

ALPOO


Teoria - Java

MVC

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

1



Disciplina: ALPOO


Tema: Arquitetura Model-View-Controller

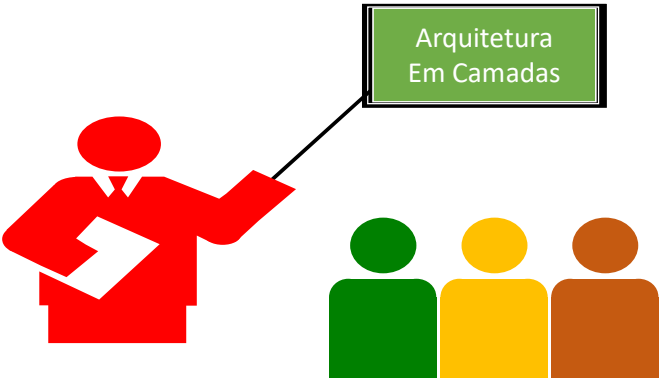
Professor: Ricardo Drudi

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

2






Arquitetura Em Camadas

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

3



Introdução

- Em aplicações OO de médio e grande porte, diversos aspectos devem ser considerados:
 - Apresentação
 - Lógica da aplicação
 - Lógica do negócio
 - Persistência de Objetos
 - Camada de Utilitários:
 - Controle de Exceções, Logging, comunicação, etc.
- **Persistência** : capacidade de uma aplicação manter suas informações entre sessões de uso.
 - Uma proporção significativa do esforço de desenvolvimento recai sobre a solução que o desenvolvedor deve dar a este problema.

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

4

Arquitetura em Camadas

UNIP UNIVERSIDADE PAULISTA

- Arquitetura em camadas visa a criação de aplicativos modulares, de forma que a camada mais alta se comunica com a camada mais baixa e assim por diante, fazendo com que uma camada seja dependente apenas da camada imediatamente abaixo.

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

Arquitetura em Camadas

UNIP UNIVERSIDADE PAULISTA

- Camada de Apresentação:** Lógica de interface do usuário (GUI). O código responsável pela apresentação e controle da página e tela de navegação forma a camada de apresentação;
- Camada de Negócios:** Código referente a implementação de regras do negócio ou requisitos do sistema;
- Camada de Persistência:** Responsável por armazenamento e recuperação dos dados quando solicitado. Objetivo é o de garantir uma independência da fonte de dados (arquivos, bancos de dados) e ao mesmo tempo manter as informações entre diferentes sessões de uso.

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

Arquitetura em Camadas

UNIP UNIVERSIDADE PAULISTA

- Banco de Dados:** O BD existe fora da aplicação Java, é a atual representação persistente do estado do sistema.
- Assistentes e Classes de Utilidade:** São classes necessária para o funcionamento ou mesmo o complemento de uma aplicação ou parte dela. Exemplos: String, ArrayList, Exception, Math, etc...

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

Arquitetura em Camadas

UNIP UNIVERSIDADE PAULISTA

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

MVC



- O padrão de arquitetura **MVC** (model-view-controller) surgiu nos anos 80 com a linguagem SmallTalk.
- Inicialmente utilizado no desenvolvimento de aplicações *Desktop* por facilitar o desenvolvimento em camadas de aplicações que usam a orientação a objetos.
- Tornou-se um padrão para desenvolvimento de ambientes Web, por permitir a utilização de ferramentas, *frameworks* e até mesmo linguagens diferentes em uma aplicação.

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

9

MVC



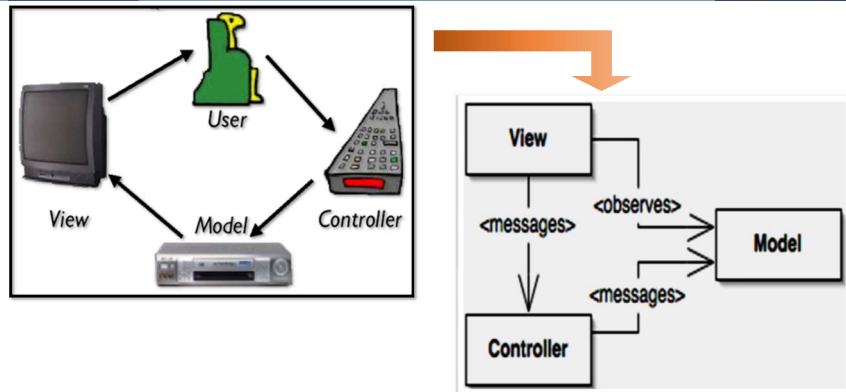
- O **MVC** é um padrão de desenvolvimento de sistemas orientados a objetos.
- É um dos Design Patterns mais utilizados em Java.
 - **Model**: dados / regras do negócio / estrutura do sistema
 - **View**: interface com usuário / telas / relatórios / site
 - **Controller**: resposta a eventos da View

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

10

MVC



Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

11

MVC



- MVC é um padrão projeto para o desenvolvimento de aplicações.
 - A implementação de aplicações usando este padrão são feitas com recurso a *frameworks* (Struts, Spring, Rails, JSF), apesar de não ser obrigatória a utilização de uma para seguir o padrão.
- Objetivo do MVC
 - Isolar mudanças na GUI, evitando que estas mudanças acarretem em mudanças na Camada de Negócios da Aplicação.
- Vantagens
 - Facilita a manutenção
 - Facilita o trabalho de equipes multidisciplinares
 - Permite o desenvolvimento de várias GUIs diferentes para o mesmo conjunto de regras de negócio

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

12

MVC Next – Steve Jobs

- A **Next** (Steve Jobs), resolveu modificar esse modelo oferecendo uma alternativa para sua linguagem de programação objective-c.
- O MVC Next delega a responsabilidade de observar o modelo para a camada de Controller que, por sua vez, envia para a camada de visão as alterações ao invés da camada de View obter esses dados do Model.

UNIP UNIVERSIDADE PAULISTA

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

13

MVC em Desktop

UNIP UNIVERSIDADE PAULISTA

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

14

MVC Desktop

UNIP UNIVERSIDADE PAULISTA

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

15

Arquitetura MVC

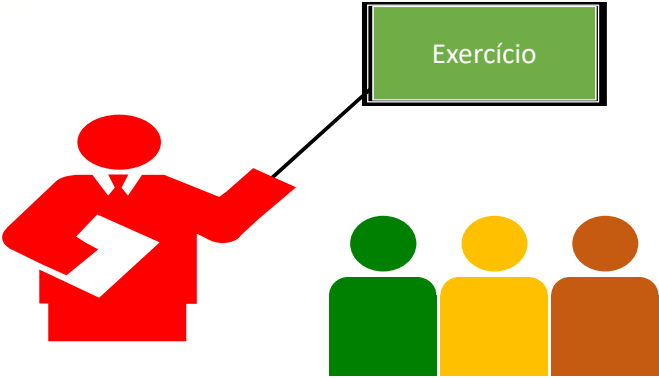
UNIP UNIVERSIDADE PAULISTA

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

16

MVC



UNIP
UNIVERSIDADE PAULISTA

Exercício

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

17

Exercício

UNIP
UNIVERSIDADE PAULISTA

Em um sistema de vendas, o cliente pode pesquisar os produtos vendo fotos, características e preço dos itens. O cliente pode incluir qualquer item pesquisado no seu “carrinho de compras”. Quando terminar, entra da tela de fechamento do pedido onde lhe é informado o valor total da compra. Nessa tela o cliente seleciona a forma de pagamento, que pode ser por boleto, cartão ou *paypal*. Após realizar o pagamento, recebe por e-mail uma cópia do pedido como comprovante da compra.

A partir da descrição acima, defina em qual camada da arquitetura MVC deve estar cada um dos eventos relacionados:

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

18

Exercício

UNIP
UNIVERSIDADE PAULISTA

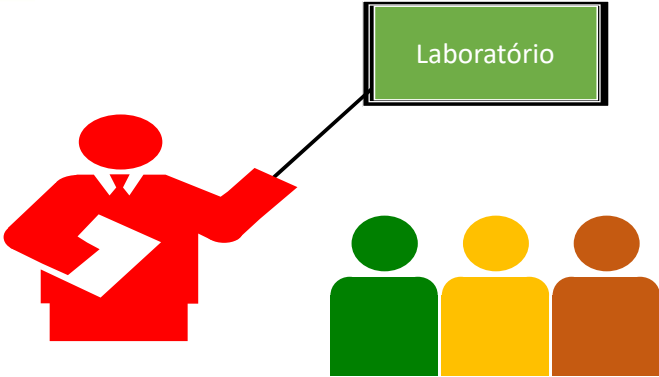
- 1) O cadastro de todos os itens do estoque da loja;
- 2) O botão “Colocar no Carrinho”, que inclui o item no carrinho de comprar do cliente;
- 3) A lista dos itens que o cliente adicionou no carrinho;
- 4) As fotos dos produtos;
- 5) Qual operação deve ser executada quando o botão “Próxima Página” for clicado;
- 6) O controle de quais formas de pagamento deverão ser exibidas como opção para o cliente;
- 7) O cálculo do total do pedido a partir dos itens que estarão listados na tela do usuário na seção “Carrinho de Compras”;
- 8) A rotina que vai solicitar o envio do e-mail após o fechamento do pedido.

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi

19

MVC



UNIP
UNIVERSIDADE PAULISTA

Laboratório

Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

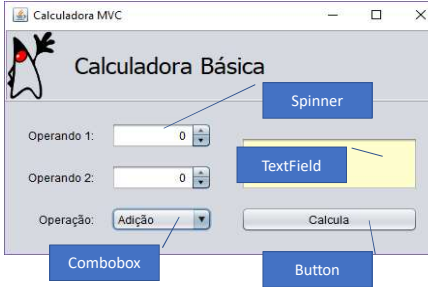
Prof. Ricardo Drudi

20

Exercício

Crie uma calculadora com as 4 operações básicas, conforme o modelo de tela abaixo. Implemente sua calculadora usando 100% do padrão MVC.

- 1) Na **View**, somente o tratamento de componentes da tela deve ser feito.
- 2) Deve haver um **Model** que tenha métodos para cada operação.
- 3) O **Controller** deve criar a **View**, adicionar os respectivos **Listeners**, capturar os valores digitados, utilizar o **Model** para efetivar a operação e devolver o resultado para a **View**.




Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi
21

Projeto Lab

Implemente o caso de uso “Cadastrar Cliente”. Sua rotina deve conter:

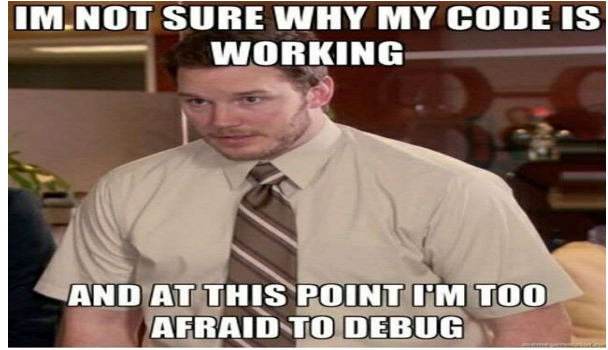
- 1) **View** para digitação dos dados
- 2) **Model** para armazenamento dos dados
- 3) Opção para listar todos os clientes cadastrados
- 4) Opção para pesquisar um cliente existente
- 5) Opção para alterar os dados de um cliente
 - Deve incluir uma opção para pesquisar um cliente
- 6) Opção para excluir um cliente
 - Deve incluir uma opção para pesquisar um cliente




Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi
22

Obrigado.



Copyright © 2018. Prof. Ricardo Drudi
Todos direitos reservados. A reprodução ou divulgação total ou parcial deste documento é expressamente proibida sem o consentimento formal, por escrito, do professor Ricardo Drudi.



Aplicações de Linguagem de Programação Orientada a Objetos
Arquitetura Model-View-Controller

Prof. Ricardo Drudi
23