Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

Programação Visual

Trabalho Prático – Estacionamento – Parte 02

Professor: Edwar Saliba Júnior Valor: 15 pontos

Contextualização:

Essa nova modalidade de estacionamento foi um sucesso nas cidades onde foi implantada e agora seus proprietários pretendem expandir o negócio. Porém, para esta nova fase do negócio eles pretendem atingir um novo público, ou seja, anteriormente o público-alvo era apenas carros pequenos, agora os proprietários da empresa querem estender o atendimento a caminhões e motocicletas. Para tanto, solicitou a sua empresa que ampliasse as funcionalidades do *software* de modo que atenda à nova demanda.

Sua empresa aceitou a proposta e, após algumas análises, decidiu que a nova estrutura de classes do *software* será a apresentada a seguir. Continuando a análise, sua empresa chegou a conclusão de que, para melhorar a capacidade de armazenamento, confiabilidade, segurança dos dados e também para que o usuário possa desligar o computador ao final do expediente, o melhor a se fazer é armazenar todos os dados em um Sistema Gerenciador de Banco de Dados (SGBD). Para tanto, sua empresa, com a experiência adquirida em outros tempos, decidiu usar o SGBD conhecido como PostgreSQL.

A nova proposta foi apresentada aos proprietários do estacionamento e os mesmos a aprovaram. No entanto, exigiram a entrega do código fonte do sistema e uma imagem do Diagrama de Entidades e Relacionamentos (DER) do banco de dados.

1) Estrutura de classes[1] (Figura 1):
    - Cliente,
    - Veículo
        1. Motocicleta
        2. Carro
        3. Caminhão
    - Pátio e
    - Conta.

2) Atributos das Classes:
    - Cliente
        1. código
        2. nome
        3. logradouro
        4. número
        5. bairro
        6. município
        7. estado
        8. cep
        9. telefone
    - Veículo
        1. marca
        2. modelo

1 Neste diagrama não estão representadas as classes de gerenciamento.

    3. ano de fabricação
    4. ano do modelo
    5. chassi
    6. placa
- Carro
    1. número de portas
    2. quantidade de passageiros
- Caminhão
    1. número de eixos
    2. capacidade de carga
- Motocicleta
    1. cilindradas
    2. quantidade de rodas
- Pátio
    1. nome
    2. logradouro
    3. número
    4. bairro
    5. município
    6. estado
    7. cep
    8. telefone
    9. capacidade de veículos
    10. valor da diária para motocicletas
    11. valor da diária para carros
    12. valor da diária para caminhões
- Conta
    1. Pessoa
    2. Veículo
    3. Pátio
    4. ano
    5. mês
    6. diárias
    7. paga

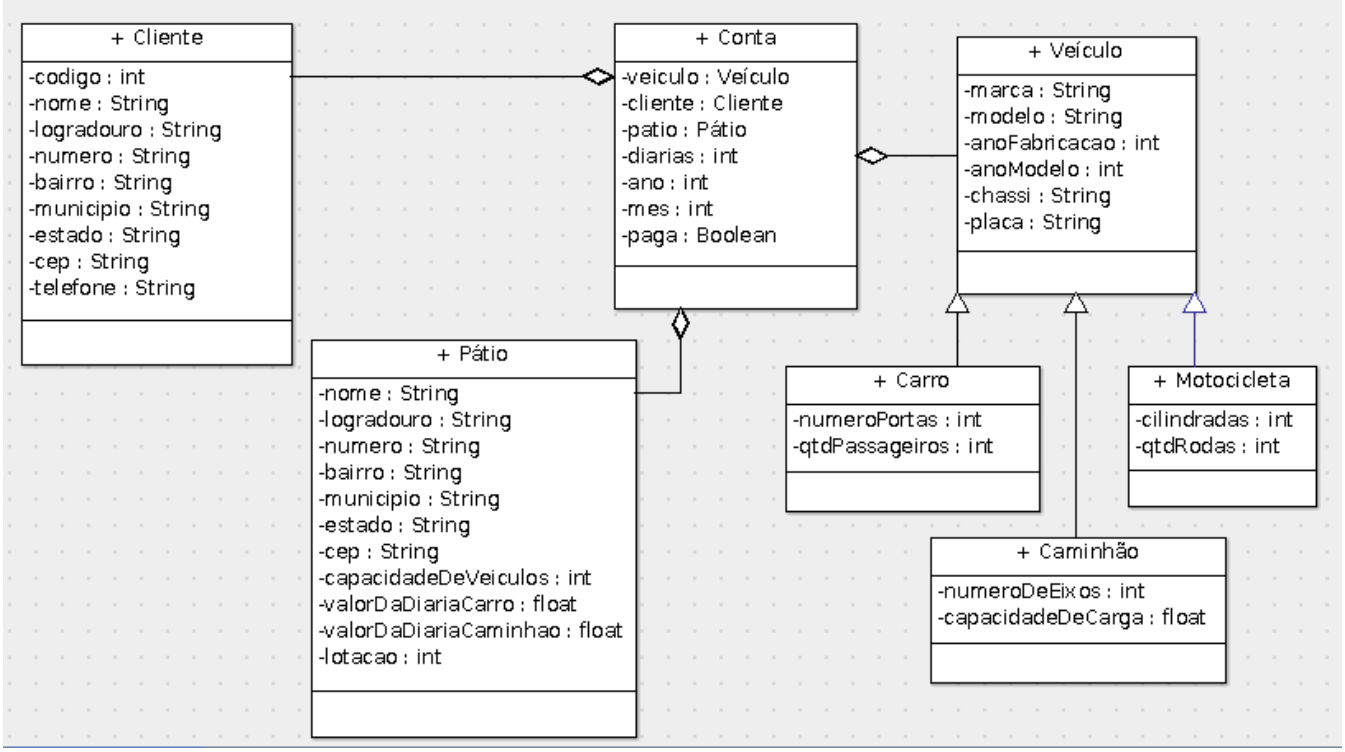

Figura 1: Diagrama UML

3) Caso seja necessário, os grupos 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) Refaça o que for necessário no *software* já existente para que o mesmo contemple a nova estrutura de classes e suporte o uso do SGBD.

5) Atenção! Crie também, os seguintes relatórios para auxiliar o usuário do *software*:

    1. **Clientes Simples** (deverá imprimir o código, o nome do cliente e o telefone de todos os clientes cadastrados);
    2. **Cliente Completo** (deverá imprimir todos os dados de todos os clientes cadastrados);
    3. **Veículo Simples** (deverá imprimir o código, a marca, o modelo e o tipo [caminhão, motocicleta ou carro] de todos os veículos cadastrados);
    4. **Veículo Completo** (deverá imprimir todos os dados dos veículos cadastrados;
    5. **Pátio Completo** (deverá imprimir todos os dados de todos os pátios cadastrados);
    6. **Contas a Receber no Período** (deverá imprimir, por pátio, o total a ser recebido (contas em aberto) no período que o usuário determinar (data início e data fim);
    7. **Contas a Receber do Cliente** (deverá imprimir todos os dias que o usuário usou o estacionamento, qual carro e em qual pátio foi, totalizando o valor da conta) e

8. **Percentual de Utilização do Estacionamento** (deverá imprimir por pátio, o percentual de utilização do estacionamento por motocicletas, carros e caminhões, no período [data início e data fim] estipulado pelo usuário).

Para criação dos relatórios acima descritos, utilize o gerador de relatórios conhecido como "**iReport**".

Como o seu *software* deverá funcionar

- A estrutura de telas, deverá proporcionar ao usuário:
  - a possibilidade de navegar entre as telas e subtelas sem efetuar qualquer operação;
  - o usuário só poderá sair do programa através do *menu* principal, ou seja, acessando a opção "Sair" ou por meio do botão de sistema "X";
- Um *software* deve ter uma boa aparência e ser de fácil utilização, para agradar e facilitar a vida de quem o utilizará.

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 entre também, um arquivo contendo o script de criação do banco de dados.**
- **O Diagrama de Entidade e Relacionamento (DER) do banco de dados.**
- 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: **IFTM – PV – Parte 02 – Nome da Empresa Fictícia**
- No corpo do *e-mail* deverá conter, única e exclusivamente, o nome 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, estruturação, uso de boas práticas de programação e etc.);
- Usabilidade das interfaces de interação com o usuário;
- **Entendimento <u>individual</u> a respeito do código-fonte apresentado.**