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 – AP2 2° semestre de 2007.**

**Nome –**

**Assinatura –**

---

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

---

Questão 1 (2,5 pontos)

Considere um sistema de controle de supermercado, onde o caso de uso de "Compra de Itens" tem a seguinte descrição e seqüencia típica de eventos.

**Descrição:**

"Um cliente chega ao caixa com um conjunto de itens que deseja comprar. O caixa registra os itens da compra e recebe o pagamento. No final, o cliente sai com os itens comprados."

**Seqüencia típica de eventos:**

| Ação do ator | Resposta do sistema |
|---|---|
| 1. Este caso de uso começa quando um cliente chega ao caixa, portando os itens que deseja comprar. | |
| 2. O caixa registra cada item. Se houver mais de um exemplar do item, o caixa pode registrar a quantidade. | |
| | 3. O sistema informa a descrição e o preço do item registrado e o acrescenta à transação em andamento. |
| 4. Ao terminar de registrar os itens, o caixa indica ao sistema que está completada a compra. | |
| | 5. O sistema informa o valor total. |
| 6. O caixa informa ao cliente o valor total. | |

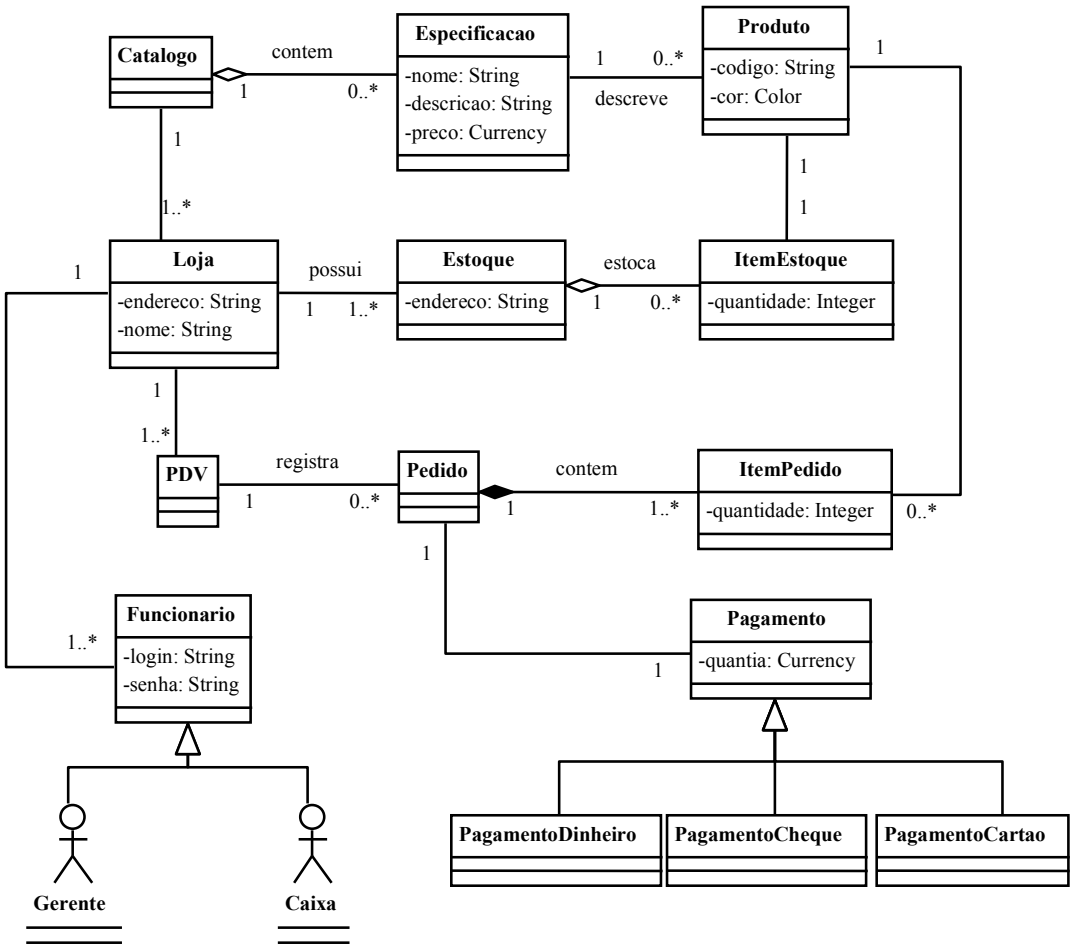
| | |
|---|---|
| 7. O cliente efetua o pagamento. | |
| | 8. O sistema registra a venda. |
| | 9. O sistema atualiza os níveis de estoque. |
| | 10. O sistema gera um recibo. |
| 11. O caixa fornece o recibo ao cliente. | |
| 12. O cliente sai com os itens comprados. | |

(a) Defina os eventos de sistema para este caso de uso (Lembrete: os eventos de sistema são os métodos que cruzam a fronteira do sistema para implementar um caso de uso).

**Resposta:**

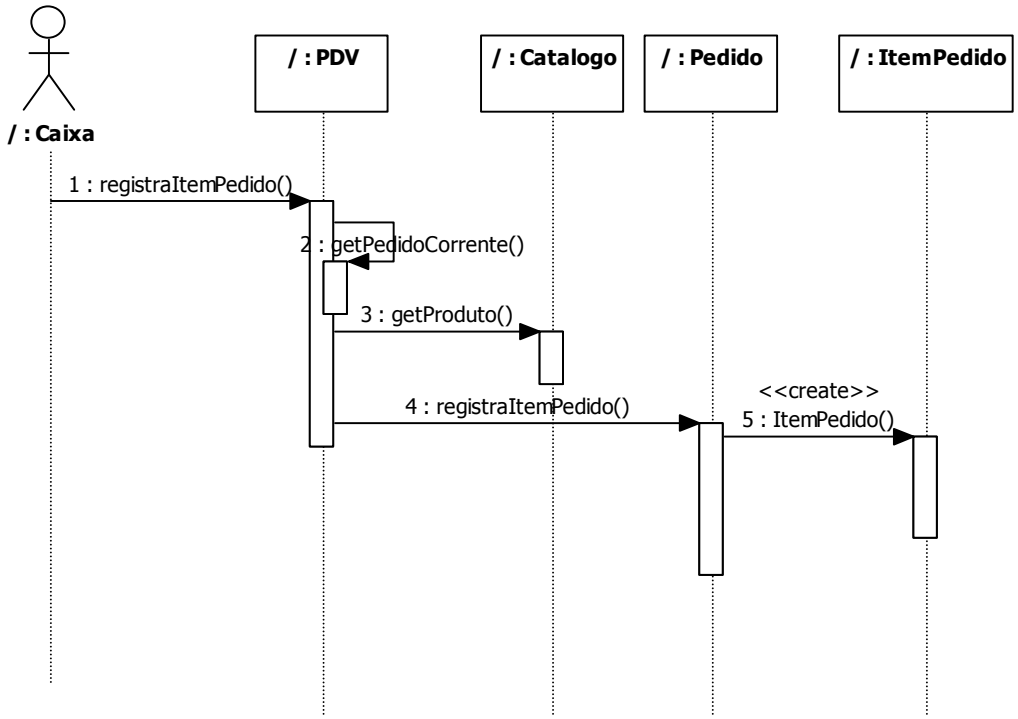
**registraItemPedido(código : String, quantidade : int) : void**
**fechaPedido() : Currency**
**registraPagamento(pagamento : Currency) : Currency**

(b) Faça diagramas de seqüencia que detalhem a atribuição de responsabilidades para cada evento de sistema, considerando o seguinte modelo de classes conceitual que foi gerado durante a análise.

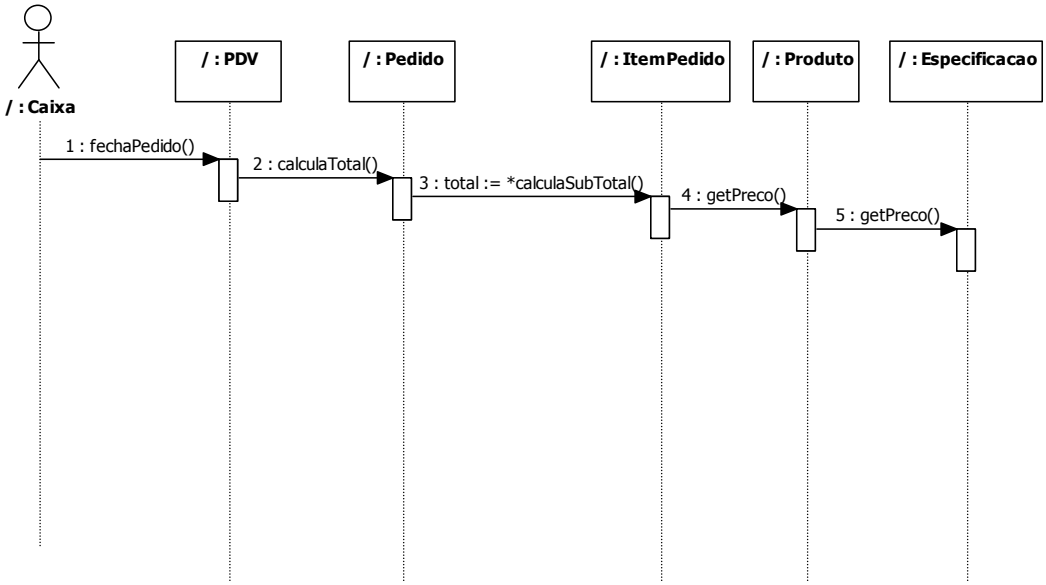


**Resposta:**

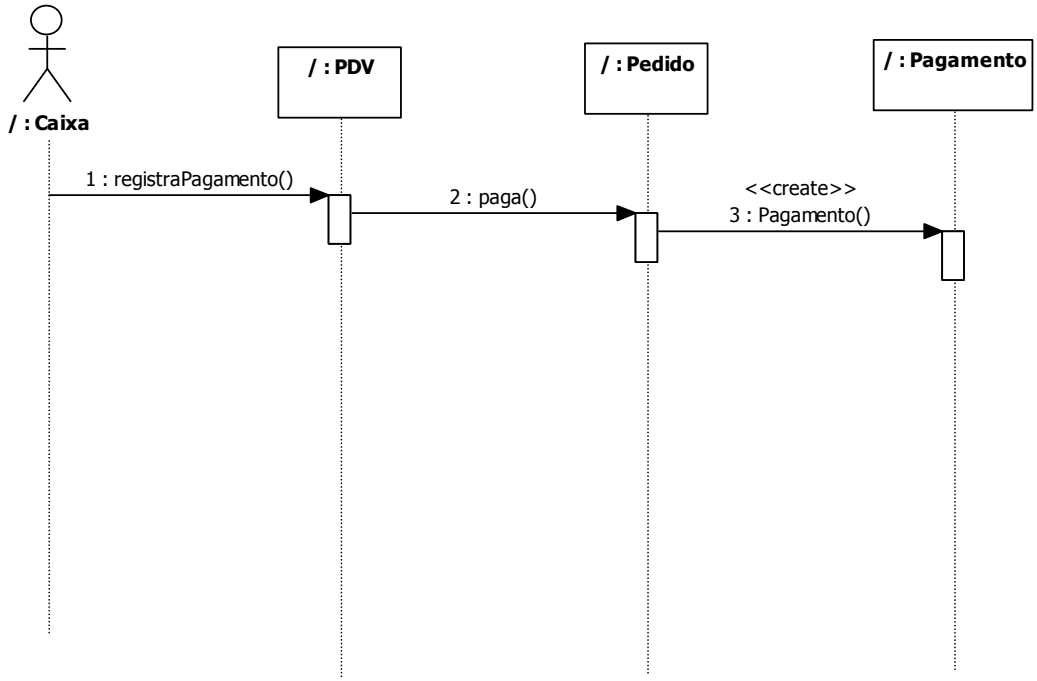
**registraItemPedido(código : String, quantidade : int) : void**




**fechaPedido() : Currency**



**registraPagamento(pagamento : Currency) : Currency**




(c) Defina um contrato com pré-condições, pós-condições e invariantes para cada evento de sistema definido anteriormente.

**Resposta:**

**registraItemPedido(codigo : String, quantidade : int)**
**Responsabilidade: Registrar um novo item de pedido ao pedido atual e exibir o preço e a descrição do item.**
**Argumentos:**
**codigo : código do produto que está sendo comprado.**
**quantidade : quantidade do produto que está sendo comprado.**
**Retorno: Não se aplica.**
**Invariante:**
**Existe um pedido corrente aberto para um determinado PDV.**
**Pré-condição:**
**O código do produto é válido no sistema (existe produto com esse código).**
**A quantidade do produto tem valor positivo e menor que a quantidade do produto em estoque.**
**Pós-condição:**
**Um item de pedido foi criado.**
**O item de pedido foi associado ao pedido corrente do PDV.**
**O item de pedido recebeu o valor da quantidade (argumento).**

**O item de pedido foi associado ao produto que tem determinado código (argumento).**

**fechaPedido();**
**Responsabilidade: Registrar o fim da entrada de itens e exibir o total do pedido.**
**Argumentos: Não se aplica.**
**Retorno: Total do pedido.**
**Invariante:**
**Existe um pedido corrente aberto para um determinado PDV .**
**Pré-condição:**
**Existe ao menos um item de pedido registrado para o pedido corrente.**
**Pós-condição:**
**O pedido passa para o estado fechado.**

**registraPagamento(quantia : Currency)**
**Responsabilidade: Registrar o pagamento, calcular o troco e imprimir o recibo.**
**Argumentos:**
**quantia : Valor pago pelo cliente.**
**Retorno: Não se aplica.**
**Invariante: Não se aplica.**
**Pré-condição:**
**Existe um pedido corrente fechado para um determinado PDV.**
**A quantia tem valor maior ou igual ao valor total do pedido.**
**Pós-condição:**
**Um pagamento foi criado.**
**O pagamento recebeu o valor total do pedido.**
**O pagamento foi associado ao pedido.**

Questão 2 (2,5 pontos)

Considere um sistema de controle bancário onde a classe "Conta" tem um método que é responsável pela transferência de dinheiro para outra conta corrente: "transfere(valor, contaDestino)". Neste cenário, somente transferências de valores menores que R$ 5000,00 podem ser feitas.

(a) Defina as pré-condições e pós-condições do método "transfere(valor, contaDestino)".

**Resposta:**

**Pré-condições:**
**Valor menor que R$ 5000,00**
**Saldo da conta origem maior ou igual ao valor**
**Pós-condição:**
**Saldo da conta origem igual ao saldo anterior menos o valor**
**Saldo da conta destino igual ao saldo anterior mais o valor**

(b) Seria correto criar uma classe que herde da classe "Conta" e somente permita transferências de até R$ 4000,00? Justifique a sua resposta.

**Resposta: Não. De acordo com a regra de contravariação, as pré-condições dos métodos polimórficos da subclasse devem ser iguais ou menos restritivas que as pré-condições dos métodos da superclasse. Contudo, o método da superclasse aceita valores de R$ 0 a R$ 5000,00, enquanto o método da subclasse aceitaria valores de R$ 0 a R$ 4000,00, sendo mais restritivo.**

(c) Se, além do método "transfere(valor, contaDestino)", existissem também os métodos "deposita(valor)" e "saca(valor)", a interface da classe "Conta" poderia ser considerada uma interface com comportamento replicado? Justifique a sua resposta.

**Resposta: Não. Apesar de ser possível simular uma transferência via saque e depósito, a ausência de um método próprio para transferência levaria a uma interface com comportamento inábil.**

Questão 3 (2 pontos)

O padrão "Observer" visa definir uma dependência de um para muitos via um mecanismo de notificação de eventos.

(a) Descreva um exemplo de situação onde esse padrão é útil.

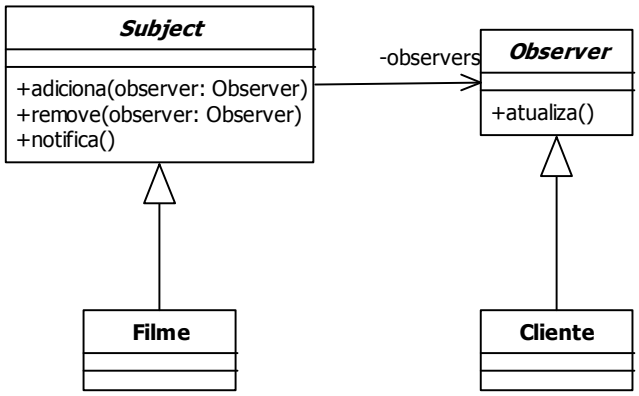
**Resposta: Em um sistema de locadora, onde os clientes podem se cadastrar como interessados em um filme e o sistema notifica os clientes assim que o filme estiver disponível.**

(b) Quais são os benefícios alcançados com o uso desse padrão?

**Resposta: Com esse padrão, os interessados na informação não têm a responsabilidade de verificar se a informação está disponível, aumentando o desempenho do sistema. Além disso, não é necessário acoplar os interessados na informação com as fontes de informação, melhorando a manutenibilidade do sistema.**

(c) Desenhe um modelo de classes exibindo como o padrão pode ser utilizado na situação descrita.

**Resposta:**




Questão 4 (2 pontos)

Enumere uma vantagem e uma desvantagem para os seguintes estilos arquiteturais:

(a) *Pipes & Filters*

(b) Camadas

(c) Processos distribuídos

(d) Orientado a objetos

**Resposta:**

**Pipes & Filters**
**Vantagens: simplicidade; e facilidades para compor e paralelizar o sistema;**
**Desvantagens: baixo desempenho (abstração de dados primitiva e gerência de buffers); e difícil interatividade e cooperação entre filtros.**

**Em camadas**
**Vantagens: flexibilidade; e reutilização de uma infra-estrutura de computação pré-existente;**
**Desvantagens: determinação do número de camadas; e comprometimento do desempenho devido à comunicação entre as camadas.**

**Processos distribuídos**
**Vantagem: decomposição do sistema em partes menores de modo a facilitar modificações necessárias;**
**Desvantagens: restrições topológicas; sujeito a problemas de sincronização de mensagens.**

**Orientado a objetos**
**Vantagens: permite a organização de programas; e ajuda a manter a integridade dos dados do sistema;**
**Desvantagem: devido à granularidade fina dos objetos e a dependência entre eles, a manutenção e a reutilização podem ser dificultadas, pois para modificar ou reutilizar um objeto pode ser necessário conhecer e atuar sobre outros objetos do sistema.**

Questão 5 (1 ponto)

Conceitue reutilização de software e cite alguns de seus benefícios e dificuldades.

**Resposta: Reutilização de software é o processo de incorporar em um novo produto artefatos de software preexistentes, ou de uma forma geral o conhecimento sobre o desenvolvimento destes artefatos. Alguns de seus benefícios esperados são: (i) melhores índices de produtividade; (ii) produtos de melhor qualidade, mais confiáveis, consistentes e padronizados; (iii) redução dos custos e tempo envolvidos no desenvolvimento de software; e (iv) maior flexibilidade na estrutura do software produzido, facilitando sua manutenção e evolução. Por outro lado, existem algumas dificuldades: (i) identificação, recuperação e modificação de artefatos reutilizáveis; (ii) compreensão dos artefatos recuperados; (iii) garantia da qualidade dos artefatos reutilizáveis; (iv) composição de aplicações a partir de componentes; (v) barreiras psicológicas, legais e econômicas; e (vi) a necessidade da criação de incentivos à reutilização.**