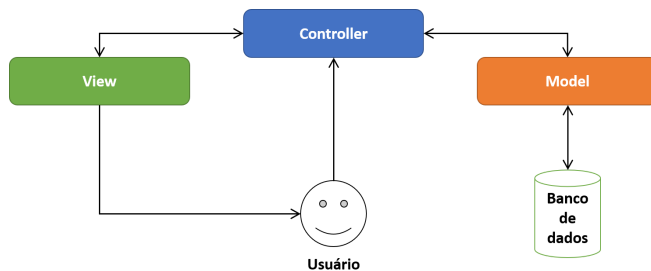


O `model` é a camada que possui a lógica da aplicação. Ele é o responsável pelas regras de negócios, persistência com o banco de dados e as classes de entidades. O `model` recebe as requisições vindas do `controller` e gera respostas a partir destas requisições.

Sempre que pensar em `model`, pense que ele terá conhecimento apenas dos dados armazenados no sistema e da lógica sobre eles. Estes dados podem estar armazenados em um banco de dados ou até mesmo em arquivos do tipo XML, TXT ou de qualquer outro tipo. É no `model` também que as operações de CRUD devem ser realizadas. Esta camada na verdade é responsável pelo motivo que a aplicação foi construída, ou seja, ela é o núcleo da aplicação.



CONTROLLER

Já sabemos que as requisições são enviadas pela `view` e a lógica de negócios é representada pelo `model`. Para que haja a comunicação entre essas duas camadas de maneira organizada, é necessário construir a camada `controller`.

Sua função é ser uma camada intermediária entre a camada de apresentação (`View`) e a camada de negócios (`Model`).

Deste modo, toda requisição criada pelo usuário deve passar pelo `controller`, e este então se comunica com o `model`. Se o `model` gerar uma resposta para essas requisições, ele enviará as respostas ao `controller` que por sua vez repassa à camada `view`.

O controlador serve como um intermediário que organiza os eventos da interface com usuário e os direciona para a camada de modelo, assim, torna-se possível um reaproveitamento da camada de modelo em outros sistemas já que não existe dependência entre a visualização e o modelo.

MODEL

```
package ejm.appdesktop.model.Service;

import ejm.appdesktop.model.dao.CalculoDao;
import ejm.appdesktop.model.entity.Calculo;

public class CalculoService{

    private CalculoDao dao;
```

```
public CalculoService() {  
    this.dao = new CalculoDao();  
}
```

```
public Calculo jurosSimples(Calculo calculo) {  
    double principal = calculo.getValorPrincipal();  
    double taxa = calculo.getTaxa() / 100;  
    int meses = calculo.getMeses();  
    double juros = principal * taxa * meses;  
    double montante = principal * (1 + (taxa * meses));  
    calculo.setTotalJuros(juros);  
    calculo.setMontante(montante);  
    salvar(calculo);  
    return calculo;  
}  
  
public Calculo jurosCompostos(Calculo calculo) {  
    double principal = calculo.getValorPrincipal();  
    double taxa = calculo.getTaxa() / 100;  
    int meses = calculo.getMeses();  
    double montante = principal * Math.pow((1 + taxa), meses);  
  
    double juros = montante - principal;  
    calculo.setTotalJuros(juros);  
    calculo.setMontante(montante);  
    salvar(calculo);  
    return calculo;  
}  
  
public void salvar(Calculo calculo) {  
    dao.salvar(calculo);  
}  
}
```

CONTROLLER

```
package ejm.appdesktop.controller;

public interface IController {

    public void executa(Object view);

}
```

The screenshot shows a web browser window with the URL `devmedia.com.br/padrao-mvc-java-magazine/21995#3`. The page is from DevMedia, a website for developers, and displays a Java code example for a controller. The code is as follows:

```
1 package ejm.appdesktop.controller;
2
3 import ejm.appdesktop.model.Service.CalculoService;
4 import ejm.appdesktop.model.entity.Calculo;
5 import ejm.appdesktop.view.CalculoForm;
6
7 import java.text.NumberFormat;
8 import java.util.Locale;
9
10 public class CalculoController implements IController {
11
12     private Calculo calculo;
13     private CalculoForm frame;
14
15     public void executa(Object view) {
16         frame = (CalculoForm) view;
17         calculo = new Calculo();
18         calculo.setValorPrincipal(
19             Double.parseDouble(frame.getTxtValorPrincipal().getText())
20         );
21         calculo.setTaxa(Double.parseDouble(frame.getTxtTaxa().getText()));
22     }
23 }
```

The browser's taskbar at the bottom shows the Windows logo, a search bar with the text "Digite aqui para pesquisar", and several application icons including Chrome, Firefox, and VS Code. The system tray on the right indicates a temperature of 21°C, a cloudy sky, and the date/time 21:28 on 07/12/2021.

Atividades de x VitorPiresKat x Trabalho capi x (8) OPMo x (2) WhatsApp x Padrão MVC x MoCo - Docu x como tirar pr x +

devmedia.com.br/padrao-mvc-java-magazine/21995#3

DEV MEDIA Tecnologias Navegue CRIE SUA CONTA

```
21 calculo.setTaxa(Double.parseDouble(frame.getTxtTaxa().getText()));
22 calculo.setMeses(Integer.parseInt(frame.getTxtMeses().getText()));
23 calculo.setTipo(frame.getRdBtnSimples().isSelected() ?
24     Calculo.TipoJuros.SIMPLES : Calculo.TipoJuros.COMPOSTOS
25 );
26
27 calcular(calculo);
28
29 String montante = doubleFormat(calculo.getMontante());
30 String juros = doubleFormat(calculo.getTotalJuros());
31
32 frame.getTxtTotJuros().setText(juros);
33 frame.getTxtMontante().setText(montante);
34 }
35
36 private Calculo calcular(Calculo calculo) {
37     CalculoService service = new CalculoService();
38
39     Calculo resultado;
40     if (calculo.getTipo().getValor() == 1) {
41         resultado = service.jurosSimples(calculo);
42     } else {
```

imagem recupera...html Exibir todos

21°C Nublado 21:29 07/12/2021

Atividades de x VitorPiresKat x Trabalho capi x (8) OPMo x (2) WhatsApp x Padrão MVC x MoCo - Docu x como tirar pr x +

devmedia.com.br/padrao-mvc-java-magazine/21995#3

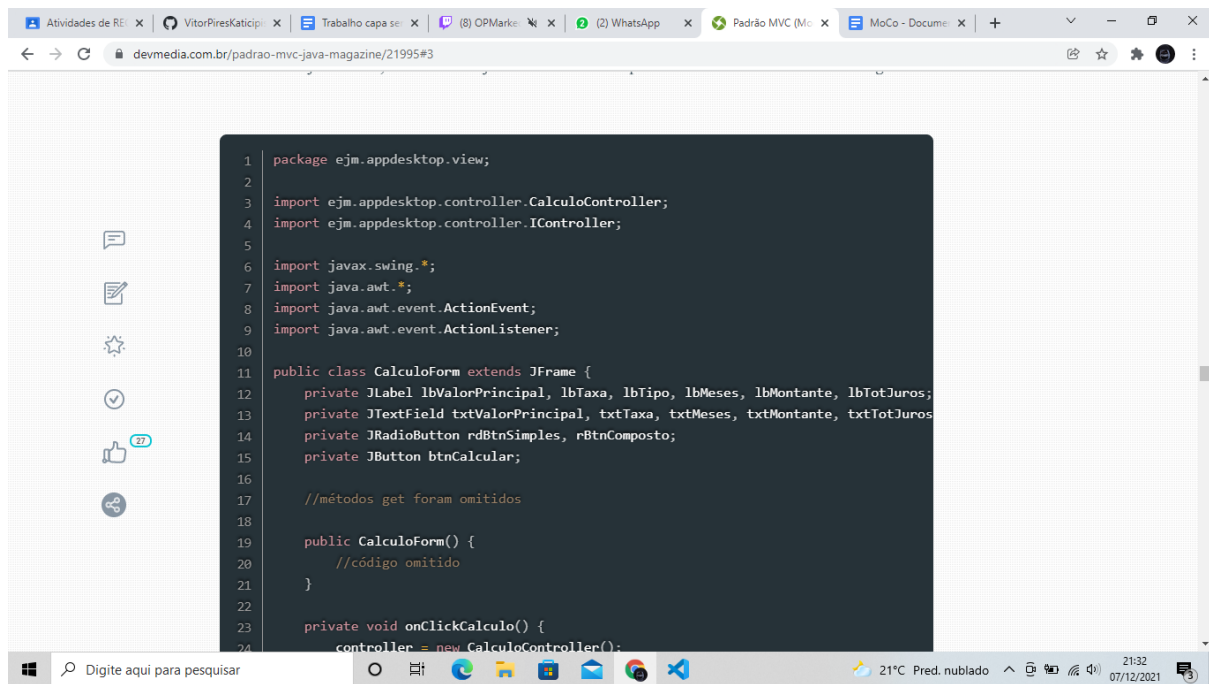
```
35
36 private Calculo calcular(Calculo calculo) {
37     CalculoService service = new CalculoService();
38
39     Calculo resultado;
40     if (calculo.getTipo().getValor() == 1) {
41         resultado = service.jurosSimples(calculo);
42     } else {
43         resultado = service.jurosCompostos(calculo);
44     }
45     return resultado;
46 }
47
48 private String doubleFormat(Double aDouble) {
49     NumberFormat nf =
50         NumberFormat.getCurrencyInstance(new Locale("pt", "BR"));
51
52     nf.setMaximumFractionDigits(2);
53     return nf.format(aDouble);
54 }
55 }
```

Listagem 6 Classe do pacote controller: CalculoController

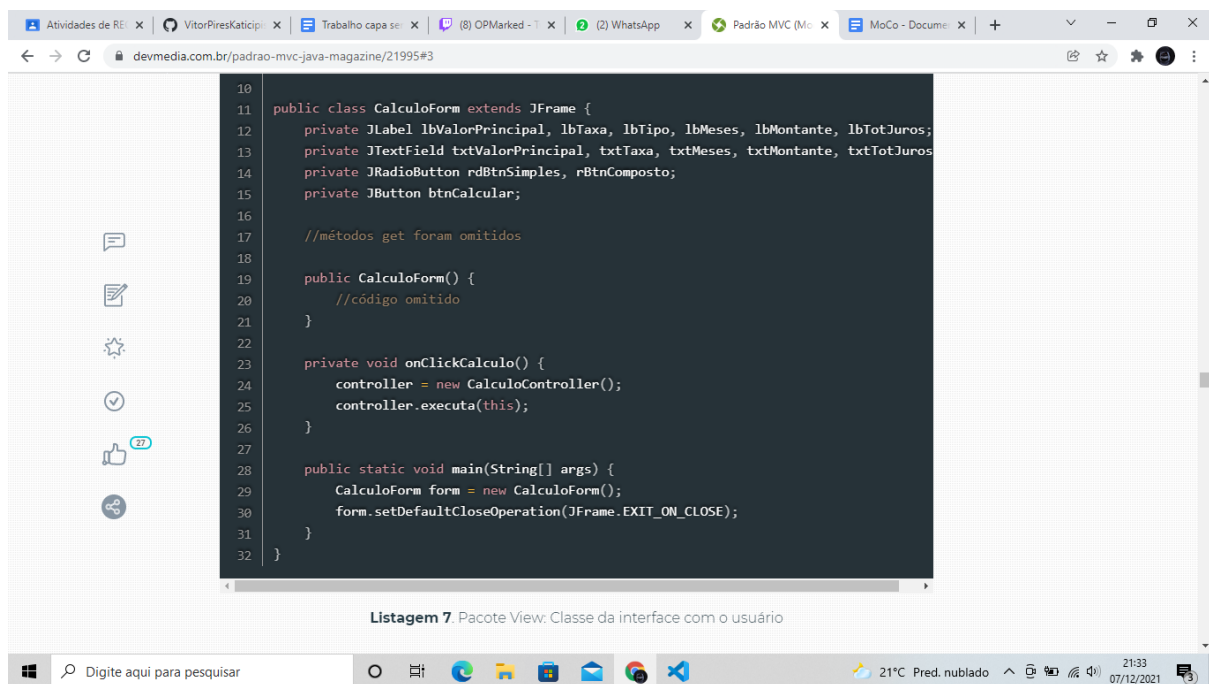
imagem recupera...html Exibir todos

21°C Nublado 21:30 07/12/2021

VIEW



```
1 package ejm.appdesktop.view;
2
3 import ejm.appdesktop.controller.CalculoController;
4 import ejm.appdesktop.controller.IController;
5
6 import javax.swing.*;
7 import java.awt.*;
8 import java.awt.event.ActionEvent;
9 import java.awt.event.ActionListener;
10
11 public class CalculoForm extends JFrame {
12     private JLabel lbValorPrincipal, lbTaxa, lbTipo, lbMeses, lbMontante, lbTotJuros;
13     private JTextField txtValorPrincipal, txtTaxa, txtMeses, txtMontante, txtTotJuros;
14     private JRadioButton rdBtnSimples, rBtnComposto;
15     private JButton btnCalcular;
16
17     //métodos get foram omitidos
18
19     public CalculoForm() {
20         //código omitido
21     }
22
23     private void onClickCalculo() {
24         controller = new CalculoController();
```



```
10
11 public class CalculoForm extends JFrame {
12     private JLabel lbValorPrincipal, lbTaxa, lbTipo, lbMeses, lbMontante, lbTotJuros;
13     private JTextField txtValorPrincipal, txtTaxa, txtMeses, txtMontante, txtTotJuros;
14     private JRadioButton rdBtnSimples, rBtnComposto;
15     private JButton btnCalcular;
16
17     //métodos get foram omitidos
18
19     public CalculoForm() {
20         //código omitido
21     }
22
23     private void onClickCalculo() {
24         controller = new CalculoController();
25         controller.executa(this);
26     }
27
28     public static void main(String[] args) {
29         CalculoForm form = new CalculoForm();
30         form.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
31     }
32 }
```

Listagem 7. Pacote View: Classe da interface com o usuário