



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
BACHARELADO INTERDISCIPLINAR EM CIÊNCIA E TECNOLOGIA
PARADIGMAS DE PROGRAMAÇÃO

Discentes: Kauan Guilherme Alves Pinheiro Santos, Virginia Maria Mondego Ferreira e Wanderson Campos Soares.

Assunto: Explicação do projeto na linguagem JAVA desenvolvido para obtenção da terceira nota avaliativa da disciplina de Paradigmas de Programação.

Primeiramente o grupo escolheu o tema/finalidade do projeto e foi definido que seria um sistema de cadastro em uma clínica. As funcionalidades deste projeto será cadastrar, atualizar, remover os pacientes e marcar a consulta desejada. Para isso, definimos, como segundo passo, as classes necessárias para a execução. As primeiras foram a classe Usuario (classe mãe) e a classe Paciente. Depois, criamos a classe Consulta, a classe Gui e por fim a classe Main para execução de tudo.

DIAGRAMA DE CASO DE USO

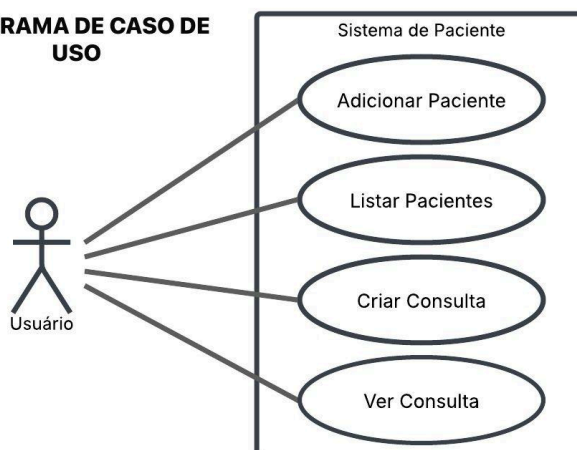
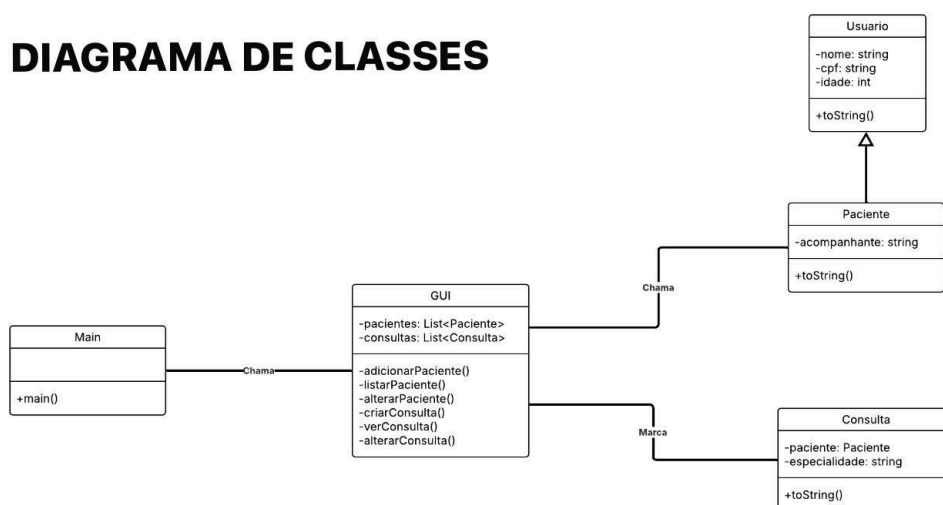


DIAGRAMA DE CLASSES



1. **Classe Usuario:** Classe abstrata que possui atributos básicos para paciente, um construtor para inicializar esses dados, os métodos getters e setters para acessar os atributos privados, e um método toString para transformar o objeto em uma string legível.

Atributos:

-*nome(String)*: nome do usuário;

-*cpf(String)*: CPF do usuário;

-*idade(int)*: Idade do Usuário.

Métodos:

+*toString()*: Retorna uma representação em texto dos atributos do usuário, formatando em uma string.

2. **Classe Paciente:** Herda os atributos da classe mãe (Usuário) e tem um atributo adicionado somente a ele, um construtor para inicializar os seus atributos e o método “super” para acessar os atributos da classe mãe. Possui métodos get e set para seu novo atributo. Também possui o método toString.

Atributos:

-*acompanhante(String)*: Nome do acompanhante do paciente.

Métodos:

+*toString()*: Retorna uma string representando os detalhes do paciente, incluindo o acompanhante.

3. **Classe Consulta:** possui dois atributos, e um desses atributos armazenam um objeto da classe paciente. Possui um construtor, getters e setters e método toString.

Atributos:

-*paciente(Paciente)*: O paciente que marcou a consulta;

-*especialidade(String)*: Área médica da consulta.

Métodos:

+*toString()*: Retorna uma string representando os detalhes da consulta.

4. **Classe GUI:** Responsável por criar a interface gráfica da aplicação, herdando do componente JFrame, sendo a janela principal da aplicação, utilizando a biblioteca Java Swing. Seus atributos são listas de objetos das classes Usuario e Consulta, possui um construtor chamado Gui que configura os títulos, tamanho e fechamento da nossa janela, além de criar o painel com o JPanel, adicionando os 4 botões e definindo o que irão fazer (chamada dos métodos). Essa classe permite com seus métodos: cadastrar, listar, alterar e excluir pacientes, além de criar, alterar, excluir e listar consultas. O uso de JDialog facilita a criação de janelas para cada funcionalidade, tornando o sistema mais interativo e intuitivo.

Atributos:

-*pacientes(List<Paciente>)*: Lista que armazena os pacientes cadastrados no sistema.

Métodos:

-*adicionarPaciente()*: Adiciona um novo paciente à lista;

-*listarPaciente()*: Exibe todos os pacientes cadastrados;

-*alterarPaciente()*: Permite modificar os dados de um paciente;

-*criarConsulta()*: Cria uma nova consulta para um paciente específico;

-*verConsulta()*: Exibe os detalhes de uma consulta agendada.

-*alterarConsulta()*: Permite modificar os dados de uma consulta existente.

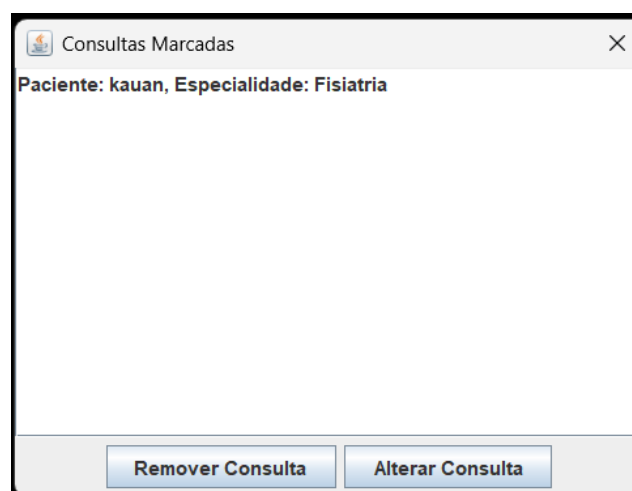
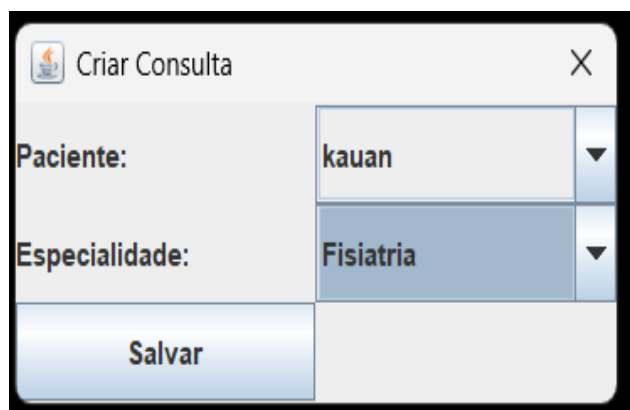
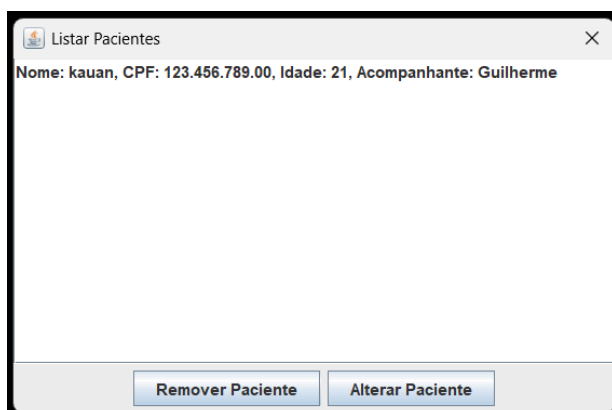
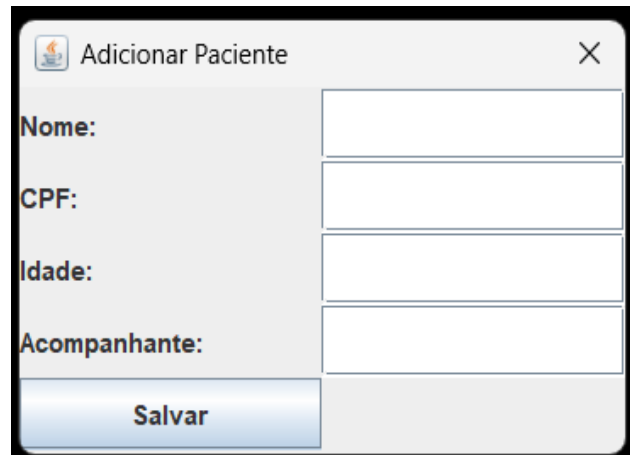
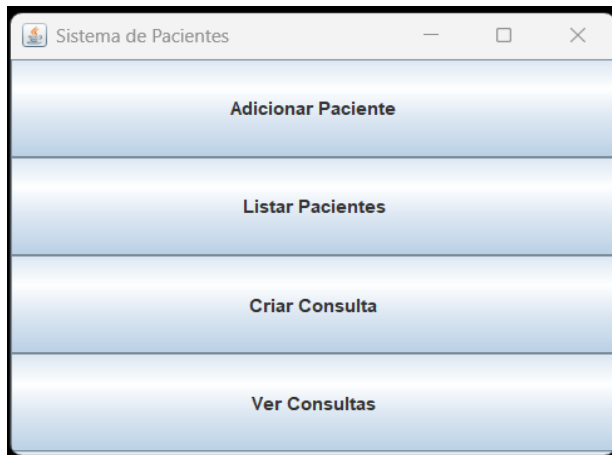
5. **Classe main:** Inicializa a interface gráfica do sistema ao criar e exibir a janela principal (Gui). Ele garante que a interface seja gerenciada corretamente na thread do Swing, proporcionando a execução do programa.

Métodos:

+*main()*: método que inicializa o programa e chama a interface gráfica.

Os dados do sistema, como informações de pacientes e consultas, são armazenados temporariamente em listas (List<Paciente> e List<Consulta>) durante a execução do programa. Não há persistência em arquivos ou banco de dados, ou seja, os dados são perdidos quando o programa é encerrado.

A interface gráfica foi desenvolvida utilizando a biblioteca Java Swing, que permite a criação de janelas interativas.



A classe GUI herda de JFrame, tornando-se a janela principal do sistema. Além disso, o programa utiliza JDialog para criar janelas modais que facilitam a interação com o usuário ao adicionar, listar ou modificar pacientes e consultas. O uso desses componentes torna a aplicação mais intuitiva e organizada, separando visualmente cada funcionalidade.

A interatividade do sistema é possível graças ao uso de ActionListeners, que são atribuídos aos botões para capturar as ações do usuário. Sempre que um botão é pressionado, um evento é acionado, executando o método correspondente. Por exemplo, ao clicar no botão "Adicionar Paciente", um JDialog é aberto para entrada de dados, e a informação é adicionada à lista de pacientes. Esse mecanismo garante que cada funcionalidade responda de forma dinâmica às ações do usuário, tornando a aplicação mais eficiente e interativa.