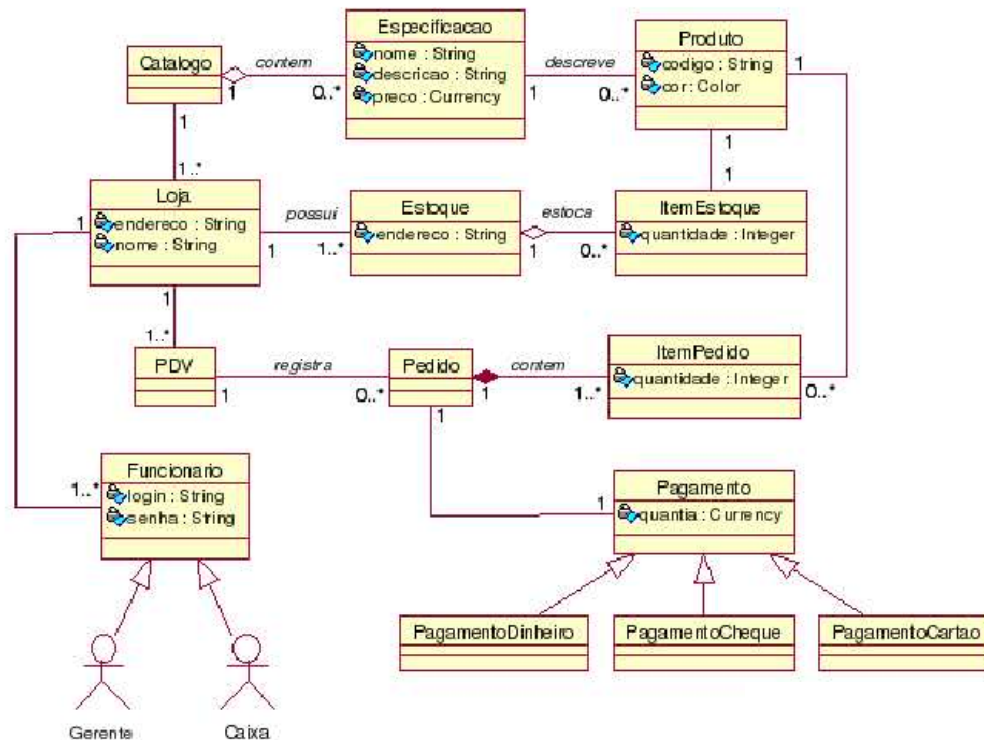
**Regras de negócio:**

RN01 – Um novo pedido deve ser criado para cada compra efetuada por um cliente, de forma que itens de pedido referentes aos produtos previamente cadastrados sejam contabilizados para pagamento pelo cliente.

b) Diagrama de Seqüência do Caso de Uso *Compra Itens*:



c) Modelo de Classes Conceitual:



d) Os Diagramas de Seqüência são diagramas da UML utilizado para representar as seqüências de processos (i.e., mensagens trocadas entre objetos) de um sistema, descrevendo a maneira como os grupos de objetos colaboram em algum comportamento ao longo do tempo. Além disso, estes diagramas ilustram as interações em um formato semelhante a “cercas”, nas quais cada objeto novo é acrescentado à direita. Já os Diagramas de Comunicação exibem interações que consistem de um conjunto de objetos e seus relacionamentos, incluindo as mensagens que podem ser trocadas entre eles, ilustrando essas interações em forma de “grafos” ou “rede”, na qual os objetos podem ser colocados em qualquer lugar do diagrama (a essência da sua vantagem como um “rascunho” para os projetistas). Os pontos fortes e fracos de cada um desses diagramas são exibidos na tabela abaixo:

TIPO	PONTOS FORTES	PONTOS FRACOS
Seqüência	Mostra com clareza a seqüência ou ordem temporal das mensagens.  Ampla conjunto de opções detalhadas.	Deve ser estendido para a direita quando são acrescentados novos objetos; consome espaço na horizontal.
Comunicação	Economia de espaço – flexibilidade de adicionar novos objetos em duas dimensões.	É mais difícil ver a seqüência das mensagens.  Menos opções de notação.

Questão 3 [1 ponto]

Explique o que são estereótipos, perfis e etiquetas em Diagramas de Classe UML.

**Resposta:**

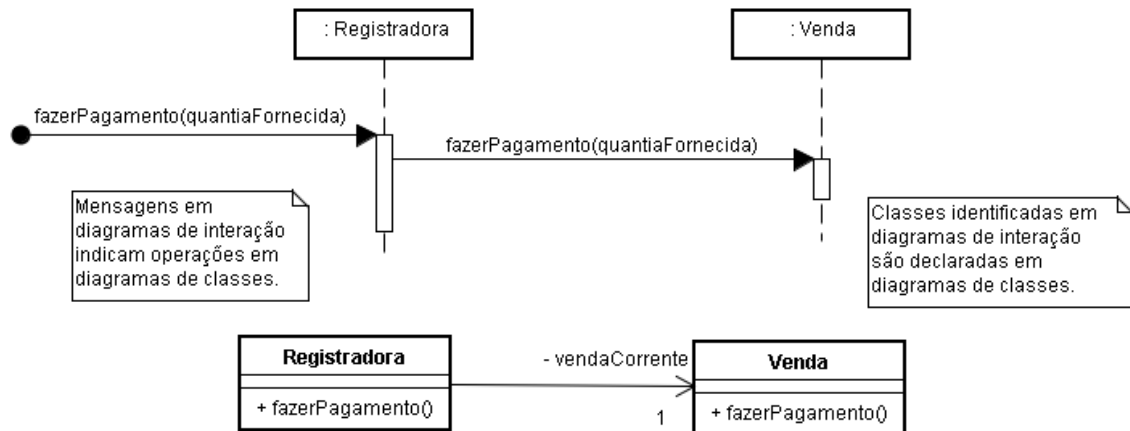
Um *estereótipo*, de forma análoga a uma palavra-chave, é mostrado com símbolos (e.g., <<autoria>>) e representa um refinamento de um conceito de modelagem existente, definido dentro de um *perfil* UML – uma coleção de estereótipos, etiquetas e restrições relacionadas para especializar o uso da UML para um domínio ou plataforma específico (e.g., um perfil UML para gestão de projetos ou para modelagem de dados). A UML predefine muitos estereótipos, como <<destroy>> (usado em diagramas de sequência), e também permite estereótipos definidos pelo usuário, fornecendo, assim, um mecanismo de extensão em UML. Adicionalmente, o estereótipo declara um conjunto de *etiquetas*, usando a sintaxe de atributo – quando um elemento é marcado com um estereótipo, todas as etiquetas se aplicam ao elemento e a elas podem ser atribuídos valores.

Questão 4 [1 ponto]

Qual é o relacionamento entre Diagramas de Interação e de Classes? Mostre a influência detectada entre eles em um exemplo, bem como explicita o sentido desta influência.

**Resposta:**

Quando diagramas de interação são desenhados, um conjunto de classes e seus métodos emergem do processo “criativo” de projeto de modelagem dinâmica de objetos. Por exemplo, ao se começar com o diagrama de sequência `fazerPagamento`, conforme a figura abaixo, pode-se perceber que uma definição das classes `Registradora` e `Venda` em um diagrama de classe podem ser notadamente derivadas. Dessa forma, a partir de diagramas de interação, as definições de diagramas de classe podem ser geradas. Isso sugere uma ordenação linear ao desenhar diagramas de interação antes de diagramas de classe – mas, na prática, especialmente quando se segue a prática de modelagem ágil de modelos em paralelo, essas visões complementares (dinâmica e estática) são desenhadas concorrentemente. Além disso, uma diretriz pode ser extraída desse fato: uma boa ferramenta UML deve apoiar automaticamente modificações em um diagrama, de maneira que esta seja refletida no outro. No “rascunho” feito pelos projetistas de sistemas, pode-se desenhar diagramas de interação ao lado de diagramas de classes para maximizar o desempenho e a veracidade da atividade de modelagem.



### Questão 5 [2 pontos]

Explique detalhadamente o que grau de dependência, bem como suas variações, e relacione este conceito com os tipos de domínios de classes de um sistema. Além disso, calcule o grau de dependência de cada uma das 7 classes apresentadas no Diagrama de Classes da Figura 2.

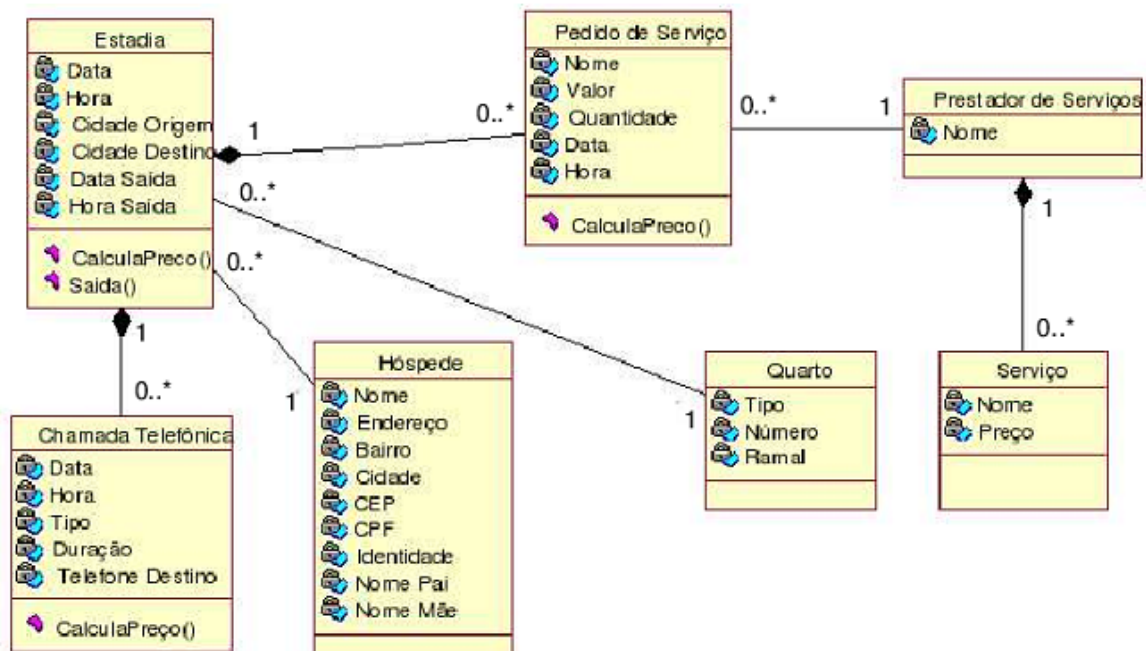


Figura 2 – Diagrama de Classes de um Sistema do Domínio de Hotelaria

### Resposta:

Grau de Dependência (GD) é uma métrica semelhante a Fan-out<sup>2</sup> de projeto estruturado e serve para verificar projetos orientados a objetos. Pode ser classificado em dois tipos: (i)

<sup>2</sup> Indica quantos módulos são acessados por um dado módulo.



*grau de dependência direto*, que indica quantas classes são referenciadas diretamente por uma determinada classe; e (iii) *grau de dependência indireto*, que indica quantas classes são referenciadas direta ou indiretamente (recursivamente) por uma determinada classe. Para identificar o grau de dependência, deve-se verificar se uma classe *A* referencia diretamente outra classe *B*, de maneira que *A* seja subclasse (ou tenha atributo, ou tenha parâmetro do método do tipo, ou tenha variáveis em métodos do tipo, ou ainda chame métodos que retornem valores do tipo) de *B*. Analisando os diferentes tipos de domínios de classes de um sistema, temos que classes de domínios mais altos (negócio e aplicação) tendem a ter alto grau de dependência indireto e as classes de domínios mais baixos (arquitetura e base) tendem a ter baixo grau de dependência indireto.

Cálculo do grau de dependência para cada uma das classes:

CLASSES	GD Direto	GD Indireto
<i>Estadia</i>	4	6
<i>Chamada Telefônica</i>	0	0
<i>Hóspede</i>	1	6
<i>Pedido de Serviço</i>	1	2
<i>Quarto</i>	1	6
<i>Serviço</i>	0	0
<i>Prestador de Serviços</i>	2	2