



Programação Visual

Trabalho Prático 01 – Hotel

Parte Única

Professor: Edwar Saliba Júnior

Valor: 30 pontos

Contextualização:

O Sr. Joseph Jones, primo do Sr. Antônio Jones, morador de um pequeno município do estado de Santa Catarina, trabalhava com seu primo em uma indústria portuária e tinha um bom salário. Porém, a crise que assola o país desde 2017 e agora juntamente à pandemia, tiraram-lhes o emprego e de muitos de seus amigos também. Sendo um homem honesto, chefe de família, empreendedor e consciente de seus deveres enquanto ser humano e cidadão, o Sr. Joseph Jones decidiu pegar o dinheiro de sua rescisão contratual e abrir um negócio para si. Sem saber que negócio iniciar, resolveu procurar o SEBRAE de sua cidade. O Sr. Joseph Jones foi muito bem recebido e atendido pela equipe do SEBRAE, onde coincidentemente encontrou seu primo fazendo uma pesquisa de negócios também. A equipe do SEBRAE informou ao Sr. Joseph Jones que eles haviam terminado uma pesquisa de mercado na cidade para apontar as necessidades locais. E para a grande surpresa do Sr. Joseph Jones, o SEBRAE o aconselhou a abrir um hotel.

O Sr. Joseph Jones, satisfeito com a ideia de abrir um hotel, então começou a providenciar a documentação necessária e procurar um local adequado. E em dois anos já estava com seu negócio funcionando, praticamente em frente do “Jones’ Pet Shop” que foi aberto pelo seu primo, Sr. Antônio Jones.

Sua clientela cresceu satisfatoriamente. Pois o Sr. Joseph Jones, no “Jones’ Hotel”, oferecia um serviço diferenciado a custos satisfatórios a seus clientes.

A fama do “Jones’ Hotel” se espalhou pela cidade e seu negócio começou a prosperar de forma expressiva. Foi então que o Sr. Joseph Jones, que controlava seu negócio de forma totalmente manual, sentiu a necessidade de ter um *software* para controlar as contas dos clientes do hotel.

Conversou com seu primo, que já tinha aberto e infomatizado seu negócio há mais tempo e pediu-lhe conselhos sobre que *software* comprar ou como mandar fazer um *software* específico para o hotel.

Prontamente o seu primo, o Sr. Antônio Jones, dono do “Jones’ Pet Shop”, indicou sua empresa, especializada em desenvolvimento de *software*, ao seu primo. Então o Sr. Joseph Jones procurou sua empresa e encomendou o desenvolvimento do aplicativo. E fez as seguintes exigências: o *software* deverá atender as especificações mínimas (Figura 1) e deverá possuir telas intuitivas, ou seja, deverá ser o mais amigável possível. Deverá fazer uso de bons padrões de implementação e utilizar o SGBD¹ conhecido como PostgreSQL na sua versão final. E por fim, exigiu também que a linguagem de programação seja Java com componentes Swing.

ATENÇÃO: não estão representadas no diagrama da Figura 1, as classes “fichário” e tampouco as DAO², esta última para manipulação do SGBD. Este diagrama está representando exclusivamente o núcleo do sistema (classes modelo). Sua empresa deverá criar fielmente as classes e os respectivos atributos de cada uma delas existentes na Figura 1. Caso seja realmente necessário, sua empresa poderá criar novos atributos e/ou novas classes para o sistema.

1 Sistema Gerenciador de Banco de Dados

2 Data Access Object

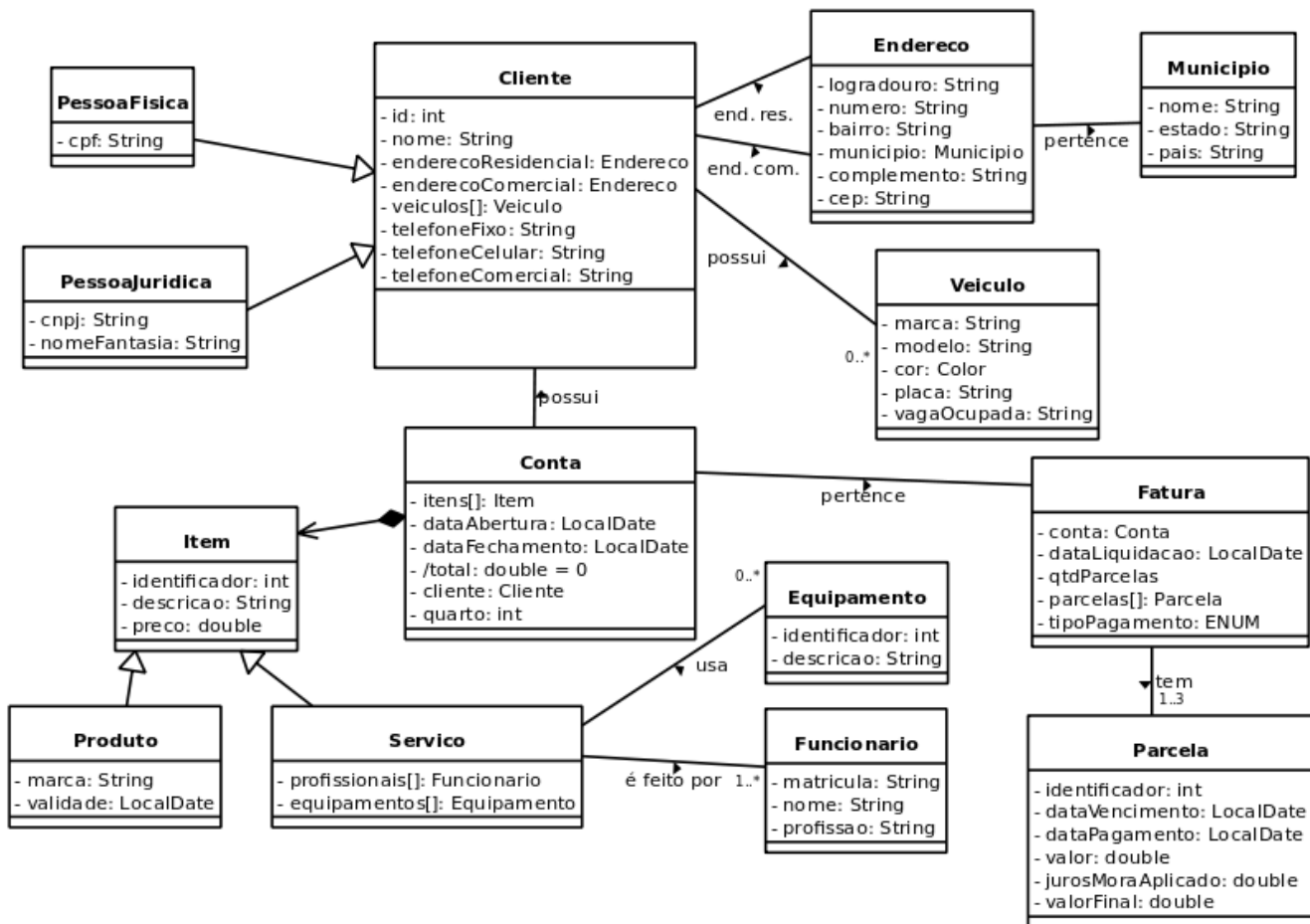


Figura 1: Diagrama de classes do sistema.

Tarefas:

- 1) o *software* deverá ser todo visual e orientado a objetos;
- 2) para facilitar o desenvolvimento do sistema, sua empresa deverá implementar o *software*, em princípio, sem SGBD. Ou seja, guardando os objetos em memória principal;
- 3) para facilitar manutenções futuras, exige-se o padrão MVC (*Model, View, Controller*) e a utilização das classes “fichário” na implementação do *software*. O assunto MVC não será lecionado neste curso, sendo parte do trabalho a pesquisa e utilização do conceito;
- 4) o *software* deverá também, possuir uma tela de login com usuário e senha e deverá fazer um controle de usuários (não especificado na Figura 1). Sendo que a senha deverá ficar criptografada (encriptação de mão única). O assunto criptografia tampouco será apresentado nesta disciplina, sendo parte do trabalho, também, a pesquisa e utilização do conceito;
- 5) as telas do *software* deverão possibilitar, para cada CRUD (*Create, Retrieve, Update e Delete*):
 - a visualização de todos os objetos cadastrados por meio de um JTable, mesmo que os dados sejam exibidos de forma resumida. Ou seja, mostrando apenas os campos principais;
 - a possibilidade de inclusão de novos objetos e
 - a possibilidade exclusão e alteração de objetos.
- 6) o *software* deverá consistir e validar as exclusões, ou seja: ao tentar excluir um determinado objeto, o *software* deverá verificar se este não está sendo referenciado/usado

por outro(s) objeto(s). Pois, se estiver deverá ser emitida uma mensagem de erro para o usuário e o objeto não deverá ser excluído. Exemplo: não se pode excluir um Município se algum Endereço já o estiver referenciando/utilizando; ou ainda, não se pode excluir um Serviço ou Produto, se estes já estiverem na Conta de algum Cliente e assim por diante;

- 7) se uma Conta estiver com o atributo “dataFechamento” preenchido, então, esta não poderá sofrer mais nenhum tipo de alteração e deverá ser emitida uma Fatura (parcelada em até 3 vezes, à escolha do Cliente). A emissão da Parcela é obrigatória mesmo que o Cliente deseje pagar à vista (1 parcela). Caso ocorra o parcelamento da “Fatura” em mais de uma vez, então deverá ser calculado juros. Que obedecerão as regras que serão estabelecidas mais adiante;
- 8) o atributo “tipoPagamento” na classe “Fatura” diz respeito ao tipo de pagamento, ou seja:
- dinheiro,
 - cartão de crédito,
 - cartão de débito ou
 - Pix.

Para a manipulação deste atributo deverá ser criado um “Tipo Enumerado” ou “Enumeração” em Java. 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.

- 9) O campo “dataLiquidacao” 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 ou não recebidas por completo.
- 10) O Cliente poderá parcelar sua “Fatura / Conta”. Porém, o mesmo arcará com as seguintes condições, de acordo com os seguintes tipos de pagamento:
- dinheiro ou Pix - deverá ser calculado 2% ao mês, juros sobre juros;
 - cartão de débito - deverá ser calculado 4% ao mês, juros sobre juros e mais 1,3% sobre o valor final;
 - cartão de crédito - deverá ser calculado 6% ao mês, juros sobre juros e mais 1,8% sobre o valor final. Estes cálculos são de responsabilidade do método “JurosPorParcelamento” que deverá existir na classe “Fatura”.

Exemplo de cálculo:

- supondo que o cliente tenha feito um conta de R\$300,00, queira dividi-la em 3 vezes (30, 60 e 90 dias) e pagará todas as parcelas com cartão de crédito:
 - ◆ mês 01 → $R\$100,00 * 1.06 = R\$106,00$
 - ◆ mês 02 → $R\$106,00 * 1.06 = R\$112,36$
 - ◆ mês 03 → $R\$112,36 * 1.06 = R\$119,10$
 - ◆ soma-se as 3 parcelas $106 + 112,36 + 119,1 = R\$337,46$
 - ◆ aplica-se o juros do cartão $337,46 * 1.018 = R\$343,53$
 - ◆ e divide-se por 3 para achar o valor final de cada parcela = **R\$114,51**

- 11) 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 do método “JurosPorAtraso” que deverá existir na classe “Fatura” e
- 12) o atributo “jurosMoraAplicado” da “Parcela” deverá constar o valor aplicado de juros sobre a parcela, caso esta seja paga com atraso. Caso contrário a este atributo deverá ser atribuído o valor zero;

- 13) ao atributo “valorFinal” da “Parcela” deverá ser atribuído o somatório do atributo “valor” e “jurosMoraAplicado”. O atributo “valorFinal” sempre deverá ser preenchido, uma vez que a parcela seja paga. Caso contrário deverá conter o valor zero.
- 14) Para o armazenamento dos objetos em memória principal, devem ser utilizados objetos *ArrayList*. Exceto para os atributos de objetos que tiverem limite de armazenamento especificado no diagrama da Figura 1), pois, para estes deverão ser utilizados vetores.
- 15) O hotel possui apenas 15 quartos (de 1 a 15), então, o software deverá fazer o controle de quartos ocupados por meio dos objetos “Conta”. Ou seja, conta com o atributo “dataAbertura” preenchido e “dataFechamento” não preenchido, significa quarto ocupado. Ambos os atributos sem preenchimento ou ambos preenchidos, significa quarto livre. O *software* deverá prover um recurso visual da ocupação dos quartos, para facilitar a utilização pelo usuário.
- 16) **uma versão prévia e funcionando deste *software* deverá ser apresentada, sem SGBD, até o dia 02/05/2022. Esta versão deverá ser enviada por e-mail para o Professor, assim como a versão final, seguindo as regras no final deste documento.**
- 17) **ATENÇÃO! Somente após a apresentação da versão do *software* sem SGBD, poderá ser iniciada a implementação do mesmo com SGBD e relatórios;**
- 18) deverão ser feitos relatórios utilizando o JasperStudio para, no mínimo, as seguintes telas:
 - listagem de Itens (Produtos e Serviços) cadastrados,
 - listagem de Clientes e seus respectivos Pets cadastrados,
 - conta por Cliente (cliente escolhido pelo usuário do *software*) e
 - listagem de Agendamentos para uma determinada data a ser definida pelo usuário na emissão do relatório;
- 19) deverá existir um ou mais botões/campos específicos, em cada tela, para a emissão dos relatórios supracitados e
- 20) **a versão final do *software*, já com o SGBD funciona implementado, deverá ser entregue até o dia 14/06/2022.**

Regras para a entrega do trabalho

- Deverá ser apresentado e entregue, o projeto (compactado) do código-fonte.
- **O código-fonte que será entregue, apresentado e não deverá possuir nenhum tipo de comentário:**
 - antes de compactar o seu projeto de *software*, lembre-se de apagar a pasta “dist”, caso ela exista na estrutura de pastas do projeto ou a pasta que contém os arquivos “.class” (“compilados”).
- **Deverá ser entregue um arquivo, na forma de *back-up* do banco de dados.**
- **Deverá ser entregue um *script* de criação do banco de dados.**
- Tudo deverá ser enviado para o e-mail: eddiesaliba2@yahoo.com (de acordo com as regras a seguir).
- **Trabalhos enviados após a data marcada para entrega, serão avaliados em 5 pontos a menos, subtraídos do valor total, para cada dia de atraso. Ou seja, se atrasar 1 dia, o trabalho será avaliado em 25 pontos. Se atrasar 2 dias, será avaliado em 20 pontos e assim por diante.**
- **A apresentação deverá ser realizada nos horários de aula da disciplina. Uma cópia do trabalho enviado, incluindo o *back-up* e o *script* do banco de dados, deverá ser levada pelo grupo em pendrive. 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 - PV - 2022/01 - Trabalho 01 - Parte XX** (substitua XX por 01 ou 02 de acordo com a parte que você estiver enviando);
- no corpo do *e-mail* deverá conter, **única e exclusivamente**, o **nome completo** de todos os integrantes do grupo (um nome por linha);
- só será aceito **UM** *e-mail* por aluno. Portanto, verifique se está tudo certo com seu *e-mail* e trabalho antes de enviá-lo.

Observação: o desrespeito a qualquer das regras ou tarefas acima implicará na perda de créditos.

Critérios de avaliação do *software*:

- conformidade do *software* em relação ao solicitado;
- legibilidade do código (**organização, endentação e etc.**);
- usabilidade das interfaces do *software* e
- **entendimento individual a respeito do código-fonte apresentado.**