

Relatório:

1. Introdução

O presente trabalho tem como objetivo desenvolver um sistema de locadora de veículos aplicando os principais conceitos de Programação Orientada a Objetos (POO).

Ao longo do projeto, buscou-se colocar em prática os pilares da POO, estruturando uma aplicação organizada, modular e eficiente, capaz de realizar operações como cadastro, carregamento de dados, reservas e devoluções de veículos.

2.Arquitetura de Classes:

1. Classe Locadora

- Atributos principais:
listaClientes, listaVeiculos, listaReservas
- Métodos principais:
cadastrarCliente(), buscarCliente(), fazerReserva()

2. Classe Cliente

- Atributos:
nome, cpf, tipoCliente

3. Classe Veiculo

- Atributos:
placa, modelo,, disponivel

4. Classe Reserva

- Atributos:
cliente, veiculo, dataInicio, dataFim

5. Associação entre classes:

- A Locadora tem listas de Cliente, Veículo e Reserva
- A Reserva liga um Cliente a um Veículo

Princípios de POO Utilizados

- **Herança:** apliquei herança ao criar uma hierarquia de veículos. As classes *CarroDeLuxo* e *CarroEconomico* herdam de *Veiculo*, reutilizando atributos e métodos da classe base e evitando duplicação de código.
- **Encapsulamento:** todos os atributos das classes foram definidos como *private*, garantindo proteção dos dados e possibilitando controle de acesso por meio de métodos públicos. Isso melhora a segurança e a manutenção do código.
- **Polimorfismo:** o método *calcularValorAluguel()* é utilizado de forma polimórfica. Cada tipo de carro implementa sua própria lógica de cálculo, permitindo que o código cliente trate todos os veículos de maneira uniforme.
- **Abstração:** a classe *Veiculo* representa o modelo genérico de um veículo, definindo apenas as características essenciais. Detalhes específicos são implementados nas subclasses, simplificando o design e deixando o sistema mais modular.

Conclusão

O desenvolvimento do sistema *Viaja Fácil* permitiu aplicar, na prática, os principais conceitos de Programação Orientada a Objetos, organização modular e manipulação de arquivos CSV para persistência de dados. Apesar dos desafios encontrados, especialmente relacionados ao carregamento e à

consistência das reservas, o sistema foi estruturado de forma funcional, permitindo cadastrar clientes e veículos, realizar devoluções, calcular valores e gerar relatórios. Algumas limitações foram identificadas, como a dificuldade em integrar corretamente a leitura das reservas com os dados de clientes e veículos. Ainda assim, o sistema cumpre seu objetivo: demonstra domínio dos fundamentos da linguagem Java, uso de classes, herança, leitura e escrita de arquivos, tratamento de erros e interação com o usuário via console