UNIVERSIDADE FEDERAL DE OURO PRETO

*CAMPUS* MORRO DO CRUZEIRO – OURO PRETO

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUISA

LEANDRO

VINICIUS NUNES DOS ANJOS

TPI – BCC221

Gerenciamento de uma ofinina mecânica

OURO PRETO

2024

LUISA

LEANDRO

VINICIUS NUNES DOS ANJOS

TPI – BCC221

Gerenciamento de uma ofinina mecânica

OURO PRETO

2024

SUMÁRIO

[1. INTRODUÇÃO 4](#_Toc137473361)

[1.1. Objetivo Geral 4](#_Toc137473362)

[1.2. Objetivos Específicos 4](#_Toc137473363)

[1.3. Justificativa 5](#_Toc137473364)

[2. DESENVOLVIMENTO](#_Toc137473365) 5

[2.1. Especificação de Requisitos 5](#_Toc137473366)

[2.2. Diagrama de Classes 5](#_Toc137473367)

[2.3. Diagrama de Entidade Relacionamento (SE HOUVER) 5](#_Toc137473368)

[3. CONCLUSÃO](#_Toc137473369) 7

# INTRODUÇÃO

No contexto da disciplina de Programação Orientada a Objetos (POO), o desenvolvimento de sistemas que utilizam conceitos fundamentais como abstração, herança e polimorfismo é essencial para a criação de software com alto nível de organização, reutilização de código e facilidade de manutenção. Este trabalho tem como foco a implementação de um sistema em C++ que permita o gerenciamento de serviços e vendas de peças em uma oficina mecânica. O sistema será capaz de realizar o cadastro de clientes e seus veículos, gerar e gerenciar ordens de serviço, além de controlar a execução de serviços e vendas de produtos.

A escolha de um sistema para uma oficina mecânica se justifica pela necessidade real de otimização dos processos de gestão de serviços, controle de estoque e atendimento ao cliente, onde um sistema informatizado pode oferecer melhorias significativas em termos de eficiência e precisão.

## Objetivo Geral

Este trabalho tem como objetivo geral desenvolver um sistema em C++ para gerenciar os serviços e as vendas de peças em uma oficina mecânica, utilizando os princípios de Programação Orientada a Objetos, com ênfase na implementação de classes coesas e com baixo acoplamento.

## Objetivos Específicos

Esse trabalho tem como objetivos específicos:

● Desenvolver uma estrutura de classes que represente os atores e processos envolvidos na gestão de uma oficina mecânica, utilizando herança e polimorfismo.

● Implementar funcionalidades para cadastro de clientes, veículos, ordens de serviço, e para o controle de execução e aprovação dos serviços.

● Criar uma interface de usuário simples e funcional que permita o acesso às funcionalidades do sistema conforme o tipo de usuário (administrador, vendedor ou mecânico).

● Garantir que o sistema mantenha um histórico detalhado das manutenções realizadas em cada veículo, promovendo rastreabilidade e controle.

## Justificativa

Este trabalho é justificado devido à importância de aplicar os conceitos de Programação Orientada a Objetos em um projeto prático, permitindo aos alunos consolidar o conhecimento teórico por meio do desenvolvimento de um sistema realista e relevante. Além disso, o projeto oferece a oportunidade de explorar a aplicação de boas práticas de design de software, como a criação de classes com alta coesão e baixo acoplamento, essenciais para o desenvolvimento de sistemas robustos e escaláveis.

# DESENVOLVIMENTO

Neste capítulo é apresentado o desenvolvimento do trabalho, detalhando as etapas seguidas para a implementação do sistema de gerenciamento de serviços e vendas de peças em uma oficina mecânica. Serão discutidos desde a especificação dos requisitos até a arquitetura e a implementação das funcionalidades principais, sempre com ênfase na aplicação dos conceitos de Programação Orientada a Objetos.

## Especificação de Requisitos

A especificação de requisitos é uma etapa fundamental no desenvolvimento de qualquer sistema, pois define as funcionalidades que o sistema deverá oferecer e as restrições sob as quais ele deverá operar. Para o sistema em questão, os requisitos foram classificados em funcionais e não funcionais:

2.1.1. Requisitos Funcionais

* Cadastro de Clientes e Veículos:

O sistema deve permitir que vendedores cadastrem clientes e seus respectivos veículos. O sistema deve associar cada veículo a um cliente específico.

* Gerenciamento de Ordens de Serviço:

O vendedor deve ser capaz de gerar uma ordem de serviço, incluindo informações como motivo da manutenção e quilometragem do veículo.

O mecânico deve visualizar ordens de serviço abertas e selecionar uma para execução

.

* Aprovação de Orçamentos:

O sistema deve permitir que ordens de serviço de orçamento aguardem a aprovação do cliente antes de serem executadas.

O vendedor deve poder marcar uma ordem de serviço como aprovada ou encerrada, dependendo da resposta do cliente.

* Execução de Serviços e Registro de Peças Utilizadas:

O mecânico deve registrar os serviços realizados e as peças utilizadas, com seus respectivos preços, após a aprovação do orçamento.

O sistema deve calcular o custo total do serviço, incluindo mão de obra e peças.

* Histórico de Manutenções:

O sistema deve manter um histórico detalhado de todas as manutenções realizadas em cada veículo, acessível por cliente e veículo.

* Gerenciamento de Usuários:

O administrador deve ser capaz de editar os dados de vendedores e mecânicos, garantindo a correta gestão de usuários no sistema.

2.1.2. Requisitos Não Funcionais

* Usabilidade:

O sistema deve possuir uma interface de usuário simples e intuitiva, facilitando o uso por vendedores, mecânicos e administradores.

* Desempenho:

O sistema deve responder rapidamente às solicitações dos usuários, garantindo fluidez no processo de cadastro e gerenciamento de ordens de serviço.

* Segurança:

O sistema deve permitir o acesso mediante login, com diferentes níveis de permissão (administrador, vendedor, mecânico), para garantir a segurança das informações.

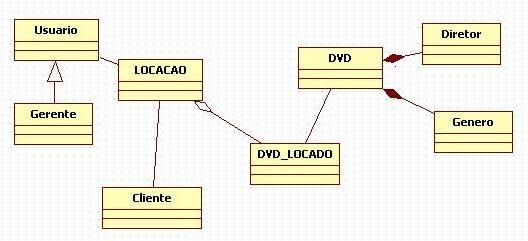
* Manutenibilidade:

O código do sistema deve ser modular e seguir boas práticas de desenvolvimento, permitindo futuras expansões e modificações com facilidade.

A partir desses requisitos, o sistema foi projetado para atender às necessidades específicas de uma oficina mecânica, com foco na eficiência, segurança e facilidade de uso, garantindo que todos os atores possam realizar suas atividades de forma integrada e eficaz.

## Diagrama de Classes

Figura 1 – Diagrama de Classes

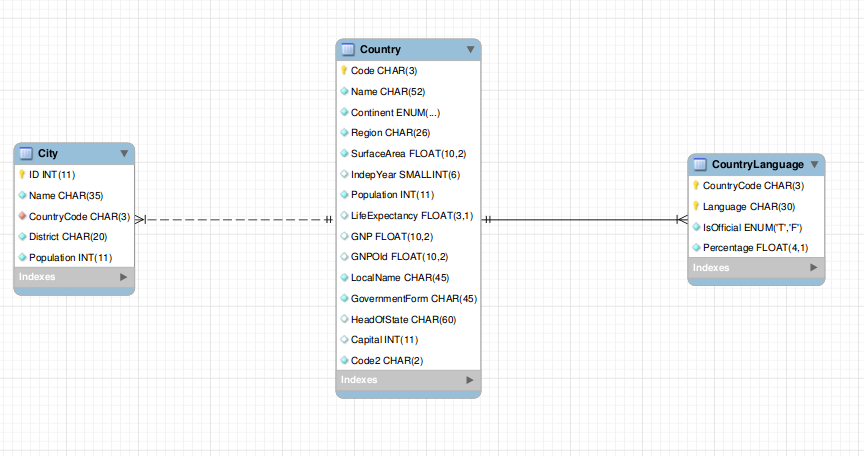


A Figura 1 apresenta as classes Lorem Lorem ipsu ….

Explicar o que cada classe representa …

## Diagrama de Entidade Relacionamento (SE HOUVER)

Figura 2 – Diagrama de Entidade e Relacionamento



A Figura 2 apresenta as tabelas do Lorem Lorem ipsu ….

Explicar o que cada tabela representa e seus relacionamentos …

# CONCLUSÃO

O desenvolvimento deste trabalho permitiu a aplicação prática dos conceitos fundamentais da Programação Orientada a Objetos (POO), como abstração, herança e polimorfismo, no contexto de um sistema de gestão para uma oficina mecânica. Através da criação de uma estrutura de classes bem definida e da implementação de funcionalidades específicas para o gerenciamento de clientes, veículos, ordens de serviço e controle de execução de serviços, foi possível construir um sistema coeso, com baixo acoplamento entre suas partes, atendendo aos requisitos propostos.

O trabalho demonstrou a importância de um bom planejamento na fase de design, onde a definição de classes abstratas e a correta utilização de herança foram essenciais para garantir a extensibilidade e a flexibilidade do sistema. Além disso, a implementação de uma interface de usuário que se adapta ao tipo de usuário logado (administrador, vendedor ou mecânico) reforçou a aplicação do princípio de encapsulamento, permitindo que cada ator tenha acesso apenas às funcionalidades relevantes para o seu papel.

Por fim, a construção do histórico de manutenções e a separação clara entre a aprovação de orçamentos e a execução dos serviços evidenciam a aplicabilidade de boas práticas de desenvolvimento, contribuindo para um sistema organizado e de fácil manutenção. Esse projeto, portanto, não só proporcionou uma experiência enriquecedora na consolidação dos conceitos de POO, como também resultou em um software funcional que pode ser expandido e adaptado a diferentes necessidades no futuro.