



# Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro

## Programação Visual

### Trabalho Prático - Estacionamento - Parte 01

**Professor:** Edwar Saliba Júnior

Valor: 15 pontos

#### Contextualização:

Precisa-se de um *software* para controle de um estacionamento. Este estacionamento é um pouco diferente dos estacionamentos tradicionais em sua forma de trabalhar. Primeiramente, este estacionamento só recebe veículos pequenos, ou seja, carros normais de passeio ou trabalho (carros, pick-ups, vans e etc.). Neste estacionamento todos os clientes, seus respectivos veículos e até os pátios existentes para estacionar são controlados. O cliente paga sempre o valor de uma diária, mesmo que seu veículo fique apenas 5 minutos no estacionamento. No entanto, os clientes deste estacionamento têm duas vantagens: a primeira é que ele só pagará pelos dias que utilizar o estacionamento e a segunda é que ele tem sua vaga garantida o mês inteiro. Os pagamentos são realizados pelos clientes no último dia de cada mês. Para desenvolver este *software* a estrutura mínima que deverá ser utilizada é a seguinte:

#### 1) Estrutura de classes<sup>1</sup> (Figura 1):

- Cliente,
- Pátio,
- Veículo 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
  3. ano de fabricação
  4. ano do modelo
  5. chassi
  6. placa
- Pátio
  1. nome
  2. logradouro
  3. número
  4. bairro
  5. município

<sup>1</sup> Neste diagrama não estão representadas as classes de gerenciamento.

- 6. estado
- 7. cep
- 8. telefone
- 9. capacidade de veículos
- 10. valor da diária
- Conta
  - 1. Pessoa
  - 2. Veículo
  - 3. Pátio
  - 4. ano
  - 5. mês
  - 6. diárias
  - 7. paga

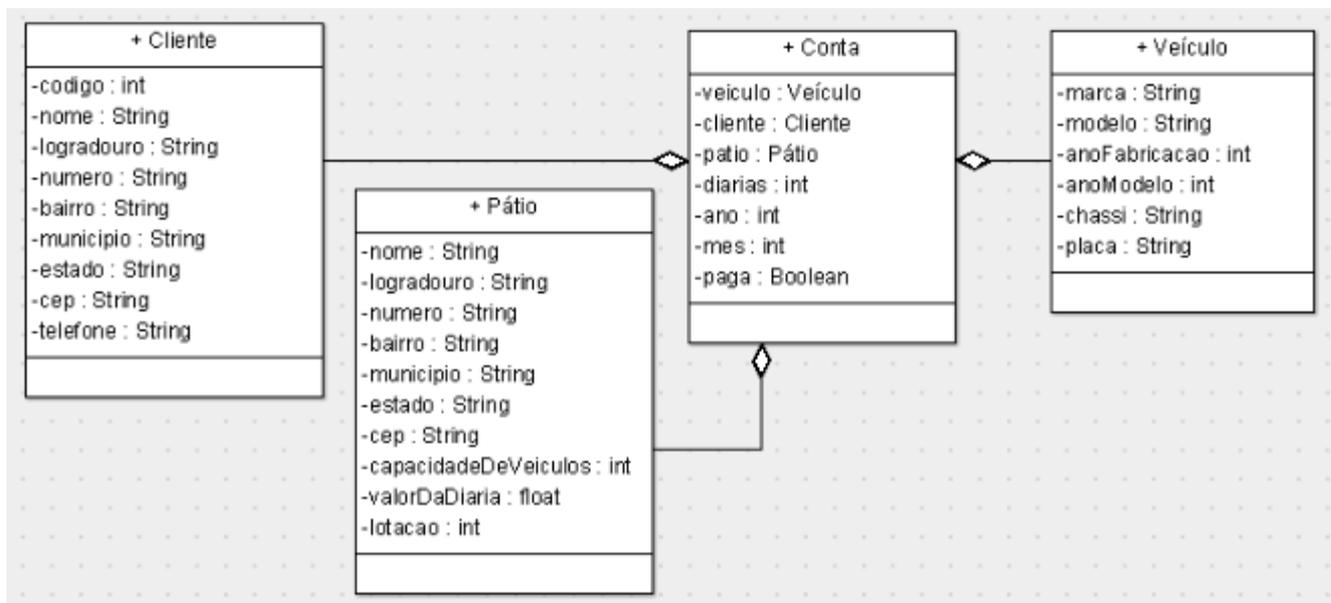


Figura 1: Diagrama UML

- 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) Caso seja necessário, os grupos poderão criar mais itens no *submenu* de Conta;
- 5) Neste *software*, o usuário deverá ser capaz de cadastrar até clientes, veículos, pátios e até contas por pátio, para tanto, faça uso de ArrayList.
- 6) Crie telas e um *menu* que permitam que os usuários do *software* possam manusear com facilidade: Clientes, Veículos, Pátios e Contas. As operações básicas de cadastro (inserção, alteração, exclusão e consulta) deverão existir para todas as classes de domínio. Sendo que a classe Conta, deverá ter também um mecanismo para armazenar as diárias dos usuários, fazer fechamento mensal, gerar débito e proporcionar o pagamento do mesmo.

### Como o seu *software* deverá funcionar

- A estrutura de telas, deverá proporcionar ao usuário:
  - a possibilidade de navegar entre as telas sem efetuar qualquer operação;
  - o usuário só poderá sair do programa através do *menu* principal, ou seja, acessando a opção “Sair” ou da tecla de sistema “X” (em sistemas Windows, no canto superior direito da tela).
- 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

- A equipe deverá criar uma empresa fictícia. Para esta empresa deverá ser criado um nome.
- 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](mailto: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 - PV - Parte 01 - Nome da Empresa Criada**
- 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 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, indentação, uso de boas práticas de programação e etc.);
- Usabilidade das interfaces de interação com o usuário;
- **Entendimento individual a respeito do código-fonte apresentado.**