



CENTRO UNIVERSITÁRIO CENAC  
SANTO AMARO

# **Sistema De Gerenciamento Farmácias**

• BRUNO HENRIQUE GOUVEIA DA SILVA

**CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

São Paulo

2024

Este programa em Java é um sistema de gerenciamento de farmácias. Ele permite que os usuários realizem várias operações em uma lista de medicamentos (Remedios) armazenada em um vetor (VetorFarmacia).

Aqui está uma análise da funcionalidade do programa:

1. O programa começa exibindo um menu ao usuário com opções para:
  - Adicionar um novo medicamento (Remedio)
  - Remover um medicamento
  - Verifique a quantidade de medicamentos registrados
  - Exibir todos os medicamentos registrados
  - Pesquisar um medicamento pelo nome
  - Pesquisar um medicamento por posição
  - Alterar dados do remédio (Modificar dados do medicamento)
  - Sair do programa
2. Quando o usuário seleciona uma opção, o programa executa a ação correspondente:
  - Adicionar um novo medicamento: solicita que o usuário insira o nome, a quantidade, o preço e o tipo do medicamento e, em seguida, o adiciona ao vetor.
  - Removendo um medicamento: solicita que o usuário insira o nome do medicamento, procura-se no vetor e o remove, se encontrado.
  - Verificação do número de medicamentos registrados: exibe o tamanho atual do vetor.
  - Exibindo todos os medicamentos registrados: exibe uma lista de todos os medicamentos no vetor.
  - Pesquisar um medicamento pelo nome: solicita que o usuário insira o nome do medicamento, pesquisa-o no vetor e exibe seus detalhes, se encontrado.
  - Procurando um medicamento por posição: solicita que o usuário insira a posição do medicamento, procura-o no vetor e exibe seus detalhes, se encontrado.
  - Modificando dados do medicamento: solicita que o usuário insira o nome do medicamento, procura-se o no vetor e permite que o usuário atualize seus dados, caso os encontre.
3. O programa continua exibindo o menu e realizando ações até que o usuário selecione a opção "Finalizar Programa".

A (**VetorFarmacia**) classe representa o vetor de medicamentos, e a (**Remedio**) classe representa um medicamento individual. A (**Executar**) classe contém o método main que direciona a execução do programa.

## Classe Executar

```
package ADO1;

import javax.swing.JOptionPane;

/**
 *
 * @author bruno.hgsilva3
 */
public class Executar {

    public static void main(String[] args) throws Exception {

        Remedio r = new Remedio();

        VetorFarmacia vetor = new VetorFarmacia();

        String opcao = "";

        do {

            opcao = (String) JOptionPane.showInputDialog(null, "Informe operação desejada:",
"Opção",

                JOptionPane.INFORMATION_MESSAGE, null, r.getOp(), r.getOp()[0]);

            switch (opcao) {

                case "Finalizar Programa" -> {

                    JOptionPane.showMessageDialog(null, "Ate a Proxima!!!");

                }

                case "1) Adiciona Novo remedio" -> {

                    String nome = JOptionPane.showInputDialog("informe nome do remedio: ");

                    String qtdDeGramaRemedio = JOptionPane.showInputDialog("informe quantas
Gramas tem a caixa: ");

                    int qtdComprimidos = Integer.parseInt(JOptionPane.showInputDialog("informe
quantidade de comprimidos: "));

                    int qtdDeCaixaRemedio = Integer.parseInt(JOptionPane.showInputDialog("informe
quantidade de caixas de remedios: "));

                    double valorRemedio =
Double.parseDouble(JOptionPane.showInputDialog("informe o valor do remedio: "));

                    String tipoRemedio = (String) JOptionPane.showInputDialog(null, "Informe o tipo do
remedio:", "Opção",

                        JOptionPane.INFORMATION_MESSAGE, null, r.getTipos(), r.getTipos()[0]);

                    Remedio dadosremedio = new Remedio(nome, qtdDeGramaRemedio,
qtdComprimidos, qtdDeCaixaRemedio, valorRemedio, tipoRemedio);
```

```

        vetor.adiciona(dadosremedio);
OptionPane.showMessageDialog(null, "Remedio Cadastrado com sucesso");
    }
    case "2) Remove remedio" -> {
        String nome = JOptionPane.showInputDialog("Informe o nome do remedio:");
        if (JOptionPane.showConfirmDialog(null, "Deseja remove esse remedio:\n" +
vetor.buscaPorNome(nome)) == 0) {
            vetor.remove(vetor.buscaPosicao(nome));
        }
    }
    case "3) Quantidade de remedio cadastrado" -> {
        JOptionPane.showMessageDialog(null, "Quantidade de remedios cadastrado é: " +
vetor.tamanho);
    }
    case "4) Mostrar remedios cadastrados" -> {
        JOptionPane.showMessageDialog(null, vetor);
    }
    case "5) Busca posição do remedio" -> {
        String nomeRmedio = JOptionPane.showInputDialog("informe nome do remedio:
");
        if (vetor.buscaPosicao(nomeRmedio) >= 0) {
            JOptionPane.showMessageDialog(null, "A posição do remedio cadastrado é: " +
vetor.buscaPosicao(nomeRmedio));
        } else {
            JOptionPane.showMessageDialog(null, "Remedio não encontrado");
        }
    }
    case "6) Busca dados do remedio" -> {
        String nomeRmedio = JOptionPane.showInputDialog("informe nome do remedio:
");
        JOptionPane.showMessageDialog(null, vetor.buscaPorNome(nomeRmedio));
    }
    case "7) Alterar dados do remedio" -> {
        String nomeRmedio = JOptionPane.showInputDialog("informe nome do remedio:
");

```

```

        Remedio remedio = vetor.buscaPorNome(nomeRmedio);

        if (remedio != null) {

            String novoNome = JOptionPane.showInputDialog("informe novo nome do
remedio: ");

            String qtdDeGramaRemedio = JOptionPane.showInputDialog("informe nova
quantidade de Gramas: ");

            int qtdComprimidos = Integer.parseInt(JOptionPane.showInputDialog("informe
nova quantidade de comprimidos: "));

            int qtdDeCaixaRemedio =
Integer.parseInt(JOptionPane.showInputDialog("informe nova quantidade de caixas de
remedios: "));

            double valorRemedio =
Double.parseDouble(JOptionPane.showInputDialog("informe novo valor do remedio: "));

            String tipoRemedio = (String) JOptionPane.showInputDialog(null, "Informe o
novo tipo do remedio:", "Opção",

                JOptionPane.INFORMATION_MESSAGE, null, r.getTipos(), r.getTipos()[0]);

            remedio.setNomeRemedio(novoNome);

            remedio.setQtdDeGramaRemedio(qtdDeGramaRemedio);

            remedio.setQtdComprimidos(qtdComprimidos);

            remedio.setQtdDeCaixaRemedio(qtdDeCaixaRemedio);

            remedio.setValorRemedio(valorRemedio);

            remedio.setTipoRemedio(tipoRemedio);

            JOptionPane.showMessageDialog(null, "Dados do remedio alterados com
sucesso");

        } else {

            JOptionPane.showMessageDialog(null, "Remedio não encontrado");

        }

    }

    default ->

        JOptionPane.showMessageDialog(null, "opção invalida, tente novamente");

    }

} while (!opcao.equalsIgnoreCase("Finalizar Programa"));

}

}

```

## Classe Remédio

```
package ADO1;

/**
 * @author bruno.hgsilva3
 */

public class Remedio {

    private String nomeRemedio;

    private String qtdDeGramaRemedio;

    private int qtdComprimidos;

    private int qtdDeCaixaRemedio;

    private double valorRemedio;

    private String tipoRemedio;

    private String[] tipos = {"Escolha um Tipo", "Alergias", "Anti-inflamatorio", "Antidepressivos",
"Calmanantes", "Asma", "Congestao nasal", "Diabetes", "Dor e Febre", "Dor de garganta", "Gripe e
resfriado", "Gastrite", "Infecções", "Insônia", "Tosse"};

    private String[] op = {"Opções", "1) Adiciona Novo remedio", "2) Remove remedio", "3) Quantidade de
remedio cadastrado", "4) Mostrar remedios cadastrados", "5) Busca posição do remedio", "6) Busca dados
do remedio", "7) Alterar dados do remedio", "Finalizar Programa"};

    public String[] getOp() {

        return op;

    }

    public void setOp(String[] op) {

        this.op = op;

    }

    public Remedio() {

    }

    public String[] getTipos() {

        return tipos;

    }

    public Remedio(String nomeRemedio, String qtdDeGramaRemedio, int qtdComprimidos, int
qtdDeCaixaRemedio, double valorRemedio, String tipoRemedio) {

        this.nomeRemedio = nomeRemedio;

        this.qtdDeGramaRemedio = qtdDeGramaRemedio;

        this.qtdComprimidos = qtdComprimidos;

        this.qtdDeCaixaRemedio = qtdDeCaixaRemedio;

        this.valorRemedio = valorRemedio;

    }

}
```

```
        this.tipoRemedio = tipoRemedio;
    }

    public String getNomeRemedio() {
        return nomeRemedio;
    }

    public void setNomeRemedio(String nomeRemedio) {
        this.nomeRemedio = nomeRemedio;
    }

    public String getQtdDeGramaRemedio() {
        return qtdDeGramaRemedio;
    }

    public void setQtdDeGramaRemedio(String qtdDeGramaRemedio) {
        this.qtdDeGramaRemedio = qtdDeGramaRemedio;
    }

    public int getQtdComprimidos() {
        return qtdComprimidos;
    }

    public void setQtdComprimidos(int qtdComprimidos) {
        this.qtdComprimidos = qtdComprimidos;
    }

    public int getQtdDeCaixaRemedio() {
        return qtdDeCaixaRemedio;
    }

    public void setQtdDeCaixaRemedio(int qtdDeCaixaRemedio) {
        this.qtdDeCaixaRemedio = qtdDeCaixaRemedio;
    }

    public double getValorRemedio() {
        return valorRemedio;
    }

    public void setValorRemedio(double valorRemedio) {
        this.valorRemedio = valorRemedio;
    }

    public String getTipoRemedio() {
        return tipoRemedio;
    }
}
```

```
public void setTipoRemedio(String tipoRemedio) {  
    this.tipoRemedio = tipoRemedio;  
}  
  
@Override  
public String toString() {  
    StringBuilder sb = new StringBuilder();  
    sb.append("\n=====\\n");  
    sb.append("Remedio: ").append(nomeRemedio).append("\\n");  
    sb.append("Grama Remedio: ").append(qtdDeGramaRemedio).append("\\n");  
    sb.append("Quantidade Comprimidos: ").append(qtdComprimidos).append("\\n");  
    sb.append("Quantidade Caixa Remedios: ").append(qtdDeCaixaRemedio).append("\\n");  
    sb.append("valor Remedio: $").append(valorRemedio).append("\\n");  
    sb.append("Tipo Remedio: ").append(tipoRemedio).append("\\n");  
    sb.append("=====\\n");  
    return sb.toString();  
}  
}
```



## Classe Vetor Remédio

```
package ADO1;

import javax.swing.JOptionPane;

/**
 * @author bruno.hgsilva3
 */

public class VetorFarmacia {

    // lista de atributos

    public Object[] elementos;

    public int tamanho;

    // metodo construtor

    public VetorFarmacia() {

        //this palavra-chave Java é usada para referenciar a instância atual do método
        //no qual é usada. Aqui, isso se refere à variável de instância. ...

        //Isso chama o construtor da mesma classe java que possui um parâmetro .

        this.elementos = new Object[20];

        this.tamanho = 0;

    }

    public void adiciona(Object elemento) throws Exception {

        this.aumentaCapacidade();

        if (this.tamanho < this.elementos.length) {

            this.elementos[this.tamanho] = elemento;

            this.tamanho++;

        } else {

            throw new Exception("o vetor ja esta cheio");

        }

    }

    public int tamanho() {

        return this.tamanho;

    }

    @Override

    public String toString() {

        StringBuilder s = new StringBuilder();
```

```

s.append("->\n");
for (int i = 0; i < this.tamanho - 1; i++) {
    s.append(i + 1);
    s.append(this.elementos[i]);
}
if (this.tamanho > 0) {
    s.append(this.tamanho);
    s.append(this.elementos[this.tamanho - 1]);
}
s.append("<-");
return s.toString();
}

public Remedio buscaPorNome(String nomeRemedio) {

    for (int i = 0; i < tamanho; i++) {
        if (elementos[i] instanceof Remedio) {
            Remedio remedio = (Remedio) elementos[i];
            if (remedio.getNomeRemedio().equals(nomeRemedio)) {
                return remedio;
            }
        }
    }

    return null;
}

public int buscaPosicao(String nomeRemedio) {
    for (int i = 0; i < tamanho; i++) {
        if (elementos[i] instanceof Remedio) {
            Remedio remedio = (Remedio) elementos[i];
            if (remedio.getNomeRemedio().equals(nomeRemedio)) {
                return i;
            }
        }
    }

    return -1;
}

```

```

public boolean adicionalInicio(int posicao, Object elemento) throws Exception {
    this.aumentaCapacidade();
    if (posicao >= 0 && posicao < tamanho) {
        for (int i = this.tamanho - 1; i >= posicao; i--) {
            this.elementos[i + 1] = this.elementos[i];
        }
        this.elementos[posicao] = elemento;
        this.tamanho++;
    } else {
        throw new Exception("posicao invalida");
    }
    return true; }

public void aumentaCapacidade() {
    if (this.elementos.length == this.tamanho) {
        Object[] elementosNovos = new Object[this.elementos.length * 2];
        for (int i = 0; i < this.elementos.length; i++) {
            elementosNovos[i] = this.elementos[i];
        }
        this.elementos = elementosNovos;
    }
}

public void remove(int posicao) throws Exception {
    if (posicao >= 0 && posicao < this.tamanho) {
        for (int i = posicao; i < this.tamanho - 1; i++) {
            this.elementos[i] = this.elementos[i + 1];
        }
        this.tamanho--;
        JOptionPane.showMessageDialog(null, "Remedio removido com sucesso");
    } else {
        throw new Exception("posicao invalida");
    }
}
}

```