

Digital Mavericks

Kauã de Melo Rodrigues
Eric de Oliveira Rodrigues
Caike Dametto

Sumário

1. Introdução

- Objetivo do Documento
- Visão Geral do Projeto

2. Descrição do Projeto

- Contexto e Motivação
- Objetivos do Projeto
- Organização do Documento

3. Diagrama de Classes

- Explicação das Classes e Relacionamentos

4. Conclusão

- Considerações Finais
- Próximos Passos

I. Introdução

Este documento apresenta a documentação para um projeto de gerenciamento de serviços automotivos, detalhando as classes e relações essenciais do sistema através de um diagrama de classes UML. O projeto visa facilitar o processo de manutenção e reparo de veículos, conectando clientes a oficinas mecânicas confiáveis e permitindo o gerenciamento eficiente de ordens de serviço.

2. Descrição do Projeto

No setor automotivo, os proprietários de veículos frequentemente enfrentam desafios ao lidar com problemas mecânicos e elétricos em seus veículos. A dificuldade em encontrar oficinas confiáveis, obter orçamentos precisos e acompanhar o progresso da manutenção são questões comuns. Este projeto visa abordar essas dores, oferecendo uma solução digital para facilitar a busca por serviços de qualidade e simplificar o processo de manutenção automotiva.

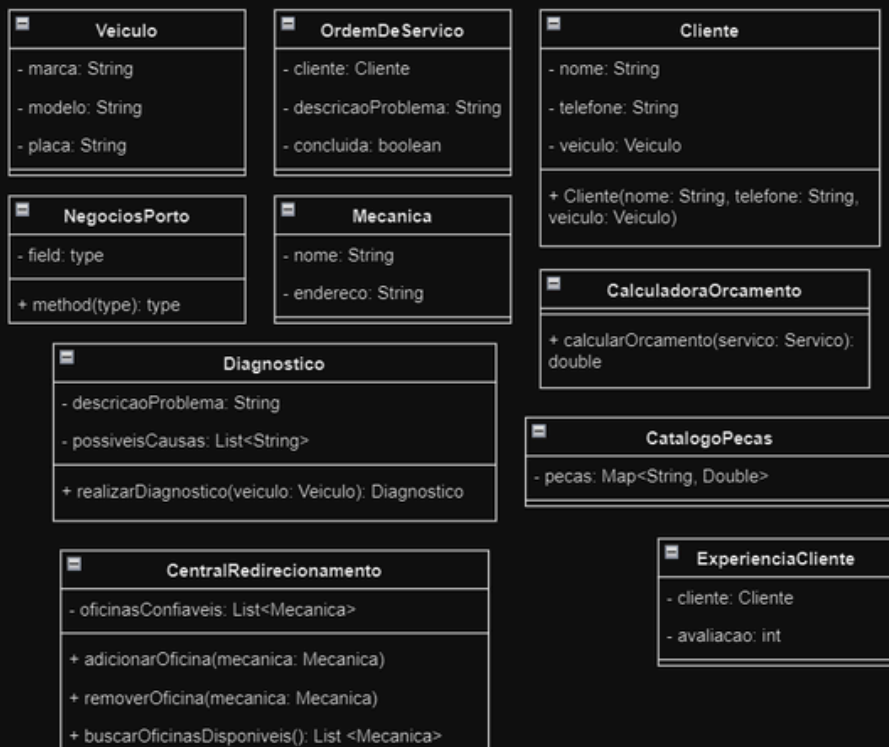
Objetivos do Projeto

Os principais objetivos deste projeto incluem:

- Criar um sistema de gerenciamento que conecte clientes a oficinas mecânicas confiáveis.
- Facilitar a criação de ordens de serviço, fornecendo descrições precisas dos problemas e permitindo o acompanhamento do status.
- Implementar uma calculadora de orçamento para estimar os custos de serviços com base nos problemas identificados.
- Melhorar a experiência do cliente ao oferecer opções de serviços eficientes e transparentes.
- Integrar um catálogo de peças para facilitar a consulta de preços e disponibilidade.

3. Diagrama de Classes

O diagrama de classes UML apresentado neste documento representa as entidades principais do sistema de gerenciamento de serviços automotivos. Ele inclui classes como Cliente, Veiculo, OrdemDeServico, Mecanica, CatalogoPecas, entre outras, mostrando suas relações e associações. Este diagrama é fundamental para entender a estrutura do sistema e as interações entre as diferentes entidades.



Cliente:

- A classe **Cliente** representa um cliente do sistema. Ela possui atributos como **nome**, **telefone** e **veiculo** (que é uma referência para a classe **Veiculo**). Um cliente pode ter várias ordens de serviço associadas.

Veiculo:

- A classe **Veiculo** descreve um veículo, com atributos como **marca**, **modelo** e **placa**. Um veículo pode estar associado a um único cliente (**Cliente**) e pode ter várias ordens de serviço relacionadas.

OrdemDeServico:

- A classe **OrdemDeServico** representa uma ordem de serviço, vinculada a um cliente específico (**Cliente**). Ela contém informações como **descricaoProblema** (descrição do problema relatado pelo cliente) e **concluida** (indicando se a ordem de serviço foi concluída ou não). Uma ordem de serviço está associada a um único cliente e pode ser realizada em uma oficina (**Mecanica**).

CentralRedirecionamento:

- A classe **CentralRedirecionamento** gerencia o redirecionamento de clientes para oficinas mecânicas confiáveis (**Mecanica**). Ela mantém uma lista de oficinas disponíveis (**oficinasConfiaveis**) e fornece métodos para adicionar, remover e buscar oficinas.

Mecanica:

- A classe **Mecanica** representa uma oficina mecânica, com atributos como **nome** e **endereco**. Várias ordens de serviço podem ser associadas a uma mesma oficina mecânica.

CatalogoPecas:

- A classe **CatalogoPecas** contém um catálogo de peças representado por um mapa (**pecas**) que mapeia o nome da peça para o seu preço. Essa classe é responsável por armazenar e fornecer informações sobre peças disponíveis e seus preços.

CalculadoraOrcamento:

- Embora não esteja explicitamente representada no diagrama, a **CalculadoraOrcamento** seria uma classe responsável por calcular o orçamento para um serviço específico (**Servico**), utilizando informações do **CatalogoPecas** e outros custos associados.

ExperienciaCliente:

- A classe **ExperienciaCliente** representa a experiência de um cliente com um serviço realizado por uma oficina (**Mecanica**). Ela pode incluir uma avaliação (**avaliacao**) do serviço prestado.

Diagnostico:

- A classe **Diagnostico** descreve o diagnóstico realizado para identificar problemas em um veículo. Ela contém uma **descricaoProblema** e uma lista de **possiveisCausas** para o problema identificado. O diagnóstico pode ser realizado com base nas informações do veículo (**Veiculo**) e do **CatalogoPecas**.

NegociosPorto

- A classe **NegociosPorto** representa a entidade responsável por gerenciar os clientes associados ao ecossistema da Porto Seguro no contexto do sistema de gerenciamento de serviços automotivos.
- Atributos:
 - **clientesPorto**: Lista de clientes que são segurados pela Porto Seguro.
- Métodos:
 - **adicionarCliente(cliente:Cliente)**: Permite adicionar um cliente ao ecossistema da Porto Seguro.
 - **removerCliente(cliente:Cliente)**: Permite remover um cliente do ecossistema da Porto Seguro.
 - **buscarClientesPorto():List<Cliente>**: Retorna a lista de clientes segurados pela Porto Seguro.

4. Conclusão

Em conclusão, este projeto busca transformar a experiência de manutenção automotiva, oferecendo uma solução abrangente para conectar clientes a serviços confiáveis. A implementação das classes e relações definidas no diagrama de classes UML é crucial para o sucesso deste empreendimento. Próximos passos incluem a implementação do sistema com base nesse design e a realização de testes para validar sua eficácia e usabilidade.

Este documento fornece uma visão abrangente do projeto de gerenciamento de serviços automotivos, explicando seu propósito, objetivos e estrutura por meio de um diagrama de classes e uma documentação detalhada.