MAPA – Material de Avaliação Prática da Aprendizagem

Acadêmico: FRANCISCO DE A. DA SILVA.	**R.A.** 1918714-5
Curso: ENGENHARIA DE SOFTWARE	
Disciplina: PROGRAMAÇÃO DE SISTEMAS II	
Valor da atividade: 3,00	**Prazo:** 09/10/23 a 08/12/23

Instruções para Realização da Atividade

1. Todos os campos acima deverão ser devidamente preenchidos;
2. É obrigatória a utilização deste formulário para a realização do MAPA;
3. Esta é uma atividade individual. Caso identificado cópia de colegas, o trabalho de ambos sofrerá decréscimo de nota;
4. Utilizando este formulário, realize sua atividade, salve em seu computador, renomeie e envie em forma de anexo no campo de resposta da atividade MAPA;
5. Formatação exigida para esta atividade: documento Word, Fonte Arial ou Times New Roman tamanho 12, Espaçamento entre linhas 1,5, texto justificado;
6. Ao utilizar quaisquer materiais de pesquisa referência conforme as normas da ABNT;
7. No campo “Material da disciplina”, no ambiente virtual da disciplina, você encontrará orientações importantes para elaboração desta atividade. Confira!
8. Critérios de avaliação: Utilização do template; atendimento ao Tema; Constituição dos argumentos e organização das Ideias; Correção Gramatical e atendimento às normas ABNT.
9. Procure argumentar claramente e objetiva, conforme o conteúdo da disciplina.

Em caso de dúvidas, entre em contato com seu Professor Mediador.

Bons estudos!

LET’s GETTING STARTED

MAPA PROGRAMAÇÃO DE SISTEMAS II

Código fonte do projeto

Classe Principal

```
package MenuPrin;

import View.Crianca_view;
import View.Estadia_View;
import View.Responsavel_view;
import java.awt.Color;
import java.awt.Font;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.time.LocalDate;
import java.time.LocalTime;
import java.util.Calendar;
import java.util.Locale;
import javax.swing.*;
/**
 * classe principal
 * @author kim
 */
public class ProgramacaoSistemas_II extends JFrame implements ActionListener {
    /**
     * @param args the command line arguments
     */
        JMenuBar barraMenu;
        JMenu cadastro, pesquisa, backup, confbanco;
        JMenuItem jmiResp, jmiCrianca, jmiEstadia, jmipesquisar, jmibackstore, jmiconfig;
        JPanel panelInferior, panelGeral;
        static JLabel welcome, labelInferior, lb_dataHora, lb_geral, lb_ead, lb_oneself;
        ImageIcon img_eduDistancia, img_unicesumar, img_ead, img_Icon;
        Locale local = Locale.getDefault();
        Calendar calend = Calendar.getInstance();

    /**
     * método construtor
     */
    public ProgramacaoSistemas_II(){

        setTitle(title: " Mapa Programação de Sistemas II ");
        setSize(width: 400, height:500);
        setLocation(x: 500, y: 180);
        setResizable(resizable:false);
        getContentPane().setLayout(mgr:null);
        setDefaultCloseOperation(operation:JFrame.EXIT_ON_CLOSE);

        setIcon();

         //Inicialização dos componebtes
        barraMenu     = new JMenuBar();
        cadastro      = new JMenu(s: "Cadastro");
        pesquisa      = new JMenu(s: "Pesquisa");
        backup        = new JMenu(s: "Backup");
        confbanco     = new JMenu(s: "Configurações");

        cadastro      .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        pesquisa      .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        backup        .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        confbanco     .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));

        jmiResp       = new JMenuItem(text: "Responsável");
        jmipesquisar  = new JMenuItem(text: "Search");
        jmibackstore  = new JMenuItem(text: "Restore");
        jmiconfig     = new JMenuItem(text: "Banco");
        jmiCrianca    = new JMenuItem(text: "Criança ");
```

```
        jmiEstadia     = new JMenuItem(text: "Estadia");

        jmiResp       .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        jmipesquisar  .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        jmibackstore  .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        jmiconfig     .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        jmiCrianca    .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));
        jmiEstadia    .setFont(new Font(name: "Arial", style: Font.BOLD,size: 14));

        img_unicesumar     = new ImageIcon(filename: "./src/img/nome.png");
        img_eduDistancia   = new ImageIcon(filename: "./src/img/educacaoADistancia.jpeg");
        img_ead            = new ImageIcon(filename: "./src/img/ead.png");
        img_Icon           = new ImageIcon(filename: "./src/img/icon.png");

        welcome         = new JLabel(text: "Welcome to System ");
        getContentPane().add(comp: welcome);
        welcome        .setBounds(x: 10, y: 0, width: 200, height:100);
        welcome        .setForeground(fg: Color.red);
        welcome        .setFont(new Font(name: "Arial", style: Font.BOLD,size: 16));

        labelInferior= new JLabel(image: img_unicesumar);
        labelInferior.setFont(new Font(name: "Arial", style: Font.BOLD,size: 16));
        labelInferior.setForeground(fg: Color.WHITE);
        labelInferior.setBounds(x: 2, y: 2, width: 150, height:25);

        lb_geral = new JLabel(image: img_eduDistancia);
        lb_geral.setFont(new Font(name: "Arial", style: Font.BOLD,size: 16));
        lb_geral.setForeground(fg: Color.WHITE);
        lb_geral.setBounds(x: 80, y: 100, width: 200, height:200);

        lb_ead         = new JLabel(image: img_ead);
        getContentPane().add(comp: lb_ead);

        lb_ead          .setBounds(x: 310, y: 0, width: 50, height:100);
        lb_ead          .setForeground(fg: Color.red);

        lb_dataHora= new JLabel();
        lb_dataHora.setFont(new Font(name: "Arial", style: Font.BOLD,size: 12));
        lb_dataHora.setForeground(fg: Color.WHITE);
        lb_dataHora.setBounds(x: 180, y: 2, width: 170, height:25);

         lb_oneself = new JLabel(text: "Francisco de A. da Silva ");
         getContentPane().add(comp: lb_oneself);
        lb_oneself.setFont(new Font(name: "Arial", style: Font.BOLD,size: 16));
        lb_oneself.setForeground(fg: Color.YELLOW);
        lb_oneself.setBounds(x: 80, y: 310, width: 220, height:25);

        //Adicionando os menu na barra de menu
        setJMenuBar(menubar:barraMenu);
        barraMenu.add(c: cadastro);
        barraMenu.add(c: pesquisa);
        barraMenu.add(c: backup);
        barraMenu.add(c: confbanco);

        cadastro.add(menuItem: jmiResp);
        cadastro.add(menuItem: jmiCrianca);
         cadastro.add(menuItem: jmiEstadia);
        pesquisa.add(menuItem: jmipesquisar);
        backup.add(menuItem: jmibackstore);
        confbanco.add(menuItem: jmiconfig);

        //Adicionar os componentes no contente
        getContentPane().add(comp: barraMenu);

        panelInferior = new JPanel();

         panelInferior.add(comp: labelInferior);
         panelInferior.add(comp: lb_dataHora);
         getContentPane().add(comp: panelInferior);
         panelInferior.setLayout(mgr:null);
         panelInferior.setBackground(bg: Color.BLACK);
         panelInferior.setBounds(x: 10, y: 400, width: 367, height:30);

         panelGeral = new JPanel();
         panelGeral.add(comp: lb_geral);
         panelGeral.add(comp: lb_ead);
         getContentPane().add(comp: panelGeral);
         panelGeral.setLayout(mgr:null);
         panelGeral.setBackground(bg: Color.BLACK);
         panelGeral.setBounds(x: 10, y: 10, width: 367, height:385);

         //Adicionando ações para
         jmiconfig .addActionListener(l: this);
         jmiResp   .addActionListener(l: this);
         jmiCrianca.addActionListener(l: this);
         jmiEstadia.addActionListener(l: this);

         lerData();
    }
    /**
     * método principal Main
     * @param args
     */
    public static void main(String[] args) {
        JFrame shop = new ProgramacaoSistemas_II();
        shop.setVisible(b: true);
    }
```

```
159     /**
160      * mpetodo responsável por disparar ações nos components
161      * @param e
162      */
163     @Override
        public void actionPerformed(ActionEvent e) {
165         if(e.getSource()==jmiResp){
166              JFrame cadPadrao = new Responsavel_view();
167             cadPadrao.setVisible(b: true);
168         }
169         if(e.getSource()==jmiconfig){
170             JFrame Configura = new Configuracoes();
171             Configura.setVisible(b: true);
172         }
173         if(e.getSource()==jmiCrianca){
174             JFrame crianca = new Crianca_view();
175             crianca.setVisible(b: true);
176         }
177         if(e.getSource()==jmiEstadia){
178             JFrame estad = new Estadia_View();
179             estad.setVisible(b: true);
180         }
181     }
182     /**
183      * método para leitura da data e hora
184      */
185     public void lerData(){
186
187         LocalDate agora = LocalDate.now();
188          LocalTime hora = LocalTime.now();
189      lb_dataHora.setText("Data: "+String.valueOf(obj:agora)+" Hora: "+ hora);
190     }
```

```
191     /**
192      * método para trocar o icone padrão do java
193      */
194     private void setIcon() {
195         setIconImage(image: Toolkit.getDefaultToolkit().getImage(url:getClass().getResource(name: "icon.png")));
196     }
197 }
```

//**

Classe Responsável Model

```
1   /*
2    * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3    * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4    */
5   package Model;
6
7   import java.io.Serializable;
8   import java.util.ArrayList;
9
10  /**
11   *
12   * @author kim
13   */
14  public class Responsavel_Model implements Serializable {
15      /**
16       * @return the
17       */
18      private String nome;
19      private String cpf;
20      private String endereco;
21      private String telefone;
22      private String email;
23      private int idade;
24
25       private ArrayList<Responsavel_Model> responsavel;
26
27      public Responsavel_Model(String nome, String cpf, String telefone, String email, String endereco, int idade) {
28          this.nome = nome;
29          this.cpf = cpf;
30          this.telefone = telefone;
31          this.email = email;
32          this.endereco = endereco;
```

```
        this.idade = idade;
    }
    public Responsavel_Model(){
      responsavel = new ArrayList<>();
    }

    public String getCpf() {
        return cpf;
    }
    /**
     * @param cpf the cpf to set
     */
    public void setCpf(String cpf) {
        this.cpf = cpf;
    }

    /**
     * @return the nome
     */
    public String getNome() {
        return nome;
    }

    /**
     * @param nome the nome to set
     */
    public void setNome(String nome) {
        this.nome = nome;
    }

    /**
     * @return the telefone
     */
    public String getTelefone() {
        return telefone;
    }

    /**
     * @param telefone the telefone to set
     */
    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }
    /**
     * @param endereco the endereco to set
     */

    public int getIdade() {
        return idade;
    }

    /**
     * @param idade the idade to set
     */
    public void setIdade(int idade) {
        this.idade = idade;
    }

    /**
     * @return the email
     */
    public String getEmail() {
        return email;
    }

    /**
     * @param email the email to set
     */
    public void setEmail(String email) {
        this.email = email;
    }

    /**
     * @return the endereco
     */
    public String getEndereco() {
        return endereco;
    }

    /**
     * @param endereco the endereco to set
     */
    public void setEndereco(String endereco) {
        this.endereco = endereco;
    }
    @Override
    public String toString() {
        return "Responsavel_Model{" + "nome=" + nome + ", cpf=" + cpf + ", telefone=" + telefone + ", email=" + email + ","
                + " endereco=" + endereco + ", idade=" + idade + '}';
    }
    public void adicionarResponsavel(Responsavel_Model responsa){
        responsavel.add(e: responsa);
    }
}
```

//**********

Classe Criança Model

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Model;

import View.Responsavel_view;
import java.io.Serializable;

/**
 *
 * @author kim
 */
public class Crianca_Model implements Serializable {

    private Responsavel_view resposnsa;
    private String nome;
    private int idade;
    private String sexo;

    public Crianca_Model() { }


    public Responsavel_view getResposnsa() {
        return resposnsa;
    }

    public void setResposnsa(Responsavel_view resposnsa) {
        this.resposnsa = resposnsa;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public int getIdade() {
        return idade;
    }

    public void setIdade(int idade) {
        this.idade = idade;
    }

    public String getSexo() {
        return sexo;
    }

    public void setSexo(String sexo) {
        this.sexo = sexo;
    }
}
```

//**********

Classe Estadia Model

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Model;

/**
 *
 * @author kim
 */
public class Estadia_Model {

    private double tempoNoBrinquedo;
    private double totalEstadia;
    final double valueMinuto = 1.50;

    /**
     * métodos getteres and setteres
     * @return
     */
    public double getTempoNoBrinquedo() {
        return tempoNoBrinquedo;
    }

    public void setTempoNoBrinquedo(double tempoNoBrinquedo) {
        this.tempoNoBrinquedo = tempoNoBrinquedo;
    }

    public double getTotal() {
        return totalEstadia;
    }

}
```

```java
    public void setTotal(double total) {
        this.totalEstadia = total;
    }
    /**
     * método que calcula o valor da estadia
     *
     * @param value1
     * @return estadia
     */
    public double calcularEstadia(double value1) {
        double estadia = 0;
        if (value1 >= 20 && value1 < 40) {
            estadia = ((value1 * valueMinuto) - ((value1 * valueMinuto) * 0.05));
        } else if (value1 >= 40 && value1 < 60) {
            estadia = ((value1 * valueMinuto) - ((value1 * valueMinuto) * 0.1));
        } else if (value1 >= 60) {
            estadia = ((value1 * valueMinuto) - ((value1 * valueMinuto) * 0.15));
        } else {
            estadia = value1 * valueMinuto;
        }
        return estadia;
    }
    /**
     * método para calcular o valor do desconto
     * @param value1
     * @return desconto
     */
    public double calcularDesconto(double value1) {
        double desconto = ((value1 * valueMinuto) - calcularEstadia(value1));
        return desconto;
    }
}
```

//**

Classe Criança controller

```java
package Controller;
/**
 *
 * @author kim
 */
import Model.Crianca_Model;
import View.Crianca_view;

public class Crianca_Controller {

    private Crianca_Model cm;
    private Crianca_view cv;

    public Crianca_Controller(Crianca_Model cmd, Crianca_view vw ){
        this.cm = cmd;
        this.cv = vw;
    }
    public void setNomeCrianca(String nome){
     cm.setNome(nome);
  }
    public void setSexoCrianca(String sexo){
        cm.setSexo(sexo);
    }
    public void setIdadeCrianca(int idade){
        cm.setIdade(idade);
    }
    public void atualizaView(){
        cv.printResponsavelDetalhes(respName: cm.getNome());

    }
}
```

//**

Classe Estadia Controller

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Controller;

import Model.Crianca_Model;
import Model.Estadia_Model;
import Model.Responsavel_Model;
import View.Estadia_View;

/**
 * classe estadia controller
 *
 * @author kim
 */
public class Estadia_Controller {

    private Estadia_View EstWiew;
    private Estadia_Model EstModel;
    private Responsavel_Model respModel;
    private Crianca_Model crianca;

    /**
     * Método construtor
     *
     * @param model
     * @param view
     */
    public Estadia_Controller(Estadia_Model model, Estadia_View view) {
        this.EstModel = model;
        this.EstWiew = view;
    }

    public void atualizaEstadiaView() {
        EstWiew.printDadosCalculos(value: EstModel.getTotal());
        EstWiew.printDadosCalculos(value: EstModel.getTempoNoBrinquedo());
    }

    /**
     * Método para autualizar a view
     */
    public void atualizaWiew() {
        EstWiew.receberResponsavel(responsa: respModel);
        EstWiew.recebeCrianca(crianca);
    }
}
```

//**********

Classe Responsável controller

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package Controller;

/**
 * classe Controller
 *
 * @author kim
 */
import View.*;
import Model.*;

public class Responsavel_Controller {

    private Responsavel_view respWiew;
    private Responsavel_Model respModel;
    private Crianca_Model crianca;

    /**
     * Método construtor
     *
     * @param model
     * @param view
     */
    public Responsavel_Controller(Responsavel_Model model, Responsavel_view view) {
        this.respModel = model;
        this.respWiew = view;
    }

    /**
```

```
33          * métodos para recuperar e setar os dados
34          *
35          * @param nome
36          */
37         public void setNomeResponsavel(String nome) {
38             respModel.setNome(nome);
39         }
40
41         public void setRespCpfResponsavel(String cpf) {
42             respModel.setCpf(cpf);
43         }
44
45         public void setRespResEnderecoponsavel(String endereco) {
46             respModel.setEndereco(endereco);
47         }
48
49         public void setResptelefoneResponsavel(String telefone) {
50             respModel.setTelefone(telefone);
51         }
52
53         public void setRespEmailResponsavel(String email) {
54             respModel.setEmail(email);
55         }
56
57         public void setRespIdadeResponsavel(int idade) {
58             respModel.setIdade(idade);
59         }
60
61         public String getResponsavelNome() {
62             return respModel.getNome();
63         }
```

```
64
65         public String getResponsavelCpf() {
66             return respModel.getCpf();
67         }
68
69         public String getResponsavelEndereo() {
70             return respModel.getEndereco();
71         }
72
73         public String getResponsavelTelefone() {
74             return respModel.getTelefone();
75         }
76
77         public String getResponsavelEmail() {
78             return respModel.getEmail();
79         }
80
81         public String getResponsavelIdade() {
82             return (String.valueOf(respModel.getIdade()));
83         }
84
85         /**
86          * Método para atualizar a view
87          */
88         public void atualizaView() {
89             respView.printDadosResponsavel(nome: respModel.getNome(), cpf: respModel.getCpf(), endereco: respModel.getEndereco(),
90                     telefone: respModel.getTelefone(), email: respModel.getEmail(), idade: respModel.getIdade());
91         }
92     }
```

//**********

Classe Responsável View

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package View;

import Controller.Responsavel_Controller;
import Model.Crianca_Model;
import Model.Responsavel_Model;
import files.ArquivoCsv;
import java.awt.Color;
import java.awt.Font;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

/**
 * classe responsável
 *
 * @author kim
 */
public class Responsavel_view extends JFrame implements ActionListener {

    JPanel panelSuperior, panelInferior, panelOrdenacao;
    JLabel lb_pesquisar, all_InfullHouse, lb_cpf, lb_nome, lb_idade, lb_endereco, lb_fone, lb_email, lb_foto, lb_logo, lb_Icon, lb_unicumar;
    static JTextField tf_pesquisar, tf_cpf, tf_nome, tf_idade, tf_endereco, tf_fone, tf_email;
    JTable table;
    JButton bt_abreFoto, bt_gravarcsv, bt_avancar, bt_lerCsv;
    JComboBox<JTextField> cb_combo;
    ButtonGroup bg_Ordenacao;
    JRadioButton rb_codigo, rb_nome;
    ImageIcon img_semFoto, img_logo, img_avancar, img_arqcsv, img_LerCsv, img_Icon, img_unicesumar;

    static Responsavel_Model model;
    Crianca_Model crianca;
    static Estadia_View estadiaView;

    private String caminho = "./src/files/Responsavel.csv";
    private String endArq = "./src/files/Responsavel.txt";
    private int maioridade = 18;

    /**
     * Método construtor
     */
    public Responsavel_view() {
        this.crianca = new Crianca_Model();
        this.model = new Responsavel_Model();
        this.estadiaView = new Estadia_View();
        setTitle(title: "Cadastro de Responsáveis");
        setSize(width: 500, height:490);
        setLocation(x: 300, y: 180);
        setResizable(resizable:false);
        getContentPane().setLayout(mgr:null);
        setDefaultCloseOperation(operation:JFrame.DISPOSE_ON_CLOSE);

        all_InfullHouse = new JLabel(text: " All InfullHouse ");
        getContentPane().add(comp: all_InfullHouse);
        all_InfullHouse.setBounds(x: 10, y: 50, width: 300, height:25);
        all_InfullHouse.setFont(new Font(name: "Arial", style: Font.BOLD, size: 20));

        lb_pesquisar = new JLabel(text: " Pesquisar: ");
```

```
        lb_pesquisar.setBounds(x: 5, y: 5, width: 90, height:25);
        lb_pesquisar.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));
        lb_pesquisar.setForeground(fg: Color.white);

        tf_pesquisar = new JTextField(columns: 20);
        tf_pesquisar.setBounds(x: 85, y: 5, width: 200, height:20);

        cb_combo = new JComboBox<>();
        cb_combo.setBounds(x: 290, y: 5, width: 170, height:20);

        panelSuperior = new JPanel();
        getContentPane().add(comp: panelSuperior);
        panelSuperior.setBounds(x: 10, y: 10, width: 465, height:30);
        panelSuperior.setBackground(bg: Color.BLACK);
        panelSuperior.add(comp: lb_pesquisar);
        panelSuperior.add(comp: tf_pesquisar);
        panelSuperior.add(comp: cb_combo);
        panelSuperior.setLayout(mgr:null);

        img_logo     = new ImageIcon(filename: "./src/img/nomeUnicesumar.png");
        img_semFoto = new ImageIcon(filename: "./src/img/usuarios.png");
        img_avancar = new ImageIcon(filename: "./src/img/avancar.gif");
        img_arqcsv  = new ImageIcon(filename: "./src/img/gravar.gif");
        img_LerCsv  = new ImageIcon(filename: "./src/img/ler.gif");
        img_Icon     = new ImageIcon(filename: "./src/img/Icon.png");
        img_unicesumar = new ImageIcon(filename: "./src/img/nome.png");

        lb_unicumar = new JLabel(image: img_unicesumar);
        lb_unicumar.setFont(new Font(name: "Arial", style: Font.BOLD, size: 16));
        lb_unicumar.setForeground(fg: Color.WHITE);
        lb_unicumar.setBounds(x: 2, y: 0, width: 150, height:30);

        lb_logo = new JLabel(image: img_logo);
        getContentPane().add(comp: lb_logo);
        lb_logo.setBounds(x: 2, y: 0, width: 250, height:80);

        lb_Icon = new JLabel(image: img_Icon);
        getContentPane().add(comp: lb_Icon);
        lb_Icon.setBounds(x: 380, y: 40, width: 90, height:80);

        lb_foto = new JLabel(image: img_semFoto);
        getContentPane().add(comp: lb_foto);
        lb_foto.setBounds(x: 320, y: 80, width: 150, height:120);

        panelInferior = new JPanel();
        panelInferior.add(comp: lb_unicumar);
        getContentPane().add(comp: panelInferior);
        panelInferior.setBounds(x: 10, y: 410, width: 465, height:30);
        panelInferior.setBackground(bg: Color.BLACK);
        panelInferior.setLayout(mgr:null);

        lb_nome = new JLabel(text: "Nome:");
        getContentPane().add(comp: lb_nome);
        lb_nome.setBounds(x: 20, y: 100, width: 50, height:20);

        tf_nome = new JTextField();
        getContentPane().add(comp: tf_nome);
        tf_nome.setBounds(x: 60, y: 100, width: 270, height:20);
        tf_nome.requestFocus();

        lb_cpf = new JLabel(text: "CPF:");
        getContentPane().add(comp: lb_cpf);
        lb_cpf.setBounds(x: 20, y: 135, width: 50, height:20);
```

```
        tf_cpf = new JTextField();
        getContentPane().add(comp: tf_cpf);
        tf_cpf.setBounds(x: 60, y: 135, width: 270, height:20);

        lb_endereco = new JLabel(text: "Endereço:");
        getContentPane().add(comp: lb_endereco);
        lb_endereco.setBounds(x: 20, y: 170, width: 70, height:20);

        tf_endereco = new JTextField();
        getContentPane().add(comp: tf_endereco);
        tf_endereco.setBounds(x: 85, y: 170, width: 360, height:20);

        lb_fone = new JLabel(text: "Fone:");
        getContentPane().add(comp: lb_fone);
        lb_fone.setBounds(x: 20, y: 205, width: 40, height:20);

        tf_fone = new JTextField();
        getContentPane().add(comp: tf_fone);
        tf_fone.setBounds(x: 60, y: 205, width: 270, height:20);

        lb_email = new JLabel(text: "Email:");
        getContentPane().add(comp: lb_email);
        lb_email.setBounds(x: 20, y: 240, width: 40, height:20);

        tf_email = new JTextField();
        getContentPane().add(comp: tf_email);
        tf_email.setBounds(x: 60, y: 240, width: 270, height:20);

        lb_idade = new JLabel(text: "Idade:");
        getContentPane().add(comp: lb_idade);
        lb_idade.setBounds(x: 20, y: 275, width: 40, height:20);

        tf_idade = new JTextField();
        getContentPane().add(comp: tf_idade);
        tf_idade.setBounds(x: 60, y: 275, width: 180, height:20);

        bt_abreFoto = new JButton(text: "...");
        getContentPane().add(comp: bt_abreFoto);
        bt_abreFoto.setBounds(x: 335, y: 135, width: 30, height:20);
        bt_abreFoto.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));
        bt_abreFoto.setToolTipText(text: "Selecionar Imagem");

        bt_lerCsv = new JButton(icon: img_LerCsv);
        getContentPane().add(comp: bt_lerCsv);
        bt_lerCsv.setBounds(x: 320, y: 300, width: 40, height:30);
        bt_lerCsv.setToolTipText(text: "Leitura do arquivo CSV");
        bt_lerCsv.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));

        bt_gravarcsv = new JButton(icon: img_arqcsv);
        getContentPane().add(comp: bt_gravarcsv);
        bt_gravarcsv.setBounds(x: 370, y: 300, width: 40, height:30);
        bt_gravarcsv.setToolTipText(text: "Gravar em Arquivo CSV");
        bt_gravarcsv.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));

        bt_avancar = new JButton(icon: img_avancar);
        getContentPane().add(comp: bt_avancar);
        bt_avancar.setBounds(x: 420, y: 300, width: 40, height:30);
        bt_avancar.setToolTipText(text: "Avança para próxima tela...!");
        bt_avancar.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));

        bg_ordenacao = new ButtonGroup();
        rb_codigo = new JRadioButton(text: "Código");
        rb_nome = new JRadioButton(text: "  Nome");

        bg_ordenacao.add(b: rb_codigo);
        bg_ordenacao.add(b: rb_nome);

        panelOrdenacao = new JPanel();
        getContentPane().add(comp: panelOrdenacao);
        panelOrdenacao.setBounds(x: 375, y: 200, width: 90, height:65);
        panelOrdenacao.setBackground(bg: Color.darkGray);
        panelOrdenacao.add(comp: rb_codigo);
        panelOrdenacao.add(comp: rb_nome);

        setIcon();

        table = new JTable();
        table.setModel(new DefaultTableModel(
                new Object[][]{
                    {null, null, null, null},
                    {null, null, null, null},
                    {null, null, null, null},
                    {null, null, null, null}
                },
                new String[]{"Código", "Nome", "Idade", "CPF"}
        ));

        getContentPane().add(comp: table);
        table.setBounds(x: 20, y: 350, width: 450, height:50);

        bt_lerCsv.addActionListener(l: this);
        bt_gravarcsv.addActionListener(l: this);
        bt_avancar.addActionListener(l: this);
        bt_abreFoto.addActionListener(l: this);
    }
```

```java
    /**
     * Método principal Main
     *
     * @param args
     */
    public static void main(String[] args) {
        JFrame respView = new Responsavel_view();
        respView.setVisible(b: true);

        Responsavel_Model model = Prop_MVC();
        Responsavel_view view = new Responsavel_view();
        Responsavel_Controller controller = new Responsavel_Controller(model, view);

        controller.atualizaView();
    }

    /**
     * Método responsável por disparar ações nos components
     *
     * @param e
     */
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == bt_avancar) {
            //  ValidaCamposResponsavel();
            if (validarCampoIdade()) {
                exportarDados();
                gravarDadosResponsavelEmObjeto();
            }
        }
        if (e.getSource() == bt_gravarcsv) {

            GravarArquivoCsv();
        }
        if (e.getSource() == bt_lerCsv) {

            lerCsv();
        }
        if (e.getSource() == bt_abreFoto) {

        }
    }

    /**
     * Método quando chamado faz a impressão no console
     *
     * @param nome
     * @param cpf
     * @param endereco
     * @param telefone
     * @param email
     * @param idade
     */
    public void printDadosResponsavel(String nome, String cpf, String endereco, String telefone, String email, int idade) {
        System.out.println(" Responsavel: " + nome);
        System.out.println(" cpf: " + cpf);
        System.out.println(" endereco: " + endereco);
        System.out.println(" Telefone: " + telefone);
        System.out.println(" email: " + email);
        System.out.println(" idade: " + idade);
    }

    /**
     * método para pegar o que for digitado nos TextField e passado para o model
```

```
     *
     * @return model
     */
    public static Responsavel_Model Prop_MVC() {
        model.setNome(nome: tf_nome.getText());
        model.setCpf(cpf:tf_cpf.getText());
        model.setEndereco(endereco: tf_endereco.getText());
        model.setTelefone(telefone: tf_fone.getText());
        model.setEmail(email: tf_email.getName());
        model.setIdade(idade: Integer.parseInt(s: tf_idade.getText()));

        return model;
    }

    /**
     * método para exportar o nome do responsável para o formulário da criança
     */
    public void exportarDados() {
        Crianca_view crianca = new Crianca_view();
        model.setNome(nome:tf_nome.getText());
        validarCampoIdade();                    //método para validar a idade esse método só pode ser chamadoa após a chamada do método validarIdade
        crianca.receberDados(responsa: model);
        crianca.setVisible(b: true);
    }

    /**
     * Método para exportar os dados do responsável para o relatório
     *
     * @return
     */
    public void exportarDadosResponsavelRelatorio() {
        model.setNome(nome: tf_nome.getText());
        model.setCpf(cpf:model.getCpf());
        model.setEndereco(endereco: model.getEndereco());
        model.setTelefone(telefone: model.getTelefone());
        model.setEmail(email: model.getEmail());
        model.setIdade(idade: model.getIdade());
    }

    /**
     * Método para validar se o responsável é maior de idade, se for menor de
     * idade sai do sistema
     *
     * @return
     */
    public boolean validarCampoIdade() {
        model.setIdade(idade: Integer.parseInt(s: tf_idade.getText()));
        if ((model.getIdade() < maioridade) || (tf_nome.getText().equals(anObject: ""))) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Você é menor de idade\n Ou deve preencher o campo nome");
            // System.exit(1);
        } else {
            //   System.exit(0);
        }
        return true;
    }

    /**
     * Método para limpar os campos
     *
     * @return true
     */
    public boolean LimparCampos() {
        tf_nome.setText(t: "");
        tf_cpf.setText(t: "");
        tf_endereco.setText(t: "");
        tf_fone.setText(t: "");
        tf_email.setText(t: "");
        tf_idade.setText(t: "");
        return false;
    }

    /**
     * Método para validar se os campos foram preenchidos, se não foram sai do
     * sistema
     */
    public void ValidaCamposResponsavel() {

        if (tf_nome.getText().isEmpty()) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Preencha o campo nome:");
        } else if (tf_cpf.getText().isEmpty()) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Preencha o campo CPF:");
        } else if (tf_endereco.getText().isEmpty()) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Preencha o campo endereço:");
        } else if (tf_fone.getText().isEmpty()) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Preencha o campo Telefone:");
        } else if (tf_email.getText().isEmpty()) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Preencha o campo Email:");
        } else if (tf_idade.getText().isEmpty()) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Preencha o campo Idade:");
        }
    }

    //método para gravar os dados em arquivo csv
    public void GravarArquivoCsv() {
        ArquivoCsv.AdicionarResponsavel(objResponsavel: model);
    }
```

```
385
386     public void lerCsv() {
387         ArquivoCsv.leitorCsv(file: caminho);
388     }
389
390     /**
391      * método de gravação dos dados do responsável em objeto
392      */
393     public void gravarDadosResponsavelEmObjeto() {
394         Responsavel_Model objRespo = new Responsavel_Model();
395         objRespo.setNome(nome: tf_nome.getText());
396         objRespo.setCpf(cpf:tf_cpf.getText());
397         objRespo.setEndereco(endereco: tf_endereco.getText());
398         objRespo.setTelefone(telefone: tf_fone.getText());
399         objRespo.setEmail(email: tf_email.getText());
400         objRespo.setIdade(idade: Integer.parseInt(s: tf_idade.getText()));
401         try {
402             FileOutputStream arquivo = new FileOutputStream(name: "./src/files/objResponsavel.txt");
403             ObjectOutputStream obj_dados = new ObjectOutputStream(out:arquivo);
404             obj_dados.writeObject(obj:objRespo);
405             obj_dados.flush();
406             // JOptionPane.showMessageDialog(null,"Parabéns, arquivo de configurações gerados com sucesso");
407         } catch (IOException e) {
408             e.printStackTrace();
409         }
410     }
```

```
411
412     /**
413      * Método para trocar o icone padrão do java
414      */
415     private void setIcon() {
416         setIconImage(image: Toolkit.getDefaultToolkit().getImage(url:getClass().getResource(name: "icon.png")));
417     }
418 }
```

//**********

Classe Criança View

```
1   /*
2    * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3    * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4    */
5   package View;                                   // pacote
6
7   import Controller.Responsavel_Controller;
8   import Model.Crianca_Model;
9   import Model.Responsavel_Model;
10  import java.awt.Color;
11  import java.awt.Font;                           // Área da importação das bibliotecas
12  import java.awt.Toolkit;
13  import java.awt.event.ActionEvent;
14  import java.awt.event.ActionListener;
15  import java.io.*;
16  import javax.swing.*;
17  import javax.swing.table.DefaultTableModel;
18  /**
19   * classe criancaView
20   * @author kim
21   */
22  public class Crianca_view extends JFrame implements ActionListener{
23
24      JPanel panelSuperior, panelInferior;
25      JLabel lb_pesquisar, lb_nomeResponsavel, lb_nome, lb_idade, lb_sexo, lb_foto, lb_logo,lb_unicesumar, lb_icon;
26      JTextField tf_pesquisar, tf_nomeResponsavel, tf_nome, tf_idade, tf_sexo;
27      JTable table;
28      JButton bt_abreFoto, bt_gravartxt, bt_avancar, bt_lertxt;
29      JComboBox<JTextField> cb_combo;
30      ImageIcon  img_semFoto, img_avancar, img_arqtxt, img_Lertxt,img_unicesumar, img_icon;
31
32      Responsavel_Model model         = new Responsavel_Model();
```

```
    static Crianca_Model modCrianca = new Crianca_Model();
    Estadia_View estadia            = new Estadia_View();

    private int IdadeDeBrincar = 10;

    /**
     * Método Construtor
     */
    public Crianca_view(){
        setTitle(title: "Cadastro da Criança");        //Titulo do formulário
        setSize(width: 385, height:450);                        //Tamanho do formulário em pixels
        setLocation(x: 150, y: 180);                    //Posiciona o formulário na tela
        setResizable(resizable:false);                      //Se o formulário pode ser redimencionado ou não
        getContentPane().setLayout(mgr:null);           //Set o Layout como nulo
        setDefaultCloseOperation(operation:JFrame.DISPOSE_ON_CLOSE); //Método usado para nnão fechar toda a aplicação apenas esse formulário

         setIcon();

        img_semFoto    = new ImageIcon(filename: "./src/img/usuarios.png");
        img_avancar    = new ImageIcon(filename: "./src/img/avancar.gif");
        img_arqtxt     = new ImageIcon(filename: "./src/img/gravar.gif");
        img_Lertxt     = new ImageIcon(filename: "./src/img/ler.gif");
        img_unicesumar = new ImageIcon(filename: "./src/img/nome.png");
        img_icon       = new ImageIcon(filename: "./src/img/Icon.png");

        lb_pesquisar = new JLabel(text: " Pesquisar: ");
        lb_pesquisar.setBounds(x: 5, y: 5, width: 70, height:25);
        lb_pesquisar .setFont(new Font(name: "Arial", style: Font.BOLD,size: 12));
        lb_pesquisar .setForeground(fg: Color.WHITE);

        tf_pesquisar = new JTextField(columns: 25);
        tf_pesquisar.setBounds(x: 75, y: 5, width: 140, height:20);
```

```
        tf_pesquisar.setBounds(x: 75, y: 5, width: 140, height:20);

        cb_combo     = new JComboBox<>();
        cb_combo     .setBounds(x: 220, y: 5, width: 125, height:20);

        lb_unicesumar = new JLabel(image: img_unicesumar);
        lb_unicesumar .setFont(new Font(name: "Arial", style: Font.BOLD,size: 16));
        lb_unicesumar .setForeground(fg: Color.WHITE);
        lb_unicesumar .setBounds(x: 2, y: 2, width: 130, height:30);

        lb_nomeResponsavel = new JLabel(text: "Nome do Responsável:");
        getContentPane().add(comp: lb_nomeResponsavel);
        lb_nomeResponsavel.setBounds(x: 20, y: 50, width: 150, height:20);

        panelSuperior = new JPanel();
        getContentPane().add(comp: panelSuperior);
        panelSuperior.setBounds(x: 10, y: 10, width: 350, height:30);
        panelSuperior.setBackground(bg: Color.BLACK);
        panelSuperior.add(comp: lb_pesquisar);
        panelSuperior.add(comp: tf_pesquisar);
        panelSuperior.add(comp: cb_combo);
        panelSuperior.setLayout(mgr:null);

        panelInferior = new JPanel();
        panelInferior.add(comp: lb_unicesumar);
        getContentPane().add(comp: panelInferior);
        panelInferior .setBounds(x: 10, y: 375, width: 350, height:30);
        panelInferior .setBackground(bg: Color.BLACK);
        panelInferior.setLayout(mgr:null);

        tf_nomeResponsavel = new JTextField();
        getContentPane().add(comp: tf_nomeResponsavel);
```

```
        tf_nomeResponsavel.setBounds(x: 20, y: 70, width: 320, height:20);
        tf_nomeResponsavel.setBackground(bg: Color.lightGray);

        lb_nome = new JLabel(text: "Nome da Criança:");
        getContentPane().add(comp: lb_nome);
        lb_nome.setBounds(x: 20, y: 90, width: 150, height:20);

         tf_nome    = new JTextField();
        getContentPane().add(comp: tf_nome);
        tf_nome.setBounds(x: 20, y: 110, width: 320, height:20);

        lb_sexo = new JLabel(text: "Sexo da Criança:");
        getContentPane().add(comp: lb_sexo);
        lb_sexo.setBounds(x: 20, y: 130, width: 150, height:20);

        tf_sexo = new JTextField();
        getContentPane().add(comp: tf_sexo);
        tf_sexo.setBounds(x: 20, y: 150, width: 180, height:20);

        lb_idade= new JLabel(text: "Idade da Crinça:");
        getContentPane().add(comp: lb_idade);
        lb_idade.setBounds(x: 20, y: 170, width: 150, height:20);

        tf_idade = new JTextField();
        getContentPane().add(comp: tf_idade);
        tf_idade.setBounds(x: 20, y: 190, width: 180, height:20);

        lb_icon = new JLabel(image: img_icon);
        getContentPane().add(comp: lb_icon);
        lb_icon.setBounds(x: 1, y: 190, width: 90, height:130);

        bt_abreFoto = new JButton(text: "...");
        getContentPane().add(comp: bt_abreFoto);
        bt_abreFoto.setBounds(x: 210, y: 150, width: 40,height:20);
        bt_abreFoto.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));

        lb_foto = new JLabel(image: img_semFoto);
        getContentPane().add(comp: lb_foto);
        lb_foto.setBounds(x: 245, y: 120, width: 100, height:80);

        bt_lertxt = new JButton(icon: img_Lertxt);
        getContentPane().add(comp: bt_lertxt);
        bt_lertxt.setBounds(x: 210, y: 240, width: 40,height:30);
        bt_lertxt.setToolTipText(text: "Leitura do arquivo texto");
        bt_lertxt.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));

        bt_gravartxt = new JButton(icon: img_arqtxt);
        getContentPane().add(comp: bt_gravartxt);
        bt_gravartxt.setBounds(x: 255, y: 240, width: 40,height:30);
        bt_gravartxt.setToolTipText(text: "Gravar em Arquivo texto");
        bt_gravartxt.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));

        bt_avancar = new JButton(icon: img_avancar);
        getContentPane().add(comp: bt_avancar);
        bt_avancar.setBounds(x: 300, y: 240, width: 40,height:30);
        bt_avancar.setToolTipText(text: "Avança para próxima tela...!");
        bt_avancar.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));

        table = new JTable();
        table.setModel(new DefaultTableModel(
             new Object[][]{
                 {null, null, null, null},
                 {null, null, null, null},
                 {null, null, null, null},
                 {null, null, null, null}
             },
             new String[]{"Código","Nome", "Idade","CPF"}
        ));

        getContentPane().add(comp: table);
        table.setBounds(x: 20, y: 290, width: 330, height:50);

        bt_lertxt       .addActionListener(l: this);
        bt_gravartxt    .addActionListener(l: this);
        bt_avancar      .addActionListener(l: this);
        bt_abreFoto     .addActionListener(l: this);

    }
    /**
     * método principal Main
     * @param args
     */
    public static void main(String[] args) {
        JFrame crianca = new Crianca_view();
        crianca.setVisible(b: true);

        Responsavel_Model model = ProprerResponsavel_MVC();
        Responsavel_view view      = new Responsavel_view();
        Responsavel_Controller controller = new Responsavel_Controller(model, view);
```

```
186             controller.atualizaView();
187
188         }
189         /**
190          * método responsável por disparar ações nos components
191          * @param e
192          */
193           @Override
194         public void actionPerformed(ActionEvent e) {
195             if(e.getSource()==bt_avancar){
196                   exportarDados();
197                   ValidaIdadeDeBrincar();
198                   exportaDadosCrianca();
199
200             }
201             if(e.getSource()==bt_gravartxt){
202                gravarDadoDaCriancaEmObjeto();
203             }
204             if(e.getSource()== bt_lertxt){
205
206             }
207         }
208         /**
209          *
210          * @param respName
211          */
212         public void printResponsavelDetalhes(String respName ){}
213         /**
214          *
215          * @return objeto responsávelModel
216          */
217         public static Responsavel_Model ProprerResponsavel_MVC(){
```

```
218             Responsavel_Model resp = new Responsavel_Model();
219             resp.setNome(nome: resp.getNome());
220             resp.setCpf(cpf:resp.getCpf());
221             resp.setEndereco(endereco: resp.getEndereco());
222             resp.setTelefone(telefone: resp.getTelefone());
223             resp.setEmail(email: resp.getEmail());
224             resp.setIdade(idade: resp.getIdade());
225             return resp;
226         }
227         /**
228          *
229          * @return objeto criancaModel
230          */
231         public static Crianca_Model properCrianca(){
232             modCrianca.setNome(nome: modCrianca.getNome());
233             modCrianca.setSexo(sexo: modCrianca.getSexo());
234             modCrianca.setIdade(idade: modCrianca.getIdade());
235
236             return modCrianca;
237         }
238         /**
239          * método para receber o mome do responsável por parâmetro
240          * @param responsa
241          */
242         public void receberDados(Responsavel_Model responsa){
243               tf_nomeResponsavel.setText(t: responsa.getNome());
244         }
245          /**
246           * método que exporta o nome do responsável para o Formulário de Estadia
247           */
248         public void exportarDados(){
249           model.setNome(nome: tf_nomeResponsavel.getText());
```

```
250             estadia.receberResponsavel(responsa: model);
251             estadia.setVisible(b: true);
252             gravarDadoDaCriancaEmObjeto();              //chama o método que grava os dados em objetos
253         }
254         /**
255          * método para exportar o nome da criança para o formulário de estadia
256          */
257         public void exportaDadosCrianca(){
258             modCrianca.setNome(nome: tf_nome.getText());
259             estadia.recebeCrianca(crianca: modCrianca);
260            // estadia.setVisible(true);
261         }
262         /**
263          * método para validar a idade da criança
264          * @return
265          */
266        public boolean ValidaIdadeDeBrincar(){
267          modCrianca.setIdade(idade: Integer.parseInt(s: tf_idade.getText()));
268          if((modCrianca.getIdade() <= IdadeDeBrincar)){
269           //   JOptionPane.showMessageDialog(null, "Você pode brincar...");
270          }else if((modCrianca.getIdade() > IdadeDeBrincar)){
271              JOptionPane.showMessageDialog(parentComponent: null, message: "Você não tem mais idade para brincar com esse tipos de brinquedos... ");
272              System.exit(status:0);
273          }
274            return true;
275     }
276     /**
277      * método genérico para limpar os campos
278      * @return
279      */
280     public boolean camposVazios(){
281         tf_nomeResponsavel.setText(t: "");
```

```
282         tf_nome.setText(t: "");
283         tf_sexo.setText(t: "");
284         tf_idade.setText(t: "");
285         return true;
286      }
287       /**
288        * método para gravar os dados da criança em arquivo de texto como objeto
289        */
290       public void gravarDadoDaCriancaEmObjeto(){
291             Crianca_Model objCria = new Crianca_Model();
292             objCria.setNome(nome: tf_nome.getText());
293             objCria.setSexo(sexo: tf_sexo.getText());
294             objCria.setIdade(idade: Integer.parseInt(s: tf_idade.getText()));
295             try
296             {
297                FileOutputStream arquivo = new FileOutputStream (name: "./src/files/objCrianca.txt");
298                ObjectOutputStream  obj_dados  = new ObjectOutputStream(out:arquivo);
299                obj_dados.writeObject(obj:objCria);
300                obj_dados.flush();
301              //  JOptionPane.showMessageDialog(null,"Parabéns, arquivo de texto gerados com sucesso");
302             }catch(IOException e){
                    e.printStackTrace();
304             }
305       }
306       /**
307        * método para trocar o ícone padrão do java
308        */
309       private void setIcon() {
310           setIconImage(image: Toolkit.getDefaultToolkit().getImage(url:getClass().getResource(name: "icon.png")));
311       }
312   }
```

//**

Classe Estadia View

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package View;                                   // pacote

import Model.Crianca_Model;
import Model.Estadia_Model;
import Model.Responsavel_Model;
import java.awt.Color;
import java.awt.Font;                           // Área da importação das bibliotecas
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.FileInputStream;
import java.io.ObjectInputStream;
import java.text.DecimalFormat;
import javax.swing.*;
/**
 *
 * @author kim
 */
public class Estadia_View extends JFrame implements ActionListener{              //classe Eatadia estendendo do JFrmame e implementando a interfa

    JPanel panelSuperior, panelInferior;
    JLabel lb_pesquisar, lb_nomeResponsavel, lb_nome, lb_tempo,lb_vlr, lb_valorTotal, lb_logo, lb_Unicesumar;  // Área de Criação dos Conteners
    static JTextField tf_pesquisar, tf_nomeResp, tf_nome, tf_tempo;
    JTable table;
    JButton bt_gravarcsv, bt_calcular;
    JComboBox<JTextField> cb_combo;
    JTextArea areaResumo;
    ImageIcon  img_logo,  img_Unicesumar, img_LerCsv;


    Crianca_Model crianca       = new Crianca_Model();
    static  Responsavel_Model model     = new Responsavel_Model();
    static Estadia_Model  estadiaModel = new Estadia_Model();

    static DecimalFormat objNumberFormat = new DecimalFormat();

    /**
     * Método Construtor
     */
    public Estadia_View(){

        setTitle(title: "Cadastro de Estadia");        //Titulo do formulário
        setSize(width: 380, height:530);                        //Tamanho do formulário em pixels
        setLocation(x: 150, y: 180);                        //Posiciona o formulário na tela
        setResizable(resizable:false);                          //Se o formulário pode ser redimencionado ou não
        getContentPane().setLayout(mgr:null);           //Set o Layout como nulo
        setDefaultCloseOperation(operation:JFrame.DISPOSE_ON_CLOSE); //Método usado para nnão fechar toda a aplicação apenas esse formulário

        img_logo            = new ImageIcon(filename: "./src/img/Icon.png");
        img_Unicesumar      = new ImageIcon(filename: "./src/img/nome.png");
        img_LerCsv          = new ImageIcon(filename: "./src/img/ler.gif");

        lb_pesquisar        = new JLabel(text: " Pesquisar: ");     //Inicialização dos componentes da panel superior
        lb_pesquisar.setBounds(x: 5, y: 2, width: 70, height:25);
        lb_pesquisar.setForeground(fg: Color.WHITE);

        tf_pesquisar        = new JTextField(columns: 20);
        tf_pesquisar.setBounds(x: 75, y: 5, width: 140, height:20);
        cb_combo  = new JComboBox<>();
        cb_combo     .setBounds(x: 220, y: 5, width: 125, height:20);

        lb_logo             = new JLabel(image: img_logo);
        getContentPane().add(comp: lb_logo);
        lb_logo        .setFont(new Font(name: "Arial", style: Font.BOLD,size: 16));
        lb_logo        .setForeground(fg: Color.BLACK);
        lb_logo        .setBounds(x: 0, y: 175, width: 100, height:35);

        lb_Unicesumar = new JLabel(image: img_Unicesumar);
        lb_Unicesumar .setFont(new Font(name: "Arial", style: Font.BOLD,size: 16));
        lb_Unicesumar .setForeground(fg: Color.WHITE);
        lb_Unicesumar .setBounds(x: 8, y: 2, width: 120, height:25);

        panelSuperior = new JPanel();                   //inicialização da panel superior
        getContentPane().add(comp:panelSuperior);           //adiciona a panel ao contener
        panelSuperior.setBounds(x: 10, y: 10, width: 350, height:30);     //define o posicionamento da panel na tela
        panelSuperior.setBackground(bg: Color.BLACK);       //atribui a cor azeul ao background da panel
        panelSuperior.add(comp: lb_pesquisar);              //adidiona os componentes na panel superior
        panelSuperior.add(comp: tf_pesquisar);
        panelSuperior.add(comp: cb_combo);
        panelSuperior.setLayout(mgr:null);

        panelInferior = new JPanel();                   //inicializa a panel inferior
        getContentPane().add(comp: panelInferior);              //adiciona a panel inferior ao content
        panelInferior.add(comp: lb_Unicesumar);             //Adiciona uma imagem na panel
        panelInferior.setBounds(x: 10, y: 450, width: 350, height:30);
        panelInferior.setBackground(bg: Color.BLACK);
        panelInferior.setLayout(mgr:null);

        lb_nomeResponsavel = new JLabel(text: "Nome do Responsável:");
        getContentPane().add(comp: lb_nomeResponsavel);
        lb_nomeResponsavel.setBounds(x: 20, y: 50, width: 150, height:20);

        tf_nomeResp = new JTextField();
```

```
97          getContentPane().add(comp: tf_nomeResp);
98          tf_nomeResp.setBounds(x: 20, y: 70, width: 320, height:20);
99          tf_nomeResp.setBackground(bg: Color.lightGray);
100
101         lb_nome = new JLabel(text: "Nome da Criança:");
102         getContentPane().add(comp: lb_nome);
103         lb_nome.setBounds(x: 20, y: 90, width: 150, height:20);
104
105          tf_nome    = new JTextField();
106         getContentPane().add(comp: tf_nome);
107         tf_nome.setBounds(x: 20, y: 110, width: 320, height:20);
108         tf_nome.setBackground(bg: Color.lightGray);
109
110         lb_tempo = new JLabel(text: "Tempo Total Utilizado:");
111         getContentPane().add(comp: lb_tempo);
112         lb_tempo.setBounds(x: 20, y: 130, width: 150, height:20);
113
114         tf_tempo = new JTextField();
115         getContentPane().add(comp: tf_tempo);
116         tf_tempo.setBounds(x: 20, y: 150, width: 140, height:20);
117
118         lb_vlr = new JLabel(text: "Valor Total:");          // Indica que o valor total será mostrado
119         getContentPane().add(comp: lb_vlr);
120         lb_vlr.setBounds(x: 170, y: 150, width: 150, height:20);
121
122         lb_valorTotal = new JLabel();            //Recebe o resultado do calculo da estadia
123         getContentPane().add(comp: lb_valorTotal);
124         lb_valorTotal.setBounds(x: 250, y: 150, width: 100, height:20);
125
126         bt_calcular = new JButton(text: "Calcular");  //botão calcuar estadia
127         getContentPane().add(comp: bt_calcular);
128         bt_calcular.setBounds(x: 190, y: 180, width: 160,height:30);
```

```
129         bt_calcular.setToolTipText(text: "Calcular");
130         bt_calcular.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));
131
132         bt_gravarcsv = new JButton(icon: img_LerCsv);        //botão para gravar no arquivo csv
133         getContentPane().add(comp: bt_gravarcsv);
134         bt_gravarcsv.setBounds(x: 105, y: 180, width: 80,height:30);
135         bt_gravarcsv.setToolTipText(text: "Gravar em Arquivo CSV");
136         bt_gravarcsv.setFont(new Font(name: "Arial", style: Font.BOLD, size: 14));
137
138         areaResumo = new JTextArea();                //mostra o resultado da aplicação
139         getContentPane().add(comp: areaResumo);
140         areaResumo       .setBounds(x: 20, y: 220, width: 330, height:220);
            bt_calcular      .addActionListener(l: this);
            bt_gravarcsv     .addActionListener(l: this);
143
144         setIcon();
145
146
147     }
148     /**
149      * método principal Main
150      * @param args
151      */
152     public static void main(String[] args) {
153         JFrame estadia = new Estadia_View();
154         estadia.setVisible(b: true);
155
156     }
157     /*
158     public static void main2(){
159         Responsavel Model model = ProperResponsavel MVC();
160         Responsavel view view = new Responsavel view();
```

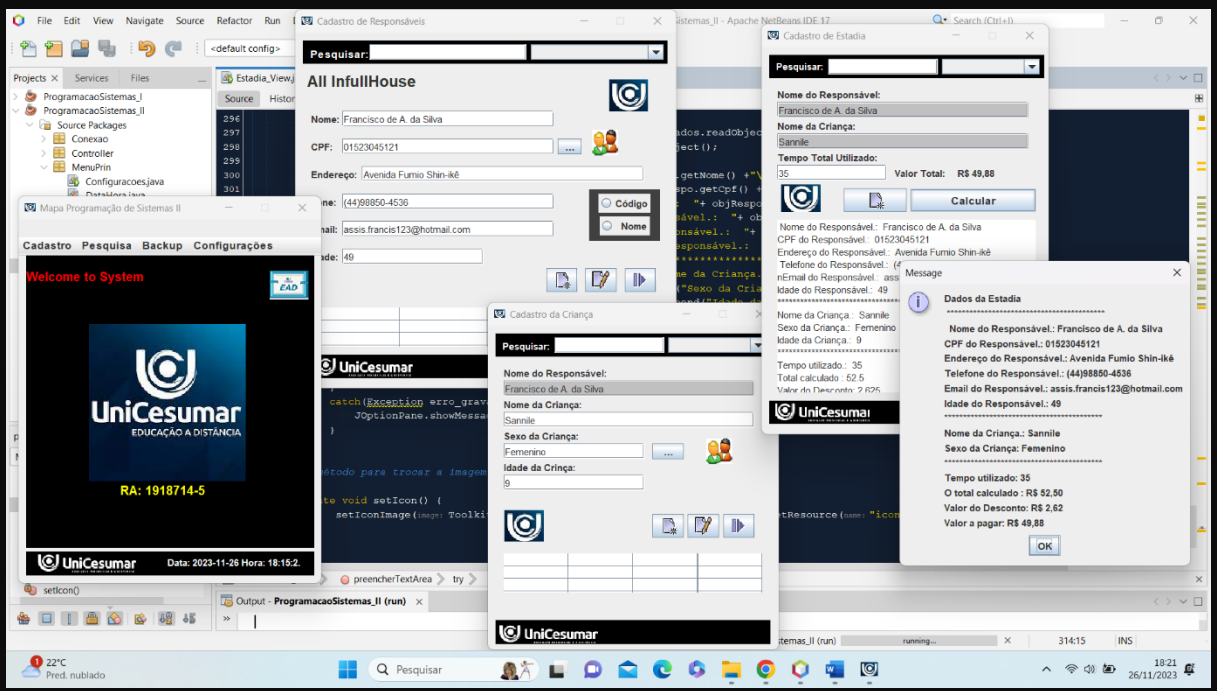
```
        Responsavel_Controller controller = new Responsavel_Controller(model, view);

        controller.atualizaView();

        Crianca_Model crianca = properCriMVC();
        Crianca_view vw = new Crianca_view();
        Crianca_Controller control = new Crianca_Controller(crianca, vw);

        control.atualizaView();
    }
    */
    public void printDadosCalculos(double value){}

    /**
     * método responsável por disparar as ações nos componentes
     * @param e
     */
      @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==bt_gravarcsv){
           areaResumo.append(str:" ");
           preencherTextArea();
        }
        if(e.getSource()==bt_calcular){
            transferirDadosCalculos();
            RelatorioDeEstadia(crianca);
        }
    }
    /**
     * método que faz a transição entre os métodos getteres and setteres e passado para o Responsavelmodel
     * @return
     */
```

```
    public static Responsavel_Model ProperResponsavel_MVC(){
        Responsavel_Model resp = new Responsavel_Model();
        resp.setNome(nome: resp.getNome());

        return resp;
    }
    /**
     * método que faz a transição entre os métodos getteres and setteres e passado para criancaModel
     * @return um objeto tipo criança model
     */
      public static Crianca_Model properCriMVC(){
        Crianca_Model cri = new Crianca_Model();
        cri.setNome(nome: cri.getNome());
        cri.setSexo(sexo: cri.getSexo());
        cri.setIdade(idade: cri.getIdade());

        return cri;
    }
    /**
     * método para receber o nome do responsável passado por parâmetro
     * @param responsa
     */
    public void receberResponsavel(Responsavel_Model responsa){    //método para receber o responsável
         tf_nomeResp.setText(t: responsa.getNome());
    }
    /**
     * método para receber o nome da criança passado por parâmetro
     * @param crianca
     */
    public void recebeCrianca(Crianca_Model crianca){
        tf_nome.setText(t: crianca.getNome());
    }
```

```
225     /**
226      * método que recebe o tempo utilizado para fazer o calculo da estadia
227      */
228     public void transferirDadosCalculos(){
229         estadiaModel.setTempoNoBrinquedo(tempoNoBrinquedo:Double.parseDouble(s: tf_tempo.getText()));
230         this.receberDadosEstadia(estModel: estadiaModel);
231     }
232     /**
233      * método para calcular o tempo utilizado na estadia
234      * @param estModel
235      */
236     public void receberDadosEstadia(Estadia_Model estModel){
237         double resultado = estModel.calcularEstadia(valuel:estadiaModel.getTempoNoBrinquedo());
238         objNumberFormat.applyPattern(pattern: "R$ #,##0.00");
239         lb_valorTotal.setText(text: objNumberFormat.format(number:resultado));
240     }
241     /**
242      * método que exibe o relatório da estadia em um componente visual JOptionPane
243      * @param crianca
244      */
245     public static void RelatorioDeEstadia(Crianca_Model crianca){
246          objNumberFormat.applyPattern(pattern: "R$ #,##0.00");
247         double value=Double.parseDouble(s: String.valueOf(d: estadiaModel.calcularEstadia(valuel:estadiaModel.getTempoNoBrinquedo())));
248         double vlrDesconto = Double.parseDouble(s: String.valueOf(d: estadiaModel.calcularDesconto(valuel:estadiaModel.getTempoNoBrinquedo())));
249         double total = value + vlrDesconto;
250             try{
251             FileInputStream arquivo = new FileInputStream (name: "./src/files/objResponsavel.txt");
252             ObjectInputStream  obj_dados  = new ObjectInputStream(in: arquivo);
253 
254             FileInputStream arqCrianca = new FileInputStream (name: "./src/files/objCrianca.txt");
255             ObjectInputStream  obj  = new ObjectInputStream(in: arqCrianca);
256 
257             Responsavel_Model objRespo = (Responsavel_Model)obj_dados.readObject();
258             Crianca_Model objCrianca = (Crianca_Model) obj.readObject();
259 
260             JOptionPane.showMessageDialog(parentComponent: null,"\tDados da Estadia" +
261                                           "\n ***************************************"+
262                                           "\n  Nome do Respons\u00e1vel.: "+ objRespo.getNome()+
263                                           "\nCPF do Responsável.: "+ objRespo.getCpf()+
264                                           "\nEndereço do Responsável.: "+ objRespo.getEndereco()+
265                                           "\nTelefone do Responsável.: "+ objRespo.getTelefone()+
266                                           "\nEmail do Responsável.: "+ objRespo.getEmail()+
267                                           "\nIdade do Responsável.: "+ objRespo.getIdade()+
268                                           "\n***************************************"+
269                                           "\nNome da Criança.: "+ objCrianca.getNome()+
270                                           "\nSexo da Criança: "+ objCrianca.getSexo()+
271                                           "\n***************************************"+
272                                           "\nTempo utilizado: "+ tf_tempo.getText()+
273                                           "\nO total calculado : "+ objNumberFormat.format(number:total)+
274                                           "\nValor do Desconto: " + objNumberFormat.format(number:vlrDesconto) +
275                                           "\nValor a pagar: "+ objNumberFormat.format(number:value));
276 
277 
278             }
279             catch(Exception erro_grava){
280                 JOptionPane.showMessageDialog(parentComponent: null,message: "Erro ao abrir o arquivo:");
281             }
282     }
283     /**
284      *  método para preencher a área de texto com os dados da estadia
285      */
286     public void preencherTextArea(){
287             double value = Double.parseDouble(s: String.valueOf(d: estadiaModel.calcularEstadia(valuel:estadiaModel.getTempoNoBrinquedo())));
288             double vlrDesconto = Double.parseDouble(s: String.valueOf(d: estadiaModel.calcularDesconto(valuel:estadiaModel.getTempoNoBrinquedo())));
289         double total = value + vlrDesconto;
290             try{
291             FileInputStream arquivo = new FileInputStream (name: "./src/files/objResponsavel.txt");
292             ObjectInputStream  obj_dados  = new ObjectInputStream(in: arquivo);
293 
294             FileInputStream arqCrianca = new FileInputStream (name: "./src/files/objCrianca.txt");
295             ObjectInputStream  obj  = new ObjectInputStream(in: arqCrianca);
296 
297             Responsavel_Model objRespo = (Responsavel_Model)obj_dados.readObject();
298             Crianca_Model objCrianca = (Crianca_Model) obj.readObject();
299 
300             areaResumo.append("Nome do Responsável.:  "+ objRespo.getNome() +"\n");
301                 areaResumo.append("CPF do Responsável.:  "+ objRespo.getCpf() +"\n");
302                     areaResumo.append("Endereço do Responsável.:  "+ objRespo.getEndereco()+"\n");
303                         areaResumo.append(" Telefone do Responsável.:  "+ objRespo.getTelefone()+"\n");
304                             areaResumo.append("nEmail do Responsável.:  "+ objRespo.getEmail()+"\n");
305                                 areaResumo.append("Idade do Responsável.:  "+ objRespo.getIdade()+"\n");
306                                     areaResumo.append(str:"***************************************\n");
307                                         areaResumo.append("Nome da Criança.:  "+ objCrianca.getNome()+"\n");
308                                             areaResumo.append("Sexo da Criança.:  "+ objCrianca.getSexo()+"\n");
309                                                 areaResumo.append("Idade da Criança.:  "+ objCrianca.getIdade()+"\n");
310                                                     areaResumo.append(str:"***************************************\n");
311                                                         areaResumo.append("Tempo utilizado.:  " +tf_tempo.getText()+"\n");
312                                                             areaResumo.append("Total calculado : "+ total + "\n");
313                                                                 areaResumo.append("Valor do Desconto: " + vlrDesconto+"\n");
314                                                                     areaResumo.append("Valor a pagar: "+ value);
315             }
316             catch(Exception erro_grava){
317                 JOptionPane.showMessageDialog(parentComponent: null,message: "Erro ao abrir o arquivo:");
318             }
319     }
320     /**
```

```
321         *  método para trocar a imagem padrão do java
322         */
323        private void setIcon() {
324                setIconImage(image: Toolkit.getDefaultToolkit().getImage(url:getClass().getResource(name: "icon.png")));
325        }
326    }
327
```

Veja o resultado da saída desse Código na imagem abaixo.



Assim chegamos ao final de mais uma atividade MAPA.