**UNIVERSIDADE FEDERAL DE OURO PRETO**

CURSO BACHARELADO EM CIÊNCIA DA COMPUTAÇÂO

**RELATÓRIO TRABALHO PRÁTICO I**

Programação Orientada a Objetos

GABRIEL FERNANDES NIQUINI

RAFAEL COELHO MONTE ALTO

Ouro Preto – Minas Gerais – Brasil

Março, 2021

**UNIVERSIDADE FEDERAL DE OURO PRETO**

CURSO BACHARELADO EM CIÊNCIA DA COMPUTAÇÂO

**Relatório de Trabalho Prático I de Programação Orientada a Objetos**

# Relatório apresentado à disciplina Programação Orientada a Objetos da matriz curricular do curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte das exigências do Trabalho Prático I.

Universidade Federal de Ouro Preto – UFOP

Departamento de Computação – DECOM

Ciência da Computação

Professor: Guilhermo Cámara Chávez

Ouro Preto – Minas Gerais – Brasil

Março, 2021

**SUMÁRIO**

**1 INTRODUÇÃO** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 02

**2 DESENVOLVIMENTO**  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 04

**3 DIAGRAMA UML** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

**4 INSTRUÇÕES** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

**5 CONCLUSÃO** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

1. **INTRODUÇÃO**

A importância de se trabalhar com linguagens orientadas a objetos e desenvolver sistemas complexos de maneira eficiente e organizada. A programação orientada a objetos facilita o reuso e manutenção do seu código através de propriedades fundamentais como encapsulamento, abstração...

Neste trabalho foi proposta a criação de um sistema de administração para uma clínica odontológica. O programa desenvolvido envolve várias grandezas e personagens, como os três dentistas que trabalham na clínica, seus respectivos assistentes e também pacientes.

É esperado que um usuário, ao entrar no sistema, tenha acesso a diferentes funções administrativas, como realizar o pagamento de contas, agendar consultas com dentistas e receber a folha de ponto dos funcionários. Para isso, são trabalhados três diferentes usuários, com permissões e acessos diferentes.

1. **DESENVOLVIMENTO**

**2.1 CLASSES**

**2.1.1 PESSOA**

Antes de qualquer classe, a classe Pessoas é definida como classe base para todas as classes que representam uma pessoa física, seja funcionário ou cliente. Ela possui apenas um atributo, nome, que é uma *string*.Ela é mãe de duas classes, Funcionário e Paciente, passa seu atributo nome, seus *getters* e *setters*.

**2.1.2 FUNCINARIO**

Dentro da clínica trabalham três dentistas especialistas, seus respectivos assistentes de consultório e um assistente/recepcionista, e para generalizar todas essas funções dentro do consultório, foi criada a classe Funcionário, filha da classe pessoa e mãe da classe dentista.

Essa classe, Funcionário, tem como atributo e funções, além das herdadas de sua classe mãe, um *double* salário e seus respectivos *getters* e *setters*.

**2.1.3 DENTISTA**

Herdando as características de Funcionário, Dentista é uma classe que possui todos os atributos, *getters* e *setters* de sua classe mãe, além de um atributo assistente, que é da classe Funcionário, e um *int* referente ao número de pacientes, e seus *getters* e *setters* particulares.

Funções essenciais para o funcionamento do programa, como folha de ponto e agenda, utilizam da classe Dentista, nem que seja apenas parar selecionar o dentista a ser trabalhado na função pelo nome.

**2.1.4 PACIENTE**

Assim como Funcionários é uma classe filha de Pessoa, herdando suas características e com uma a mais, a classe Paciente possui um atributo *string* referente ao pagamento se está ou não já efetuado. Ela é utilizada nas funções referentes a consultas, que são responsáveis por tanto agendar as consultas, quanto por atualizar o status de pagamento do paciente.

**2.1.5 USUÁRIOS DO SISTEMA**

Recebemos o desafio de implementar três diferentes usuários: Administrador, Assistente Administrativo e Usuário Geral. Cada um possui permissões diferentes, mas uma coisa que todos tem em comum são os atributos *login* e *senha*. Para acessar o sistema, todos esses três usuários precisam entrar com um nome de usuário e uma senha, sem exceção.

Para evitar a reescrita de código, foi criado uma classe User, que continha os atributos login e senha, e as funções de *get* e *set* correspondentes. Assim, os três usuários do sistema herdam a classe User e cada um possui outros atributos e métodos especializados.

**2.1.6** **CONTA**

Conta é uma classe que possui 3 atributos, sendo todas *strings*, descrição, data de vencimento e data de pagamento. Ela tem também seus respectivos *getters* e *setters*, e é utilizada para fazer o controle dos gastos da clínica.

**2.2 FUNÇÕES**

As funções principais foram separadas em quatro pastas diferentes, sendo a divisão feita entre agenda, consulta, contas e folha de ponto. Sendo elas responsáveis pelo funcionamento do programa como um todo, elas que permitem salvar, alterar e mostrar na tela informações.  
 Agenda é separada em 5 funções diferentes e que se complementam, sendo elas *menuAgenda*, apresenta um menu responsável por mostrar as opções disponíveis ao usuário referentes as ações q pode ser tomadas na agenda, *editAgenda*, permite selecionar uma das consultas na agenda e alterar a data da mesma, *printAgenda*, imprimi a agenda de um determinado dentista na tela, *addToAgenda*, adiciona consulta a agenda, sendo necessário data e nome como parâmetros de entrada, e r*emoveFromAgenda*, procura na agenda o nome e se encontrado ele é retirado da agenda.

Consulta é separada em 3 funções correlacionadas, sendo elas *receberConsulta*, que recebe a consulta e atualiza o dado de pagamento do paciente, *printConsultasPagasRecentes*, que imprime na tela as 10 últimas consultas pagas, e *addConsulta*, que adiciona a consulta a lista de consultas.

Conta, além de uma classe, também é detentora da função *fazerPagamentoContas*, que a partir das informações dadas, descrição, data de vencimento e data de pagamento, da conta, inclui ela em uma lista de contas já pagas.

A função de Folha De Ponto deve apresentar ao usuário uma lista dos colaboradores da clínica. Ao escolher o funcionário desejado, imprime-se a folha de ponto contendo as datas de trabalho e alguma observação pertinente daquele dia. O usuário pode adicionar mais linhas a folha de ponto através da função *adicionarFolhaDePonto*.

1. **UML**

A Linguagem de Modelagem Unificada (Unified Modeling Language) foi oficializada no ano 2000 como padrão na área de Orientação a Objetos. É extremamente útil e necessário na organização de projetos e sistemas com muitas variáveis que se relacionam.

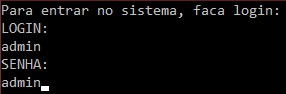
Foto do Diagrama UML

Imagem 1 – Diagrama UML do Trabalho Prático I

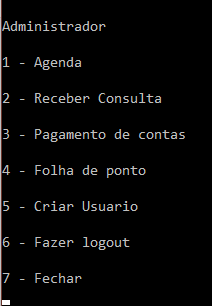
1. **INSTRUÇÕES**

As instruções de compilação e execução do programa são as seguintes:

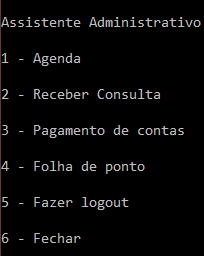
Menu de login:



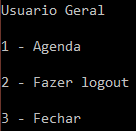
Menu Administrador:



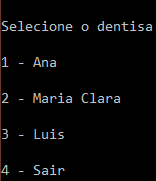
Menu Assistente Administrativo:



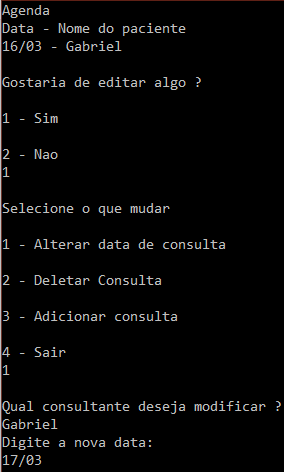
Menu Usuário Geral:



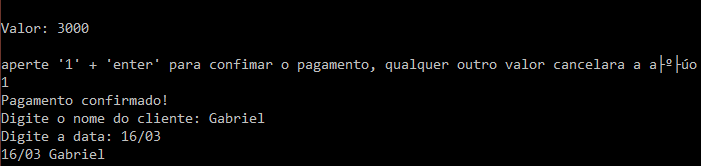
Menu seleção de dentista para acesso a agenda:



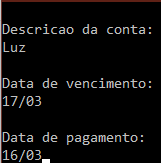
Impressão da agenda na tela e menu de edição:



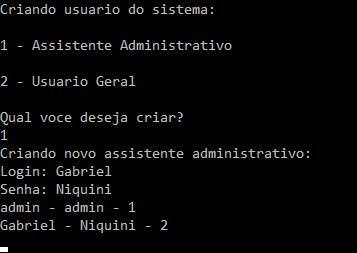
Recebimento de consulta e impressão das consultas marcadas e pagas recentemente:



Pagamento de contas:



Criação de usuário no sistema:



1. **CONCLUSÃO**

Sabe-se que o aluno de Ciência da Computação em sua vida acadêmica deve aprender a trabalhar com a programação orientada a objetos com muita familiaridade e atenção. Assim, destaca-se, novamente, a importância de atividades práticas como a apresentada neste relatório. São elas que permitem ao aluno não apenas compreender melhor as aulas teóricas, principalmente aquelas voltadas à este paradigma, como também aprimorar suas habilidades de trabalho em equipe, fundamentais no mercado de trabalho.