

Contextualização:

A locadora de veículos Tabajara (LVT) 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 de locação de veículos. Este sistema é essencial para a empresa e ela investirá um valor substancial 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 LVT.

A LVT publicou um edital de ampla concorrência esclarecendo a especificação do sistema que ela precisa, para tanto ela teve o auxílio dos analistas de sistemas 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 LVT, deverá ser feito em duas etapas. E para a primeira etapa ela pede que sejam criadas as seguintes classes apresentadas na Figura 1.

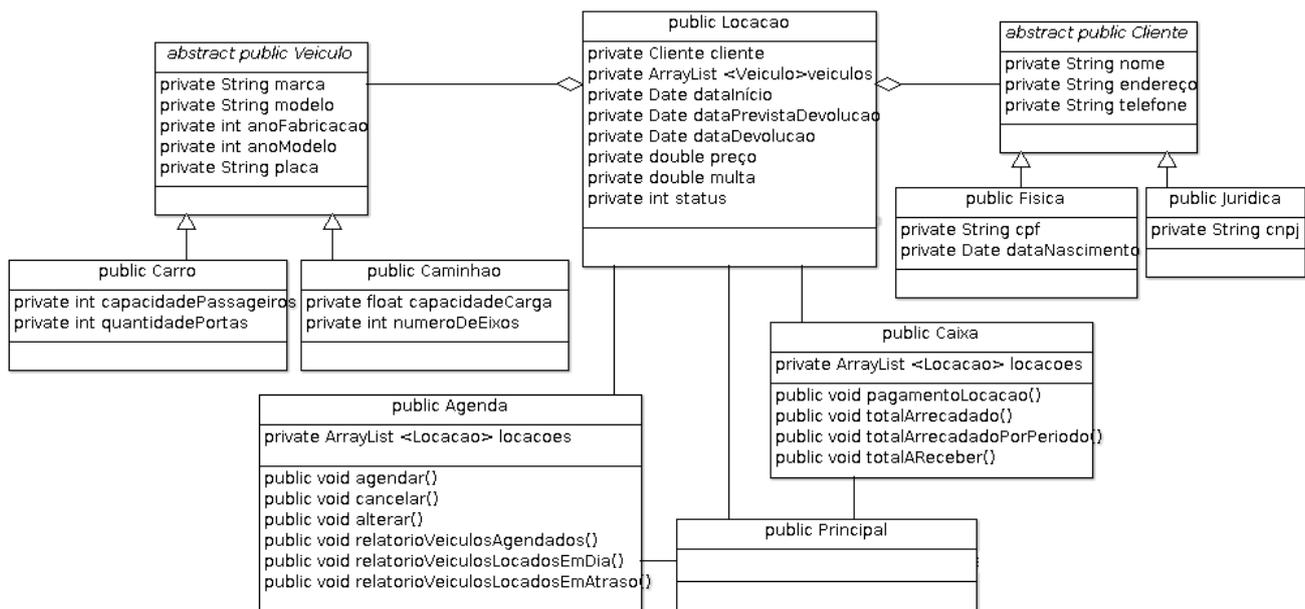


Figura 1: Diagrama de Classes do Sistema

A LVT esclarece que:

- 1) faz parte do projeto a interpretação, pelo grupo, do diagrama UML da Figura 1. Podendo o grupo, questionar sobre qualquer problema que nele for encontrado;
- 2) o tipo "Date" existente em diversos atributos especificados no diagrama da Figura 1 deverá ser substituído pelo tipo "LocalDate", que 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 tipo "LocalDate" e aplicá-lo devidamente no sistema em questão;

- 3) não foram especificados nas classes nenhum método construtor, assim sendo, as empresas poderão desenvolver quantos forem necessários para seus sistemas;
- 4) nenhum atributo poderá ser adicionado às classes do diagrama da Figura 1 sem prévia comunicação e autorização do Professor;
- 5) na classe “Agenda” o método:
 - “agendar” cria um objeto “Locação” para um determinado “Cliente” e um ou mais “Veículos”;
 - “cancelar” cancela um agendamento que tenha sido realizado. Porém, o agendamento não deve ser excluído da base de dados;
 - “alterar” dá a possibilidade de alteração do período de agendamento;
 - “relatorioVeiculosAgendados” lista todos os veículos que foram agendados;
 - “relatorioVeiculosLocadosEmDia” lista todos os veículos que estão locados;
 - “relatorioVeiculosLocadosEmAtraso” lista todos os veículos que foram locados e que ainda não foram devolvidos mas já passaram da data limite de devolução;
- 6) na classe “Caixa” o método:
 - “pagamentoLocacao” efetua o pagamento com ou sem multa de carros que foram locados. A multa, se houver, deverá ser calculada da seguinte forma: 0,28% por dia de atraso (juros sobre juros);
 - “totalArrecadado” deverá calcular e imprimir o total arrecadado deste a abertura da empresa;
 - “totalArrecadadoPorPeríodo” deverá calcular o valor arrecadado, ou seja, os pagamentos que foram efetuados entre um período determinado pelo usuário do *software* (data início e data fim);
 - “totalAReceber” deverá calcular o valor que ainda se tem para receber dos carros que estão alugados e que ainda não foram devolvidos. Lembrando que o cliente só faz o pagamento no momento da devolução do veículo e
- 7) todos os objetos do tipo ArrayList, que serão utilizados no sistema como base de dados deverão ser instanciados na classe “Principal”. As demais utilizações do mesmo, por outras classes, deverão ser apenas referências a estes objetos.

Como o seu *software* deverá funcionar

- crie uma estrutura de *menus* e *submenus* que seja:
 - simples de ser entendida e
 - fácil de usar;
- a última opção do *menu* principal deverá ser “sair”. E somente acessando esta opção do *menu* principal o usuário conseguirá sair do *software*. Nos *submenus* deverão existir uma opção “voltar ao *menu* anterior” para que o usuário possa navegar entre os *menus* existentes sem efetuar nenhuma operação no *software*;

- 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 objeto. Objeto este, escolhido pelo usuário para que o mesmo possa alterá-lo;
 - Exclusão
 - antes de excluir um objeto, deverá ser mostrado todos os dados do objeto que será excluído e emitida uma mensagem de confirmação da operação para o usuário. Caso o usuário confirme a operação, então o objeto deverá ser apagado e uma mensagem de sucesso emitida. Caso contrário o objeto não será apagado;
 - Consulta
 - deverá possibilitar ao usuário do *software* a visualização dos dados de um determinado objeto por ele escolhido;
 - Relatório
 - deverá possibilitar ao usuário do *software* a visualização de todos os dados de todos os objetos cadastrados;
 - Voltar ao *menu* anterior
 - esta opção deverá existir em todos os *submenus*. E possibilitará ao usuário a volta ao *menu* anterior ao qual ele se encontra.

Dicas:

- um *software* deve ter uma boa aparência e ser de fácil utilização, para agradar e facilitar a vida de quem o utilizará e
- 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: eddiesaliba3@gmail.com (de acordo com as regras a seguir);
- o e-mail deverá ser enviado, no máximo, até UM dia antes da data marcada para apresentação;
- **trabalhos entregues após a data marcada para entrega serão avaliados em 5 pontos a menos por dia de atraso e**
- **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.

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;
- conformidade do *e-mail* com o que foi solicitado e
- **entendimento individual a respeito do código-fonte apresentado.**