

Programação Orientada a Objeto

Trabalho Prático – Controle de Contas a Receber

Parte 02

Professor: Edwar Saliba Júnior

Valor: 19 pontos

Contextualização:

A empresa Tabajara Comércio e Varejo (TCV) fez um pronunciamento afirmando que o *software* que ela deseja ainda está longe de ficar como o desejado. Mas, mesmo assim estaria disponibilizando os pré-requisitos para a segunda etapa do desenvolvimento do mesmo. A TCV lembra também que, **todos os erros, faltas e/ou falhas** que foram apresentados / descobertos na primeira apresentação do *software*, sejam corrigidos ou implementados antes da implementação desta segunda parte.

A empresa destaca novamente o diagrama UML do *software* já com algumas melhorias:

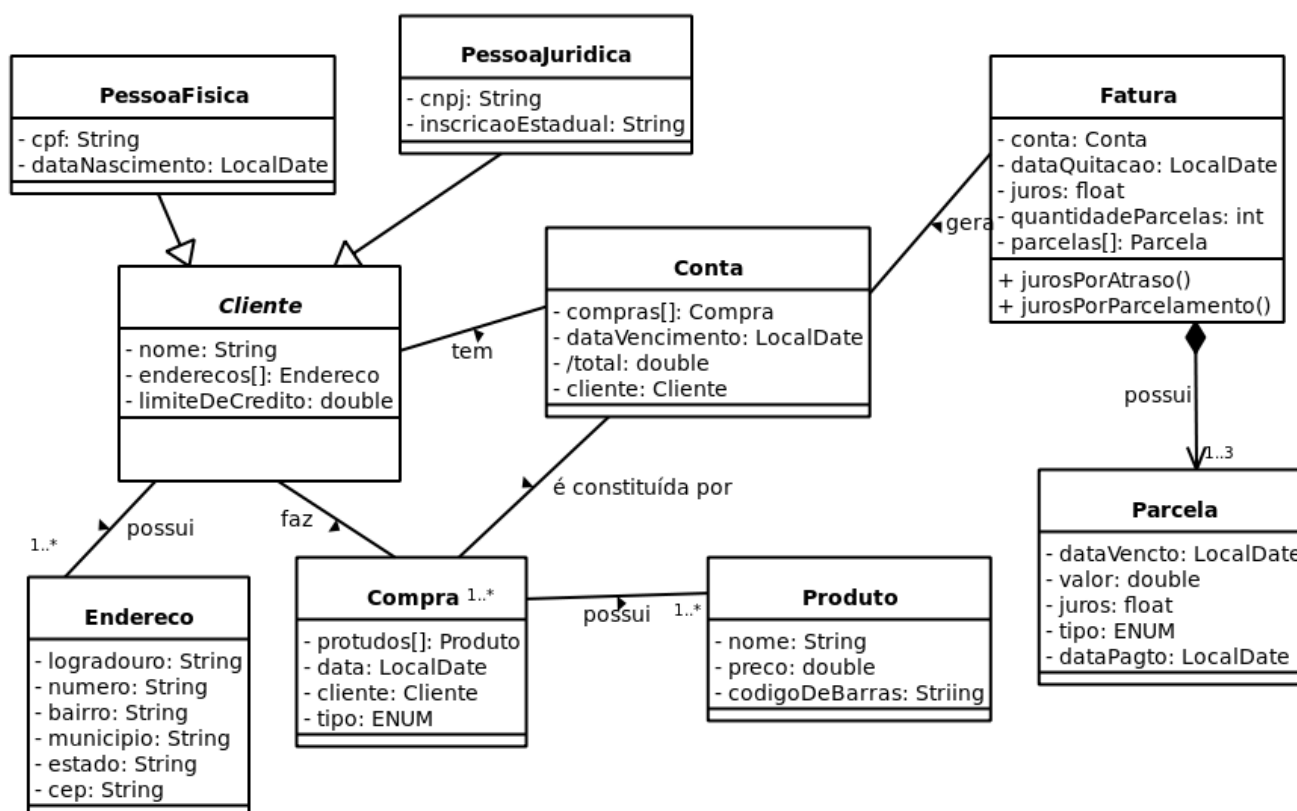


Figura 1: Diagrama de Classes do Sistema

Tarefas:

- 1) Faz parte do trabalho, a interpretação, por parte do grupo, do diagrama UML da Figura 1. Podendo o grupo, questionar sobre qualquer dúvida ou problema que nele for encontrado;
- 2) o atributo “parcelas[]”, da classe “Fatura”, continua sendo um vetor com no máximo 3 posições. Pois, uma fatura continuará podendo ser dividida em, no máximo, 3 parcelas. Observe que os demais atributos representados no diagrama da Figura 1 com “[]” em sua frente; que na versão passada deste *software* eram vetores; agora nesta nova versão deverão ser substituídos por “ArrayList’s” e não deverão haver limites de cadastros. Por exemplo: na versão anterior do *software* um *Cliente* poderia ter, no máximo, 3 endereços cadastrados. Nesta versão o *Cliente* poderá ter cadastrado quantos endereços ele tiver/quiser. O “*” (asterisco) em UML significa “muitos”, ou seja, um número não definido;

- 3) a classe *Cliente* deve ser definida como uma classe abstrata nesta nova versão do *software*;
- 4) para facilitar a manutenção futura do *software*, a TCV exige que se use o padrão MVC (*Model, View, Controller*) no desenvolvimento do *software*. O padrão MVC não será ensinado em sala de aula. É parte deste trabalho, o grupo pesquisar o padrão MVC e aplicá-lo no desenvolvimento do trabalho em questão e
- 5) toda a estrutura de *menus* e as funcionalidades utilizadas na primeira versão do *software* deverão ser mantidas nesta segunda versão.

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. Ou seja, sua interface deve ser intuitiva.

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.**
- O trabalho deverá ser feito e entregue funcionando na IDE Eclipse e com a JDK 11.

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 02**
- 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 em 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.**