

Contextualização:

A empresa Tabajara Comércio e Varejo (TCV) iniciará, em breve, suas operações no Brasil. Mas, para que possa começar a sua atuação neste mercado superconcorrido, ela precisa ter em funcionamento um sistema para controle e venda. Este sistema é essencial para a empresa e ela investirá um valor altíssimo na compra deste *software*. Sua empresa, já no ramo de desenvolvimento de *software*, viu a oportunidade e pôs-se a desenvolver o produto a fim de superar a concorrência e faturar o contrato com a TCV.

A TCV publicou um edital esclarecendo a especificação do sistema que ela precisa, para tanto ela teve o auxílio dos analistas de sistema que já trabalham na empresa e que eventualmente avaliarão os *softwares* apresentados pelas diversas empresas existentes no mercado e que atenderão ao edital publicado. O sistema, segundo a TCV, deverá ser feito em duas etapas. E para a primeira etapa ela pede que sejam criadas as seguintes classes:

1) Estrutura de classes¹ (Figura 1):

- Cliente,
- Endereço,
- Pessoa Jurídica,
- Pessoa Física,
- Conta,
- Compra,
- Produto,
- Fatura,
- Pagamento e
- Cálculo de Juros.

2) Atributos das Classes:

- Cliente
 1. nome
 2. endereço
 3. limite de crédito
- Endereço
 1. logradouro
 2. número
 3. complemento
 4. bairro
 5. município
 6. estado
- Pessoa Jurídica
 1. cnpj
 2. nome de fantasia
- Pessoa Física
 1. cpf
- Conta

¹ Neste diagrama não estão representadas as classes de gerenciamento.

- 1. compras
- 2. data de vencimento
- 3. total
- 4. clientes
- Compra
 - 1. produtos
 - 2. total
 - 3. data
- Produto
 - 1. nome
 - 2. preço
 - 3. código de barras
- Fatura
 - 1. conta
 - 2. data de quitação
 - 3. juros
 - 4. quantidade de parcelas
 - 5. parcelas
- Pagamento
 - 1. data
 - 2. valor
 - 3. tipo
 - 4. juros
- Cálculo de Juros

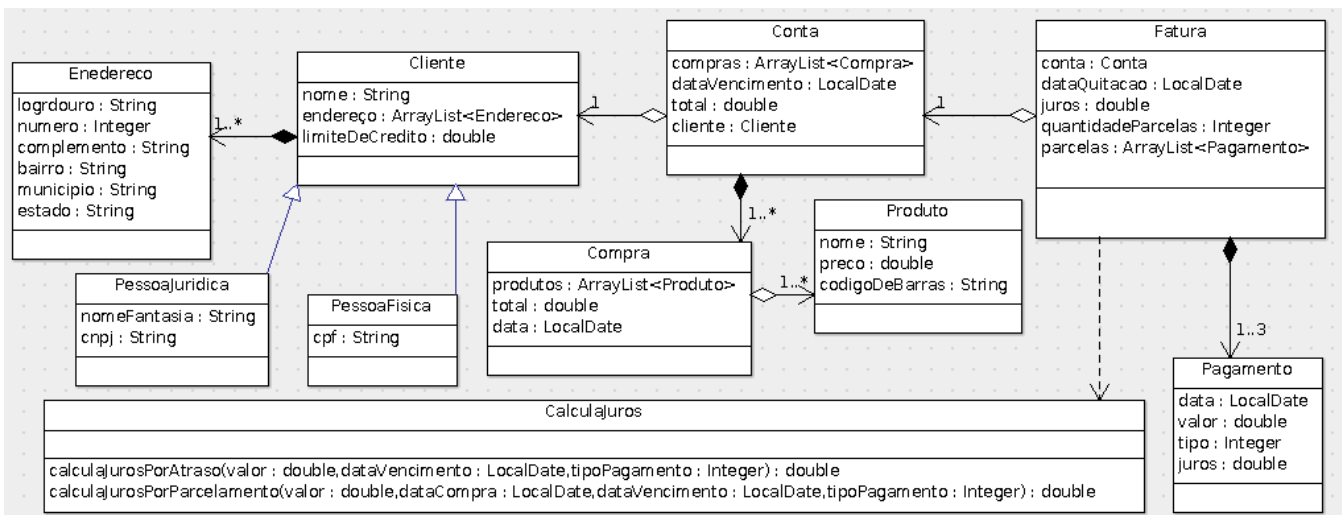


Figura 1: Diagrama de Classes do Sistema

- 3) Faz parte do trabalho, a interpretação, por parte do grupo, do diagrama UML da Figura 1. Podendo o grupo, questionar sobre qualquer problema que nele for encontrado.
- 4) O tipo “LocalDate” é o mais moderno recurso Java para se trabalhar com datas. Este tipo não foi e nem será ensinado em sala de aula. É parte deste trabalho, o grupo pesquisar o funcionamento do LocalDate e aplicar no trabalho em questão.

- 5) Entendendo o diagrama da Figura 1: Uma “Fatura” é gerada para uma “Conta”. Um “Conta” é gerada para uma ou mais “Compra”s de um mesmo “Cliente”. Uma “Compra” é o conjunto de “Produto”s comprados por um “Cliente”.
- 6) O atributo “tipo” na classe “Pagamento” diz respeito ao seguinte: dinheiro, cartão de crédito ou cartão de débito. Para a manipulação deste atributo deverá ser criado um “Tipo Enumerado” ou “Enumeração”. Este conceito de “Tipo Enumerado” também não foi ensinado em sala de aula e nem será. O grupo deverá pesquisar o funcionamento e aplicar o conceito no trabalho.
- 7) Ao faturar uma compra o cliente poderá parcelá-la em até 3 vezes;
- 8) Caso ocorra o parcelamento da fatura, então deverá ser calculado juros. Que obedecerão às regras que serão estabelecidas adiante.
- 9) O campo “dataQuitação” mostrado na classe “Fatura” só deverá ser preenchido quando a última parcela da fatura for paga. As faturas sem este campo preenchido significam faturas em aberto, ou seja, não recebidas.
- 10) A ideia da “Conta” é que o “Cliente” possa fazer diversas compras durante o mês e acertar tudo no final, assim que receber seu salário.
- 11) O Cliente poderá parcelar sua conta. Porém, o mesmo arcará com as seguintes condições, de acordo com os seguintes tipos de pagamento:
 - dinheiro - deverá ser calculado 10% ao mês, juros sobre juros;
 - cartão de débito - deverá ser calculado 10% ao mês, juros sobre juros e mais 1,3% sobre o valor final;
 - cartão de crédito - deverá ser calculado 10% ao mês, juros sobre juros e mais 1,8% sobre o valor final.Estes cálculos são de responsabilidade da função “calculaJurosPorParcelamento” da classe “CalculaJuros”.
- 12) Caso o “Cliente” atrase no pagamento das parcelas da “Fatura”, então deverá ser calculado juros de mora (atraso). Estes juros deverão ser aplicados da seguinte maneira:
 - até 15 dias de atraso - 3% a mais sobre o valor da parcela;
 - de 16 a 30 dias - 9% a mais sobre o valor da parcela;
 - acima de 30 dias - 0,33% por dia decorrido desde a data do vencimento.Estes cálculos são de responsabilidade da função “calculaJurosPorAtraso” da classe “CalculaJuros”.
- 13) Caso seja necessário, os grupos poderão criar mais atributos nas classes. E também, caso queiram, poderão criar outras classes. Os métodos deverão ser criados de acordo com a necessidade de cada *software*, por este motivo não foram especificados no diagrama UML (Figura 1), com exceção da classe “CalculaJuros”;
- 14) Sua empresa deverá seguir fielmente os nomes das classes, os nomes dos atributos, métodos e tipos que foram especificados pela TCV na Figura

1. Caso isto não aconteça, haverá uma grande penalização por parte dos avaliadores.

- 15) Neste *software*, o usuário deverá ser capaz de cadastrar objetos sem limite. Sendo que, para armazenar Pessoa Física e Pessoa Jurídica deverá haver um único ArrayList e este deverá ser gerenciado por meio do recurso de Polimorfismo. Este ArrayList deverá ter o nome de “clientes”.
- 16) Crie um *menu* principal com os itens: Cliente, ..., ... e Sair.
- 17) Para os itens que requerem operação de: Cadastro, Alteração, Exclusão, Consulta e Relatório, deverão ser respeitadas as seguintes restrições:
- Cadastro
 - sem restrição;
 - Alteração
 - esta operação deverá dar a possibilidade do usuário ver os valores já cadastrados para um determinado item do ArrayList. Item este, escolhido pelo usuário para que o mesmo possa alterá-lo;
 - Exclusão
 - antes de excluir um item do ArrayList, deverá ser mostrado todos os dados do item que será excluído e emitida uma mensagem de confirmação da operação para o usuário. Caso o usuário confirme a operação, então o item será apagado. Caso contrário o item não será apagado;
 - Consulta
 - deverá possibilitar ao usuário do *software* a visualização de um item do ArrayList;
 - Relatório
 - deverá possibilitar ao usuário do *software* a visualização de todos os itens do ArrayList;
 - Voltar ao *menu* principal
 - deverá possibilitar ao usuário a volta ao *menu* principal, caso estes itens estejam num *submenu*.

Como o seu *software* deverá funcionar

- A estrutura de *menus*, deverá proporcionar ao usuário:
 - a possibilidade de navegar entre o *menu* principal e seus *submenus* sem efetuar qualquer operação;
 - o usuário só poderá sair do programa através do *menu* principal, ou seja, acessando a opção “Sair”;
- Um *software* deve ter uma boa aparência e ser de fácil utilização, para agradar e facilitar a vida de quem o utilizará;
- O *software* que será entregue não deverá fazer uso de vetores e tampouco matrizes. No lugar destes use qualquer objeto das *Collections* de Java.

Regras para a entrega do trabalho

- Deverá ser apresentado e entregue, o projeto (compactado) do código-fonte.
- **O código-fonte que será entregue e apresentado não deverá possuir nenhum tipo de comentário.**
- Deverá ser enviado para o e-mail: eddiesaliba2@yahoo.com (de acordo com as regras a seguir).

- **Não serão recebidos trabalhos após a data marcada para entrega.**
- **Para a apresentação no laboratório deverá ser levado pelo grupo, em *pendrive*, uma cópia do arquivo que foi enviado por *e-mail*. Caso o grupo possua alguma restrição ou dificuldade no cumprimento desta regra, então, deverá avisar ao professor com antecedência mínima de 24 horas da data de apresentação.**

Regras para envio do *e-mail* com o trabalho

- No assunto do *e-mail* deve constar apenas o título: **IFTM - POO - ADS3PA - Trabalho 01 - Parte 01**
- No corpo do *e-mail* deverá conter, **única e exclusivamente**, o nome completo de todos os integrantes do grupo (**um em cada linha**).
- Só será aceito **UM** *e-mail* por grupo. Portanto, verifique se está tudo certo com seu *e-mail* e trabalho antes de enviá-lo.
- **O *e-mail* deverá ser enviado, no máximo, até UM dia antes da data marcada para apresentação.**

Obs.: O desrespeito a qualquer das regras acima implicará na perda de créditos para o grupo.

Critérios de Avaliação no Laboratório:

- Conformidade do *software* em relação ao solicitado;
- Legibilidade do código (**organização, endentação e etc.**);
- Usabilidade das interfaces de interação com o usuário;
- **Entendimento individual a respeito do código-fonte apresentado.**