



JOÃO PAULO SANTOS SOUZA

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

**BETIM - MG
2024**

JOÃO PAULO SANTOS SOUZA

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

Relatório de aula prática avaliativo do curso de Engenharia de Software da disciplina Análise Orientada a Objetos.

Tutor: Vinicius Camargo Prattes

**BETIM - MG
2024**

LISTA DE ILUSTRAÇÕES

Imagem 2-1-Print do Site	4
Imagem 2-2-Print do Projeto no Site.....	5
Imagem 3-1-Diagrama UML do Projeto	6

SUMARIO

1	INTRODUÇÃO	4
2	MÉTODOS	4
2.1	PREPARAÇÃO DO AMBIENTE:.....	4
2.2	CONSTRUÇÃO DO DIAGRAMA DE CLASSES:.....	5
2.3	RESOLUÇÃO DO PROBLEMA PROPOSTO:	5
3	RESULTADOS.....	6
1.	CLASSE MODELO.....	6
2.	CLASSE FABRICANTE.....	7
3.	CLASSE AUTOMÓVEL.....	7
4.	CLASSE CLIENTE	8
5.	CLASSE LOCAÇÃO	8
4	CONCLUSÃO	9
	REFERÊNCIAS	3

1 INTRODUÇÃO

A análise orientada a objetos (OOA) é uma abordagem essencial no desenvolvimento de sistemas de software modernos. Este trabalho aborda o desenvolvimento de um diagrama de classes utilizando a Unified Modeling Language (UML), com o objetivo de modelar um sistema de locação de veículos. A atividade prática, desenvolvida em aula, explorou os conceitos fundamentais da modelagem orientada a objetos, proporcionando uma experiência prática no uso de ferramentas como o *Visual Paradigm Online*. A construção de um diagrama de classes auxilia na identificação das entidades principais, suas relações e atributos, garantindo clareza no processo de desenvolvimento de sistemas.

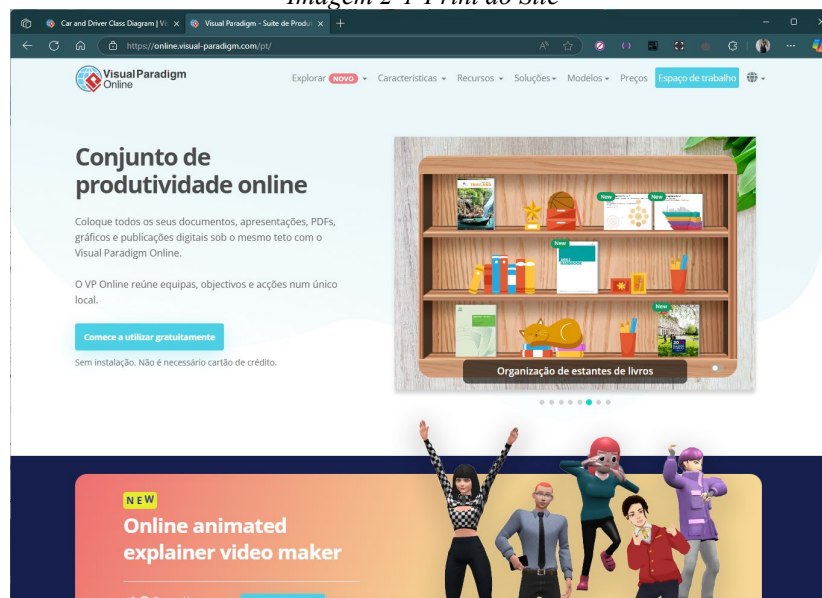
2 MÉTODOS

A metodologia seguiu um roteiro previamente estruturado, incluindo as seguintes etapas:

2.1 Preparação do Ambiente:

- Utilização de computadores com acesso à internet.
- Adoção do software *Visual Paradigm Online* (versão gratuita), acessível em <https://online.visual-paradigm.com/>.

Imagem 2-1-Print do Site



Fonte: Elaborada pelo autor (2024)

2.2 Construção do Diagrama de Classes:

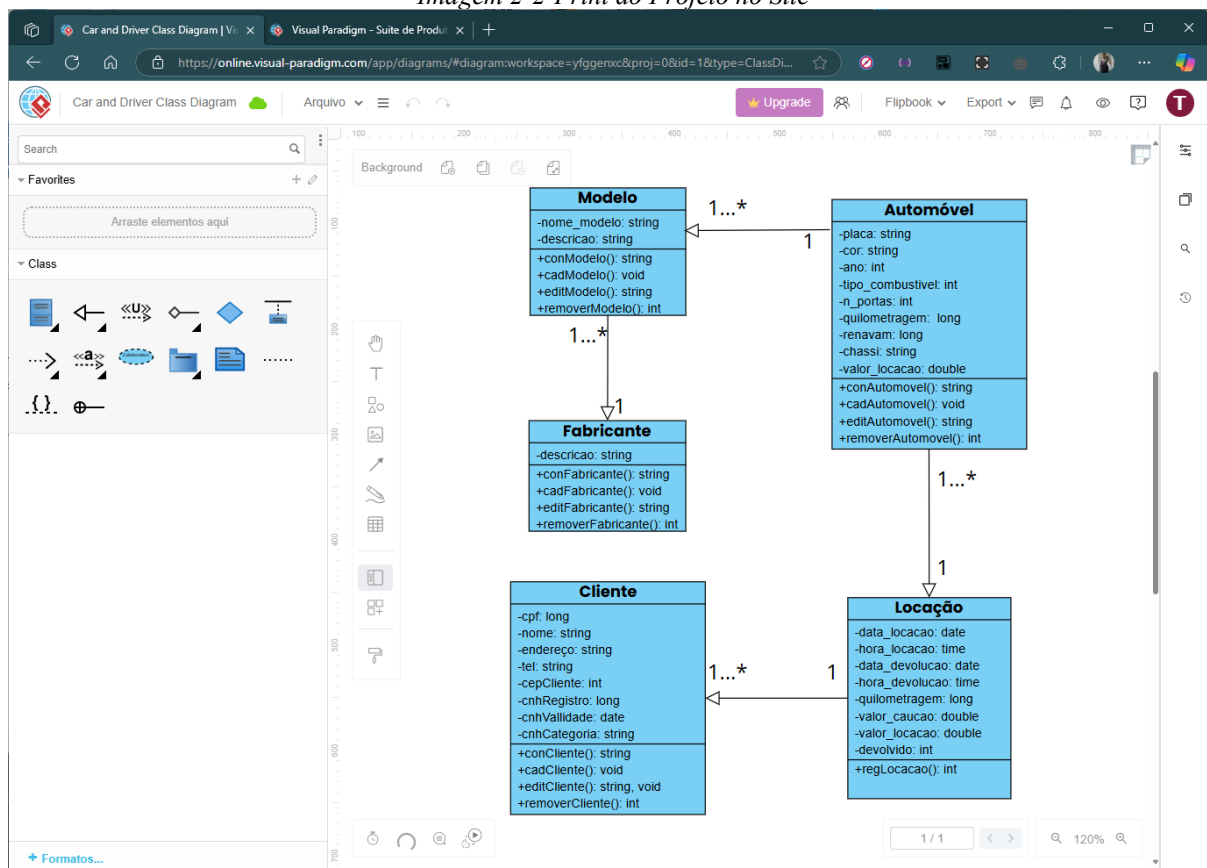
- Acessar o software e criar um novo projeto. Na tela inicial, selecionar a opção "Class Diagram" (Diagrama de Classes).
- Inserir as classes identificadas no problema proposto, adicionando seus atributos e métodos.
- Estabelecer relações entre as classes com associações, generalizações e composições, conforme necessário.

2.3 Resolução do Problema Proposto:

O problema envolvia modelar um sistema de locação de veículos, atendendo aos seguintes requisitos:

- Atributos específicos de automóveis, como placa, cor, ano, tipo de combustível, número de portas, entre outros.
- Relação entre carros, modelos e marcas (um modelo pode pertencer a várias marcas, mas cada modelo está associado a apenas uma marca específica).
- Gestão de locações, incluindo o histórico de aluguéis, datas de início e devolução.

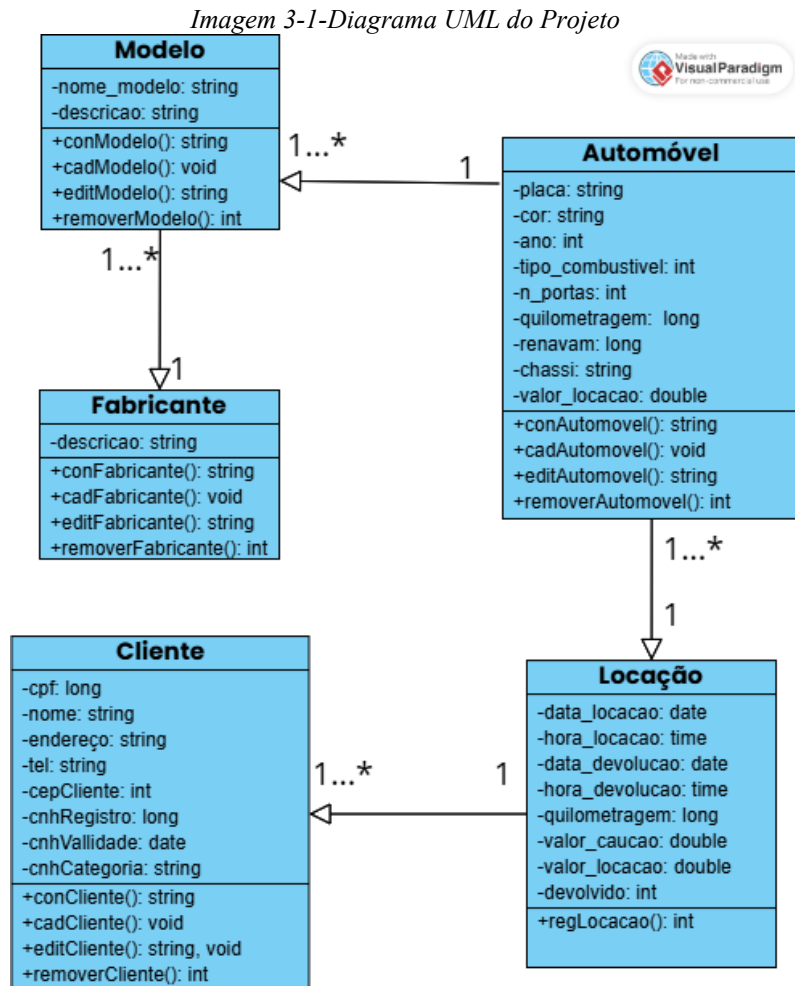
Imagem 2-2-Print do Projeto no Site



Fonte: Elaborada pelo autor (2024)

3 RESULTADOS

O diagrama UML abaixo ilustra a estrutura do sistema, incluindo as classes, atributos, métodos e as relações entre elas:



Fonte: Elaborada pelo autor (2024)

1. Classe Modelo

Representa os modelos dos automóveis disponíveis no sistema.

- **Atributos:**

- **nome_modelo:** Identifica o nome do modelo (ex.: Corolla, Civic).
- **descricao:** Breve descrição do modelo.

- **Métodos:**

- **conModelo():** Consulta os modelos cadastrados.
- **cadModelo():** Realiza o cadastro de um novo modelo.
- **editModelo():** Edita informações de um modelo existente.
- **removerModelo():** Remove um modelo do sistema.

2. Classe Fabricante

Armazena os fabricantes dos modelos de automóveis.

- **Atributos:**
 - **descricao:** Nome do fabricante (ex.: Toyota, Honda).
- **Métodos:**
 - **conFabricante():** Consulta os fabricantes cadastrados.
 - **cadFabricante():** Realiza o cadastro de um novo fabricante.
 - **editFabricante():** Edita informações de um fabricante existente.
 - **removerFabricante():** Remove um fabricante do sistema.

3. Classe Automóvel

Detalha os veículos disponíveis para locação.

- **Atributos:**
 - **placa:** Placa do veículo.
 - **cor:** Cor do veículo.
 - **ano:** Ano de fabricação.
 - **tipo_combustivel:** Tipo de combustível (ex.: gasolina, etanol).
 - **n_portas:** Numero de Portas.
 - **quilometragem:** Quilometragem atual do veículo.
 - **renavam:** Registro Nacional de Veículos Automotores
 - **chassi:** Número do Chassi
 - **valor_locacao:** Valor da locação por dia.
- **Métodos:**
 - **conAutomovel():** Consulta os automóveis cadastrados.
 - **cadAutomovel():** Cadastra um novo automóvel no sistema.
 - **editAutomovel():** Edita informações de um automóvel existente.
 - **removerAutomovel():** Remove um automóvel do sistema.

4. Classe Cliente

Gerencia os dados dos clientes que realizam locações.

- **Atributos:**
 - **cpf:** CPF do cliente.
 - **nome:** Nome completo.
 - **endereco:** Endereço residencial.
 - **tel:** Telefone de contato.
 - **cepCliente:** Código de Endereçamento Postal (ex.: 12345-67)
 - **cnhRegistro:** Número da CNH.
 - **cnhValidade:** Data de validade da CNH.
 - **cnhCategoria:** Categoria da CNH (ex.: B, C).
- **Métodos:**
 - **conCliente():** Consulta os clientes cadastrados.
 - **cadCliente():** Realiza o cadastro de um novo cliente.
 - **editCliente():** Edita informações de um cliente existente.
 - **removerCliente():** Remove um cliente do sistema.

5. Classe Locação

Gerencia as transações de locação de veículos.

- **Atributos:**
 - **data_locacao:** Data da locação.
 - **hora_locacao:** Hora da locação.
 - **data_devolucao:** Data prevista para devolução.
 - **hora_devolucao:** Hora prevista para devolução.
 - **quilometragem:** Quantos quilômetros rodado pelo cliente.
 - **valor_caucao:** Valor caução da locação.
 - **valor_locacao:** Valor total da locação.
 - **devolvido:** Status da devolução (0 para não devolvido, 1 para devolvido).
- **Métodos:**
 - **regLocacao():** Registra uma nova locação.
 - **conLocacao():** Consulta as locações cadastradas.
 - **editLocacao():** Edita informações de uma locação existente.
 - **removerLocacao():** Remove uma locação do sistema.

4 CONCLUSÃO

O diagrama UML e a descrição detalhada das classes apresentadas demonstram a importância de uma modelagem bem estruturada para o desenvolvimento de sistemas de locação de veículos. A partir deste modelo, é possível implementar um sistema que atenda às necessidades do negócio, proporcionando eficiência no gerenciamento de clientes, automóveis e locações. Além disso, a modularidade e a organização do sistema garantem escalabilidade para futuras melhorias.

REFERÊNCIAS

PISANI DA ARCH. **Diagrama De Classes | UML**. YouTube. Disponível em: <https://www.youtube.com/watch?v=niGtsDAaqzc>. Acesso em: 23 nov. 2024.

VISUAL PARADIGM. **Modelagem UML e ferramentas online**. Disponível em: <https://online.visual-paradigm.com/pt/>. Acesso em: 23 nov. 2024.