

# CIN20204 – TÉCNICAS DE PROGRAMAÇÃO

## Atividade 01 - Orientação à Objetos

(PDF gerado em 22/03/2024 às 03:41:00)

**Prazo de entrega: 28/03/2024 até às 12:00. Enviar para o e-mail: [rogerio.vieira@unialfa.com.br](mailto:rogerio.vieira@unialfa.com.br). Apenas um por Squad contendo o nome de todos os integrantes.**

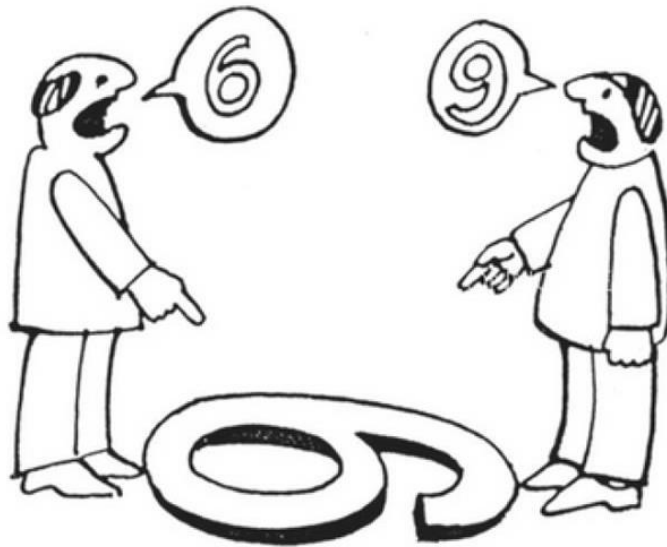
### Observação

Essa atividade deve ser realizada pelo grupo (Squad)

Antes de iniciar o desenvolvimento de qualquer software, o analista de sistemas/engenheiro de software deve compreender os requisitos do software, ou seja, é necessário que o desenvolvedor entenda o que o cliente quer que o software faça. Para isso, inicialmente, deve identificar as classes, seus atributos, seus métodos e as associações (ligações) que existem entre as classes.

Na sequência, deve ser capaz de estabelecer um modelo de trabalho apoiado em padrões, convenções e ferramentas que o auxiliem a alcançar o objetivo que é a satisfação do cliente com uma entrega que atenda as necessidades pré-determinadas pelo cliente (requisitos) e com a qualidade de seu trabalho.

**Questão 01.** Uma boa comunicação é fundamental em qualquer ambiente para se mantenha a produtividade em alta. Porém, muitas vezes, essa comunicação fluida não é possível pela falta de um padrão claro e objetivo. A ilustração demonstra um cenário onde a comunicação é ruidosa pela ausência de um padrão que determine se o número apresentado é um 6 ou um 9. Isso poderia ser facilmente resolvido se, por exemplo, um traço fosse feito em um dos lados do número e esse padrão fosse comunicado e informado a todos os envolvidos.



No desenvolvimento de software não é diferente. É comum vermos projetos fracassarem por falta de clareza e entendimento entre os envolvidos. Uma forma de mitigar isso, ou seja, de minimizar os problemas de comunicação é estabelecendo padrões, sejam eles padrões de fala, padrões aplicados no código ou ambos. Tomando como referência esse contexto, crie um padrão de nomes para:

#### Dica!

Como auxílio, você pode utilizar um dos “**Coding Conventions**” abaixo para se inspirar:

- a) **Java:** <https://www.oracle.com/docs/tech/java/codeconventions.pdf>
- b) **C#:** <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions>
- c) **Python:** <https://peps.python.org/pep-0008/>

Também é recomendável que você procure e entenda os padrões de nomenclatura que são usadas nessas mesmas linguagens, a saber:

- a) PascalCase
- b) camelCase
- c) snake\_case

#### Lembrete

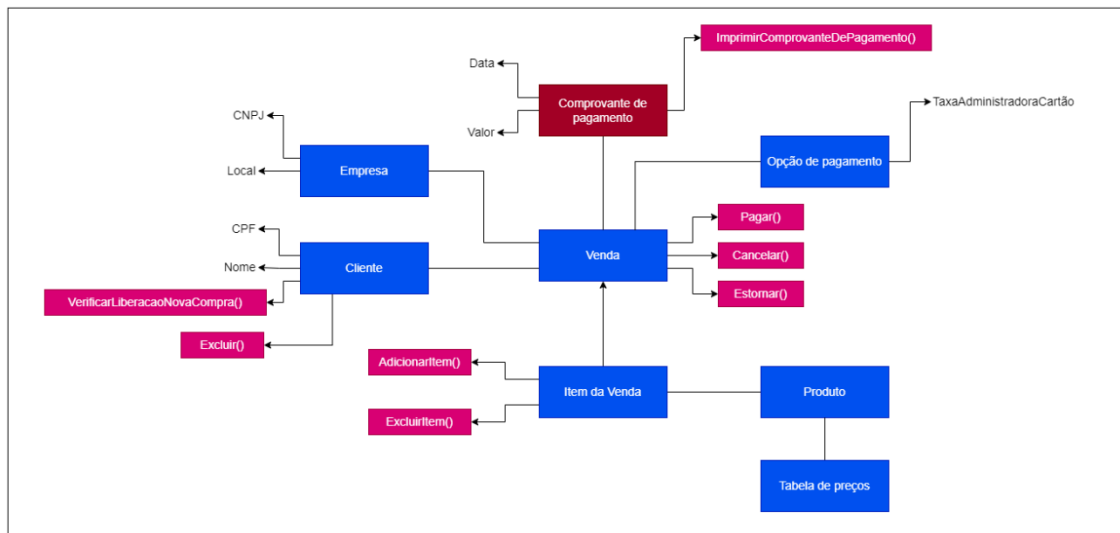
**Estrutura de controle de fluxo:** Permitem escolher entre diferentes caminhos de execução, de acordo com uma condição.

**Estrutura de repetição:** Permitem executar um bloco de instruções várias vezes.

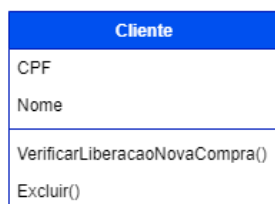
- a) Nomes de pastas e arquivos que serão utilizados no software
- b) Nomes de variáveis.
- c) Nomes de constantes.

- d) Nomes de métodos (procedimentos ou funções).
- e) Verifique se existe alguma recomendação para escrita de estrutura de controle de fluxo e estruturas de repetição.

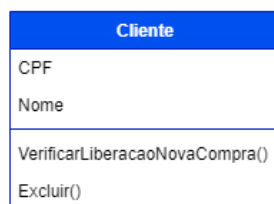
**Questão 02.** Dando continuidade ao modelo apresentado em sala de aula, faça o que se pede:



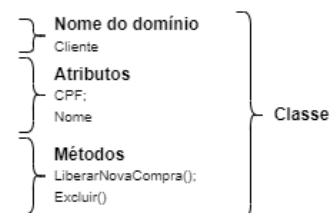
- a) Com suas palavras, defina o que é domínio, classe, método, comportamento e atributo.
- b) Crie mais 12 classes para que o modelo totalize 20 classes.
- c) Para cada classe, adicione 5 atributos.
- d) Para cada classe, adicione 3 métodos.
- e) Verifique se existem erros ou inconsistências de modelagem, ou seja, existe algo que pode ser melhorado? Caso exista, faça as devidas adequações.
- f) Em seguida, separe as classes por área de negócio (setor).
- g) Após realizar os itens **a**, **b**, **c**, **d**, **e** e **f**, refaça o modelo para que cada classe tenha no formato **(a)**, o formato **(b)** é apenas um lembrete:



(a)



(b)



**Questão 03.** A Programação Orientação à Objetos (POO) está apoiada sobre 4 pilares fundamentais, são eles:

- Abstração
- Encapsulamento
- Polimorfismo e
- Herança

Em sala de aula falamos a respeito de um exemplo de polimorfismo. Recupere de suas anotações ou da memória esse exemplo e, com suas palavras, explique o que é polimorfismo.

**Questão 04.** Com base no exposto na introdução do exercício, o proprietário de uma pequena rede de Pet Shop entrou em contato para que um software seja criado para a sua empresa. Após uma boa conversa, foram identificados os seguintes requisitos:

- 1) Deve existir um cadastro para a matriz e suas filiais;
  - a. A matriz não deve ser excluída.
  - b. A filial só poderá ser excluída caso não exista vendas ou reservas para ela.
  - c. A filial pode ser inativada.
  - d. A matriz é a única unidade capacitada a receber animais de grande porte.
- 2) Deve existir um cadastro para o cliente;
  - a. O cliente não deve ser excluído.
  - b. O cadastro do cliente pode ser inativado.
  - c. Deve ficar registrado o código e o nome do usuário que cadastrou o cliente.
  - d. Deve ficar registrado o código e o nome do usuário que alterou o cadastro do cliente.
  - e. O cliente que não compra a mais de 90 dias deve ser inativado.
- 3) Deve existir um cadastro para o pet;
  - a. O cadastro do pet deve ser associado ao cliente;
  - b. O cadastro deve permitir cadastrar animais de pequeno, médio e grande porte.
- 4) Deve existir um cadastro para os serviços prestados pelo Pet Shop;
  - a. O serviço não pode ser excluído.
  - b. Os principais serviços oferecidos pelo Pet Shop são banho e tosa.
  - c. O Pet Shop também oferece o serviço de vacinação.
  - d. E o serviço de hotel para pet.
- 5) Deve existir um cadastro para os produtos vendidos pelo Pet Shop;
  - a. O produto não deve ser excluído.
  - b. O produto só pode ser vendido se existir quantidade.
  - c. O produto só pode ser vendido se estiver dentro do período de validade.
  - d. Deve existir estoque mínimo e estoque máximo para o produto.
- 6) Deve existir um cadastro para os funcionários;
  - a. Os funcionários devem ser vinculados a uma filial específica ou a matriz.
- 7) Deve existir um cadastro de usuários;
  - a. O usuário deve ser associado ao funcionário.

- b. O usuário só deve ser excluído se o funcionário estiver inativo.
- 8) Deve existir um cadastro para os fornecedores;
  - a. O cadastro de fornecedor deve identificar quando o fornecedor for um prestador de serviço.
- 9) Deve existir um cadastro para as despesas;
- 10) Deve existir um cadastro para reservas;
  - a. Das 10 unidades da empresa, apenas 2 oferecem o serviço de hospedagem.
  - b. A reserva deverá ter a data de entrada e a data de saída do pet.
  - c. A reserva deve ter o nome do tutor do pet e o nome do pet.
  - d. Deve ficar registrado o código e o nome do usuário que realizou as reservas.
  - e. Só pode ser criada uma reserva para clientes ativos.
- 11) O sistema deve registrar as vendas dos produtos e serviços;
  - a. Só pode ser realizada uma venda para clientes ativos.
  - b. Ao finalizar a venda, deverá ser emitido o documento fiscal, nota fiscal quando for uma venda de produto ou nota de serviço quando for prestado um serviço.
  - c. Caso o cliente tenha pago valor maior do que o total da venda, deverá ser devolvido o troco.
  - d. A venda deve ter o desconto oferecido ao cliente.
    - i. Quando o total da venda for inferior a R\$ 500,00 o cliente não terá direito a desconto.
    - ii. Quando o total da venda for superior a R\$ 501,00 o cliente tem direito a 10% de desconto.
  - e. A venda deve ter a comissão do funcionário
    - i. Quando o total da venda for inferior a R\$ 100,00 o funcionário não ganhará comissão.
    - ii. Quando o total da venda estiver entre R\$ 100,00 e R\$ 500,00 o funcionário ganhará 1% de comissão.
    - iii. Quando o total da venda estiver entre R\$ 501,00 e R\$ 1.000,00 o funcionário ganhará 2% de comissão.
    - iv. Quando o total da venda for maior do que R\$ 1.001,00 o funcionário ganhará 5% de comissão.

Sobre esse estudo de caso, faça o que se pede:

- a) Identifique todos os domínios do estudo de caso.
- b) Após identificar os domínios, separe-os por área de negócio.
- c) Modele as classes identificando os atributos e comportamentos informados pelo cliente. Não se esqueça de representar (mostrar) as associações entre as classes. Observação, considere que nem sempre o cliente informa todos os atributos necessários à construção do sistema. Sendo assim, preencha o máximo de lacunas que seu Squad conseguir.