

Curso - Padrões de Projeto

Módulo 5: Model-View-Controller

Vítor E. Silva Souza
vitorsouza@gmail.com

<http://www.javablogs.com.br/page/engenh>

<http://esjug.dev.java.net>



Sobre o Instrutor

- **Formação:**
 - Graduação em Ciência da Computação, com ênfase em Engenharia de Software, pela Universidade Federal do Espírito Santo (UFES);
 - Mestrado em Informática (em andamento) na mesma instituição.
- **Java:**
 - Desenvolvedor Java desde 1999;
 - Especialista em desenvolvimento Web;
 - Autor do blog Engenho – www.javablogs.com.br/page/engenho.
- **Profissional:**
 - Consultor em Desenvolvimento de Software Orientado a Objetos – Engenho de Software Consultoria e Desenvolvimento Ltda.

Estrutura do Curso

✓ **Módulo 1**

Introdução

✓ **Módulo 2**

Padrões de Criação

✓ **Módulo 3**

Padrões de Estrutura

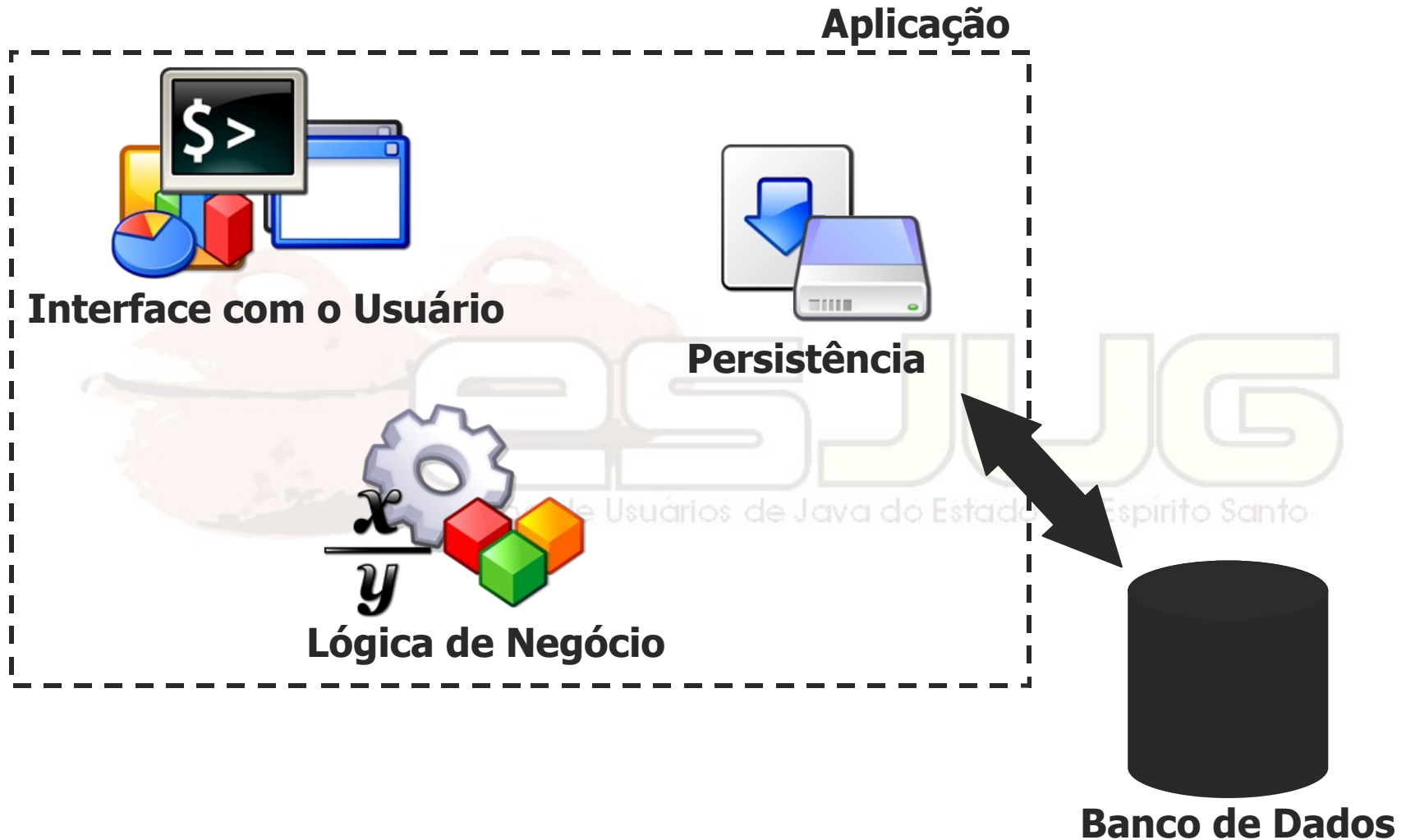
✓ **Módulo 4**

Padrões de Comportamento

➔ **Módulo 5**

O Padrão Model-View-Controller

Sistema em uma camada



Exemplo

```
<html><body>
```

```
<%
```

```
sql = "SELECT * FROM ...";
```

```
rset = stmt.executeQuery(sql);
```

```
/* ... */
```

```
sql = "UPDATE Tabela SET ...";
```

```
stmt.executeUpdate(sql);
```

```
%>
```

```
</body></html>
```

Execução de queries ao banco de dados diretamente da página web.

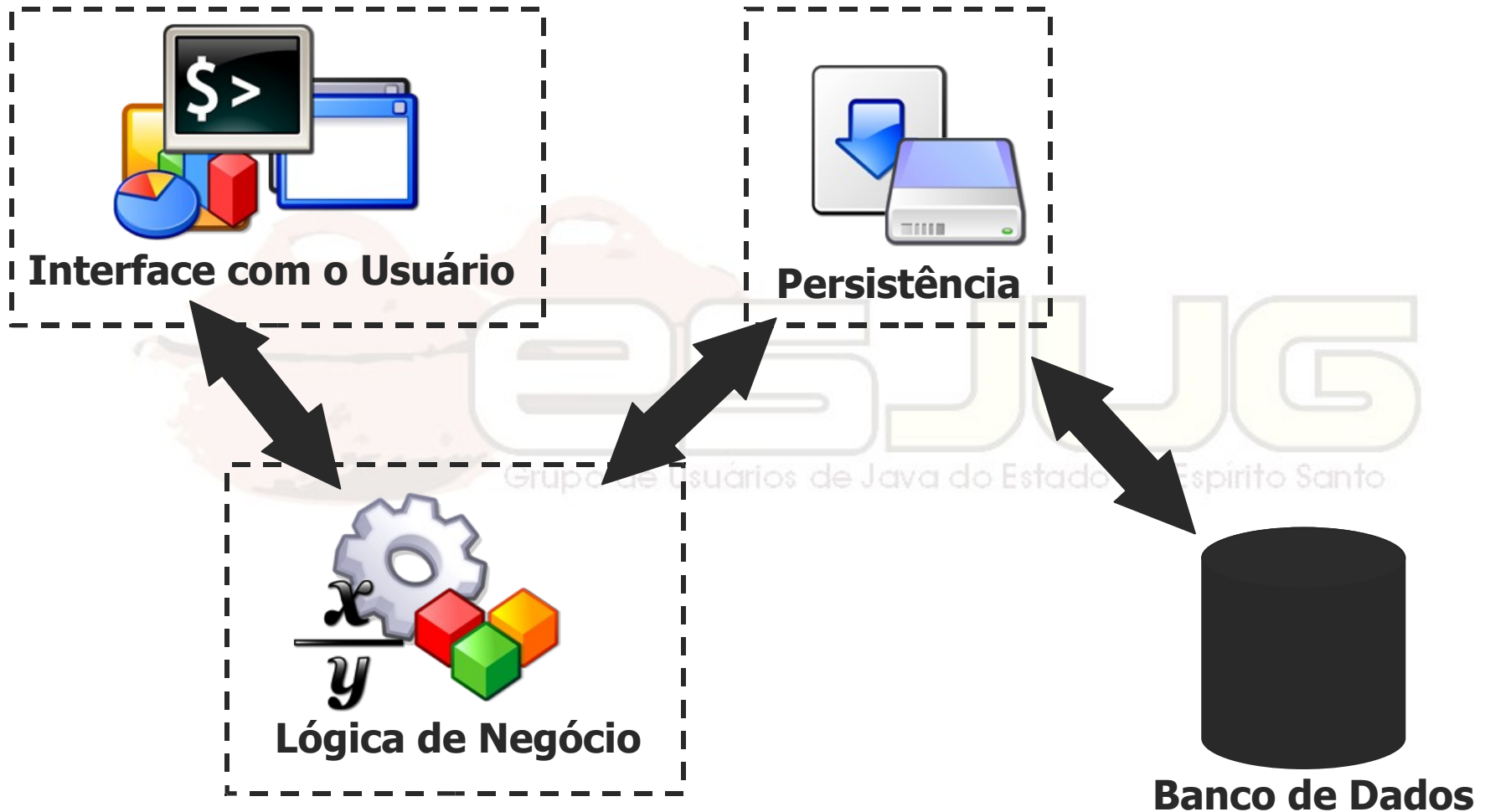


Grupo de Usuários de Java do Estado do Espírito Santo

Vantagens e desvantagens

- Sem complicações para desenvolver;
- Difícil manutenção:
 - Código desorganizado;
 - Difícil depuração;
 - Alterações em qualquer camada afetam todas as outras.
- Difícil reuso.

Sistema em várias camadas



Vantagens e desvantagens

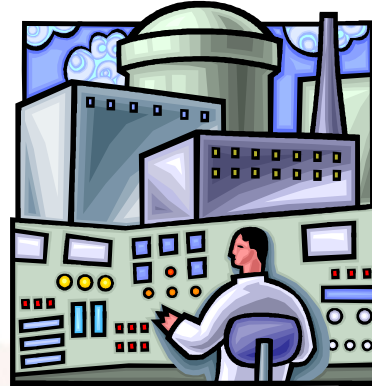
- Maior complexidade no desenvolvimento;
- Manutenção mais simples:
 - Código organizado;
 - Depuração isolada de camadas;
 - Alterações numa camada não afetam outras.
- Facilita o reuso.

[O MVC]

- Desenvolvido pelo Xerox PARC para o Smalltalk, em 1978;
- Objetivo:
 - mapear entrada-processamento-saída em GUIs para OO: controle-modelo-visão;
- Usado para:
 - Criação de componentes GUI reutilizáveis (proósito inicial);
 - Estruturação da aplicações (pós-Web).

MVC em GUIs

JSlider



Controller

2) Envia estímulos do usuário

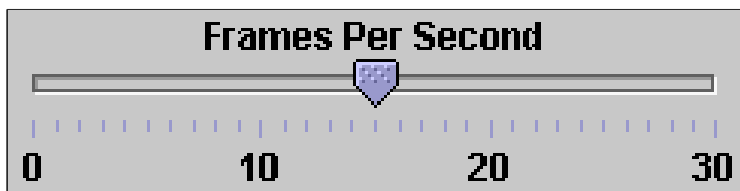
3) Altera os dados no modelo.

DefaultBoundedRangeModel

- minimum : int
- maximum : int
- value : int

+ getMinimum() : int
+ setMinimum() : int
+ getMaximum() : int
+ setMaximum() : int
+ getValue() : int
+ setValue() : int

Model

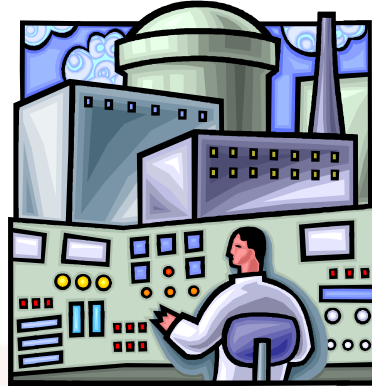


View

1) Usuário manipula componente gráfico

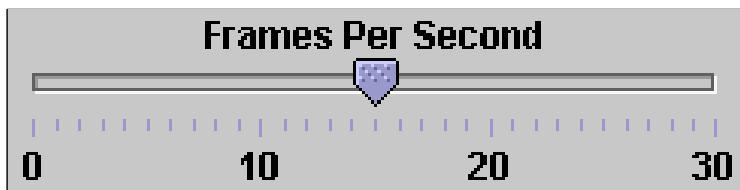
MVC em GUIs

JSlider



Controller

3) Atualiza seu estado, consultando os valores.



View

DefaultBoundedRangeModel

- minimum : int
- maximum : int
- value : int

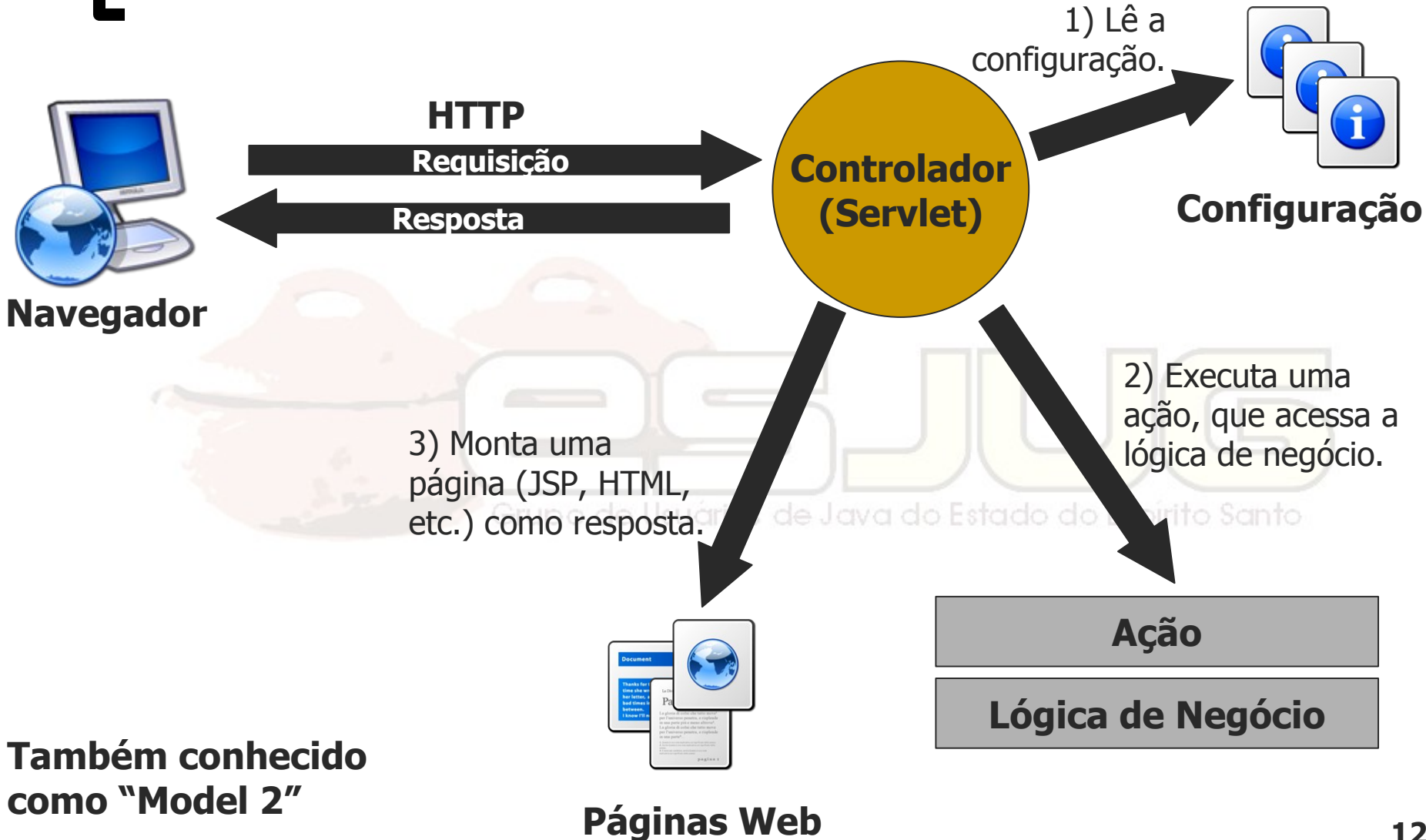
+ getMinimum() : int
+ setMinimum() : int
+ getMaximum() : int
+ setMaximum() : int
+ getValue() : int
+ setValue() : int

Model

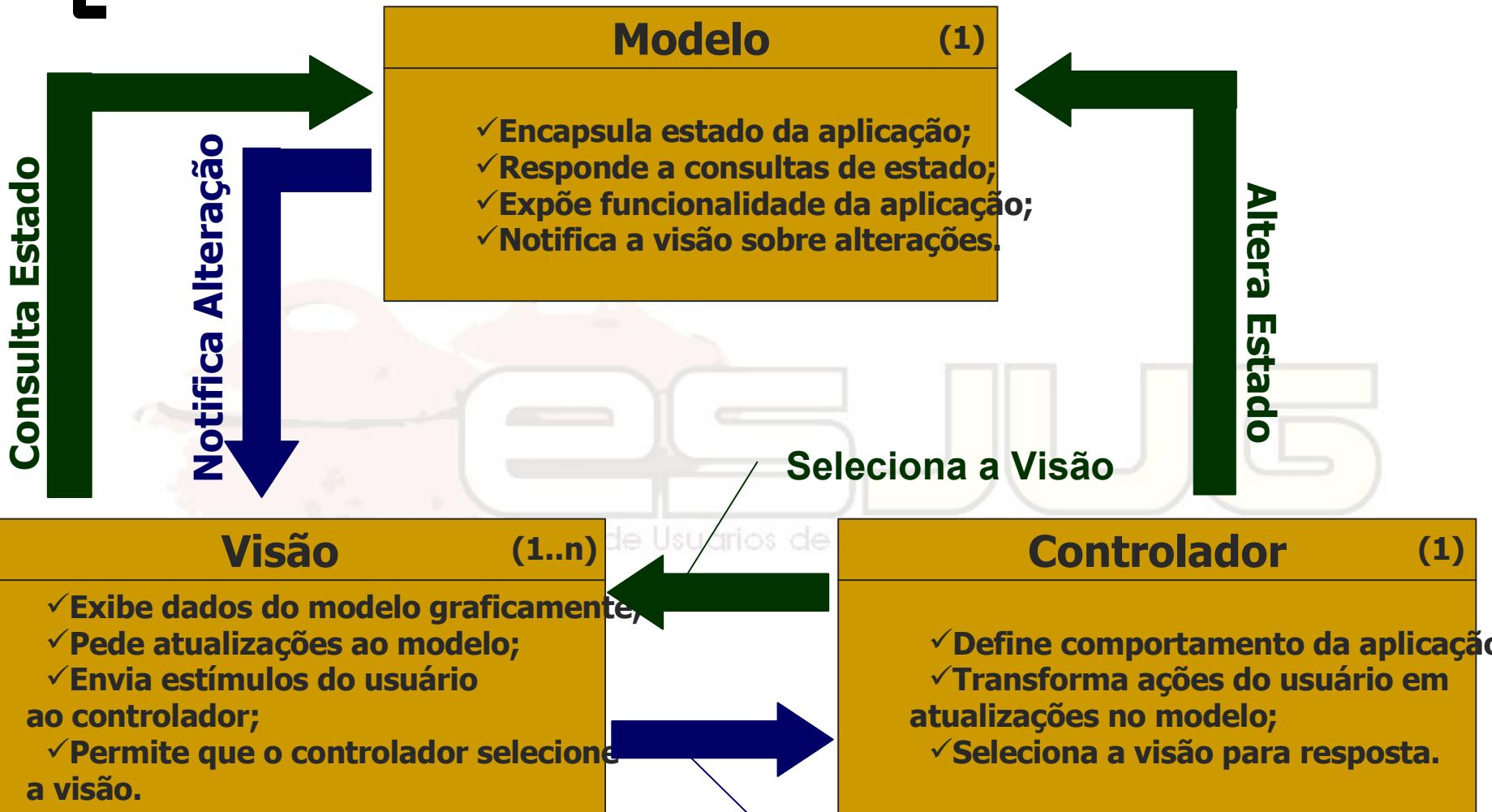
2) Notifica visão sobre a alteração

1) Aplicação altera dados no modelo.

MVC em Aplicações Web



Estrutura



MVC na API Swing

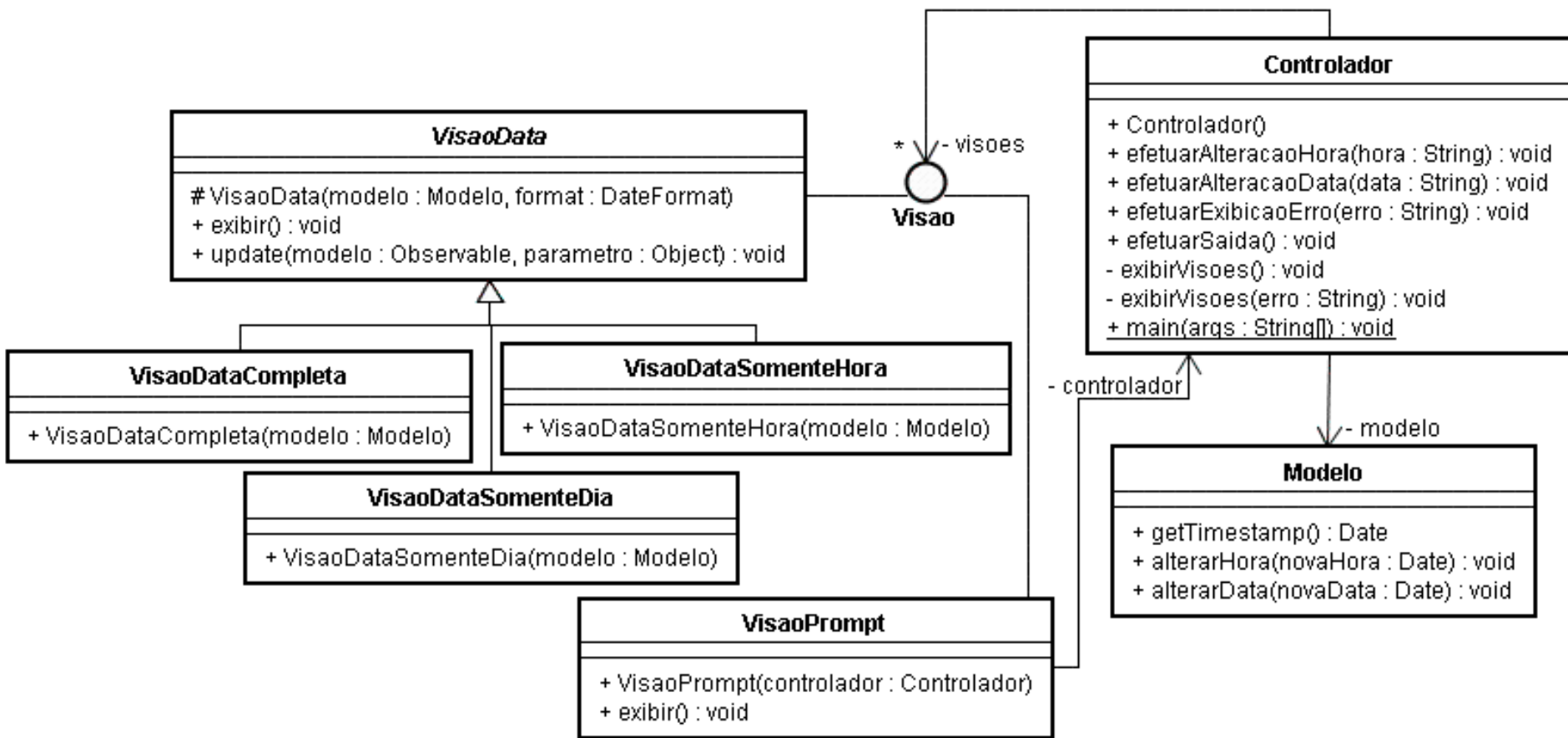
- Model:
 - classe que encapsula um valor que o componente gráfico representa.
- View:
 - classe que “renderiza” o componente gráfico no Look & Feel adequado.
- Controller:
 - classe que recebe estímulos do usuário no componente gráfico exibido e altera o modelo.

[MVC na Web]

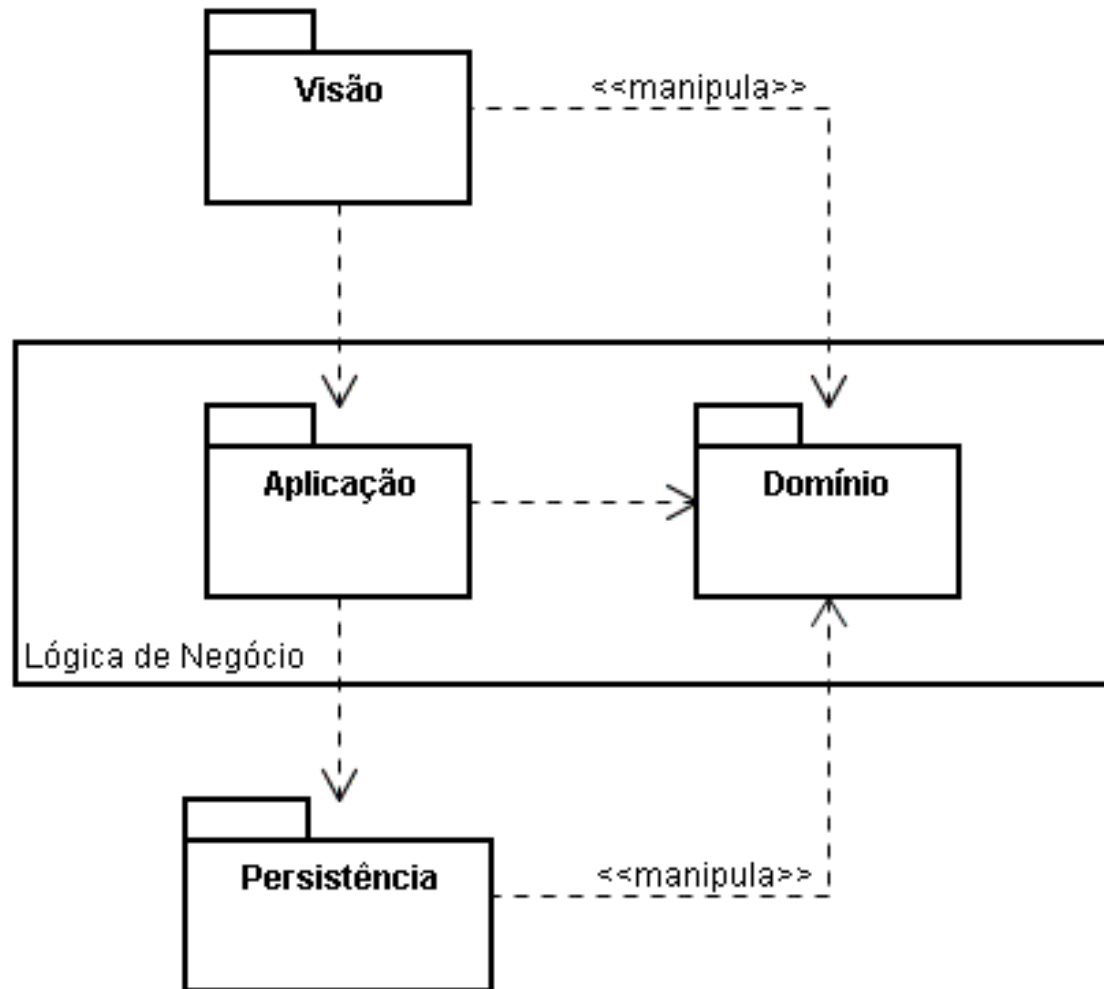
- **Model:**
 - classes de lógica de negócio.
- **View:**
 - páginas HTML, JSP e similares.
- **Controller:**
 - Servlet que recebe as requisições HTTP, chama algum método de negócio e, dependendo do retorno, escolhe uma view e redireciona.

Grupo de Usuários de Java do Estado do Espírito Santo

Demonstração



Arquitetura de 3 camadas



- Utilizada em sistemas *enterprise* ;
- Promove independência física e lógica;
- Também conhecido como "3-tier" ou "cliente magro".

Vantagens do MVC

- Facilita a manutenção do software;
- Simplifica a inclusão de um novo elemento de visão (ex.: cliente);
- Melhora a escalabilidade;
- Possibilita desenvolvimento das camadas em paralelo, se forem bem definidas.

Desvantagens do MVC

- Requer análise mais aprofundada (mais tempo);
- Requer pessoal especializado;
- Não aconselhável para aplicações pequenas (custo x benefício não compensa).

Frameworks Web para Java

- Diversos frameworks open-source implementam o MVC para a Web:
 - JSF (futuro padrão JCP);
 - Struts (Apache);
 - Tapestry (Apache);
 - WebWork (OpenSymphony);
 - Dentre outros...

Aplicações

- É importante conhecer o MVC para entender melhor a API Swing;
- Ao utilizar MVC para estruturar suas aplicações elas produzirá componentes reutilizáveis e de manutenção mais simples;
- MVC já se tornou um padrão para a camada Web, e é importante conhecê-lo para utilizar bem os frameworks.

Curso - Padrões de Projeto

Módulo 5: Model-View-Controller

Vítor E. Silva Souza
vitorsouza@gmail.com

<http://www.javablogs.com.br/page/engenh>

<http://esjug.dev.java.net>

