

Alessandro D'Angelo - 18.2.4054  
Fábio Henrique Soares Dias - 19.1.4008

## **Trabalho Prático de Programação Orientada a Objetos**

Universidade Federal de Ouro Preto - UFOP  
Departamento de Computação - DECOM  
Ciência da Computação

Orientador: Guillermo Cámara Chávez

Ouro Preto  
Setembro de 2020

# Sumário

<b>Introdução</b>	<b>2</b>
<b>Desenvolvimento</b>	<b>3</b>
<b>Programa iniciado</b>	<b>5</b>
<b>Conclusão</b>	<b>9</b>

# Introdução

A programação orientada a objetos (POO) é uma metodologia de desenvolvimento de software. Ela tem como principal objetivo facilitar a modelagem e desenvolvimento de sistemas, através da interação entre objetos. Sua principal vantagem é a proximidade com a forma que os seres humanos visualizam e entendem o mundo ao seu redor.

A ideia fundamental das linguagens orientadas a objetos é combinar, em uma única unidade, tanto os dados como as funções que operam sobre esses dados. Tal unidade denomina-se um *objeto*. O programa é organizado como um conjunto finito de objetos que contenham dados e operações que chamam a esses dados e que se comunicam entre si por meio de relações.

Existem diversas características relacionadas à orientação a objetos. Todas as propriedades consideradas não são exclusivas desse paradigma, mas em seu conjunto definem claramente as linguagens orientadas a objetos. Essas propriedades são: *Abstração de dados, Encapsulamento de dados, Herança e Polimorfismo*.

## Desenvolvimento

Este relatório consiste em apresentar os paradigmas citados, a partir da implementação de um programa que utiliza dos conceitos fundamentais de orientação a objetos. O desenvolvimento das classes atende aos critérios definidos na descrição do problema, que baseia-se na necessidade de se administrar e otimizar o funcionamento de uma oficina mecânica, realizando principalmente as funções de um vendedor e mecânico.

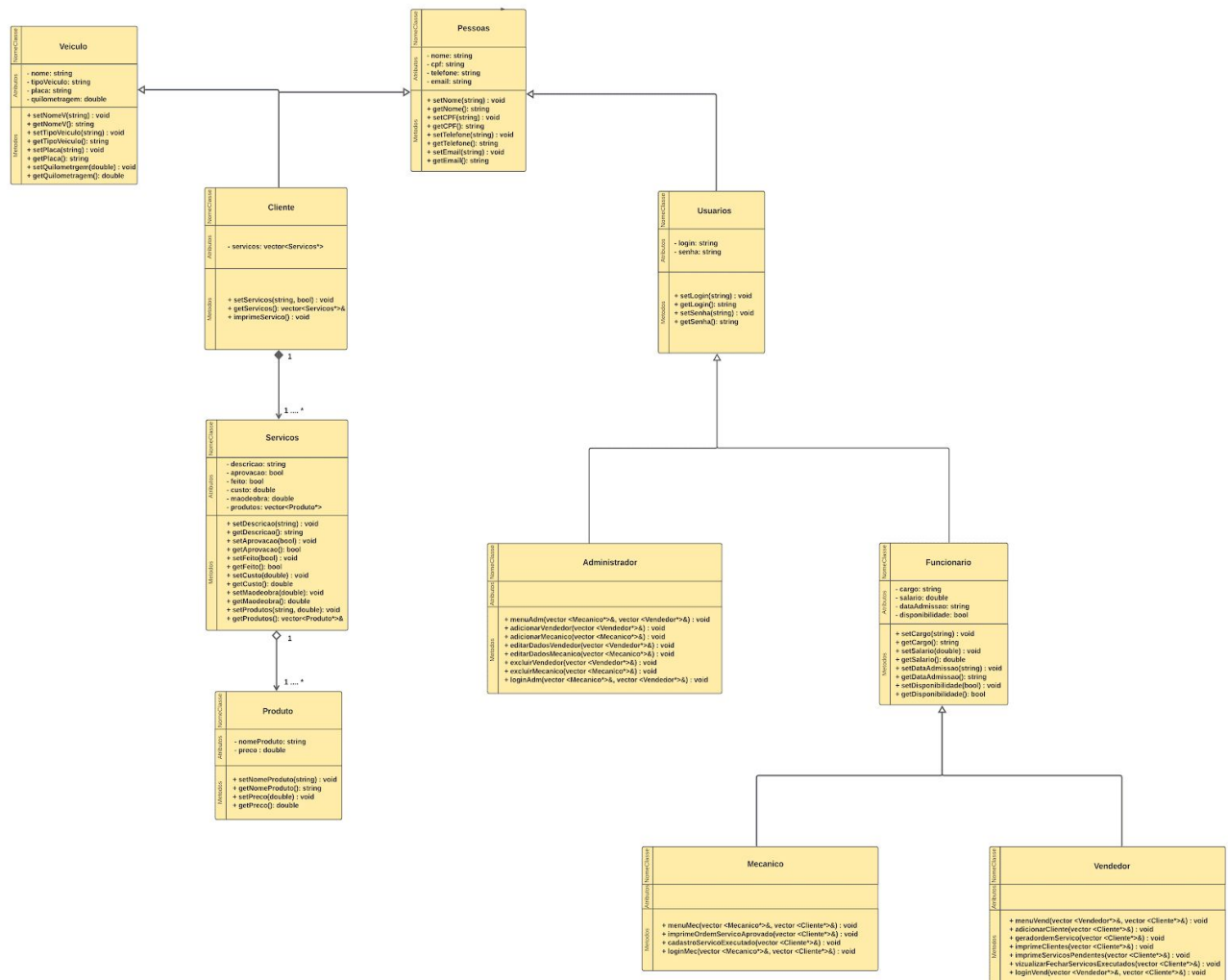
A primeira parte do trabalho consistiu no levantamento de ideias e análises de necessidades para que facilitasse a confecção das classes, além disso, verificamos como realizar a aplicação através da divisão dos requisitos não funcionais, a fim de alcançar o nível de abstração de dados para formação dos aspectos. Em segundo instante, foi criado o programa a princípio na linguagem C++.

Para a implementação, foi utilizado os seguintes conceitos:

- Abstração: diferenciação entre propriedades externas de uma entidade e os detalhes da sua composição interna.
- Encapsulamento: processamento de agrupar dados e operações relacionadas na mesma unidade de programação.
- Herança: mecanismo pelo qual elementos mais específicos incorporam a estrutura e comportamento de elementos mais gerais.
- Polimorfismo: propriedade que permite que um operador ou uma função atue de modo diferente em função do objeto sobre o qual se aplicam.

Objetivando, assim, a redução da complexidade no desenvolvimento de software e aumentar sua produtividade.

## Diagrama de Classes (UML)



Caso a imagem esteja muito pequena, enviamos também o pdf na pasta do trabalho.

## Execução do Código

Para a execução do código no terminal, utilize o seguinte comando:

*compilando:*

```
g++ main.cpp Pessoas.cpp Cliente.cpp Veiculo.cpp Usuarios.cpp  
Administrador.cpp Funcionarios.cpp Vendedor.cpp Mecanico.cpp  
Servicos.cpp Produto.cpp Funcoes.cpp -o main
```

*executando:*

```
./main
```

Para inicializar o programa, já foi definido um usuário Administrador por meio do método construtor do mesmo.

O login do administrador: **AleDangelo**

senha do administrador: **123**

Da mesma forma que o administrador, o programa já inicializa com dois funcionários já cadastrados, um mecânico e um vendedor.

O login do vendedor: **AleDangelo**

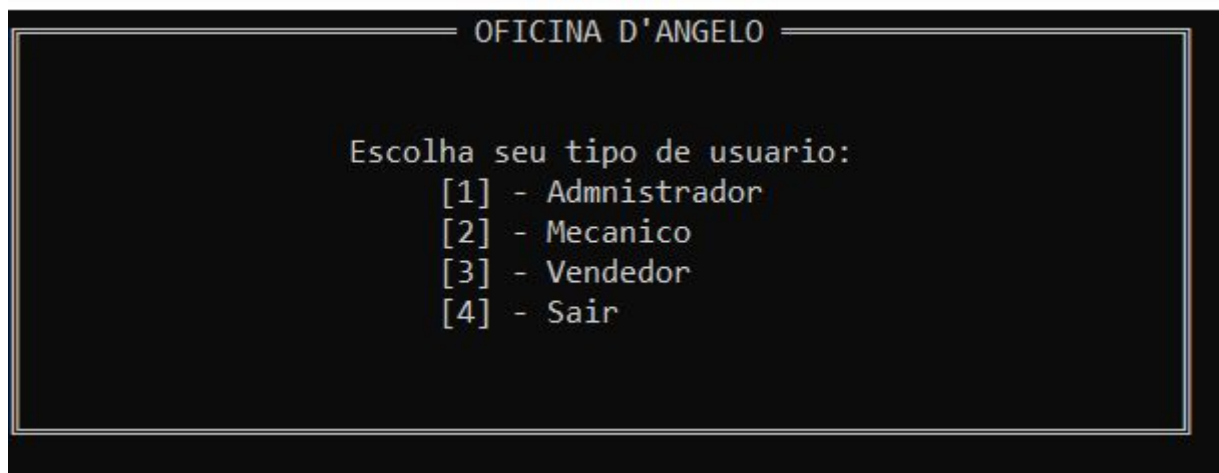
senha do vendedor: **123**

O login do mecânico: **Fabioh**

senha do mecânico: **1234**

## Programa iniciado

No início aparecerá um menu onde o usuário encontrará algumas opções para realizar o login.



Além dos funcionários já iniciados, há a escolha de um administrador cadastrar um novo funcionário seja ele mecânico ou vendedor, com um login e senha. Ao voltar no menu inicial, o funcionário já cadastrado poderá entrar com seu login e senha no mecânico caso ele seja um mecânico ou em vendedor caso ele seja um vendedor.

```
===== OFICINA D'ANGELO =====

Escolha seu tipo de usuario:
    [1] - Administrador
    [2] - Mecanico
    [3] - Vendedor
    [4] - Sair

1
Digite seu login
AleDangelo
Digite sua senha
123
Seja bem-vindo Alessandro D'Angelo!
===== ADMINISTRADOR =====

                ESCOLHA SUA TAREFA:
    [1] - Adicionar Vendedor
    [2] - Adicionar Mecanico
    [3] - Excluir Vendedor
    [4] - Excluir Mecanico
    [5] - Editar dados de Vendedores
    [6] - Editar dados de Mecanicos
    [7] - Sair
```

Por exemplo, se eu escolher adicionar um vendedor:

```
ADMINISTRADOR

ESCOLHA SUA TAREFA:
[1] - Adicionar Vendedor
[2] - Adicionar Mecanico
[3] - Excluir Vendedor
[4] - Excluir Mecanico
[5] - Editar dados de Vendedores
[6] - Editar dados de Mecanicos
[7] - Sair

1
Digite o nome do Vendedor:
Naruto
Digite o cpf do Vendedor:
12345678910
Digite o telefone do Vendedor:
35520023
Digite o e-mail do Vendedor:
teste@teste.com
Digite o login do Vendedor:
Naruto
Digite a senha do Vendedor:
123
Digite o cargo do Vendedor:
Vendedor
Digite o salario do Vendedor:
200
Digite a data de admissao do Vendedor:
20/09/2020
Digite a disponibilidade do Vendedor:
1

ADMINISTRADOR

ESCOLHA SUA TAREFA:
[1] - Adicionar Vendedor
[2] - Adicionar Mecanico
[3] - Excluir Vendedor
[4] - Excluir Mecanico
[5] - Editar dados de Vendedores
[6] - Editar dados de Mecanicos
[7] - Sair
```

A disponibilidade do vendedor ou mecânico quando criados devem ser 1 ou 0, pois o atributo é booleano.

Agora, escolhendo a opção 7, voltamos ao menu inicial e podemos escolher a opção 3 e entrar com o login e senha do vendedor cadastrado, o mesmo funciona se tivesse sido criado um mecânico, porém escolheríamos a opção 2 em vez da 3 para chamar o menu de login do mecânico.



ADMINISTRADOR

ESCOLHA SUA TAREFA:

[1] - Adicionar Vendedor

[2] - Adicionar Mecanico

[3] - Excluir Vendedor

[4] - Excluir Mecanico

[5] - Editar dados de Vendedores

[6] - Editar dados de Mecanicos

[7] - Sair

7

OFICINA D'ANGELO

Escolha seu tipo de usuario:

[1] - Admnistrador

[2] - Mecanico

[3] - Vendedor

[4] - Sair

3

Digite seu login

Naruto

Digite sua senha

123

Login invalido

Seja bem-vindo Naruto!

VENDEDOR

ESCOLHA A SUA TAREFA:

[1] - Adicionar Cliente

[2] - Gerar ordem de servico

[3] - Servicos pendentes

[4] - Fechar Servico

[5] - Sair

A partir do login podemos começar a explorar as funcionalidades do sistema de acordo com suas funcionalidades, conseguimos implementar todas que foram requisitadas no trabalho.

## **Conclusão**

Pode-se inferir, portanto, que com o avanço da tecnologia e a necessidade de se otimizar operações, cresceram também as demandas por metodologias que possam abstrair e modularizar as estruturas básicas de um programa, surgindo assim o conceito em desenvolvimento de softwares chamado de Programação Orientada a Objetos, facilitando a abstração e o desenvolvimento de aplicações complexas.