



UNIVERSIDADE ANHANGUERA

TABOÃO DA SERRA

**PORTIFÓLIO – RELATÓRIO DE AULA PRÁTICA
ANÁLISE ORIENTADA A OBJETOS**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

Helen Barros Lopes / RA: 3474905901

Tutor Presencial: Welton

Tutor a Distância: Cristiano Giroldo

SUMÁRIO

INTRODUÇÃO.....	3
MÉTODO E DESENVOLVIMENTO.....	4
ATIVIDADE PROPOSTA.....	6
RESOLUÇÃO DO PROBLEMA.....	7
CONCLUSÃO.....	9
REFERÊNCIAS.....	10

1 INTRODUÇÃO

O conceito de orientação a objetos surgiu com o intuito de minimizar os problemas encontrados até então na criação de softwares complexos, projetados por meio de decomposição funcional e sub-rotinas.

Podemos identificar como um dos maiores problemas a não existência de encapsulamento lógico para operações e dados, o que leva a não existência da divisão de tarefas por responsabilidades. O que leva a construção de longos trechos de código, muitas vezes difíceis de compreender devido ao acúmulo de responsabilidade que lhe é atribuído.

Por consequência, quanto mais complexo o software se torna, mais difícil se torna também a sua manutenção. Com isso aumentam os custos e o risco de confiabilidade do mesmo.

Nessa atividade irei mostrar o diagrama de classe, que é muito utilizado em orientação a objetos, e, como ele funciona, suas vantagens e desvantagens de utiliza-los. Irei desenvolver um diagrama de uma locadora de veículos com todas as classes que o compõe e seus atributos.

2 MÉTODO E DESENVOLVIMENTO

Em UML, diagramas de classes são um dos seis tipos de diagramas estruturais. Os diagramas de classe são fundamentais para o processo de modelagem de objetos e modelam a estrutura estática de um sistema.

Dependendo da complexidade de um sistema, é possível utilizar um único diagrama de classe para modelar um sistema inteiro ou vários diagramas de classe para modelar os componentes de um sistema.

Os diagramas de classe são as cópias do sistema ou subsistema. Você pode utilizar os diagramas de classe para modelar os objetos que compõem o sistema, para exibir os relacionamentos entre os objetos e para descrever o que esses objetos fazem e os serviços que eles fornecem.

Em um projeto de software orientado a objetos, os diagramas de classe criados durante os estágios iniciais do projeto contêm classes que normalmente são convertidas em classes e objetos de software reais quando você grava o código. Posteriormente, é possível refinar a análise e os modelos conceituais anteriores em diagramas de classe que mostrem as partes específicas do sistema, interfaces com o usuário, implementações lógicas e assim por diante. Os diagramas de classe tornam-se, então, uma captura instantânea que descreve exatamente como o sistema funciona, os relacionamentos entre os componentes do sistema em vários níveis e como planeja programar esses componentes.

Os seguintes tópicos descrevem elementos de modelos nos diagramas de classes:

- Classes

Uma classe representa um objeto ou um conjunto de objetos que compartilham uma estrutura e um comportamento comuns.

- Objetos

Os objetos são elementos de modelo que representam instâncias de uma classe ou de classes. Você pode incluir objetos no modelo para representar instâncias concretas e prototípicas.

- Pacotes

Os pacotes agrupam elementos de modelos relacionados de todos os tipos, incluindo outros pacotes.

- Sinais

Sinais são elementos do modelo independentes dos classificadores que os manipulam. Os sinais especificam comunicações assíncronas de uma via entre objetos ativos.

- Enumerações

Enumerações são elementos do modelo em diagramas de classes que representam tipos de dados definidos pelo usuário.

- Tipos de Dados

Tipos de dados são elementos de modelos que definem valores de dados. Você geralmente usa tipos de dados para representar tipos primitivos, como tipos inteiros ou de cadeia, e enumerações, como tipos de dados definidos pelo usuário.

- Artefatos

Artefatos são elementos de modelo que representam as entidades físicas em um sistema de software, como por exemplo, arquivos executáveis, bibliotecas, componentes de software, documentos e bancos de dados.

- Relacionamentos em Diagramas de Classe

Um relacionamento UML é um tipo de elementos de modelo que inclui semântica em um modelo, definindo a estrutura e o comportamento entre os elementos de modelo.

- Qualificadores em Extremidades da Associação

Qualificadores são propriedades de associações binárias e são uma parte opcional de extremidades de associação. Um qualificador mantém uma lista de atributos de associações, cada um com um nome e um tipo. Os atributos de associação modelam chaves que são usadas como um subconjunto de instâncias de relacionamento.

3 ATIVIDADE PROPOSTA

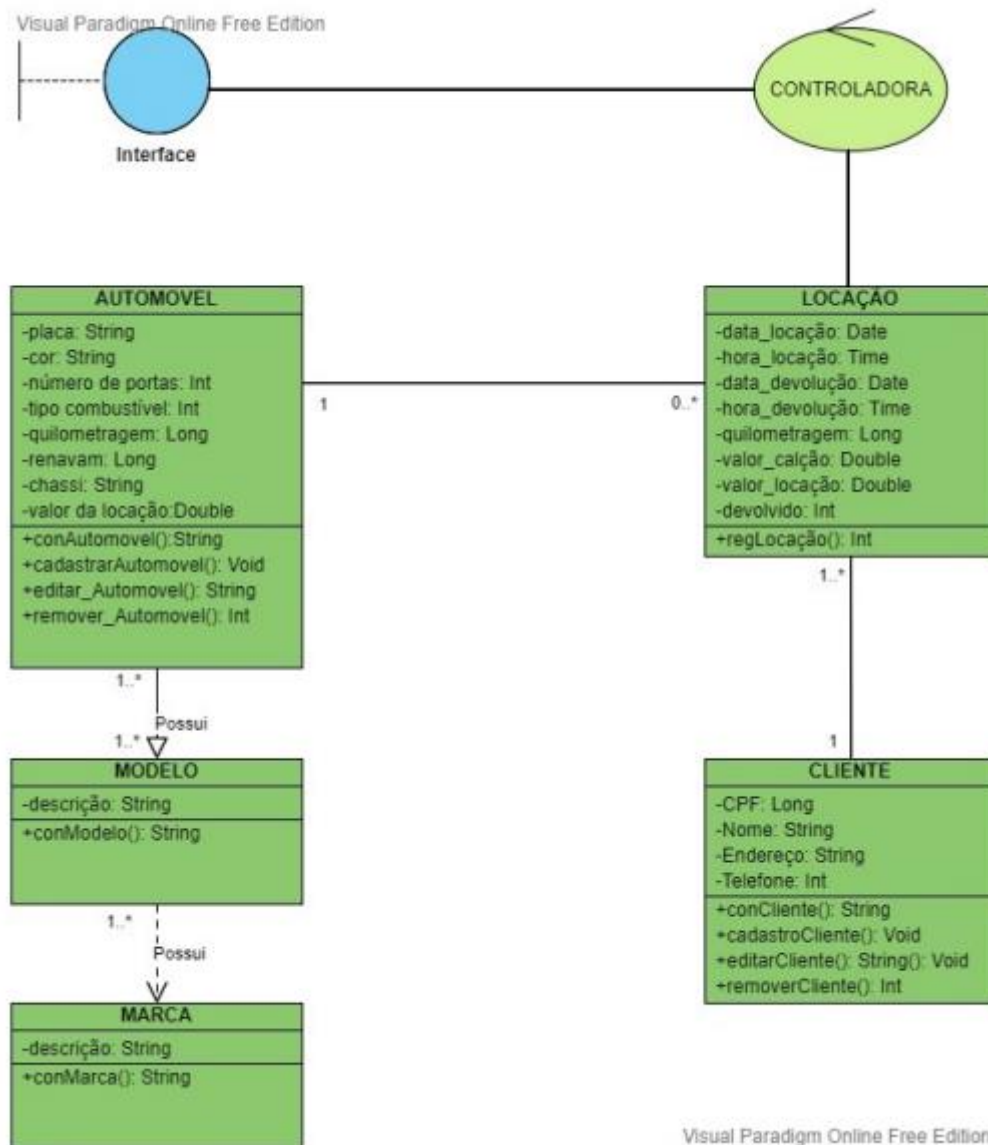
Desenvolva um diagrama de classes para um sistema de locação de veículos, levando em consideração os seguintes requisitos:

- A empresa tem muitos automóveis. Cada automóvel tem atributos como número da placa, cor, ano, tipo de combustível, número de portas, quilometragem, RENAVAM, chassi, valor de locação etc.
- Cada carro tem um modelo e uma marca, mas um modelo pode relacionar-se a muitos carros e uma marca pode referir-se a muitos modelos, embora cada modelo só tenha uma marca específica.
- Um carro pode ser alugado por muitos clientes, em momentos diferentes, e um cliente pode alugar muitos carros. É preciso saber quais carros estão locados ou não. Sempre que um carro for locado é preciso armazenar a data e hora de sua locação e, quando for devolvido, a data e hora de devolução.

4 RESOLUÇÃO DO PROBLEMA

Para fazer esse diagrama de classes, foi utilizada a plataforma Visual Paradigma Online.

A seguir tem a resolução do diagrama de classes:



Foram definidas todas as classes que compõe um sistema de locação.

Um automóvel ele possui um ou mais modelos, e um ou mais modelos podem ter mais de uma marca. Mas só pode ter uma locação de automóvel por vez pelo cliente.

Mas um cliente pode fazer uma ou mais locações se ele não tiver nenhuma pendência. Isso pode ser definido pelo atributo `conCliente` (Consultar Cliente).

Pode ter muitos outros atributos para compor essas classes, mas aqui foi colocado essencial para um sistema ser executado de forma funcional para aquilo que foi proposto.

As classes são compostas por nome (obrigatório), atributos e operações.

Classes são descritas via suas propriedades, que podem ser primitivas – representadas via atributos – e composta – representada como associação para outras classes. Quando transformadas para código, as propriedades se tornam sempre campos de classe.

Pode ser observado também que em cada atributo tem um tipo, que corresponde o tipo que será utilizado no código fonte (string, date, void, int, double, etc).

Fique ciente de que o nome utilizado para o atributo corresponde ao nome que será utilizado no código fonte. É aceitável utilizar nomes com espaços e acentos na fase de análise.

Mas qual a vantagem e desvantagem de utiliza-lo?

A vantagem é que você precisa conhecer uma pequena parte da linguagem para usa-la. Apesar de existir muitos tipos de diagramas UML, os desenvolvedores utilizam apenas três ou quatro para documentar um sistema de software.

Os diagramas de classe, diagramas de sequências e diagrama de casos de uso ainda são os mais comuns. O que isso implica é que você precisa conhecer 20% da linguagem para explicar 80% das suas necessidades de modelagem. Não é necessário conhecer ou compreender toda a notação, para se comunicar de forma eficaz usando o diagrama UML.

A desvantagem é que geralmente não são documentos mantidos sempre atualizados para que funcionem bem como documentação em longo prazo, até porque, muito das vezes são feito em um quadro branco para uso imediato, e em seguida são apagados, perdendo toda aquela documentação.

5 CONCLUSÃO

Com essa atividade prática podemos ver qual o objetivo de usar um Diagrama de Classes, que nada mais é que descrever o modelo geral de informação de um sistema, que resultam de um processo de abstração através do qual se identificam os objetos relevantes no contexto que se pretende modelares e se procuram descrever características comuns em termos de propriedades (atributos) e comportamentos (operações).

6 REFERÊNCIAS

GOÉS, Wilson Moraes. **Aprenda UML por Meio de Estudos de Caso**. 1ª edição. Novatec Editora. 2014.

SIGNIFICADOS. **Diagrama de Classes**. Disponível em:
<https://www.significados.com.br/diagrama-de-classes/>

EDISCIPLINAS. **Programação Orientada a Objetos**. Disponível em:
[https://edisiplinas.usp.br/pluginfile.php/4247716/mod_resource/content/1/Aula3_UML.p
df](https://edisiplinas.usp.br/pluginfile.php/4247716/mod_resource/content/1/Aula3_UML.pdf)

DEVMEDIA. **Artigo Engenharia de Software 2 – Análise Orientada a Objetos**. Disponível em: <https://www.devmedia.com.br/artigo-engenharia-de-software-2-analise-orientada-a-objetos/9150>