



Programação Orientada a Objeto

Trabalho Prático - Controle e Venda de Passagens Aéreas

Parte 01

Professor: Edwar Saliba Júnior

Valor: 15 pontos

Contextualização:

A empresa aérea Tabajara Airlines (TA) iniciará, em breve, suas operações no Brasil. Mas, para que possa começar a sua atuação no mercado de voos, ela precisa ter em funcionamento um sistema para controle e venda de passagens aéreas. 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 TA. A TA 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 TA, deverá ser feito em duas etapas. E para a primeira etapa a TA pede que sejam criadas as seguintes classes:

1) Estrutura de classes¹ (Figura 1):

- Cliente,
- Aeroporto,
- Voo,
- Avião e
- Passagem.

2) Atributos das Classes:

- Cliente
 1. identificação
 2. nome
 3. logradouro
 4. número
 5. bairro
 6. município
 7. estado
 8. cep
 9. telefone
 10. cpf
- Avião
 1. identificação
 2. modelo
 3. quantidade de turbinas
 4. capacidade de passageiros
 5. capacidade de carga
- Aeroporto
 1. identificação
 2. nome
 3. município
 4. estado

¹ Neste diagrama não estão representadas as classes de gerenciamento.

- 5. país
- Voo
 1. Avião
 2. Aeroporto de partida
 3. data de partida
 4. horário de partida
 5. Aeroporto de chegada
 6. data de chegada
 7. horário de chegada
 8. lotação (quantos passageiros compraram passagem para este voo)
 9. peso da carga embarcada
 10. valor da viagem
- Passagem
 1. Voo
 2. Cliente
 3. data da venda
 4. hora da venda
 5. preço final da viagem

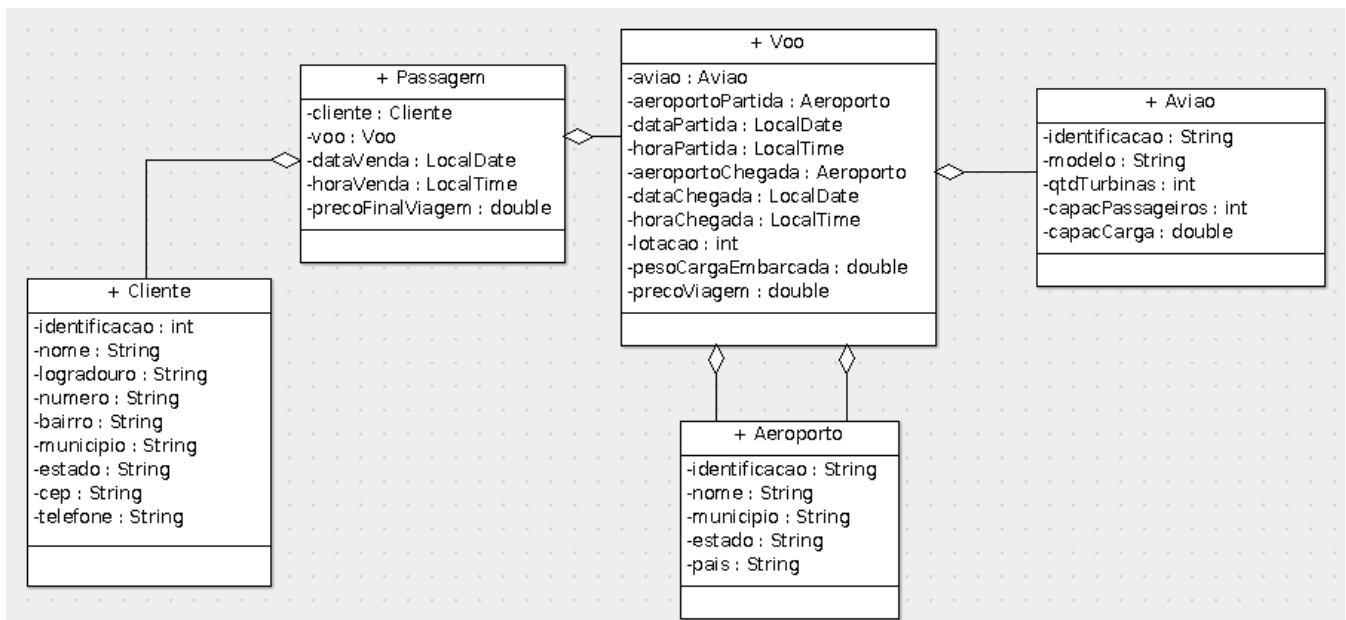


Figura 1: Diagrama de Classes do Sistema

- 3) Caso seja necessário, os grupos (empresas) poderão criar mais atributos nas 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);
- 4) Sua empresa deverá seguir fielmente os nomes das classes, os nomes dos atributos e tipos que foram propostos pela TA na Figura 1. Caso isto não aconteça, haverá uma grande penalização por parte dos avaliadores da TA.
- 5) Neste *software*, o usuário deverá ser capaz de cadastrar clientes, voos, aviões, aeroportos e passagens. Não deverá haver limite de cadastrados e:

- antes de cadastrar uma passagem deverá ser verificado se no voo cabem mais passageiros, pois, não poderá existir *overbooking*². Também deverá ser verificado antes do cadastro, se a carga levada pelo passageiro poderá ser levada no avião, pois, o mesmo também tem um limite de peso que pode levar;
- se a data da compra da passagem for até 10 dias antes da data da viagem, então, ao cliente deverá ser fornecido um desconto de 5,3% no valor da viagem. Se a data da compra da passagem tiver mais de 10 dias de antecedência à data da viagem, então ao cliente deverá ser concedido um desconto de 7,4% no valor da viagem.

6) Crie um *menu* principal com os itens: Cliente, Avião, Aeroporto, Voo, Passagem e Sair. E para cada um destes itens, com exceção do item “Passagem”, deverá existir as seguintes opções (*submenu*):

- Cadastro,
- Alteração
 - A operação de alteração deverá dar a possibilidade do usuário ver os valores já cadastrados para um determinado item do vetor. Item este, escolhido pelo usuário para que o mesmo possa alterá-lo;
- Exclusão
 - Antes de excluir um item do vetor, deverá ser 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 vetor;
- Relatório
 - Deverá possibilitar ao usuário do *software* a visualização de todos os itens do vetor.
- Voltar ao *menu* principal
 - Deverá possibilitar ao usuário a volta ao *menu* principal.

7) Para o item “Passagem”, deverá existir as seguintes opções (*submenu*):

- Venda,
- Cancelamento da Venda,
- Relatórios que mostrem:
 - quantidade de lugares disponíveis num voo;
 - quantidade de carga que ainda é possível colocar em um voo;
 - total arrecadado até o momento com as passagens vendidas para um voo;
 - listagem dos passageiros incluídos em um voo;
 - listagem dos passageiros incluídos em um voo e o valor pago pela viagem de cada um e
 - o valor que a empresa deixará de ganhar:
 - para este cálculo deverá ser multiplicado a capacidade de passageiros do avião pelo preço da viagem (sem descontos) e diminuído do somatório do preço final da viagem de todas as passagens vendidas para um determinado voo.
- Voltar ao *menu* principal
 - Deverá possibilitar ao usuário a volta ao *menu* principal.

Como o seu *software* deverá funcionar

2 Excesso de passageiros, ou seja, venda de passagens acima da capacidade que o avião suporta.

- 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.
- A parte de data deverá ser tratada com objetos do tipo *LocalDate*. Porém, o uso deste objeto não será explicado em sala de aula, cabe ao grupo pesquisar e se inteirar sobre como usar este objeto para manipulação de datas.

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 - Paracatu - ADS3PA - 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 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.**