

Treinamento

Struts²

Professor: Maurício Giacomini Penteado

Conceitos

Clientes



Web-App

APIs

Servlet

HTTPServlet

JSP

Container

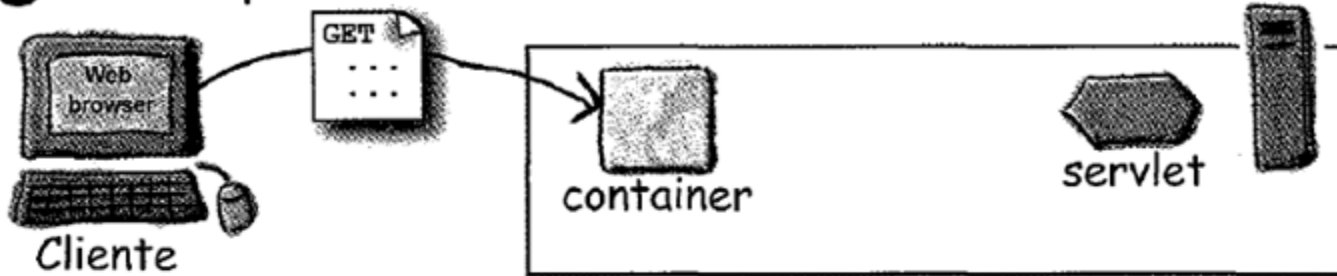


Apache
Tomcat

Processo de Solicitação

1

request HTTP



O usuário clica em um link que contém uma URL para um servlet, em vez de uma página estática.

2



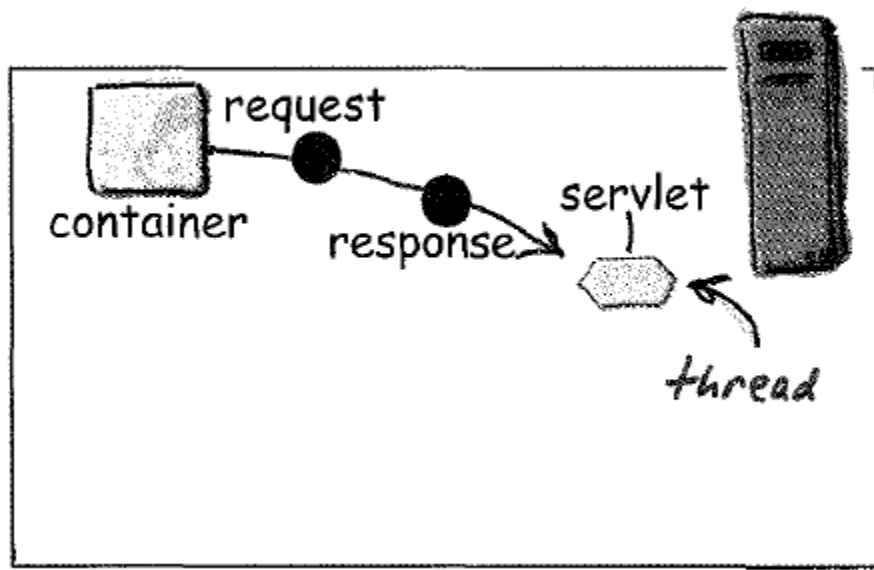
O container “vê” que a request é para um servlet e então ele cria dois objetos:

- 1) HttpServletResponse
- 2) HttpServletRequest

3



Cliente

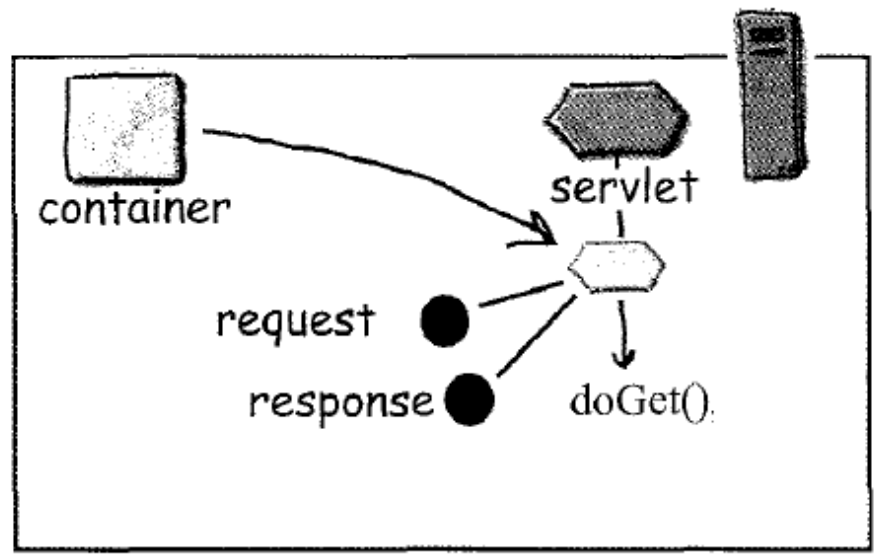


O container encontra o servlet correto baseado na URL da request, cria ou aloca uma thread para essa request, e passa os objetos request e response para a thread do servlet.

4

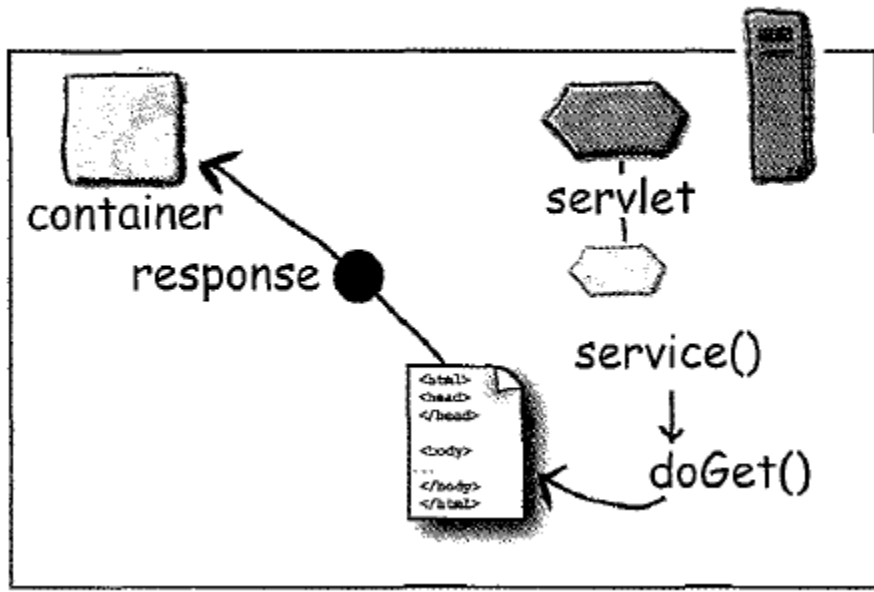


Cliente



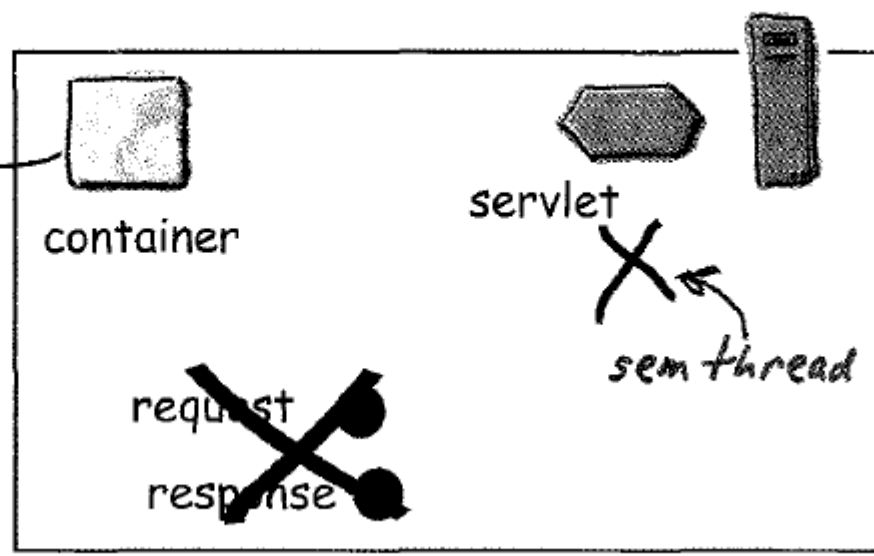
Dependendo do tipo de request, O container chama ou o método doGet(), ou o método doPost().

5



O método doGet gera uma página dinâmica e a insere no objeto response. Lembre-se, o container ainda tem uma referência do objeto response!

6



O thread termina, o container converte o objeto response em uma response HTTP, envia de volta ao cliente e apaga os objetos request e response.






Objetos do processo de solicitação:

- Headers
- Request
- Response
- Contexto
- Config
- Cookies (opcional)
- Sessions (opcional)



Nestes objetos podem ser armazenados e/ou consultados parâmetros e atributos

HttpServlet

- [-]  ExampleServlet :: HttpServlet
 -  doGet(HttpServletRequest request, HttpServletResponse response)
 -  doPost(HttpServletRequest request, HttpServletResponse response)
 -  getServletInfo() : String
 -  processRequest(HttpServletRequest request, HttpServletResponse response)

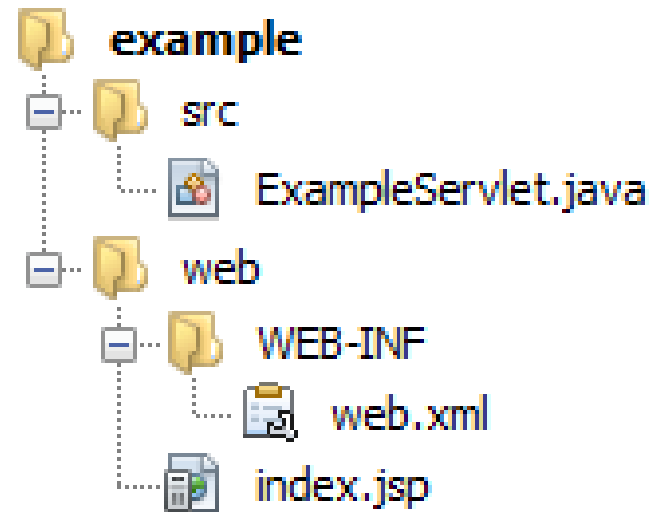
O que pode ser encontrado em um JSP

- Tags HTML
- Scriptlet: _____ `<% %>`
- Diretiva: _____ `<%@ %>`
- Expressão: _____ `<%= %>`
- Declaração: _____ `<%! %>`
- Comentário: _____ `<%-- --%>`
- Expression Language: _____ `${ }`
- Tags JSP: _____ `<jsp:xxx />`
- Tags JSTL: _____ `<c:xxx <fmt:xxx <sql:xxx ... />`
- Complementos ... JavaScript ... CSS ... Outros.

Web-Application – API Servlets 2.0

- No Web.xml pode-se:
 - Criar parâmetros init de contexto ou config
 - Declarar Listeners e Filters
 - Mapear Urls para Servlets
 - Configurar tempo da sessão
 - Configurar pag inicial
 - Aceitação de Scriplet
 - Restrições de segurança

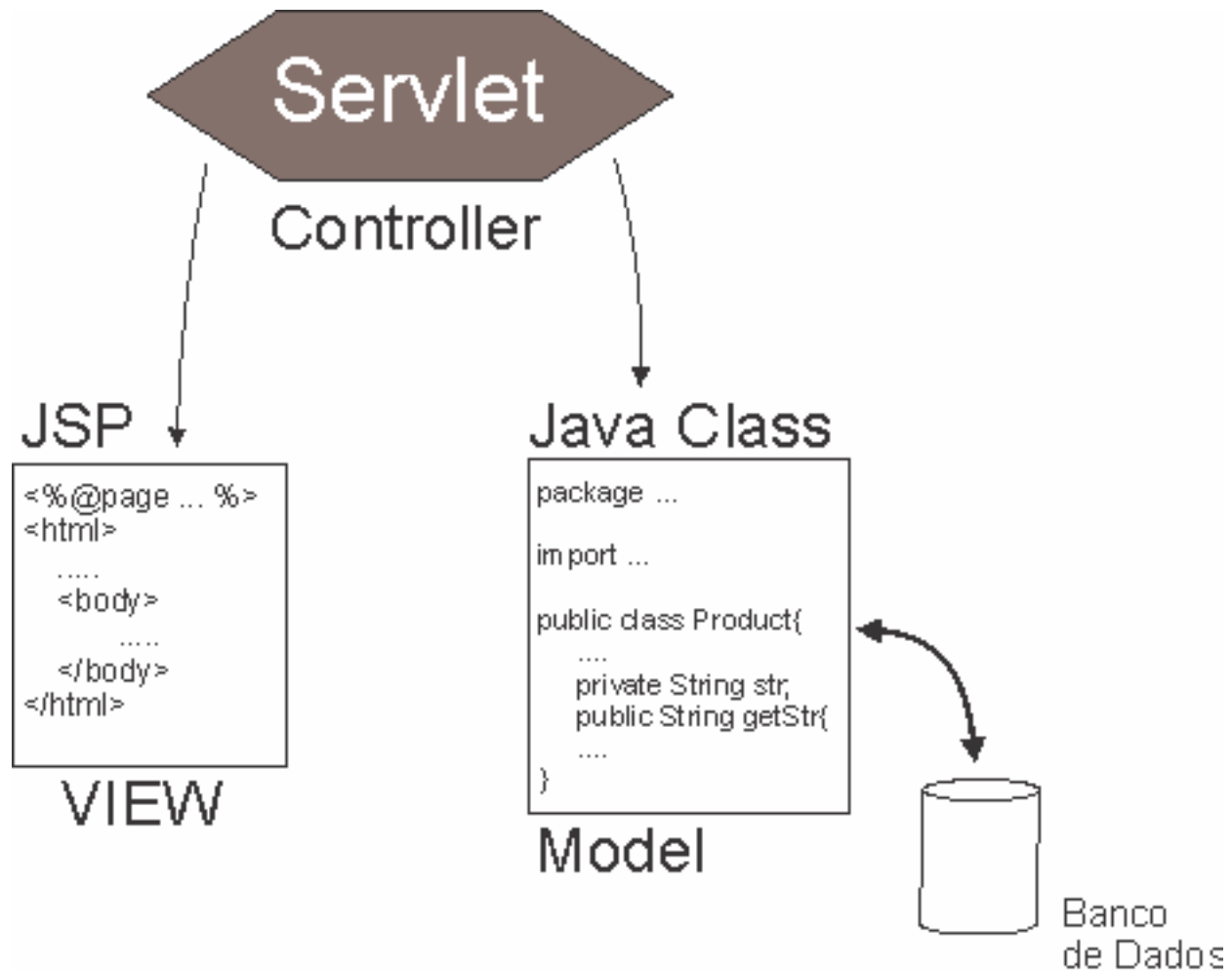
Infra-estrutura:



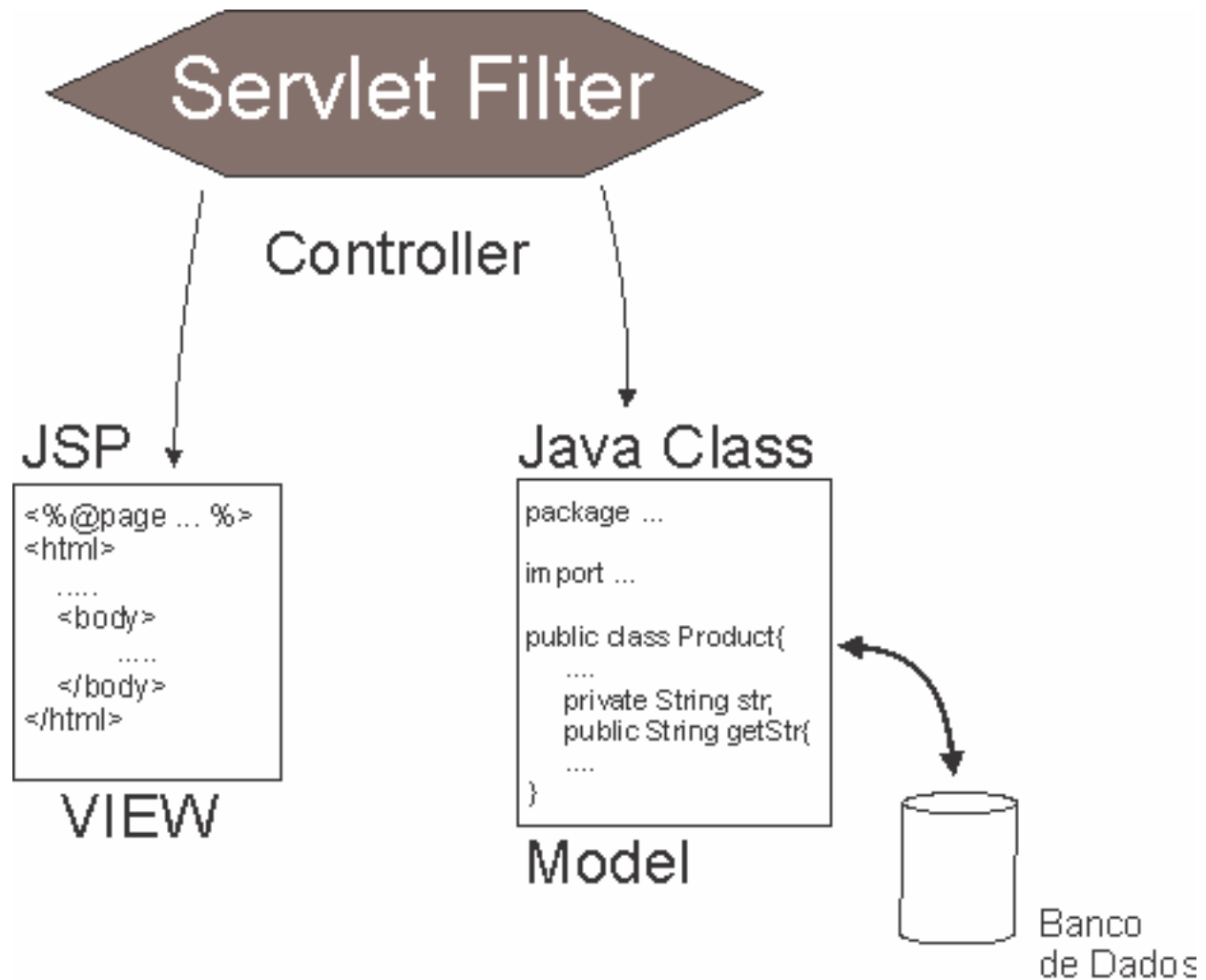
Aplicação compilada:



MVC – Model 1

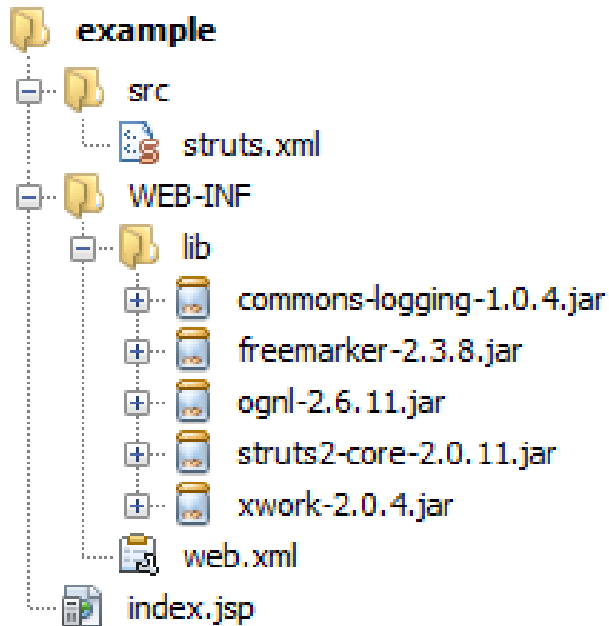


MVC – Model 2



Struts 2 – molde para Web-apps

MVC Model 2

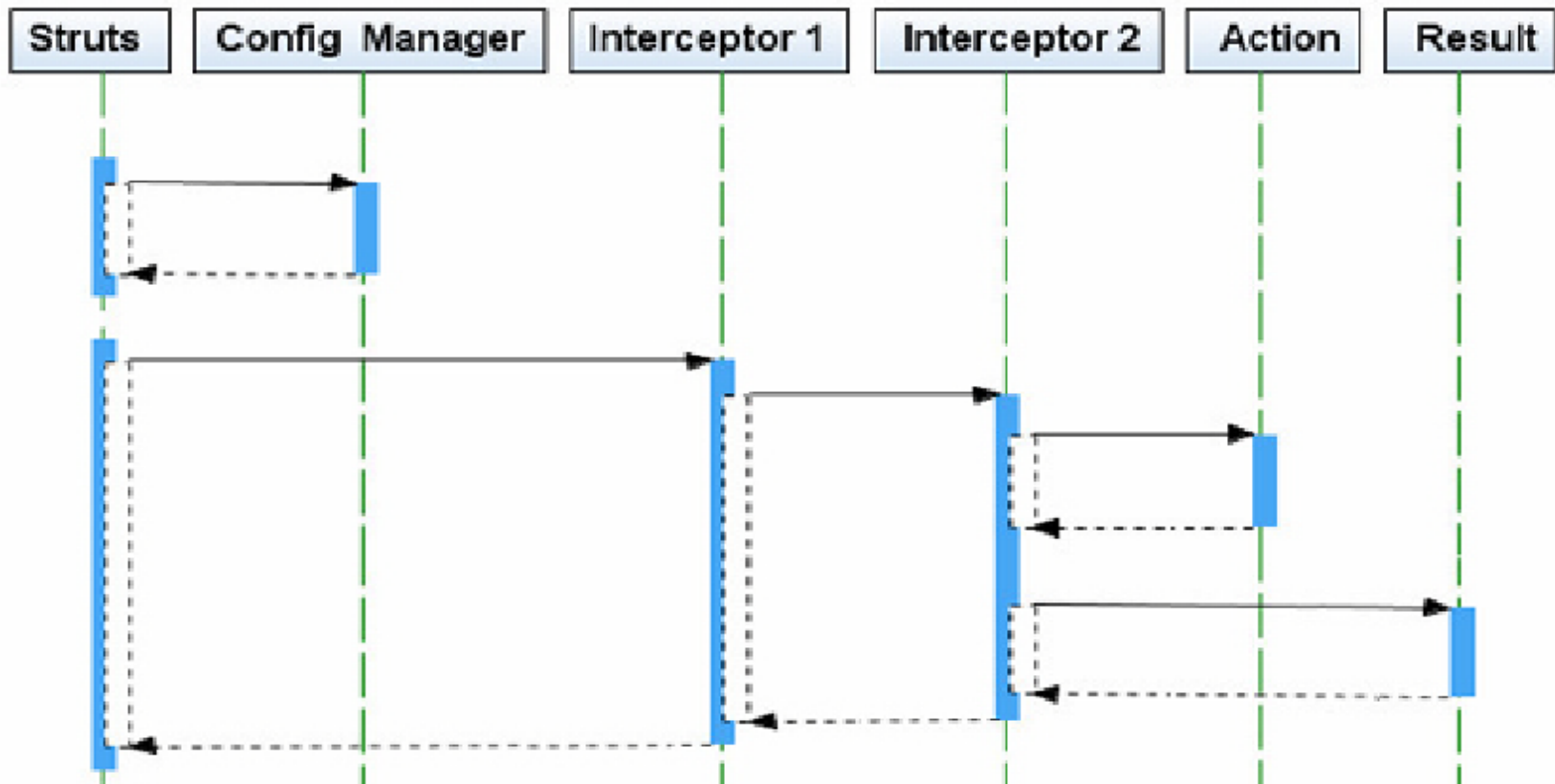


- O arquivo struts.xml estende web.xml, neste arquivo é efetuado principalmente o mapeamento entre Actions e ResultSets.
- Actions são basicamente HTTPServlets de controle porém, herdam classes do núcleo do framework Struts 2. Isso tende a facilitar a programação
- ResultSets definem o tipo de resposta a ser enviado de acordo com determinada Action executada. O uso de plugins do Struts variam bastante o tipo de ResultSet a ser criado.

Interceptors

Mapeados no arquivo web.xml, possuem processamento intercalado antes da invocação de uma Action.

Interceptors são Servlets/Filters padrões do framework Struts 2 ou compilados como extensões ao mesmo. (Mantendo fidelidade ao MVC model 2).



Actions

Actions recebem herança da classe **ActionSupport** no núcleo do framework.

Campos estáticos herdados:

- **SUCCESS**: Indica que a execução da ação foi bem sucedida e uma view de resultado deve ser mostrada para o usuário.
- **NONE**: Indica que a execução da ação foi bem sucedida, mas nenhuma view de resultado deve ser mostrado para o usuário.
- **ERROR**: Indica que falhou a execução da ação e uma exibição de erro deve ser enviada ao usuário.
- **INPUT**: Indica que a validação de um formulário de entrada de dados falhou e o formulário deve ser enviado ao usuário novamente.
- **LOGIN**: Indica que a ação não pode ser executada porque o usuário não foi autenticado. Assim, a view de login deve ser mostrada novamente.

Objetos Implícitos

- Struts 2 provém 4 interfaces para serem implementadas caso seja necessário acesso aos objetos implícitos da aplicação:

ServletContext

HttpServletRequest

HttpServletResponse

HttpSession.

- Essas interfaces são respectivamente:

ServletContextAware

ServletRequestAware

ServletResponseAware

SessionAware

OGNL – Object Graph Navigation Language

- Provém uma linguagem para JSPs que facilita o acesso aos objetos implícitos da aplicação, além de permitir a invocação de métodos nas classes de ações. Ex:

#application.code

#parameters.id

#request.getHeader('host')

@java.util.Calendar@DECEMBER

Form Tags

CheckBox
CheckBox2
CheckBoxList
ComboBox
DoubleSelect
Label
OptGroup

OptionTransfer
Select
Radio
Select
TextField
TextArea
UpDownSelect

Generic Tags

Data Tag:

Property
action
bean
date
debug
i18n
include
param
push
set
text
url

Control Tag:

if
elseif
else
append
generator
iterator
merge
sort
subset

Type Conversion

Funciona como um Framework interno ao Struts 2.

A classe que possui a lógica de conversão deve estender a classe **StrutsTypeConverter**.

A Conversão é útil para que dados enviados em um formulário possam ser pré-transformados em objetos antes do processamento de **actions** ou **views** de resposta.

Type Validation

Também funciona como um Framework interno ao Struts 2.

A **validação de campos** é codificada quando existe **em formulários** a necessidade de:

- **campos obrigatórios;**
- **campos inseridos de acordo com máscaras;**
- **tipo correto do campo;**
- **validações customizadas;**
- **entre outras (urls, emails, stringlength, etc ...).**

Internationalization

Com internacionalização a língua em que uma aplicação é apresentada ao cliente final pode ser modificada sem grandes alterações no código fonte da mesma.

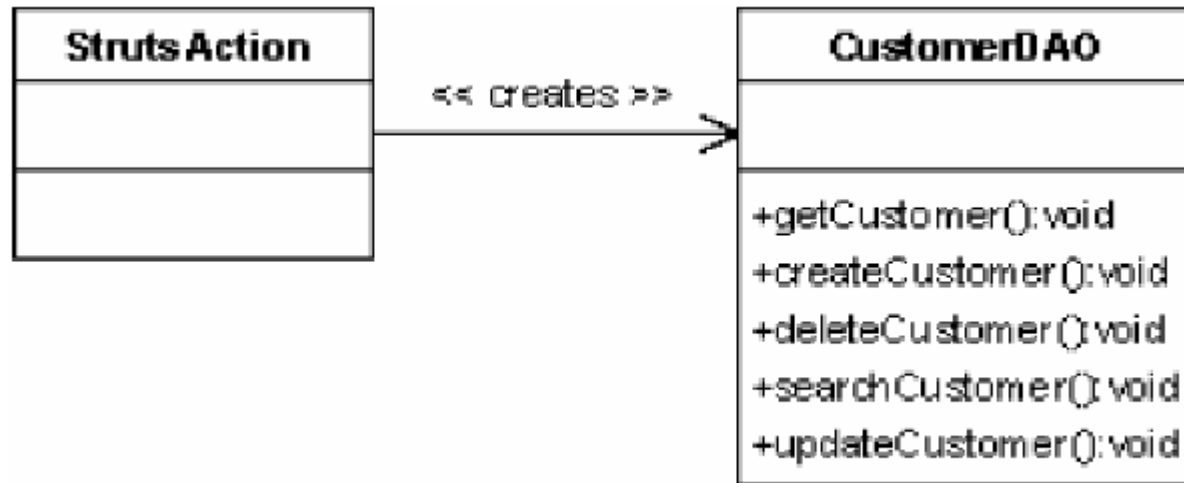
Model Driven e Prepare Interceptors

Model Driven: padrão de projeto que permite simular a camada de persistência de dados na memória (proporciona armazenamento e listagem dos dados).

Prepare Interceptors: padrão de projeto complementar ao Model Driven (proporciona edição, atualização e remoção dos dados).

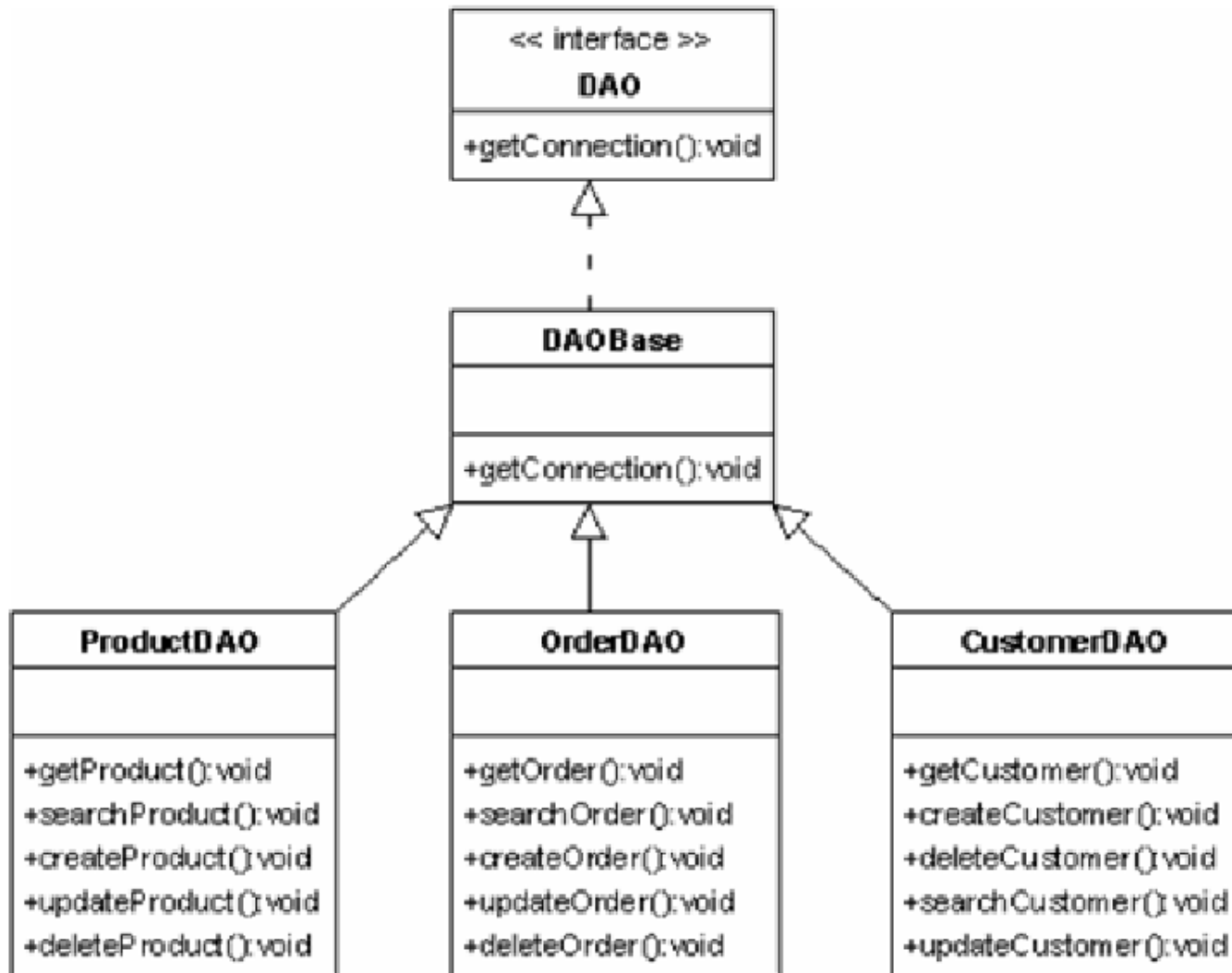
Persistência

Padrão de projeto DAO (Data Access Object)



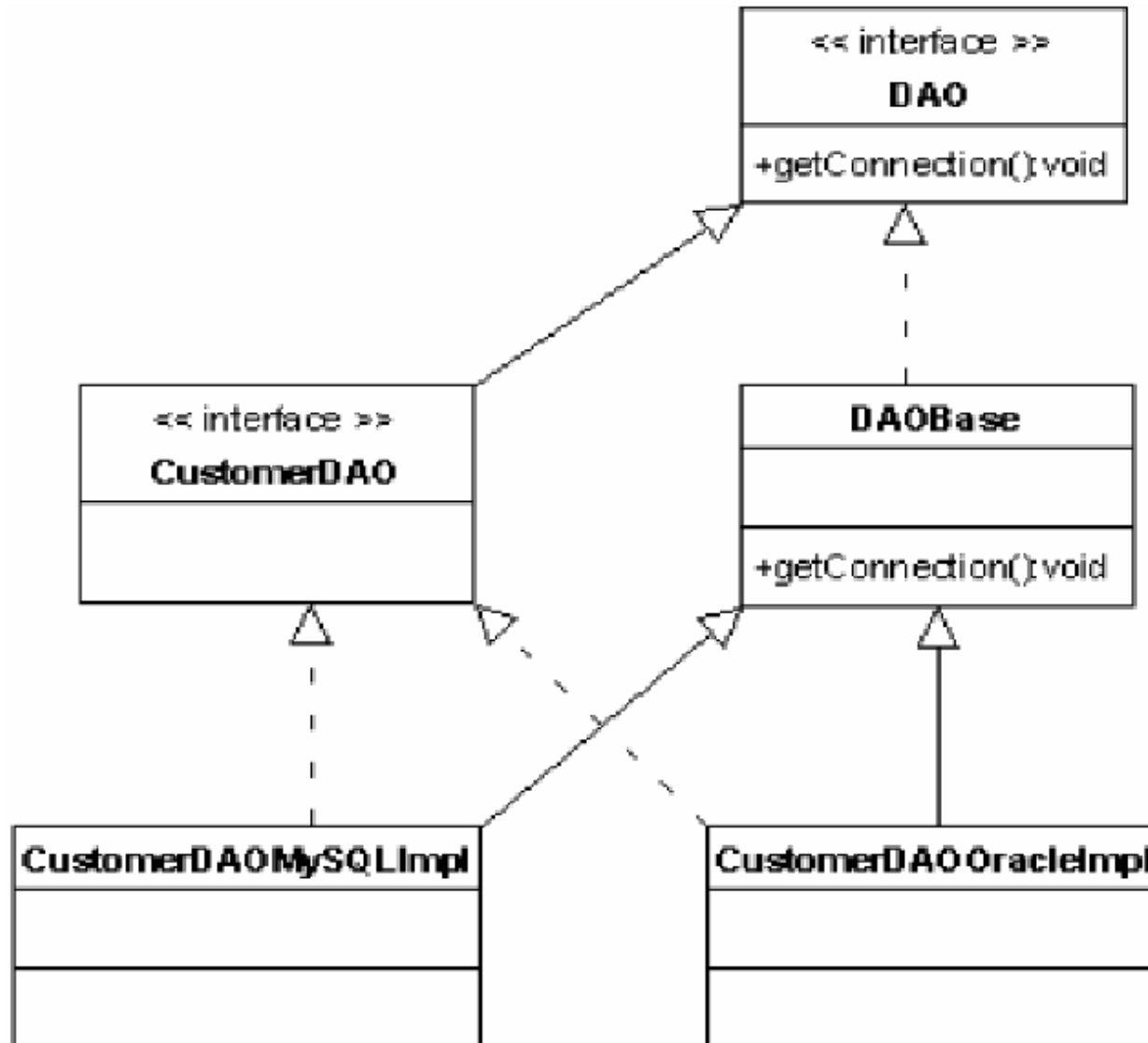
Persistência

Padrão de projeto DAO com interface DAO



Persistência

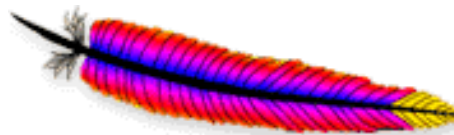
Padrão de projeto DAO com Abstract Factory



File Upload

Struts 2 File Upload Interceptor incorpora biblioteca Jakarta Commons FileUpload as aplicações.

Isso vem a permitir fácil desenvolvimento de páginas destinadas ao upload de arquivos.



Apache Commons

<http://commons.apache.org/>

File Download

Struts 2 provém fácil codificação quando existe a necessidade de disponibilizar arquivos para download.

Securing Struts Applications

Além das configurações para restrição de acesso a conteúdo que podem ser efetuadas no Tomcat e no arquivo web.xml relacionando grupos de usuários , usuários e senhas. Struts 2 também permite mapear por meio do arquivo struts.xml a permissão de execução das ações.

Preventing Double Submits

Normalmente acontecem acidentalmente por usuários que não sabem as informações requisitadas demoram a ser processadas.

Podem acarretar consequências drásticas em alguns sistemas Web.

A prevenção a Double Submits é prevista no framework Struts 2, recurso de fácil codificação.

Debugging and Profiling

A Tag `<s:debug>` e o console OGNL são auxiliares na codificação com o framework Struts 2.

Progress Meters

Struts 2 possui Interceptadores que permitem calcular a porcentagem de carga da página requisitada de forma progressiva.

Custom Interceptors

Interceptadores customizados podem ser criados para pré-processamento de certo código.

Como a conexão a um servidor de banco de dados por exemplo.

Custom Result Types

A customização de Result Types pode ser exemplificada com a codificação de sistemas de captcha em logins.

Velocity

Extensão para a
criação de Views por
meio de Templates

FreeMarker

Extensão para a
criação de Views por
meio de Templates

XSLT Results

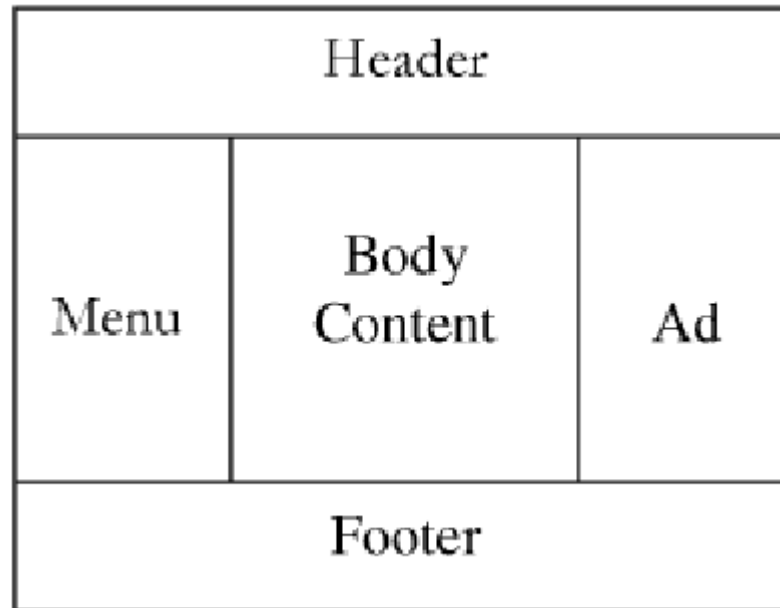
Por meio de XSLT Results a representação de objetos java em forma de arquivos XML é facilitada.

Plugins Customizados

A criação de plugins customizados no Struts 2 permite a reutilização de código.

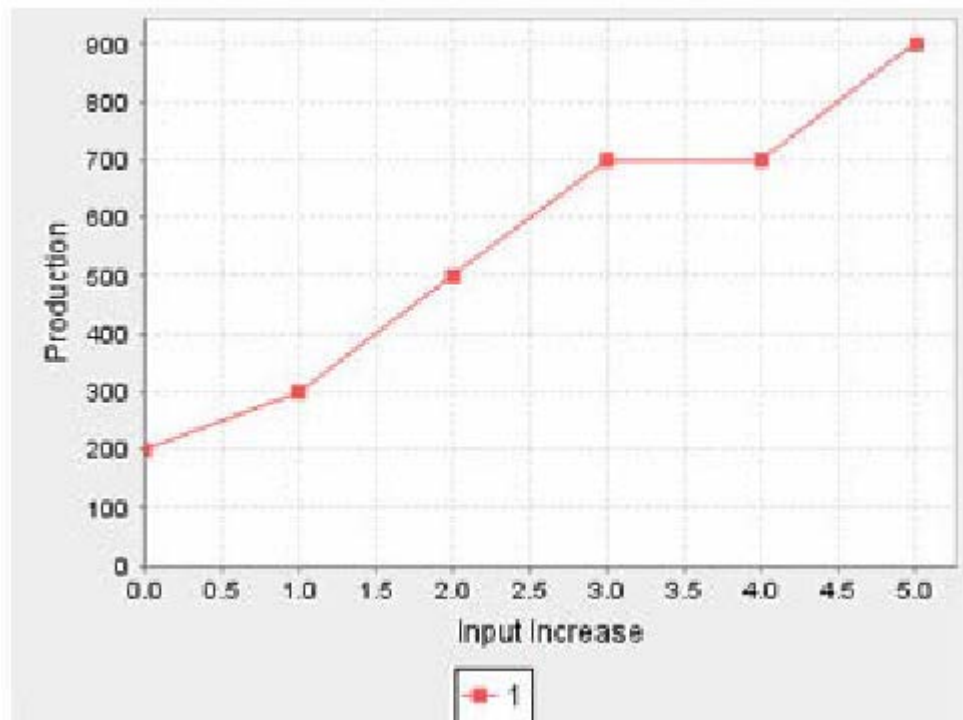
Tiles Plugin

Permite a edição do Layout de páginas JSP.



JFreeChart Plugin

Permite a geração de gráficos de acordo com valores a serem processados.



Zero Configuration

Permite o mapeamento de **actions** e **views** diretamente na codificação das classes, por meio de **anotations** (anotações). Assim o mapeamento no arquivo `struts.xml` não é mais necessário.

Componentes AJAX

Struts Dojo plugin permite que o framework Struts 2 possa manipular diversos componentes AJAX, o que pode enriquecer as **views** da aplicação.

Struts 2 Design and Programming: A Tutorial

Todos os exemplos e assuntos abordados nesse mini-curso podem ser encontrados em no livro **“Struts 2 Design and Programming: A Tutorial”**, vale a pena ler e aprender!!!

Aqui encerra o mini-curso.
Obrigado a todos!
Bons estudos!