



UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA DE
CONTROLE E AUTOMACAO - DECAT



PAULO FELIPE POSSA PARREIRA

Programação Orientada a Objetos: 1º Trabalho de
Implementação

Ouro Preto, 2021

SUMÁRIO

	SUMÁRIO	2
1	Objetivo	3
2	Softwares Utilizados	3
3	Requisitos do Sistema	3
4	Principais Recursos Utilizados	4
5	Dica de Utilização	5
6	Exemplo de Utilização	5
7	Diagrama UML	5

1 Objetivo

O objetivo deste primeiro trabalho, desenvolvida para a disciplina de Programação Orientada a Objetos, é de assimilar os conhecimentos obtidos na disciplina, modelando e implementando um sistema de gerenciamento para uma clínica de odontologia. O trabalho é desenvolvido na linguagem C++ e também tem como objetivo o aprendizado das ferramentas que essa linguagem oferece, como a biblioteca padrão e o suporte a manipulação de arquivos.

2 Softwares Utilizados

Durante o desenvolvimento do trabalho nós utilizamos alguns softwares e tecnologias, abaixo citamos esses e damos uma breve descrição do seu papel:

- *C++/11* - A versão do C++ escolhida foi a 11, principalmente devido a adoção de recursos mais modernos a partir dessa versão.
- *Apache NetBeans IDE 11.1* - A escolha da IDE foi tomada com base na experiência prévia do autor e também na capacidade de integração com a linguagem C++.
- *Visual Paradigm Online* - Interface web utilizada para montar o diagrama UML.

3 Requisitos do Sistema

A seguir estão listados os requisitos do sistema e uma breve descrição de como eles foram implementados:

- *Agenda* - Uma funcionalidade para definir a agenda de cada especialista da clínica, permitindo o cadastro, edição e remoção de elementos da agenda. Foi implementada como um conjunto de funções dentro da classe Especialista, utilizando um vector para guardar os registros da classe RegistroAgenda.
- *Recebimento de Consultas* - Funcionalidade para guardar o recibo de consultas realizadas, implementado dentro da classe SistemaClinica, também guardado como um vetor de registros.
- *Pagamento de Contas* - Funcionalidade para guardar o recibo de despesas pagas, implementado dentro da classe SistemaClinica, também guardado como um vetor de registros.

- *Folha de Ponto* - Um registro de ponto para os funcionários da clínica, implementada dentro da classe `Funcionario` e utilizável pelas classes derivadas `Administrador`, `Especialista` e `AssistenteAdministrativo`.
- *Cadastro de Usuários* - Função administrativa para cadastrar novos usuários no sistema, guardando os novos usuários em arquivo e também em um vetor da classe `Usuario`, atributo da classe `SistemaClinica`.

4 Principais Recursos Utilizados

Como o objetivo do trabalho foi a fixação dos conceitos de orientação a objetos e das funcionalidades disponíveis na linguagem C++, optou-se por utilizar o maior número de recursos disponíveis. Abaixo estão listados os recursos mais importantes e o motivo da escolha dos mesmos.

- *Herança e Polimorfismo* - Todas as classes relacionadas ao sistema de usuários utilizam de herança e polimorfismo para cumprir seus requisitos. A classe `Usuario` é estendida por todas as classes relacionadas aos papéis, que possuem alguns métodos sobrescritos. O polimorfismo foi possível através do cast dinâmico e de ponteiros inteligentes, aumentando a performance das manipulações e garantindo a consistência dos dados.
- *Biblioteca Padrão e Containers* - Todas as listagens do software são feitas através da biblioteca padrão do C++, essa opção foi escolhida de forma a facilitar a exibição e manipulação dos dados.
- *Arquivos e Streams* - A forma da linguagem lidar com arquivos é através de streams, definidos na biblioteca padrão de entrada e saída. Esses streams garantem que os dados sejam persistidos sempre que o programa for reiniciado. Também é utilizado o stream do IO padrão para a exibição dos dados.
- *Ponteiros* - Ponteiros foram amplamente utilizados no software, sendo uma forma segura de realizar o polimorfismo e diminuindo o uso de memória total.
- *Encapsulamento* - Todas as variáveis das classes estão encapsuladas e só podem ser acessadas através dos getters e setters, garantindo alguma validação desses dados.
- *Exceções e Try/Catch* - Algumas partes do código foram implementadas de forma a tratar exceções, possibilitando o funcionamento do software mesmo que ocorra algum erro durante o tempo de execução.

5 Dica de Utilização

O sistema foi projetado de forma a ser intuitivo para usuários leigos. Para começar a utilizá-lo, insira o login: admin e senha: admin, entrando com a conta de administrador e tendo acesso a todas as funcionalidades implementadas.

6 Exemplo de Utilização

Abaixo estão exibidas algumas das telas do sistema.

```

Criando adminUsuários inicializados com sucesso
TERM environment variable not set.
Criando adminCriando admin=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
=====
Digite:
1 - Fazer Login
2 - Sair
■

Criando adminCriando admin=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
Usuário Logado: Administrador Tipo: Administrador
=====
Digite:
1 - Agenda
2 - Receber Consulta
3 - Pagamento de Contas
4 - Folha de Ponto
5 - Gerenciar Usuários
6 - Sair
■

Editar o registro da agenda do especialista: Especialista 1
Informações atuais:
Data e Hora: 01/02/2020 às 03:00
Observação: Consulta
=====
Insira a nova data do registro: ■

=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
Usuário Logado: Administrador
=====
Módulo de Agenda
Selecione um especialista para visualizar sua agenda:
0 - Especialista 1
1 - Especialista 2
2 - Especialista 3
■

```

7 Diagrama UML

Abaixo está representado o diagrama de classes do sistema, o diagrama geral foi recortado para melhor visualização.

```

Mostrando agenda do especialista: Especialista 1
=====
0 - 01/02/2020 às 03:00 | Consulta
1 - 15/05/2020 às 09:00 | dasd
=====
Opções:
1 - Editar Registro | 2 - Adicionar Registro | 3 - Remover Registro | 4 - Voltar

```

```

Adicionar novo registro da agenda do especialista: Especialista 1
=====
Insira a data do registro: 15/05/2020
Insira a hora do registro: 09:00

```

```

Mostrando a folha de ponto do funcionário: Especialista 3
=====
05/10/2020 | Atrasado
=====
Opções:
1 - Adicionar Registro | 2 - Voltar

```

```

=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
Usuário Logado: Administrador
=====
Módulo de Recebimento de Consultas
Listar Consultas Recebidas
=====
Digite a data de início (0 -> mostra todos os registros) - formato: dd/mm/yyyy: 03/02/2020
Digite a data fim - formato: dd/mm/yyyy: 05/09/2021
- Dt: Tue Mar 16 | asd | R$5000
=====
Insira uma opção:
1 - Fazer nova pesquisa | 2 - Voltar | 3 - Menu Principal:

```

```

=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
Usuário Logado: Administrador
=====
Módulo de Recebimento de Consultas
Adicionar nova consulta
=====
Digite o valor da consulta: 5000.00
Digite o nome do cliente: asd
Data de registro: Tue Mar 16
Aperte 0 + enter para continuar.

```

```

=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
Usuário Logado: Administrador
=====
Adicionar novo usuário
=====
Digite o tipo de usuário -> 1 - Administrador | 2 - Assistente Administrativo | 3 - Especialista: 2
Digite o nome do usuário: atest
Digite o cpf do usuário: 121.144.4444-94
Digite o email do usuário: pasd@asd.com
Digite o login do usuário: addd
Digite a senha do usuário: addd

```

```

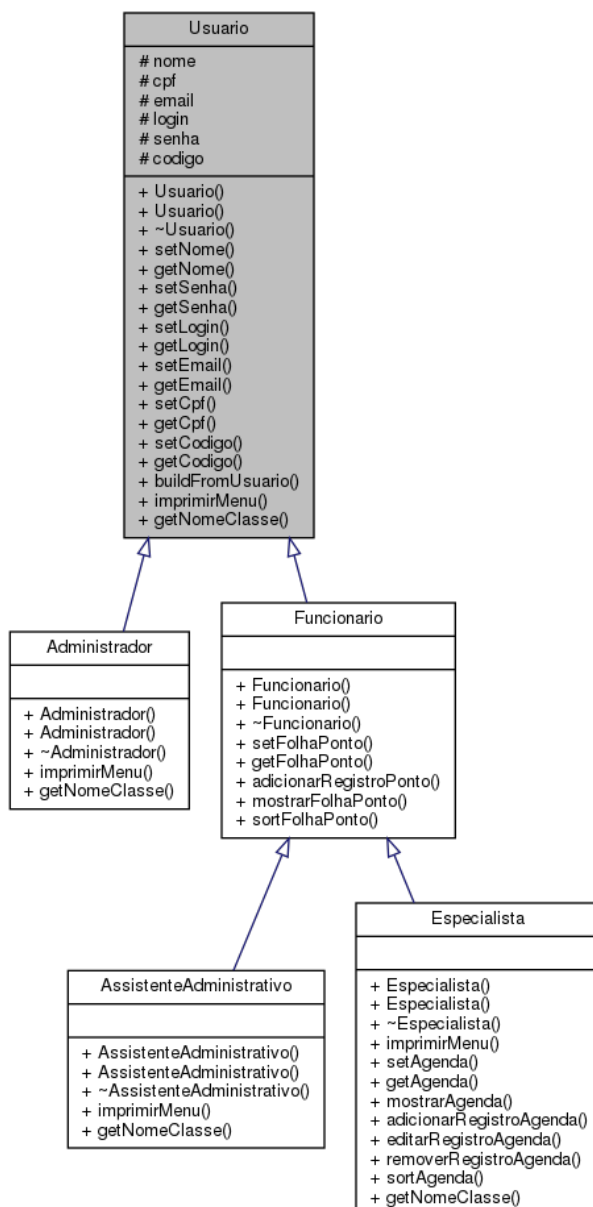
=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
Usuário Logado: Administrador
=====
Módulo de Cadastro de Usuarios
=====
Escolha uma das opções: 1 - Adicionar Usuário | 2 - Remover Usuário | 3 - Voltar

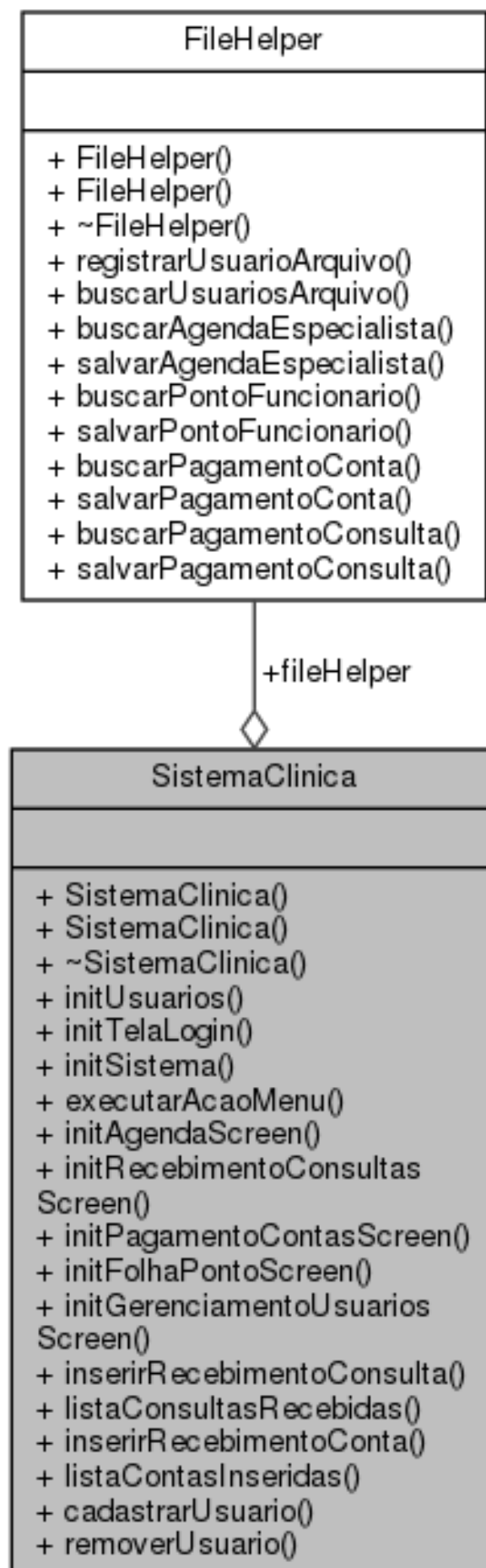
```

```

=====
Bem vindo(a) ao sistema de gerenciamento da clinica odontologica
Usuário Logado: Administrador
=====
Módulo de Recebimento de Consultas
Listar Contas Pagas
=====
0 - adasd | 01/01/2020 | 01/01/2020
1 - Luz | 01/10/2020 | 01/10/2020
=====
Insira uma opção:
1 - Voltar | 2 - Menu Principal: █

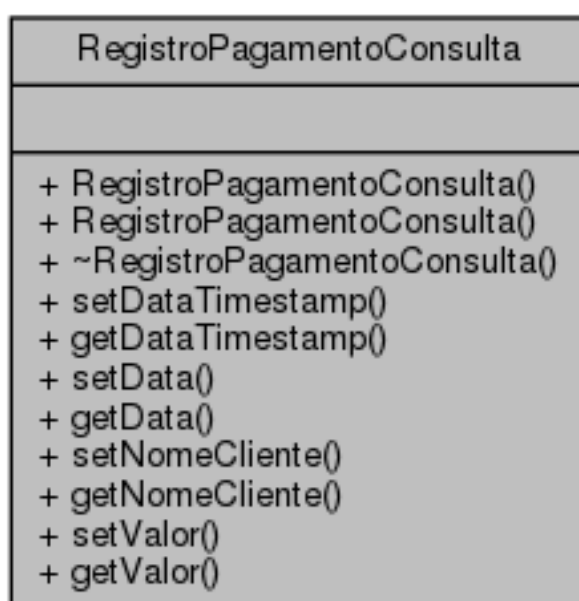
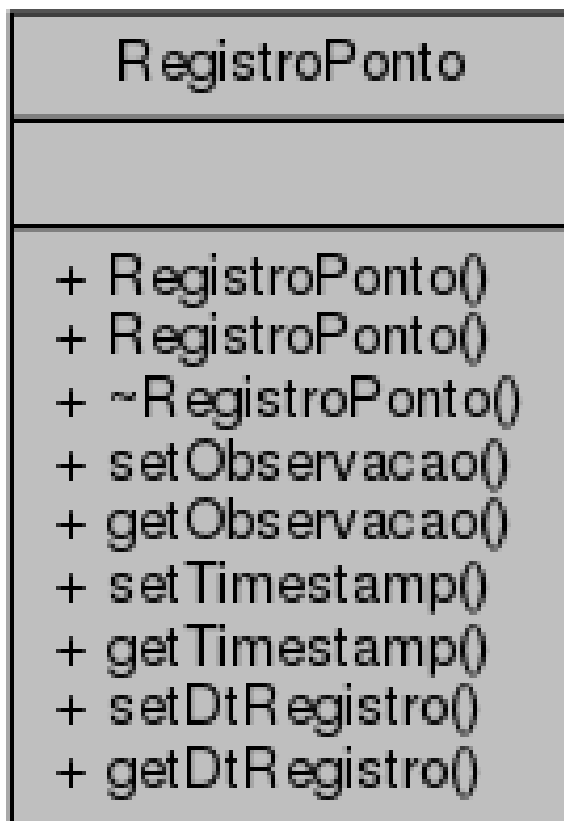
```





FileHelper
<ul style="list-style-type: none">+ FileHelper()+ FileHelper()+ ~FileHelper()+ registrarUsuarioArquivo()+ buscarUsuariosArquivo()+ buscarAgendaEspecialista()+ salvarAgendaEspecialista()+ buscarPontoFuncionario()+ salvarPontoFuncionario()+ buscarPagamentoConta()+ salvarPagamentoConta()+ buscarPagamentoConsulta()+ salvarPagamentoConsulta()

RegistroAgenda
<ul style="list-style-type: none">+ RegistroAgenda()+ RegistroAgenda()+ ~RegistroAgenda()+ setObservacao()+ getObservacao()+ setData()+ getData()+ setHora()+ getHora()+ getTimestamp()+ setTimestamp()



RegistroPagamentoConta
+ RegistroPagamentoConta() + RegistroPagamentoConta() + ~RegistroPagamentoConta() + setDtPagamento() + getDtPagamento() + setDtVencimento() + getDtVencimento() + setDescricao() + getDescricao()