



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 da AD2 – 2º semestre de 2015.

Nome:

Polo:

Matrícula:

Observações:

1. Prova com consulta.

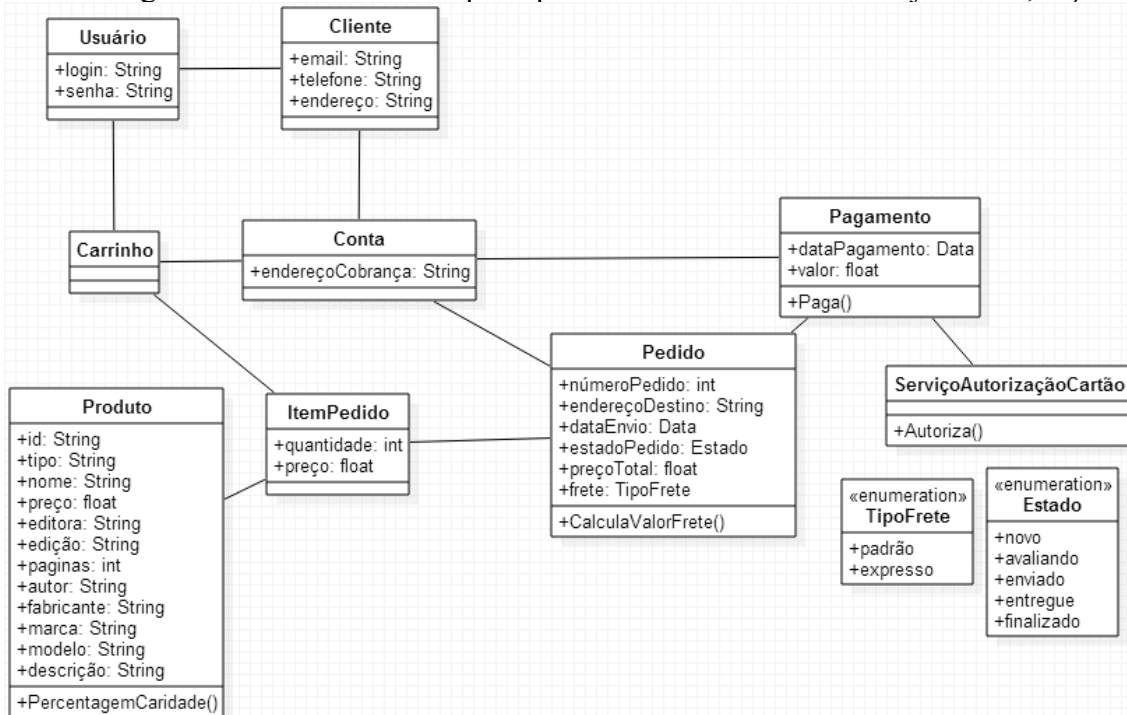
LER ATENTAMENTE AS INSTRUÇÕES A SEGUIR:

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. Apenas ADs **enviadas pela plataforma e dentro do prazo estabelecido** serão corrigidas.
 5. 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 um sistema de uma loja virtual onde o usuário realiza as compras através de um site. Essa loja possui apenas livros e eletrônicos em seu estoque. Além disso, ela possui um diferencial em relação a concorrência: um percentual dos lucros obtidos das vendas são destinadas a instituições de caridade. Esse percentual doado para uma instituição de caridade varia de acordo com o tipo de produto vendido. Ou seja, se a loja vender um livro, então 10% do lucro obtido pela venda do livro é doado para uma instituição de caridade. Caso ela venda um eletrônico, o valor doado é de 5% do lucro obtido pela venda do eletrônico.

Dado o diagrama de classes abaixo que representa o atual sistema da loja virtual, faça:



- [1.75 pontos] Altere o diagrama de classes do enunciado para resolver o problema que a classe *ItemPedido* possui referente às alternativas em função de cada tipo de produto disponível na loja. Justifique citando o padrão GRASP utilizado para realizar essa alteração.
- [1.5 pontos] Modifique o diagrama do item (a) de tal forma que deixe claro qual classe é responsável pela criação de objetos do tipo *Pedido*. Justifique citando o padrão GRASP utilizado e elabore um diagrama de sequência que demonstra essa relação de responsabilidade.
- [1.0 pontos] Suponha que objetos precisam ser capazes de referenciar um *Pedido* em particular, dado seu número de pedido. Qual classe consegue identificar o objeto *Pedido* através do número de pedido? Justifique citando um padrão GRASP.
- [1.0 pontos] Evolua o diagrama de classes gerado no item (b) referente a loja virtual para permitir que o sistema suporte formas de pagamento em *Cartão de Crédito*, *PayPal* e *Boleto Bancário*. Justifique a alteração citando os padrões GRASP utilizados. Dica: Não se esqueça dos serviços de autorização.
- [2.0 pontos] O *PayPal* possui um sistema de autorização que é incompatível com a interface disponível pela loja virtual. Dessa forma, evolua o diagrama de classes gerado no item (d) para permitir que o sistema fique compatível com a classe de autorização do *PayPal*. Justifique a alteração do diagrama citando o padrão GoF utilizado.
- [1.75 pontos] Altere o diagrama de classes para permitir que a classe *Pedido* utilize diferentes estratégias de cálculo do valor de frete de acordo com o seu tipo de envio

(padrão ou expresso). Justifique citando o padrão de comportamento utilizado para realizar a alteração.

- g) [1.0 pontos] Em uma arquitetura web de 3 camadas, quais são as camadas e como essas camadas devem interagir para atender às requisições do cliente? Além disso, descreva os componentes esperados da loja virtual apresentada que pertencem a cada camada.

Gabarito

a.

[1.75 pontos]

0.5 pela herança das classes (*Livro* e *Eletrônico* herdam de *Produto*)

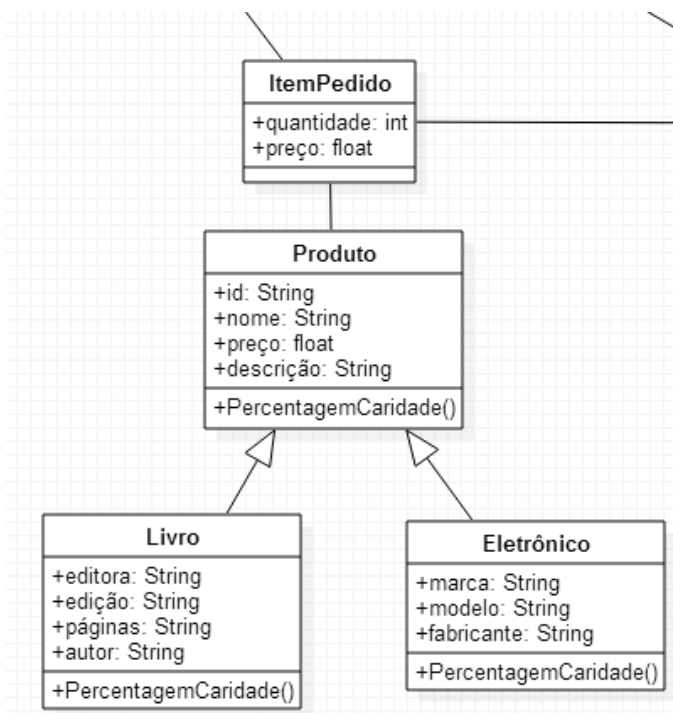
0.25 por ter removido o atributo *tipo* da classe *Produto*

0.25 sem a repetição dos atributos e alocados corretamente nas classes

0.25 sobrecarga do método *PorcentagemCaridade*

0.5 pontos pelo Padrão *Polymorphism*

Padrão *Polymorphism*.



b.

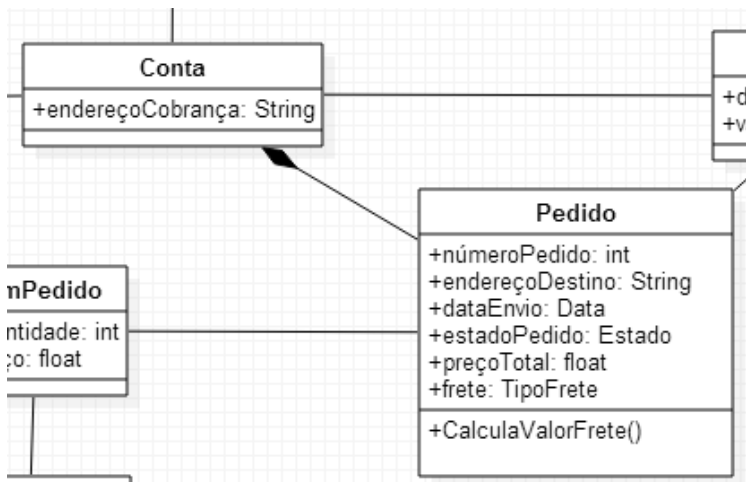
[1.5 pontos]

0.5 pontos diagrama de classe (composição Conta-Pedido)

0.5 pontos diagrama de sequencia

0.5 pontos padrão *Creator*

Padrão *Creator*.



C.

[1.0 ponto]

0.5 pontos pelo padrão *Information Expert*

0.5 pontos pela responsabilidade da classe *Conta* (apenas se acertou o padrão).

Resposta: Como a classe *Conta* agrega todos os pedidos, então pelo padrão *Information Expert*, a classe *Conta* possui as informações necessárias para cumprir essa responsabilidade.

d.

[1.0 ponto]

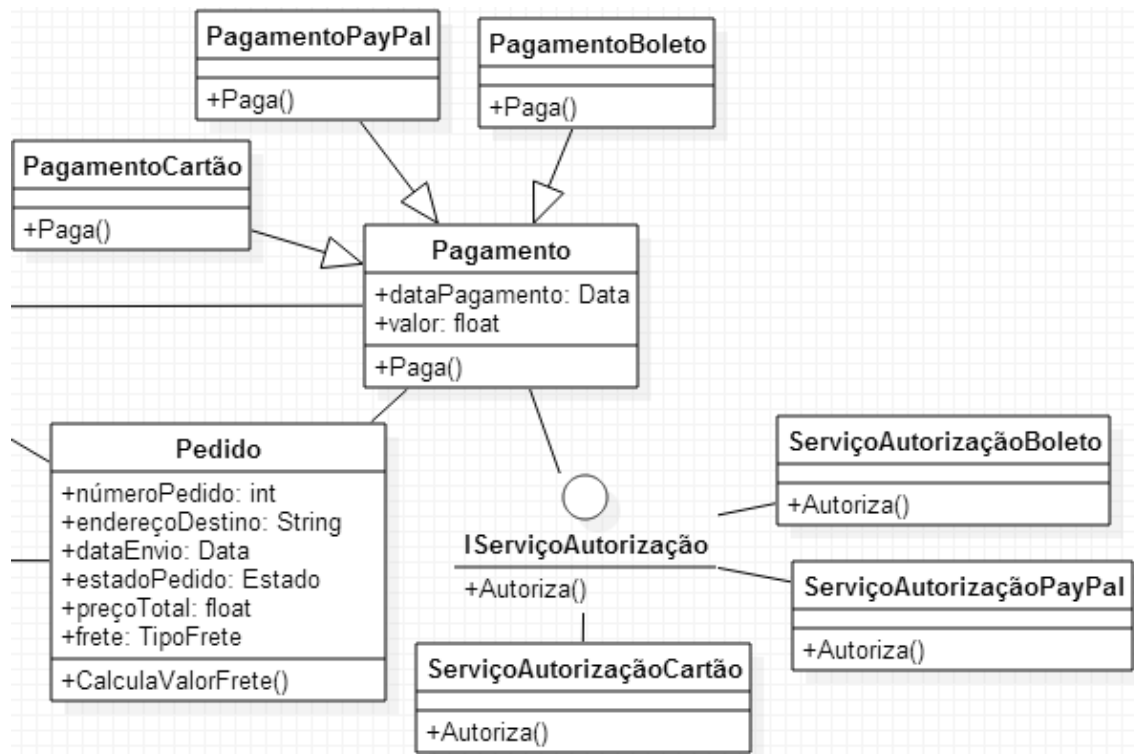
0.25 pontos pela herança para os tipos de Pagamento

0.25 pontos pelos novos serviços de autorização através da interface

0.25 pontos pelo padrão *Indirection*

0.25 pontos pelo padrão *Polymorphism*

Padrão *Indirection* e *Polymorphism*.



[2.0 pontos]

0.5 pontos pelo padrão *Adapter*

```

classDiagram
    class Pagamento {
        +dataPagamento: Data
        +valor: float
        +Paga()
    }
    class PagamentoPayPal {
        +Paga()
    }
    class PagamentoBoleto {
        +Paga()
    }
    class PagamentoCartao {
        +Paga()
    }
    class Pedido {
        +numeroPedido: int
        +enderecoDestino: String
        +dataEnvio: Data
        +estadoPedido: Estado
        +precoTotal: float
        +frete: TipoFrete
        +CalculaValorFrete()
    }
    class IServicoAutorizacao {
        +Autoriza()
    }
    class ServicoAutorizacaoCartao {
        +Autoriza()
    }
    class ServicoAutorizacaoBoleto {
        +Autoriza()
    }
    class AdapterPayPal {
        +Autoriza()
    }
    class ServicoAutorizacaoPayPal {
        +Autoriza(valor : float)
    }

    PagamentoPayPal --|> Pagamento
    PagamentoBoleto --|> Pagamento
    PagamentoCartao --|> Pagamento
    Pagamento --o-- IServicoAutorizacao
    Pedido --o-- IServicoAutorizacao
    ServicoAutorizacaoCartao --|> IServicoAutorizacao
    ServicoAutorizacaoBoleto --|> IServicoAutorizacao
    AdapterPayPal --|> ServicoAutorizacaoBoleto
    AdapterPayPal --|> ServicoAutorizacaoPayPal

```


f.

[1.75 pontos]

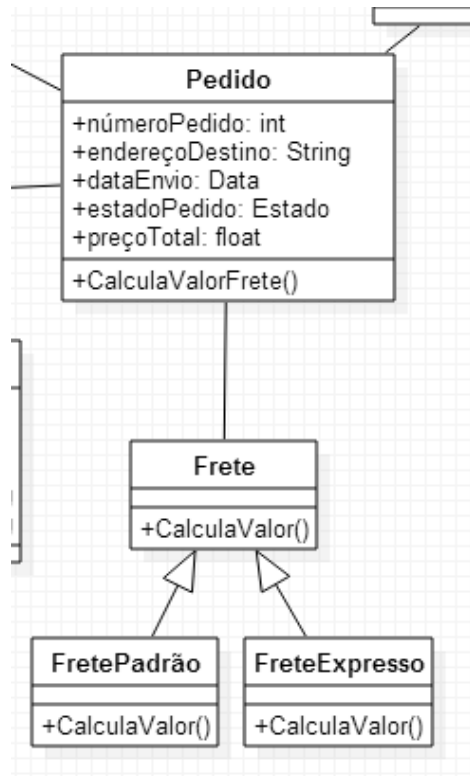
0.5 pela classe que representa os possíveis algoritmos de cálculo (*Frete*)

0.5 pela herança dos algoritmos a classe *Frete*.

0.25 por ter removido atributo *frete* de *Pedido* e a classe *TipoFrete*(considerar apenas se ambos tiverem sido removidos)

0.5 pontos pelo padrão *Strategy*

Padrão *Strategy*.



g.

[1.0 ponto]

0.2 pontos pelas 3 camadas (apresentação, aplicação, armazenamento)

0.2 pontos pela descrição da interação das camadas

0.2 pontos para o componente da camada de apresentação

0.2 pontos para o componente da camada de aplicação

0.2 pontos para o componente da camada de armazenamento

Resposta:

As camadas de uma arquitetura em três camadas são: *apresentação*, *aplicação* e *armazenamento*. Usualmente, o servidor recebe solicitações do cliente via camada de aplicação, que consulta a camada de armazenamento para processar os dados de acordo com as regras de negócio. Finalmente, a camada de apresentação é acionada para construir a página com a resposta referente à solicitação.

Divisão das camadas da loja virtual:

- Camada de Apresentação: é o site da loja virtual, onde o cliente faz a interação com a loja e realiza a compra dos produtos.
- Camada de Aplicação: é a aplicação descrita nessa AD, ou seja, o software que administra os usuários, os pedidos e compras dos produtos.
- Camada de Armazenamento: é o banco de dados da loja, contendo as tabelas dos produtos registrados.