

Universidade Federal de Ouro Preto

Decom

Programação Orientada a objeto BCC 221



Trabalho Prático II

Oficina Mecânica

Nome: Hugo Ziviani

Matrícula: 16.2.5849

Nome: Renato Tavares Goia

Matrícula: 16.1.9999

Professor: Guillermo Cámara Chávez

Introdução

O presente trabalho envolve o desenvolvimento de um software para gerenciar os serviços e vendas de peças e serviços em uma oficina mecânica. O sistema possui como atores:

- Os funcionários da oficina mecânica, que são os mecânicos e atendentes(vendedores);
- Os clientes de oficina;
- Os veículos dos clientes;
- Os itens de serviços e produtos de venda.

Desenvolvimento

Para realizar essa tarefa, implementamos todas classes baseando no diagrama abaixo, sendo ela necessária para o funcionamento básico do programa.

Para o primeiro ator(funcionário) nós implementamos três classes, que são **Funcionário**, **Atendente** e **Mecânico**. Para cada uma das classes, criamos atributos e métodos necessários e ainda implementamos os conceitos de herança e polimorfismo, onde as classes **Atendente** e **Mecânico** herdam métodos e atributos da classe **Funcionário**. Isso acontece porque tanto os atendentes como os mecânicos são classes derivados da classe Funcionário que é a classe mãe.

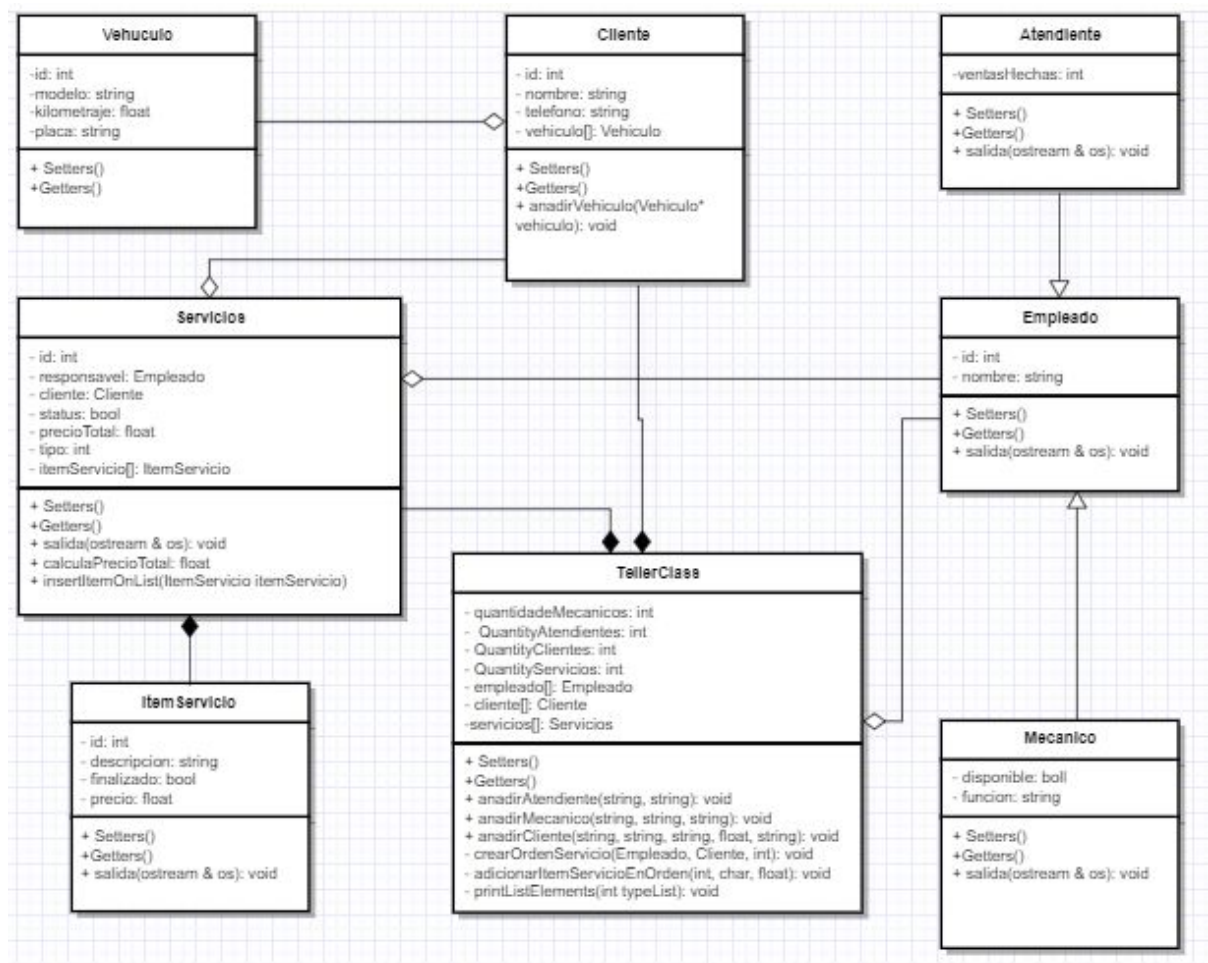
Para o segundo ator que são os clientes, desenvolvemos uma classe **Cliente** e nela temos os respectivos métodos e atributos. Dentre o atributos dessa classe, consta uma lista de objetos da classe **Veículo** (terceiro ator), concretizando assim os conceitos de agregação e encapsulamento. Foi necessário fazer isso, pois no cadastro de cada cliente que é feito, é necessário cadastrar o seu veículo também.

No terceiro ator ou seja os veículos dos clientes, criamos a classe **Veículo**, com os seus respectivos atributos e métodos. E essa classe é agregada a classe **Cliente**. Vale ressaltar que cliente está preparado para receber uma lista de carros, entretanto pelo curto prazo para elaboração do trabalho nem todas as funcionalidades estão disponíveis

ao usuário final. Entretanto a interface gráfica exige um tempo muito maior para ser produzida.

No quarto ator, que são os itens de serviço e produtos de venda, desenvolvemos duas classes: **Servicos** e **ItemServico** com seus respectivos métodos e atributos. Repare que entre os atributos da classe **Servicos** consta um objeto da classe **Funcionario** que representa o responsável por um determinado serviço, e também uma lista dos objetos da classe **ItemServico** que representa a lista dos serviços cadastrados e seus respectivos estados. Aqui nós usamos os conceitos de agregação e composição, pois a classe **Funcionario** agrega e classe **ItemServico** compõe a classe **Servicos**. Classe **Servico** e responsável de gerar uma ordem de serviço para solicitação de orçamento ou manutenção.

Diagrama de classes representando a estrutura do trabalho



Conclusão

Para gerenciar os serviços e vendas de peças em uma oficina mecânica nós criamos a classe **Administracao**. observe que essa classe ela é composta pelas classes **Cliente** e **Serviços**, e ainda ela tem a classe **Funcionario** que faz a sua composição. Aqui podemos adicionar atendente, mecânico ou cliente a sua respectivas listas e ainda criar ordem de serviço entre outras funcionalidades pedidas no presente trabalho.

Para a execução do nosso trabalho é necessário rodá-lo no Netbeans e escolher como classe principal de execução a classe `OficinaMecanica.`

A partir daí todo o fluxo de telas é realizado. Ressalto que sentimos certa dificuldade de criar os objetos a partir dos dados advindos da tela, uma vez que tivemos pouco tempo para familiarizar-mos com o conteúdo.

Execução

Para logar no sistema existem três possibilidades:

1- como Administrador:

login: adm

senha: 123

2- como Vendedor

login: vend

senha: 123

3- como Mecânico:

login: mec

senha: 123