Curso de Java 8 para Web

Professor
Antonio Benedito Coimbra Sampaio Jr

abc | Treinamentos

www.abctreinamentos.com.br

Quarta

Disciplina

JEE – Java Servlets e JSP

- **UNIDADE 1:** Introdução à Internet, WEB e HTML
- **UNIDADE 2:** Java Servlets
- **UNIDADE 3:** JSP
- **UNIDADE 4: Padrão de Projeto MVC (Integração Servlet e JSP)**

UNIDADE 4

PADRÃO DE PROJETO MVC (INTEGRAÇÃO SERVLET E JSP)

Padrões de Projeto

Modelo de Aplicação JEE

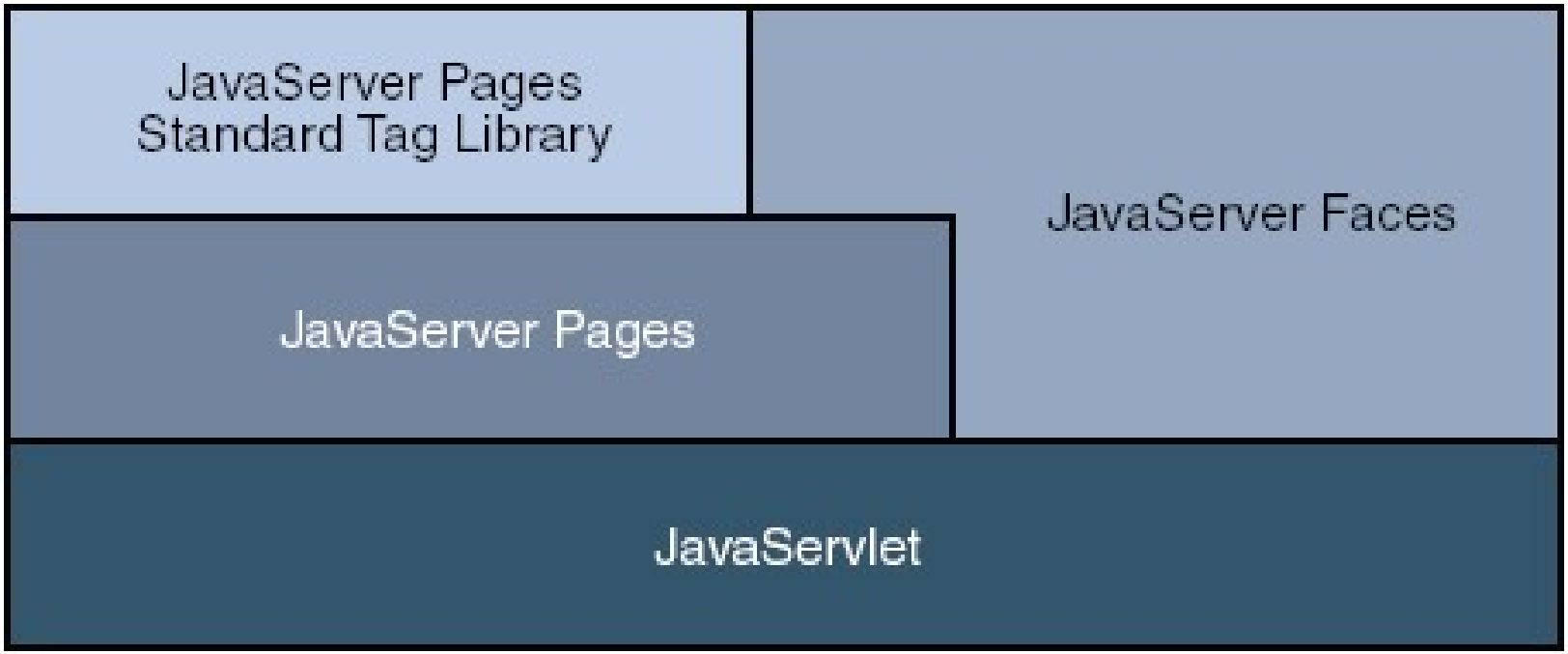
- A plataforma JEE utiliza um modelo de aplicação distribuída multicamada.

- A lógica da aplicação é dividida em componentes de acordo com a sua função.

- Os vários componentes que constituem uma aplicação JEE são instalados em diferentes equipamentos.

COMPONENTES WEB JEE

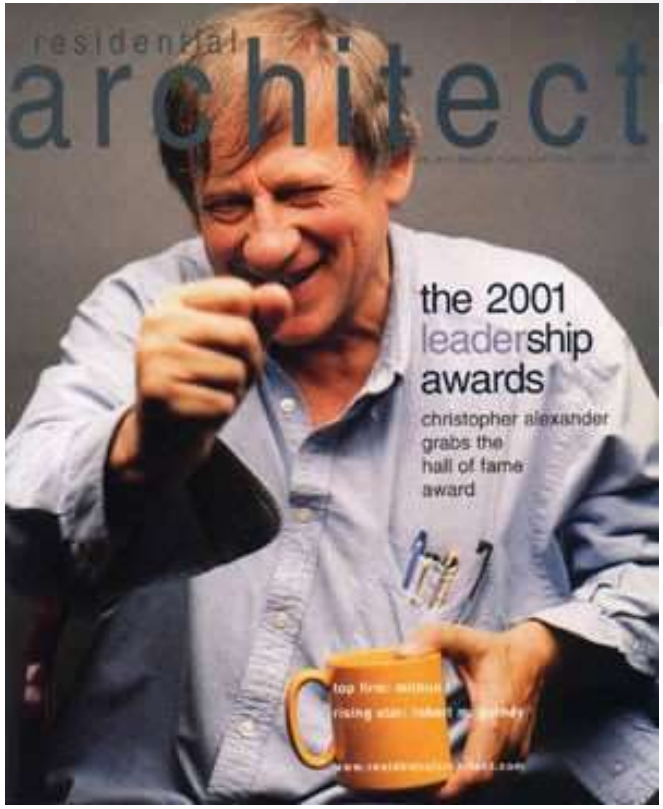
- A Plataforma JEE define quatro tecnologias básicas para a construção de Aplicações WEB:
  - **Java Servlets 3.1**
  - **JavaServer Pages 2.2**
  - **JavaServer Faces 2.2**
  - **JavaServer Pages Standard Tag Library 1.2.1**



PADRÕES DE PROJETO

HISTÓRICO

- Em 1977 o arquiteto, matemático e urbanista austríaco Christopher Alexander publicou um catálogo com mais de 250 padrões para a construção civil, que discutiam questões comuns da arquitetura, descrevendo em detalhe o problema e as justificativas de sua solução.



- Christopher Alexander descobriu que diminuindo o foco, ou seja, procurando estruturas que resolvam problemas similares, ele pode discernir similaridades entre projetos de alta qualidade. Ele chamou essas similaridades de “padrões”.

PADRÕES DE PROJETO

HISTÓRICO

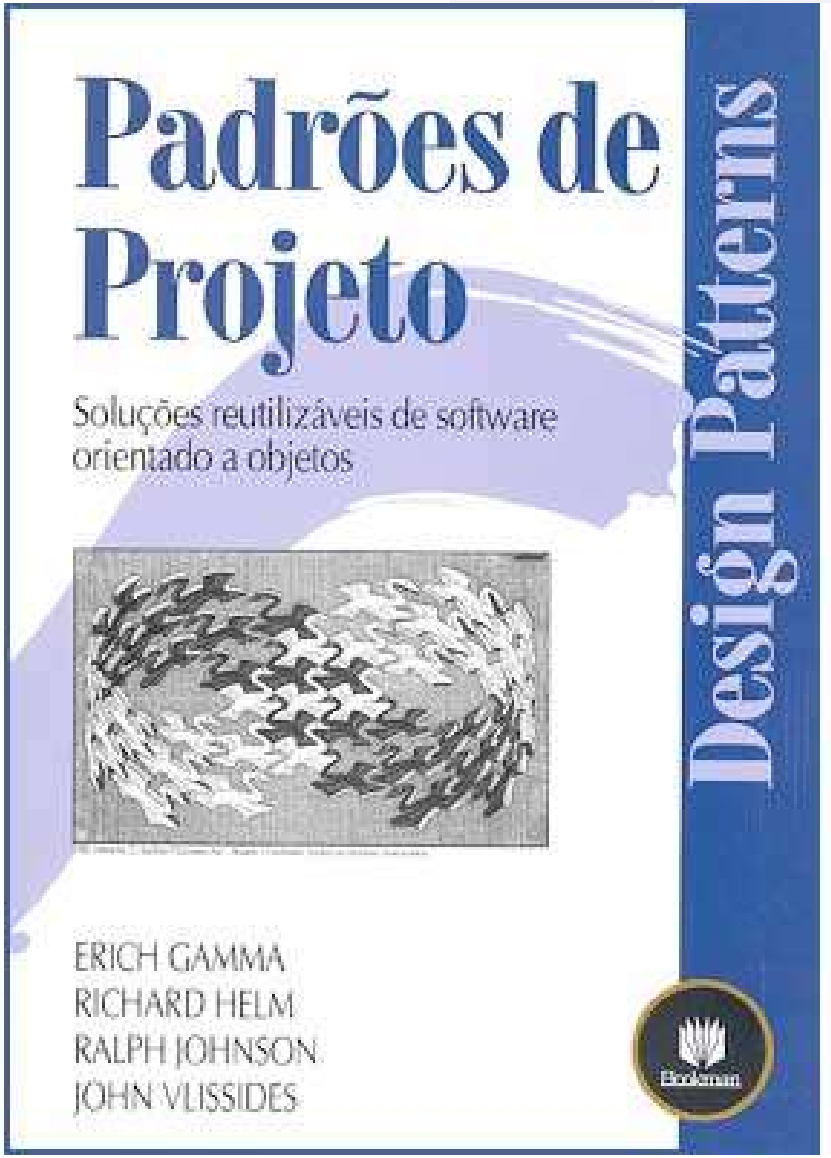
- Algum tempo depois ele formaliza seu método de descrição dos padrões, defendendo que seu uso não limitaria os arquitetos às soluções prescritas, mas garantiria a presença de elementos fundamentais, e a possibilidade de aperfeiçoa-la através das experiências adquiridas.

- Esse método chamou atenção da comunidade de software, fazendo que o tema ganhasse destaque nas conferências sobre orientação a objetos.

PADRÕES DE PROJETO

- Em 1995 Erich Gama, Richard Helm, Ralph Johnson, John Vlissides, conhecidos como os quatro amigos [Gang of Four - GoF], publicaram o livro sobre o título: "***Design patterns – elements of reusable object-oriented software***", que ganhou uma versão na língua portuguesa sobre o título de "**Padrões de Projeto – Soluções reutilizáveis de software orientado a objetos**".

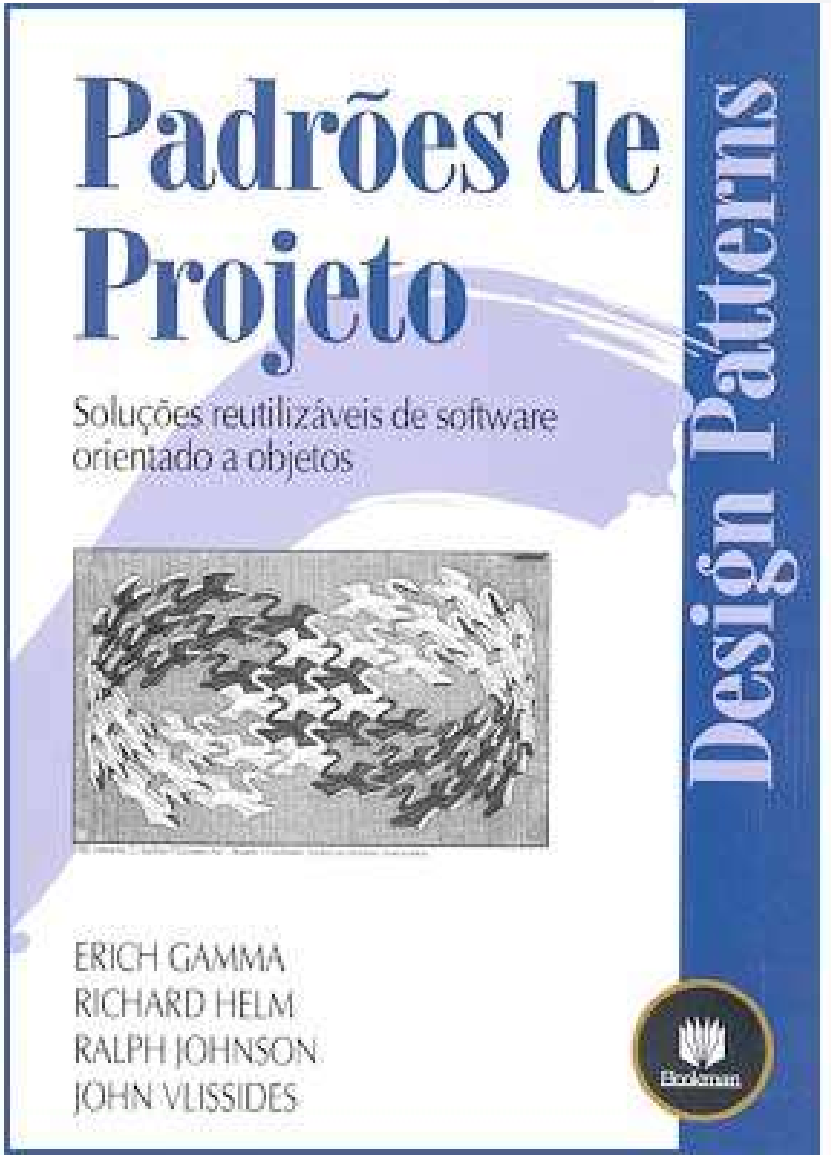
- Durante o desenvolvimento de quaisquer sistemas, projetistas tendem a se valer de decisões de projeto tomadas anteriormente em outros sistemas.



PADRÕES DE PROJETO

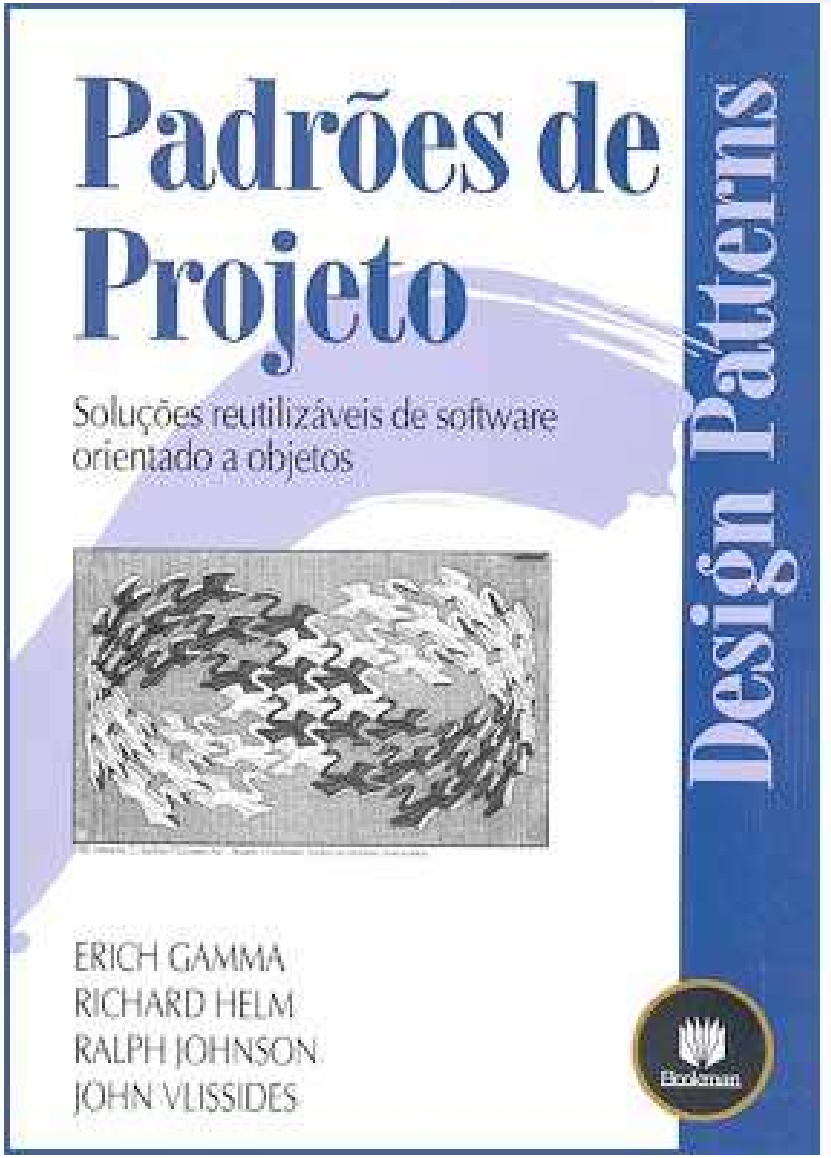
- Padrões de Projeto (*design patterns*) podem ser vistos como uma forma de documentar conjuntos de regras que descrevem decisões de projeto recorrentes em vários sistemas, facilitando assim a reutilização de soluções já existentes.

- O livro ao lado é a referência no estudo de padrões de projeto na área de engenharia de software.



PADRÕES DE PROJETO

- Este livro apresenta um **catálogo de 23 padrões**.
- Não apresenta padrão para um domínio de aplicação específico.
- Estes padrões representam o estado-da-prática em boas construções de projeto orientado a objetos.
- É comum encontrar no detalhamento de padrões específicos de domínio a ocorrência de algum destes padrões.



PADRÕES DE PROJETO

CATEGORIAS

- Segundo o livro de referência, os padrões de projeto na área de orientação a objetos estão organizados em **03 grupos**:

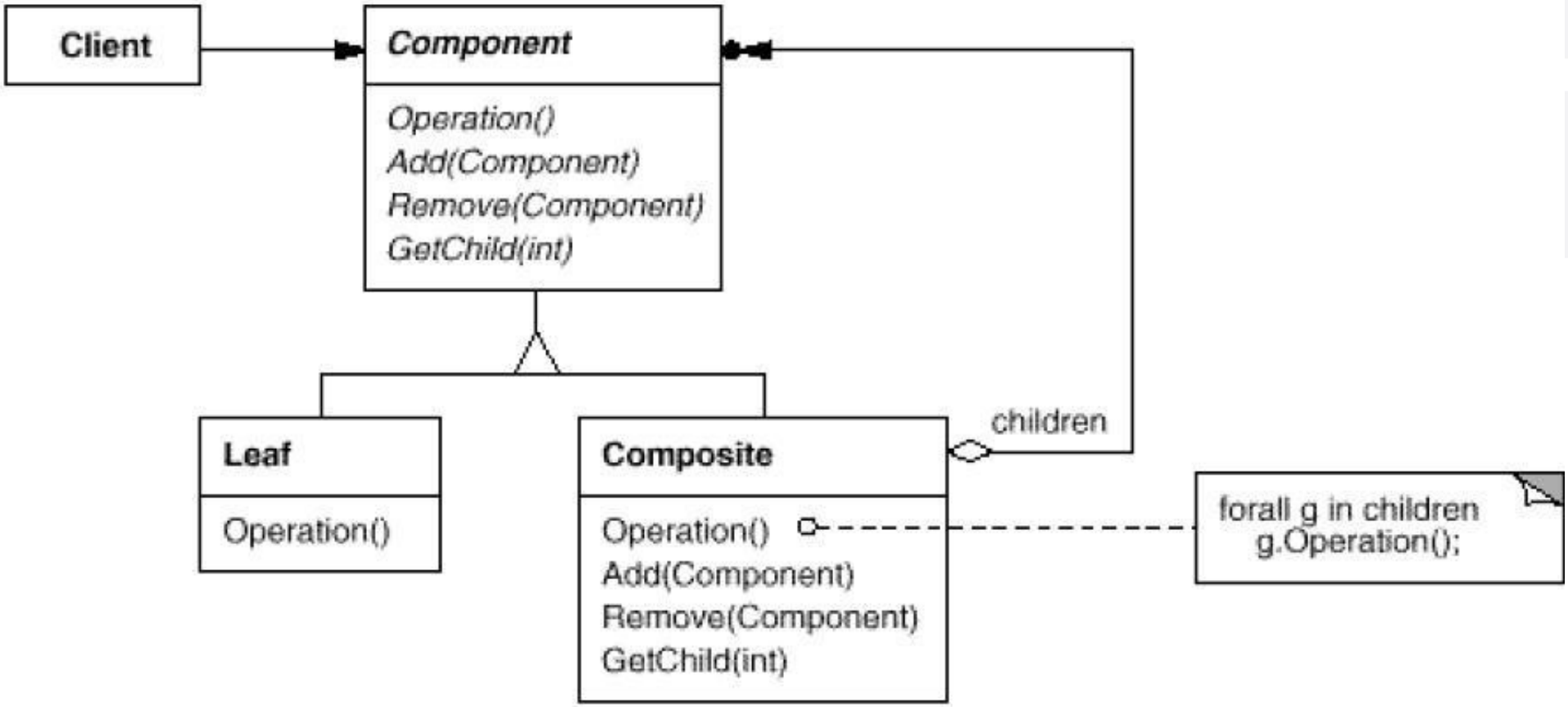
- **Padrões de Criação**
  - Abstraem o processo de instanciação de objetos. Eles ajudam a tornar um sistema independente de como seus objetos são criados, compostos e representados. Eles dão muita flexibilidade para o que é criado, quem cria, como e quando cria.
- **Padrões de Estrutura**
  - Preocupam-se com a forma como as classes e objetos são compostos para formar estruturas maiores.
- **Padrões de Comportamento**
  - Preocupam-se com algoritmos e a atribuição de responsabilidades entre os objetos, dando um grande foco na comunicação entre eles.

OS 23 PADRÕES DE PROJETO

Criação	Estutural	Comportamental
Abstract Factory	Adapter	Chain of Responsibility
Builder	Bridge	Command
Factory Method	Composite	Interpreter
Prototype	Decorator	Iterator
Singleton	Facade	Mediator
	Flyweight	Memento
	Proxy	Observer
		State
		Strategy
		Visitor
		Template Method

http://abrindoojogo.com.br/padroes-de-projeto-em-games-introducao

EXEMPLO: COMPOSITE



© Wikipedia

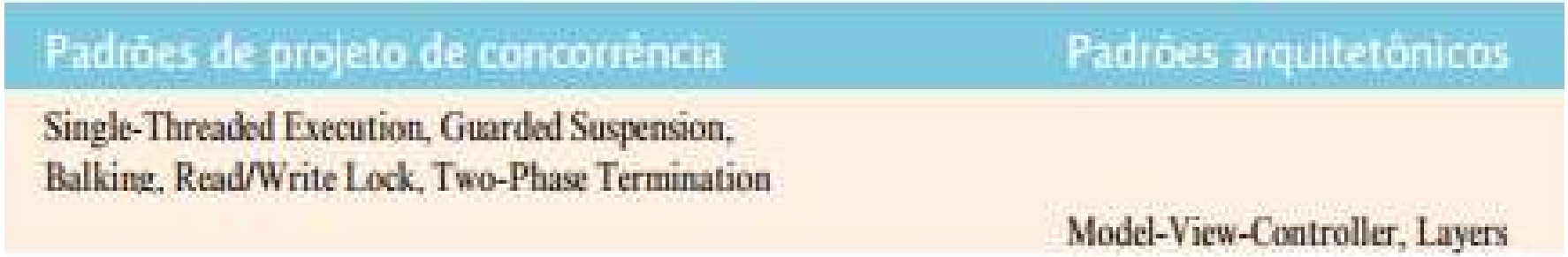
PADRÕES DE PROJETO

OUTRAS CATEGORIAS

- Após a popularização do livro de referência, diversos outros padrões foram criados e documentados, estando organizados em mais 02 grupos:

- **Padrões de Concorrência**
  - Especialmente úteis no projeto de sistemas de múltiplas threads.

- **Padrões Arquitetônicos**
  - Especificam como subsistemas interagem entre si.



© Java Como Programar - Deitel

PADRÕES DE PROJETO

BENEFÍCIOS

- **Confiabilidade Crescente**
  - Toda vez que um software é utilizado, ele é novamente testado.
  - Componentes já utilizados e testados em outros sistemas são mais confiáveis que novos componentes.
- **Risco de Processo Reduzido**
  - Margem de erro dos custos de reuso menor que dos custos de desenvolvimento.
- **Uso Efetivo de Especialistas**
  - Especialista desenvolve software reutilizável encapsulando seu conhecimento, ao invés de desenvolver as mesmas funcionalidades repetidas vezes em diferentes projetos.

Exercícios

1) [FCC – 2010 – TRT] Sobre design pattern considere:

I. No framework pode incluir código de programação e conter vários design patterns.

II. No design pattern pode incluir código de programação e conter vários frameworks.

III. Os design patterns são bastante abstratos e os frameworks menos abstratos.

Está correto o que consta em

a) I e III, apenas.
b) I e II, apenas.
c) II e III, apenas.
d) III, apenas.
e) I, II e III.

Exercícios

2) [CONSULPLAN – 2012 – TSE] O desenvolvimento de software é uma atividade que apresenta dificuldades, ligada ao entendimento do problema. Design Patterns surgiram na busca de soluções para as dificuldades, tornando-se um mecanismo eficiente no compartilhamento de conhecimento entre os desenvolvedores. Gamma propõe um modo de categorização dos DESIGN PATTERNS, definindo famílias de padrões relacionados, descritos a seguir.

I. Abrange a configuração e inicialização de objetos e classes.

II. Lida com as interfaces e a implementação das classes e dos objetos.

III. Lida com as interações dinâmicas entre grupos de classes e objetos.

Essas famílias são denominadas, respectivamente,

a) Structural Patterns, Standard Patterns e Creational Patterns.
b) Behavioral Patterns, Structural Patterns e Standard Patterns.
c) Creational Patterns, Structural Patterns e Behavioral Patterns.
d) Standard Patterns, Creational Patterns e Structural Patterns.

Exercícios

1) [FCC – 2010 – TRT] Sobre design pattern considere:

a) I e III, apenas.

2) [CONSULPLAN – 2012 – TSE] O desenvolvimento de software é uma atividade que apresenta dificuldades, ligada ao entendimento do problema. Design Patterns surgiram na busca de soluções para as dificuldades, tornando-se um mecanismo eficiente no compartilhamento de conhecimento entre os desenvolvedores. Gamma propõe um modo de categorização dos DESIGN PATTERNS, definindo famílias de padrões relacionados, descritos a seguir.

d) Standard Patterns, Creational Patterns e Structural Patterns.

Padrão de Projeto MVC

MVC

DEFINIÇÃO

- MVC é um padrão de projeto que separa a lógica da aplicação em **03 camadas**: **Model, View e Controller**.

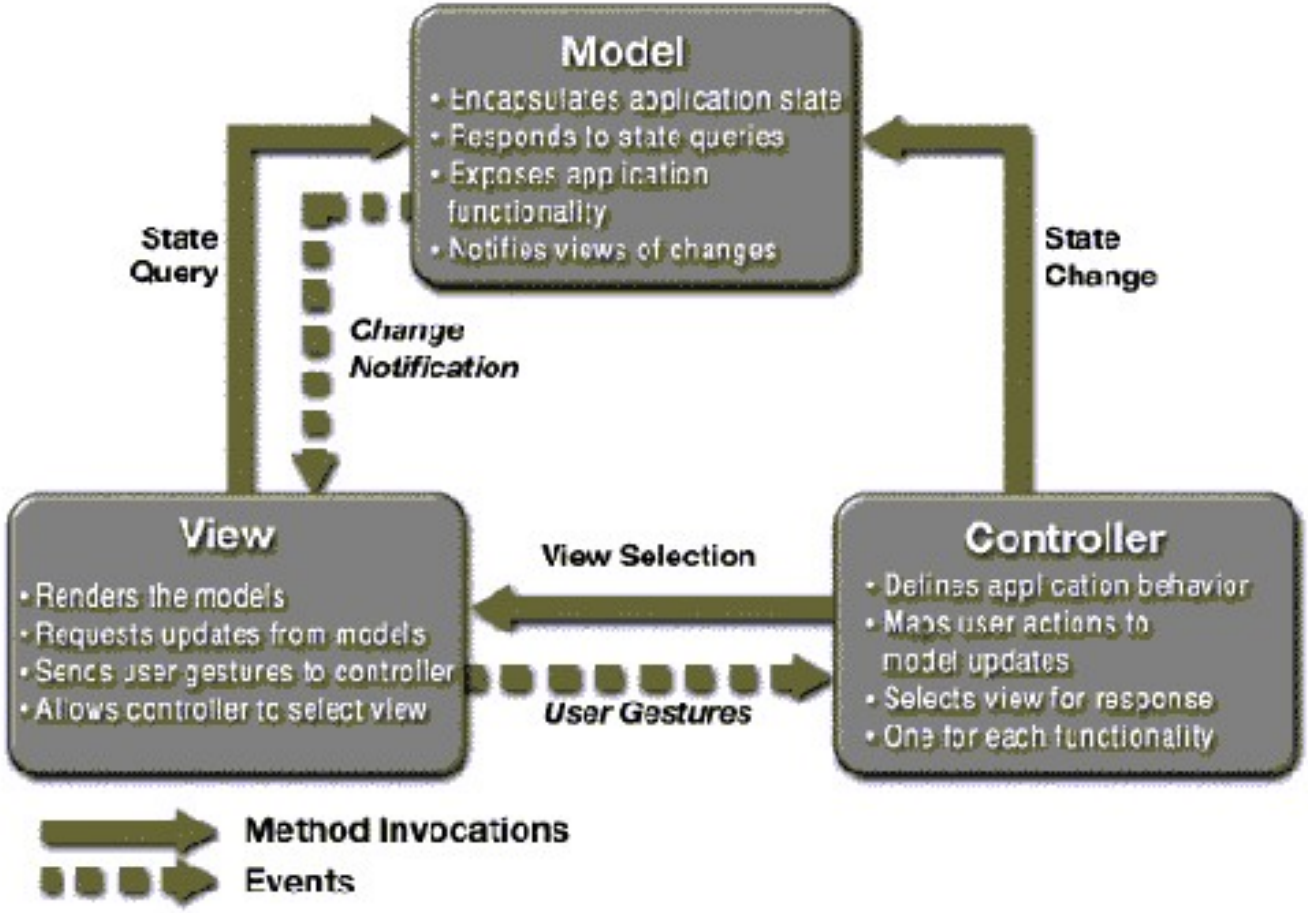
- O **Model** representa a camada responsável por gerenciar o acesso aos dados da aplicação e as **regras do negócio** que governam o acesso e a modificação dos dados.

- O **View** representa a camada responsável por mostrar as informações aos usuários (telas), sem estar preocupada onde e como a informação foi obtida, apenas exibe a informação nos formatos HTML, JSP, JSF, etc;

- O **Controller** representa a camada responsável por definir o funcionamento da aplicação, interpretando as ações do usuário e mapeando-as para chamadas na camada modelo.

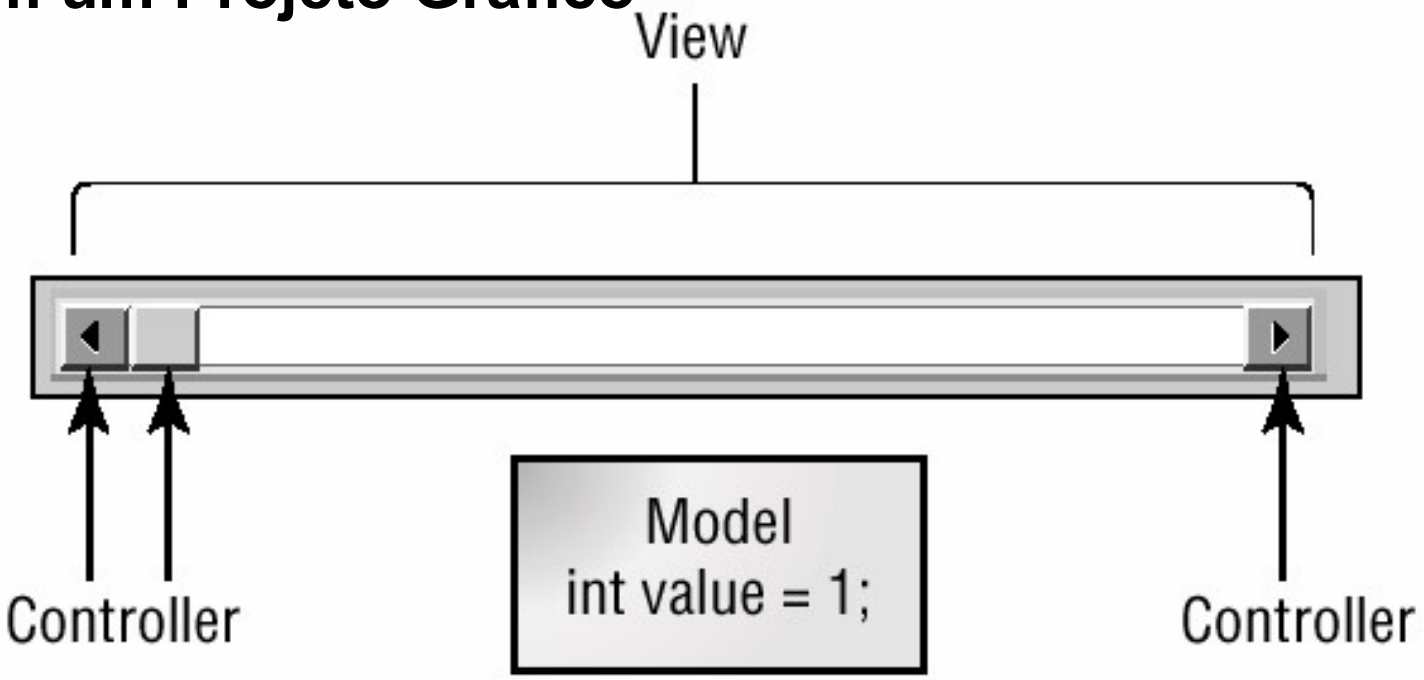
MVC

- Quando o **Controller** é acionado, este comunica mudanças para o Model. O **Model** faz mudanças nos dados e disponibiliza novos dados para o View. O **View** recebe os novos dados e os apresenta na tela.



EXEMPLO DE USO

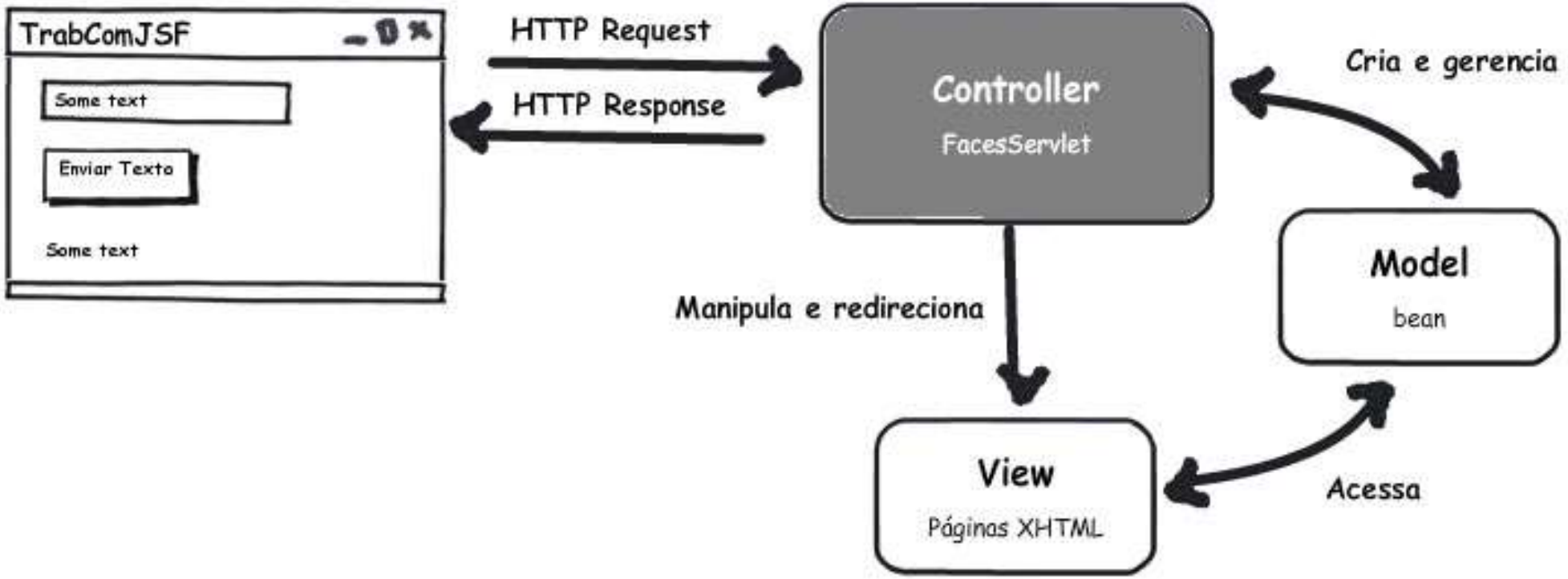
MVC em um Projeto Gráfico



- O **Controller** determina a mudança no componente;
- O **Model** gerencia os dados a serem acessados/modificados;
- O **View** oferece a visualização desses dados.

EXEMPLO DE USO

MVC em um Projeto WEB



VANTAGENS

- Como o modelo MVC gerencia múltiplos visualizadores usando o mesmo modelo, é fácil de manter, testar e atualizar.
- Aumenta a produtividade da equipe de desenvolvimento de software, pois possibilita o desenvolvimento em paralelo para cada uma das camadas.
- A aplicação construída com MVC é naturalmente escalável, pois novos componentes no **Controller** e no **View** não afetam o código já existente.
- Mudanças no **Controller** não afetam os componentes do **View**.
- Mudanças no **View** não afetam as classes do **Controller**.

VANTAGENS

- Vários frameworks de desenvolvimento Web populares são baseados no MVC. **E são gratuitos!**

DESVANTAGENS

- Requer uma quantidade maior de tempo para analisar e modelar o sistema.
- Requer pessoal especializado.
- Não é aconselhável para pequenas aplicações.

Exercícios

1) [FCC - 2007 - TRF - 4ª REGIÃO] No modelo multicamadas MVC, considere as seguintes propriedades e suas prováveis e respectivas características, estas últimas relacionadas em negrito:

I. gerenciamento de múltiplos visualizadores usando mesmo modelo - facilidade/dificuldade de manutenção, teste e atualização de sistemas múltiplos;

II. desenvolvimento em paralelo para o modelo, visualizador e controle - possível/impossível;

III. uso em pequenas aplicações - aconselhável/ desaconselhável em razão do custo/benefício.

Respectivamente a I, II e III, as características corretas são

a) facilidade, possível e desaconselhável.
b) facilidade, impossível e aconselhável.
c) dificuldade, possível e aconselhável.
d) dificuldade, impossível e desaconselhável.
e) dificuldade, possível e desaconselhável.

Exercícios

2) [NCE-UFRJ - 2008 - UFRJ] Considere as seguintes afirmativas sobre o padrão Modelo-Visão-Controle (MVC) como utilizado no estilo de projeto orientado a objetos:

I- Os objetos do Modelo devem ter conhecimento direto de objetos da Visão.

II- O padrão MVC permite o desenvolvimento em separado das camadas de Modelo e Visão.

III- O padrão MVC aumenta o acoplamento entre a camada Modelo e a camada Visão.

A(s) afirmativa(s) correta(s) é/são somente:

a) I
b) II
c) III
d) I e II
e) I, II e III

Exercícios

1) [FCC - 2007 - TRF - 4ª REGIÃO] No modelo multicamadas MVC, considere as seguintes propriedades e suas prováveis e respectivas características, estas últimas relacionadas em negrito:

a) facilidade, possível e desaconselhável.

2) [NCE-UFRJ - 2008 - UFRJ] Considere as seguintes afirmativas sobre o padrão Modelo-Visão-Controle (MVC) como utilizado no estilo de projeto orientado a objetos:

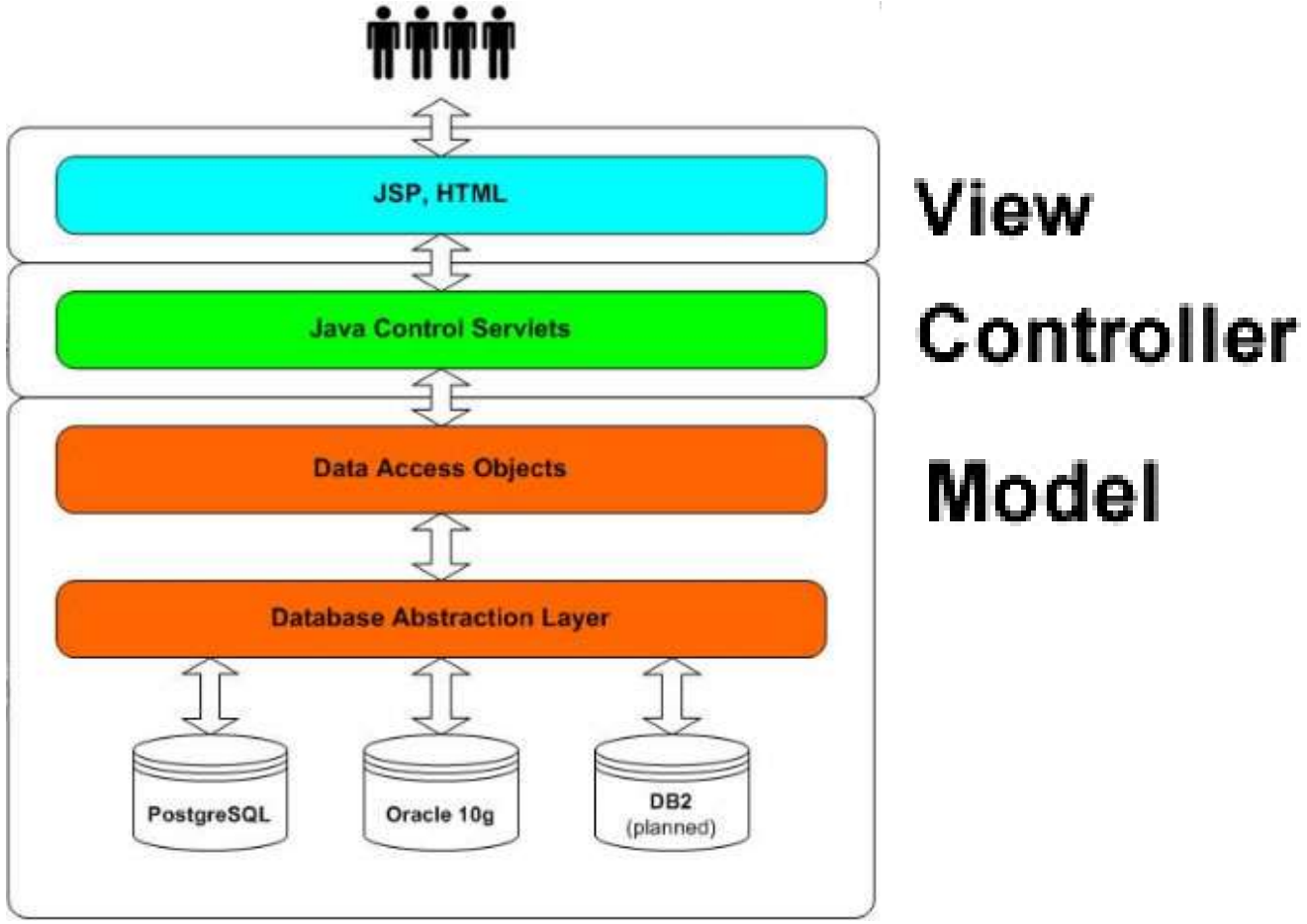
b) II.

Projeto Prático (Parte 4)

Projeto Prático

Adequação do Projeto ao Padrão MVC:

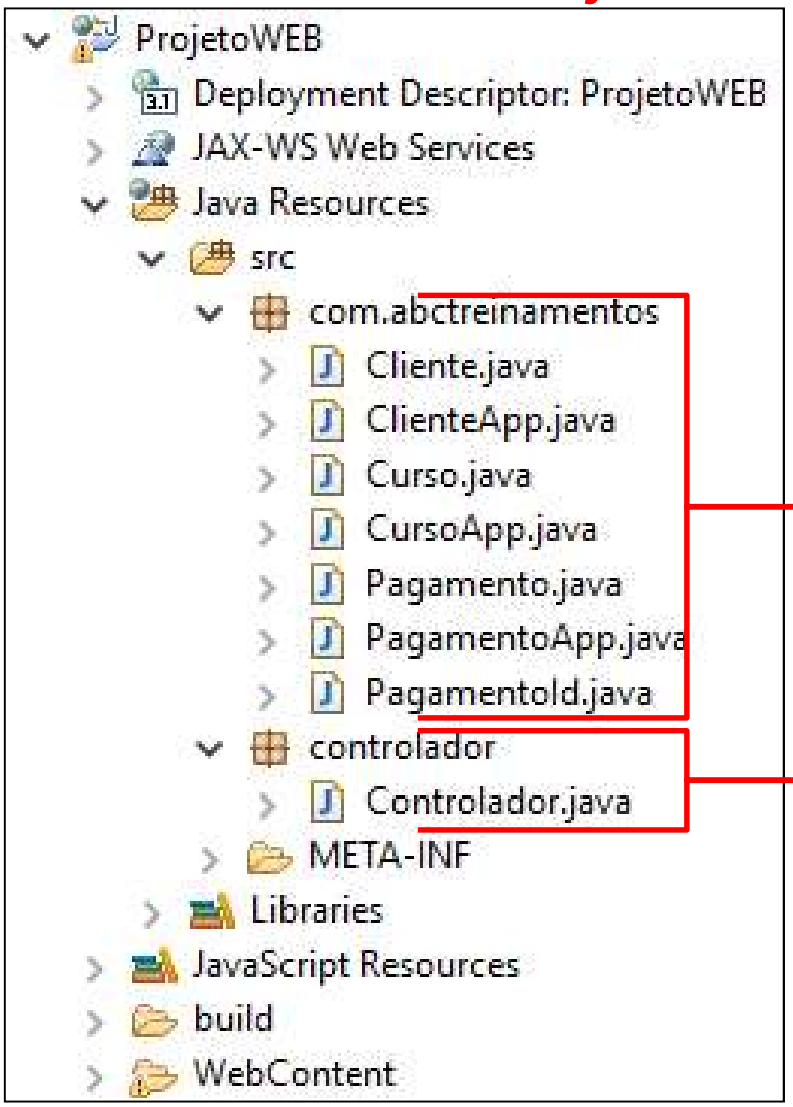
- Faz uso das tecnologias **Java Servlet** (Camada **Controller**), **JSP** (Camada **View**) e **POJO** (Camada **Model**).



Projeto Prático

Adequação do Projeto ao Padrão MVC:

- Estrutura final do **ProjetoWEB:**



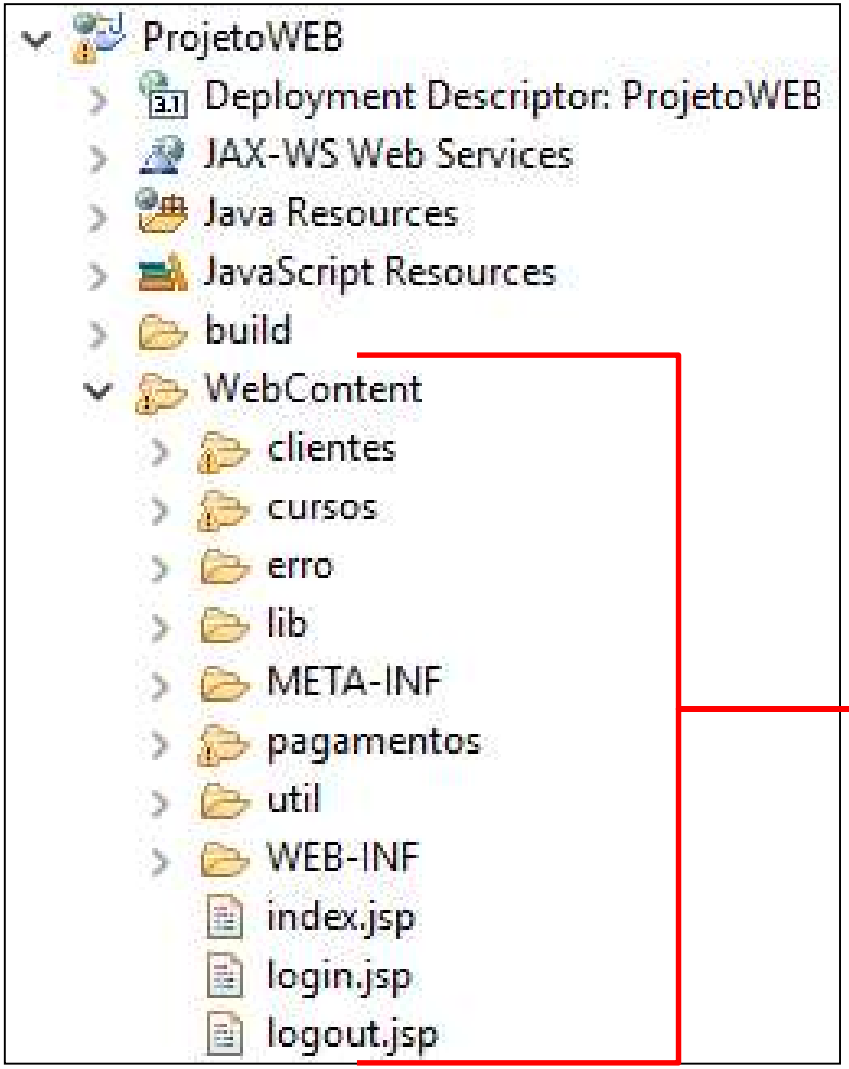
Camada Model (Classes **POJO**)

Camada Controller (Servlet **Controlador**)

Projeto Prático

Adequação do Projeto ao Padrão MVC:

- Estrutura final do **ProjetoWEB:**

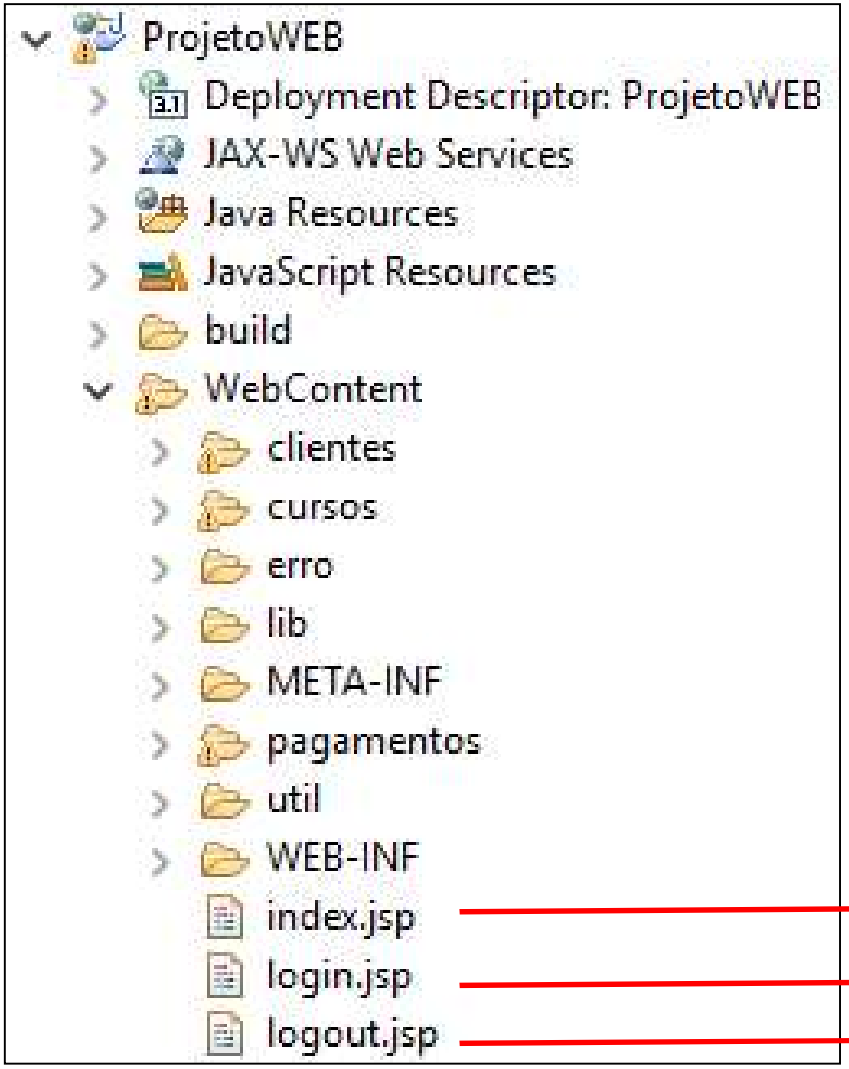


Camada View (Páginas **JSP**)

Projeto Prático

Adequação do Projeto ao Padrão MVC:

- Estrutura final do **ProjetoWEB:**



Página Inicial

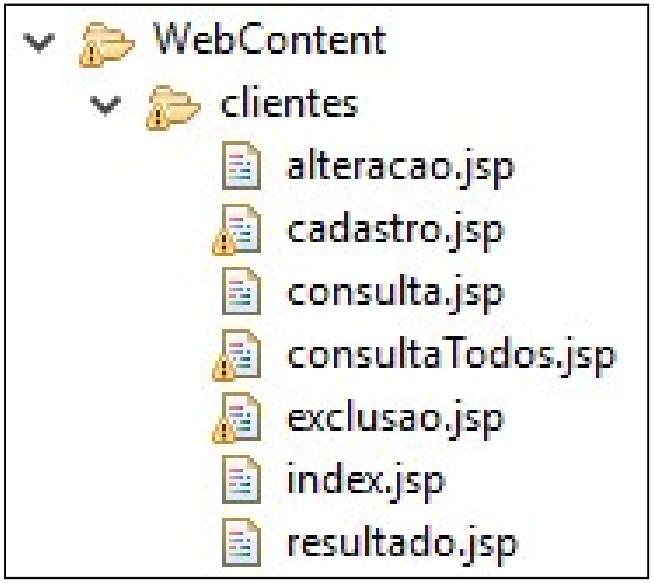
Página de Autenticação

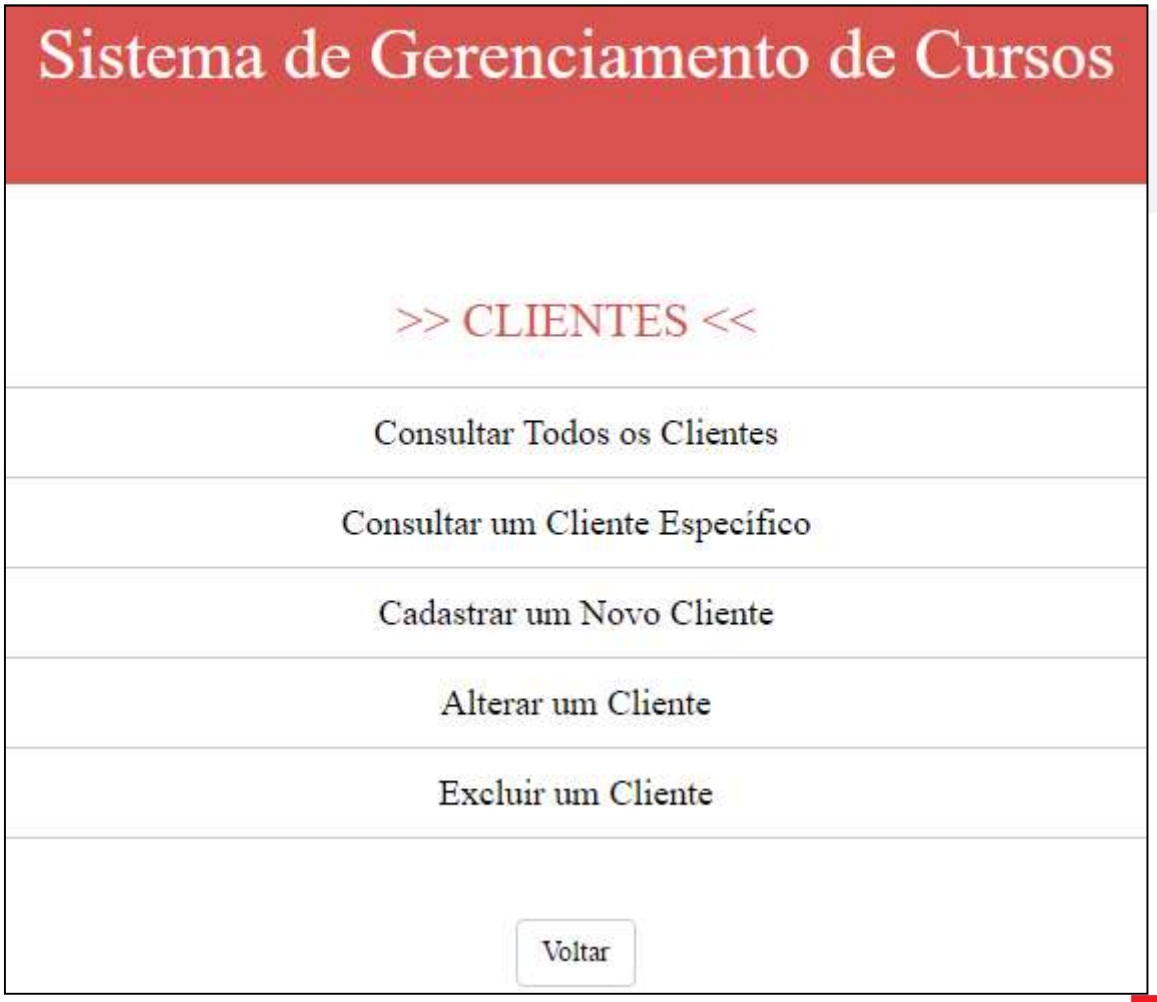
Página de Sair

Projeto Prático

Adequação do Projeto ao Padrão MVC:

- Estrutura final do **ProjetoWEB:**

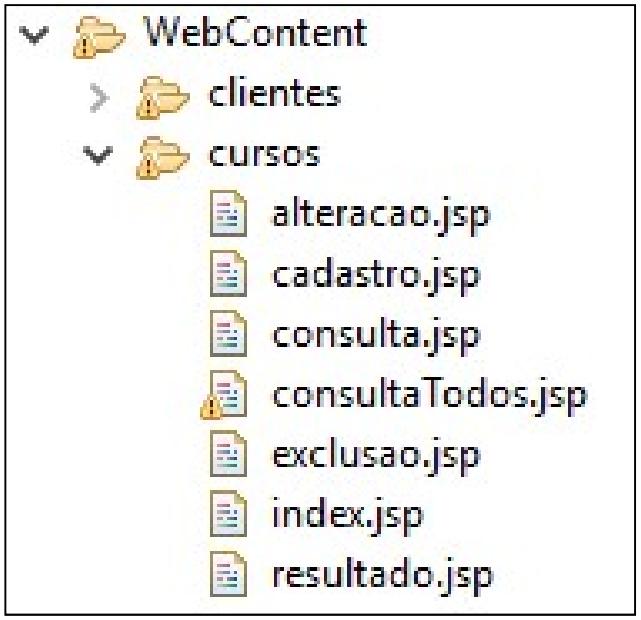


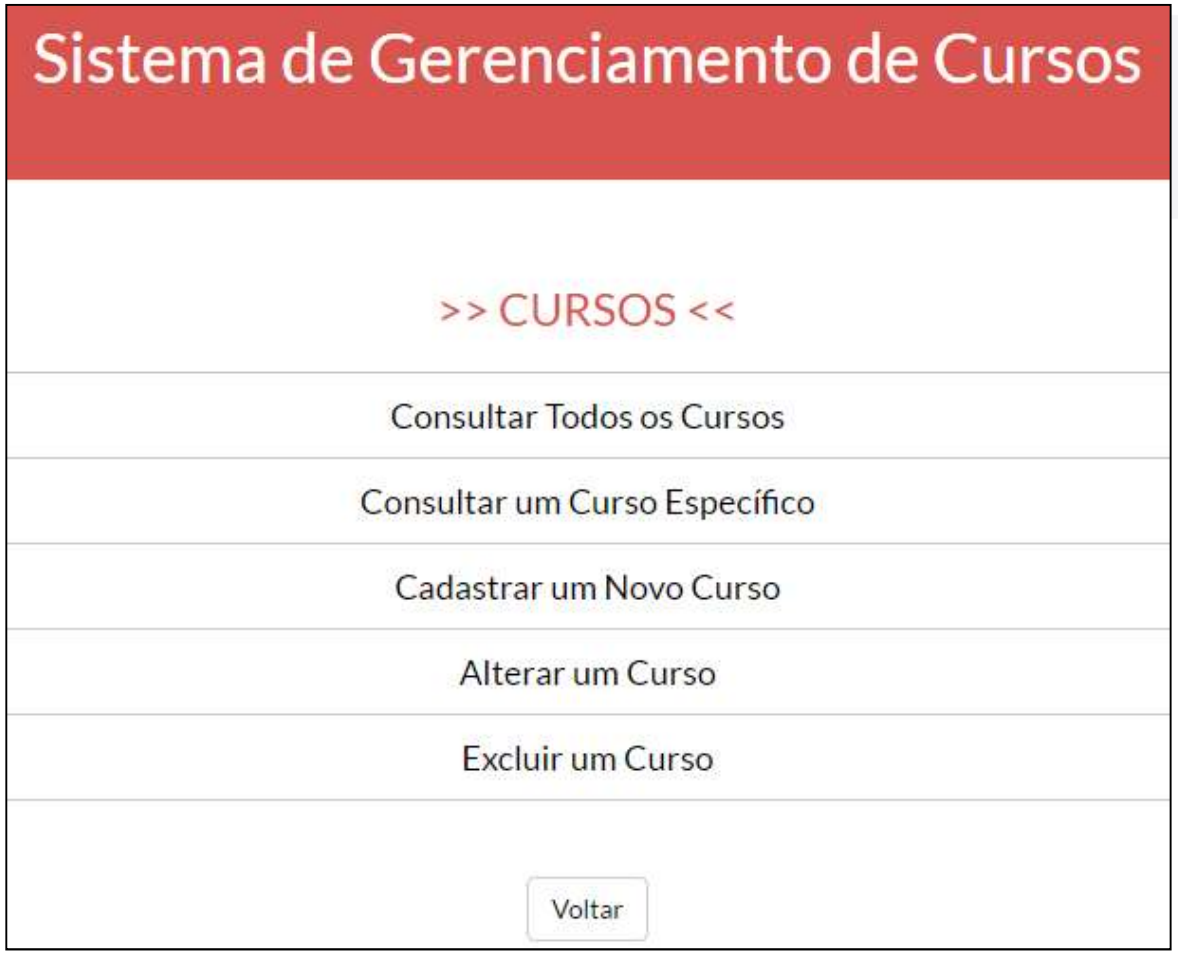


Projeto Prático

Adequação do Projeto ao Padrão MVC:

- Estrutura final do **ProjetoWEB:**

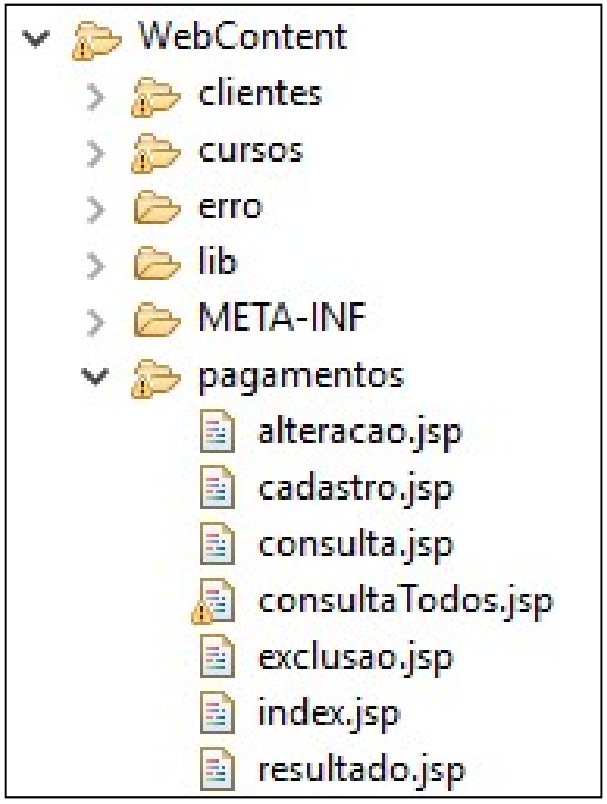


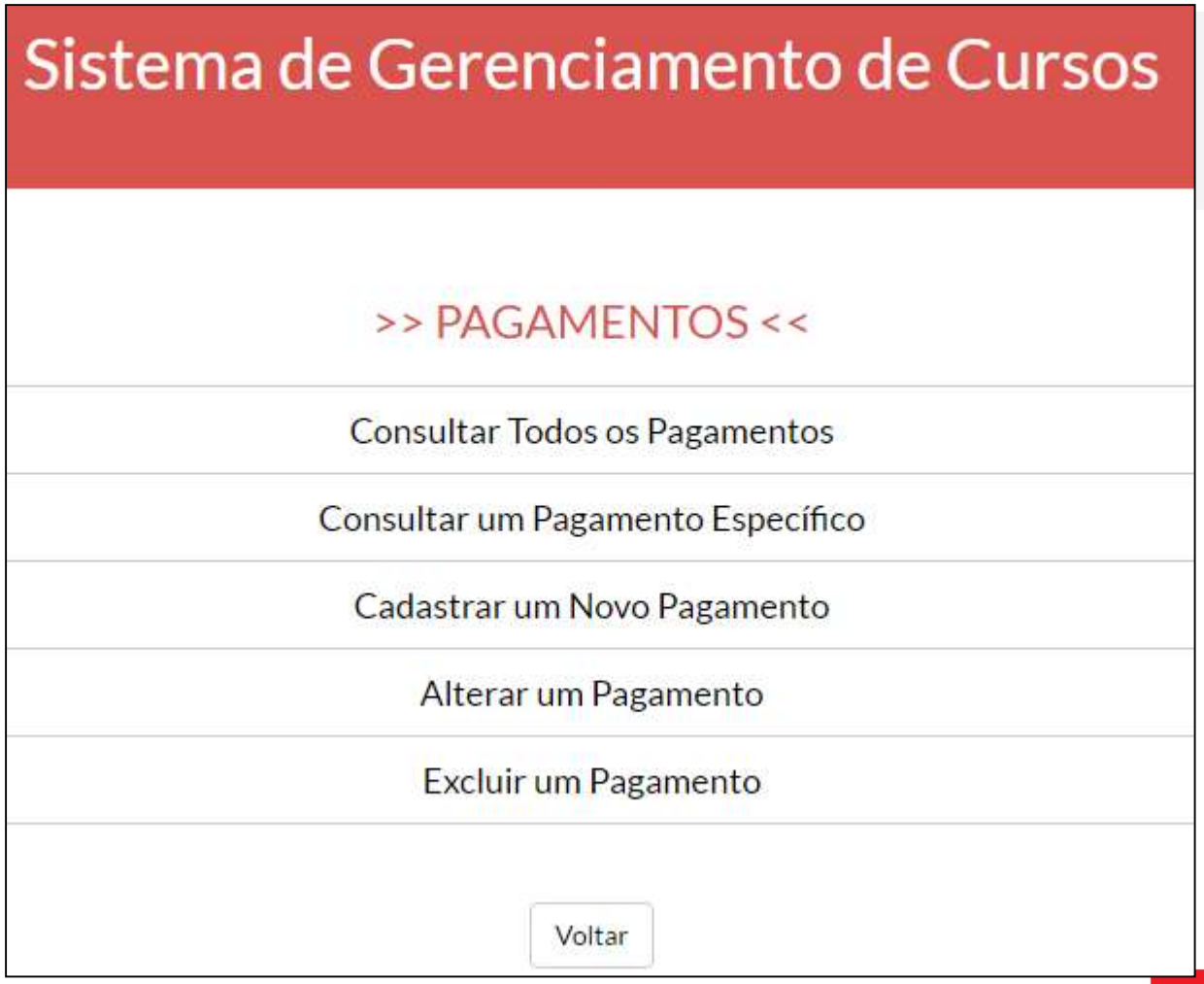


Projeto Prático

Adequação do Projeto ao Padrão MVC:

- Estrutura final do **ProjetoWEB:**





RESUMO

TÓPICOS APRESENTADOS

- Nesta aula nós estudamos:
  - **Padrões de Projeto**
  - **Padrão de Projeto MVC**
  - **Projeto Prático (Parte 4)**

ATIVIDADES PARA SE APROFUNDAR

- **1) Foi criada uma página JSP (autenticacao.jsp) para realizar a autenticação do usuário do Sistema de Gerenciamento de Cursos. Para seguir a risca o modelo preconizado pelo padrão MVC, essa autenticação deveria ser executada por uma classe Java na camada Model (com.abctreinamentos).**
  - **Faça essa mudança no ProjetoWEB e analise as vantagens e as desvantagens dessa abordagem.**
- **2) Faça um estudo comparativo dos 05 frameworks de desenvolvimento Web populares listados abaixo:**
  - **SpringMVC**
  - **JSF 2**
  - **Demoiselle**
  - **Struts2**
  - **Vraptor 3**