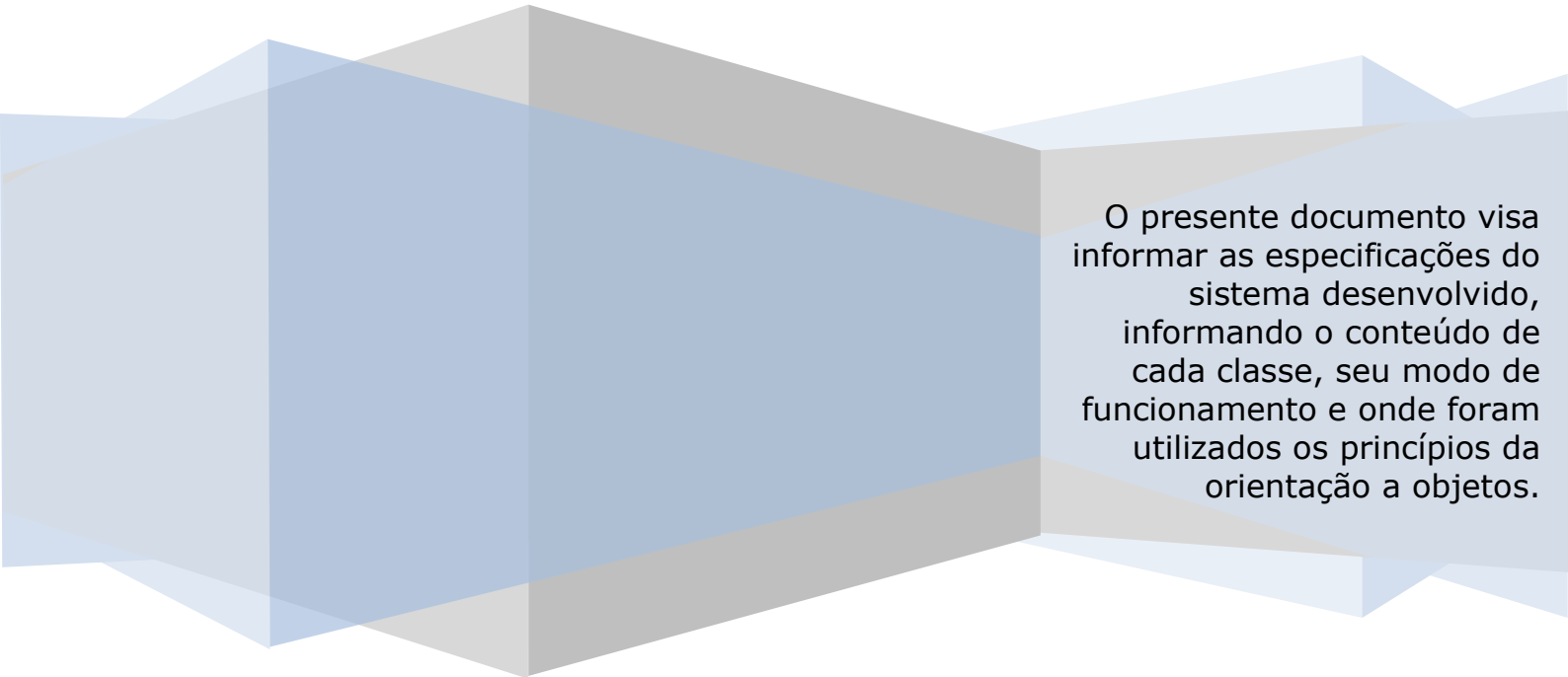


SGH

Sistema Gerenciador de Hospital

Ana Caroline Silva Nogueira

Giliarde Reis Pereira



O presente documento visa informar as especificações do sistema desenvolvido, informando o conteúdo de cada classe, seu modo de funcionamento e onde foram utilizados os princípios da orientação a objetos.

Sumário

1. Objetivo	5
2. Classes	5
[2.1] Classe Lobby	6
[2.2] Classe Login	6
[2.3] Endereço	7
[2.4] Pessoa	8
[2.5] Paciente	10
[2.6] Funcionários	11
[2.7] Médicos	12
[2.8] Enfermeiros	13
[2.9] Medicamentos	15
3. Princípios da Orientação a Objetos	16

1. Objetivo

O sistema SGH tem como principal objetivo realizar o gerenciamento de pacientes, medicamentos e funcionários de um hospital, disponibilizando as funcionalidades necessárias a cada tipo de usuário de acordo com seu acesso.

Tal disponibilização ocorre no momento de login, onde é necessário informar nome e senha para conseguir acessar o sistema. A senha utilizada define qual será o nível de acesso do usuário que pode ser: secretário, médico, enfermeiro ou diretor.

2. Classes

O SGH foi montado sobre nove classes distintas. São elas:

- Lobby;
- Login;
- Endereço;
- Pessoa;
- Paciente;
- Funcionário;
- Médico;
- Enfermeiro;
- Medicamentos;

Todas foram montadas seguindo o princípio de encapsulamento para maior segurança dos dados informados.

[2.1] Classe Lobby

Função: realiza o gerenciamento das demais classes e apresenta a interface do programa para o usuário.

Na classe Lobby é solicitado ao usuário que se faça o login entrando com seu nome e a senha, havendo quatro opções de acesso:

- **Secretário** – Nesse nível de acesso o usuário poderá realizar quatro tipos de ações: cadastrar paciente, excluir paciente, trocar de usuário e fechar o programa.
- **Médico** - Nesse nível de acesso o usuário poderá realizar quatro tipos de ações: listar paciente, atender paciente, trocar de usuário e fechar o programa.
- **Enfermeiro** - Nesse nível de acesso o usuário poderá realizar quatro tipos de ações: listar paciente, atender paciente, trocar de usuário e fechar o programa.
- **Diretor** - Nesse nível de acesso o usuário poderá realizar treze tipos de ações: cadastrar pacientes, cadastrar médicos, cadastrar enfermeiros, cadastrar medicamentos, excluir pacientes, excluir médicos, excluir enfermeiros, listar pacientes dos médicos, listar pacientes dos enfermeiros, médico atender paciente, enfermeiro atender paciente, trocar de usuário e fechar o programa.

[2.2] Classe Login

Função: realizar o login do usuário. Exige nome e senha para o acesso.

As possibilidades de senha para login são as seguintes:

Corpo: elaborada com variáveis privadas, possui métodos getters e setters, bem como seu construtor vazio.

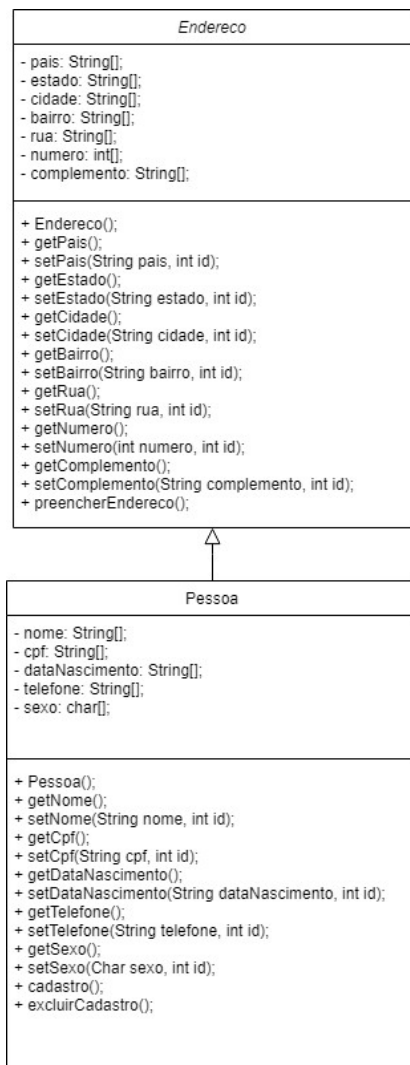
- **111** – Ao utilizar essa senha o usuário entrará no sistema como secretário e terá as possibilidades que lhe forem ofertadas.
- **222** – Neste modo o login é feito como médico;
- **333** – Utilizando esta sequência entra-se como enfermeiro;

- **444** – Acesso de administrador, permite ao usuário lançar mão de todas as possibilidades fornecidas pelo sistema.

[2.3] Endereço

Função: realizar o cadastro do endereço de pacientes, médicos e enfermeiros.

Essa classe é utilizada numa relação de herança com a classe Pessoa, sendo ela a classe mãe.

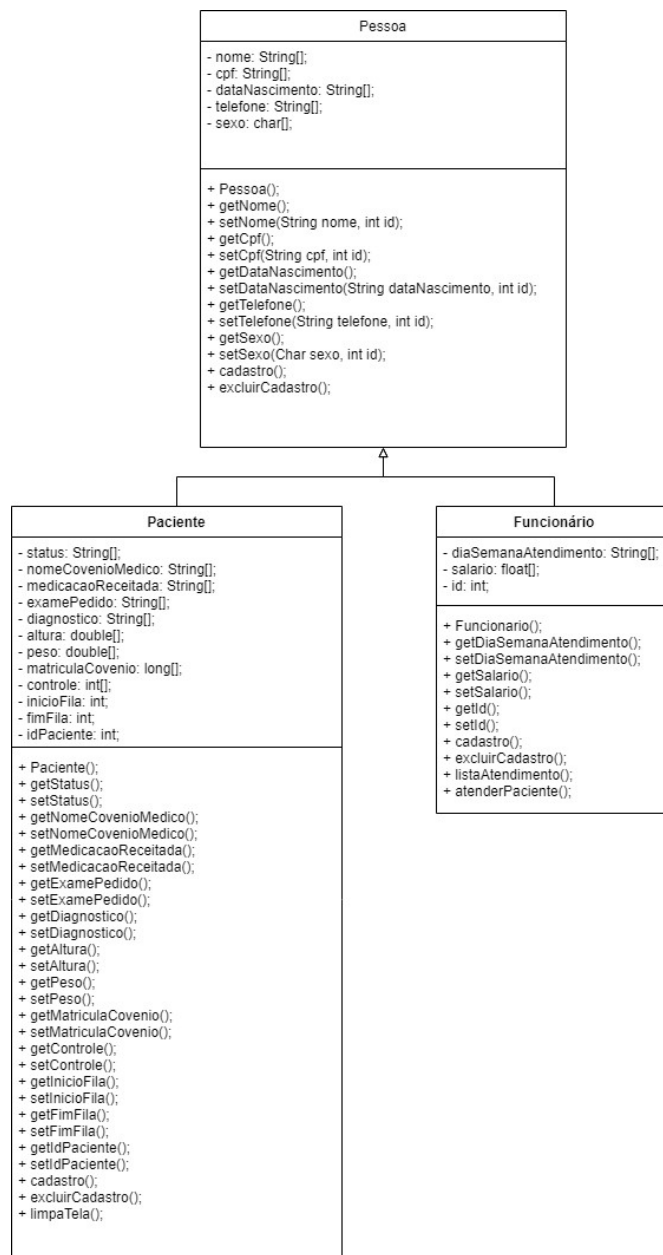


Corpo: a classe segue o padrão de encapsulamento, possuindo seu construtor, getters e setters. Ela possui ainda um método para preencher o endereço que será chamado posteriormente por outras classes.

Ao ser executado esse método pede ao usuário as informações comuns de seu endereço, tais quais: país, estado, cidade, bairro, rua, número e complemento.

[2.4] Pessoa

Função: a classe Pessoa tem como função fornecer uma base para a construção de suas classes herdeiras, que são: Paciente e Funcionários.



Corpo: a classe segue o padrão de encapsulamento, possuindo seu construtor, getters e setters. Ela possui ainda os métodos cadastro e excluirCadastro, que serão chamados por suas classes filhas e utilizados com polimorfismo.

[2.5] Paciente

Função: a classe Paciente tem como função realizar o gerenciamento dos pacientes, fazendo seu cadastro e sua exclusão.

Paciente
<pre>- status: String[]; - nomeCovenioMedico: String[]; - medicacaoReceitada: String[]; - examePedido: String[]; - diagnostico: String[]; - altura: double[]; - peso: double[]; - matriculaCovenio: long[]; - controle: int[]; - inicioFila: int; - fimFila: int; - idPaciente: int;</pre>
<pre>+ Paciente(); + getStatus(); + setStatus(); + getNomeCovenioMedico(); + setNomeCovenioMedico(); + getMedicacaoReceitada(); + setMedicacaoReceitada(); + getExamePedido(); + setExamePedido(); + getDiagnostico(); + setDiagnostico(); + getAltura(); + setAltura(); + getPeso(); + setPeso(); + getMatriculaCovenio(); + setMatriculaCovenio(); + getControle(); + setControle(); + getInicioFila(); + setInicioFila(); + getFimFila(); + setFimFila(); + getIdPaciente(); + setIdPaciente(); + cadastro(); + excluirCadastro(); + limpaTela();</pre>

Corpo: a classe segue o padrão de encapsulamento, possuindo seu construtor, getters e setters. Ela possui ainda os métodos cadastro e excluirCadastro.

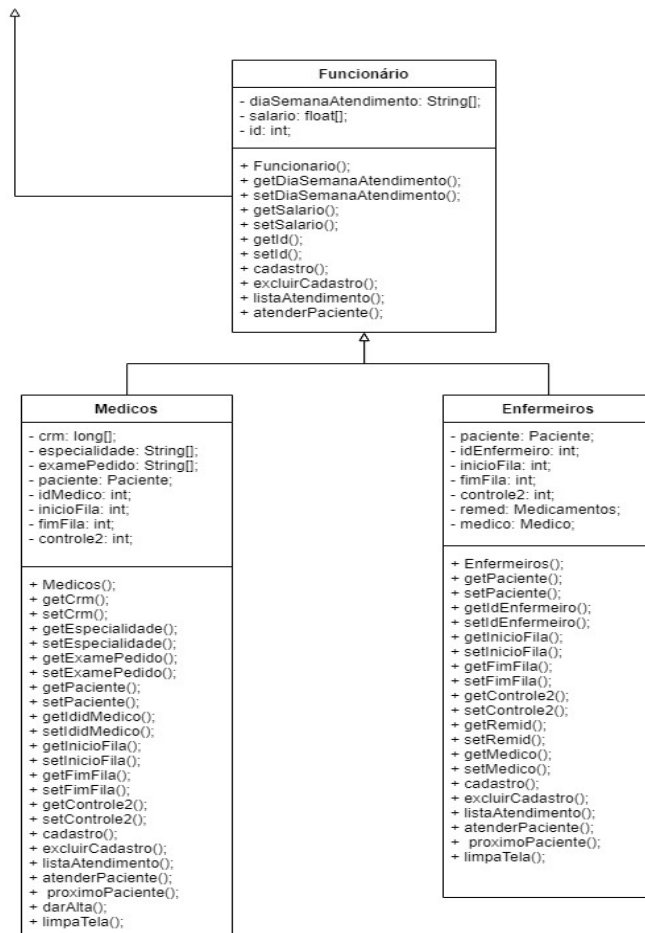
- **Cadastro** – pede as informações do paciente, tais quais: nome, CPF, data de nascimento, sexo, número para contato, status, altura e peso. Esse método ainda perguntará se o paciente possui convênio médico, em caso negativo será chamado o método

preencherEndereço da classe Endereço. Em caso afirmativo será solicitado que se informe o nome e a matrícula do convênio médico, para em seguida chamar o método **preencherEndereço** da classe Endereço.

- **Excluir cadastro** – é um método que removerá o paciente da lista de pacientes.

[2.6] Funcionários

Função: a classe Funcionários tem como função fornecer uma base para a construção de suas classes herdeiras, que são: Médicos e Enfermeiros.



Corpo: a classe segue o padrão de encapsulamento, possuindo seu construtor, getters e setters. Ela possui ainda os métodos cadastro,

excluirCadastro, listaAtendimento e atenderPaciente, que serão herdados por suas classes filhas e utilizados com polimorfismo.

[2.7] Médicos

Função: a classe Médicos tem como função realizar o gerenciamento dos médicos, fazendo seu cadastro, sua exclusão, listando, atendendo e dando alta para os pacientes.

Medicos
- crm: long[]; - especialidade: String[]; - examePedido: String[]; - paciente: Paciente; - idMedico: int; - inicioFila: int; - fimFila: int; - controle2: int;
+ Medicos(); + getCrm(); + setCrm(); + getEspecialidade(); + setEspecialidade(); + getExamePedido(); + setExamePedido(); + getPaciente(); + setPaciente(); + getIdMedico(); + setIdMedico(); + getInicioFila(); + setInicioFila(); + getFimFila(); + setFimFila(); + getControle2(); + setControle2(); + cadastro(); + excluirCadastro(); + listaAtendimento(); + atenderPaciente(); + proximoPaciente(); + darAlta(); + limpaTela();

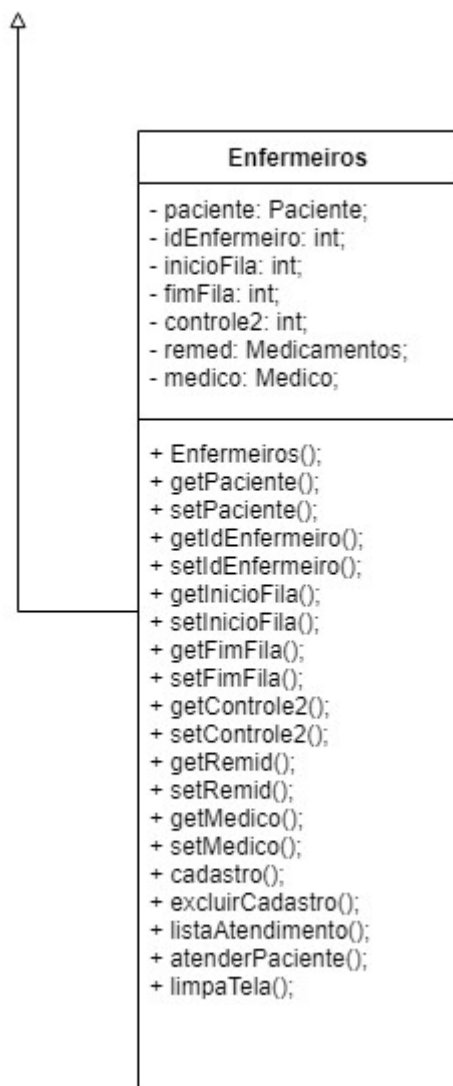
Corpo: a classe segue o padrão de encapsulamento, possuindo seu construtor, getters e setters. Ela possui ainda os métodos cadastro,

excluirCadastro, listaAtendimento, atenderPaciente, próximoPaciente e darAlta.

- **Cadastro** – pede as informações do paciente, tais quais: nome, CPF, data de nascimento, sexo, número para contato, dias de atendimento, salário, número de CRM e especialidade, para em seguida chamar o método **preencherEndereço** da classe Endereço.
- **Excluir cadastro** – é um método que removerá o médico da lista de médicos.
- **Lista atendimento** – vai listar todos os pacientes que ainda precisam ser atendidos.
- **Atender paciente** – realiza o atendimento do paciente informando seu diagnóstico, a medicação receitada e o exame passado.
- **Próximo paciente** – atende o próximo paciente da lista.
- **Dar alta** – define se o paciente terá alta ou não.

[2.8] Enfermeiros

Função: a classe Enfermeiros tem como função realizar o gerenciamento dos enfermeiros, fazendo seu cadastro, sua exclusão, listando e atendendo os pacientes.



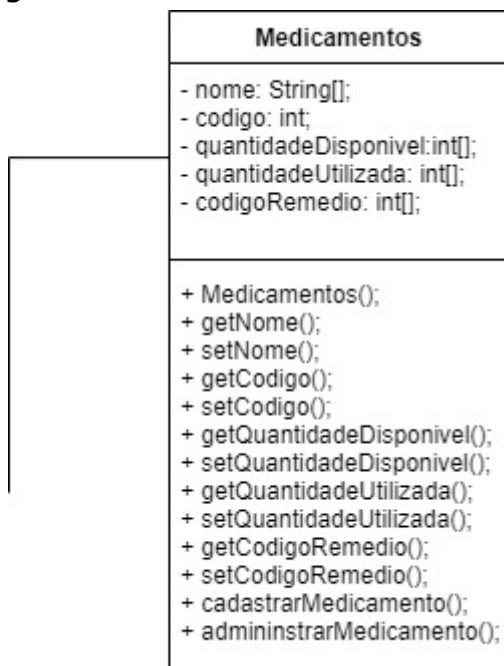
Corpo: a classe segue o padrão de encapsulamento, possuindo seu construtor, getters e setters. Ela possui ainda os métodos cadastro, excluirCadastro, listaAtendimento, atenderPaciente, próximoPaciente e darAlta.

- **Cadastro** – pede as informações do paciente, tais quais: nome, CPF, data de nascimento, sexo, número para contato, dias de atendimento, salário, para em seguida chamar o método **preencherEndereço** da classe Endereço.

- **Excluir cadastro** – é um método que removerá o enfermeiro da lista de enfermeiros.
- **Lista atendimento** – vai listar todos os pacientes que ainda precisam ser atendidos.
- **Atender paciente** – realiza o atendimento do paciente administrando a medicação receitada, isso será feito através do método `administrarMedicamento` que funciona por meio de uma associação entre a classe `Enfermeiros` e a classe `Medicamento`. Após isso o enfermeiro pode solicitar a alta do paciente ao médico, ou atender o próximo paciente.

[2.9] Medicamentos

Função: a classe `Medicamentos` tem como função realizar o gerenciamento dos medicamentos, fazendo seu cadastro e seu uso.



Corpo: a classe segue o padrão de encapsulamento, possuindo seu construtor, getters e setters. Ela possui ainda os métodos `cadastrarMedicamento` e `administrarMedicamento`.

-
- **Cadastrar medicamento** – pede as informações do medicamento, tais quais: nome, código e quantidade.
 - **Administrar medicamento** – método que faz o uso do medicamento nos pacientes através da associação com a classe Enfermeiros.

3. Princípios da Orientação a Objetos

O projeto conta com todos os princípios da orientação a objetos.

Abstração – Utilizada principalmente para verificar o que haveria de funcionalidade no sistema;

Encapsulamento – Presente em todas as classes que possuam algum tipo de variável;

Herança – Visível entre as classes.

Endereço -> Pessoa -> Paciente.

Endereço -> Pessoa -> Funcionário -> Médicos.

Endereço -> Pessoa -> Funcionário -> Enfermeiros.

Polimorfismo – Utilizada em diversos momentos onde foi necessário realizar cadastro e exclusão de alguma instância.

Vale também mencionar que, embora não seja um dos pilares da orientação a objetos, também foi utilizada associação (entre a classe Enfermeiros e Medicamentos).