



**PLATAFORMA DE DESENVOLVIMENTO PINHÃO PARANÁ**

**PADRÃO PARA ORGANIZAÇÃO FÍSICA DE ARQUIVOS EM**

**APLICAÇÕES JAVA WEB**

**Abril – 2010**

## Sumário de Informações do Documento

**Tipo do Documento:** Documento para Padronização.

**Título do Documento:** Padrão para Organização Física de Arquivos em Aplicações Java Web

**Estado do Documento:** Elaborado

**Responsáveis:** Cleverson Budel, Fábio Sgoda, Danielle Mayer

**Palavras-Chaves:** organização arquivos, gerência de configuração

**Resumo:** Descrição do padrão, utilizado na Celepar, para organização física das aplicações J2EE armazenadas nos sistemas de controle de versões (CVS/SVN).

**Número de páginas:** 7

**Software utilizado:** BrOffice Writer

<b>Versão</b>	<b>Data</b>	<b>Mudanças</b>
1.0	12/04/2010	<p>Desmembramento dos documentos "Padrão de Versionamento e Organização Física de Arquivos"</p> <p>O padrão de versionamento foi separado para fazer parte dos documentos da Metodologia de desenvolvimento da Celepar.</p> <p>O padrão para organização física de arquivos diz respeito apenas a tecnologia Java, por isso houve o desmembramento dos documentos.</p> <p>Alterada a estrutura de diretório dos arquivos devido a utilização da ferramenta Apache Maven.</p>

## **Sumário**

<b>1. INTRODUÇÃO.....</b>	<b>4</b>
<b>2.ORGANIZAÇÃO FÍSICA DE ARQUIVOS.....</b>	<b>4</b>
2.1 NOMENCLATURA DE PÁGINAS WEB.....	5

## 1. INTRODUÇÃO

Este documento visa detalhar a padronização adotada pela CELEPAR para organização física dos arquivos de uma aplicação Java Web.

A estrutura adotada é a mesma proposta pela ferramenta Apache Maven, amplamente utilizada pela comunidade de desenvolvedores em aplicações modernas.

Este é um padrão utilizado mundialmente que visa melhorar a organização de diversos tipos de aplicações escritas na Plataforma Java. Neste documento esta descrito apenas parte do padrão, utilizado em aplicações Java Web, no contexto da CELEPAR.

## 2. ORGANIZAÇÃO FÍSICA DE ARQUIVOS

A organização física, interna, dos arquivos da aplicação devem seguir a estrutura descrita na tabela abaixo:

Diretório	Descrição
src/main/java	Código fonte da aplicação. Arquivos com extensão .java (ponto java) organizados em pacotes. Seguindo o padrão de pacotes: <ul style="list-style-type: none"> <li>▪ gov.pr.[nome-do-cliente].[nome-do-pacote]</li> <li>▪ gov.pr.celepar.[nome-do-pacote]</li> </ul> Exemplos: <ul style="list-style-type: none"> <li>▪ gov.pr.celepar.reuso.cep</li> <li>▪ gov.pr.detran.veiculo.facade</li> <li>▪ gov.pr.detran.habilitacao.dao</li> </ul>
src/main/resources	Arquivos utilizados pela aplicação. Ex: hibernate.cfg.xml, arquivos.hbm.xml.
src/main/webapp	Estrutura raiz da aplicação web.
src/main/webapp/images	Arquivos de imagens.
src/main/webapp/META-INF	Arquivos de configuração de DataSources. Deve haver um para cada ambiente. Devem estar no formato:  <nome_aplicacao>-<desenv homolog prod>-ds.xml
src/main/webapp/css	Arquivos de folha de estilo (extensão css).
src/main/webapp/js	Arquivos JavaScript utilizados pelas páginas web da aplicação (extensão js).

src/main/webapp/pages	Páginas web (html ou jsp) da aplicação. Criar sub-diretórios dentro desta pasta visando separar-las logicamente.
src/main/webapp/reports	Arquivos utilizados para geração/apresentação de relatórios. Criar sub-diretórios dentro desta pasta visando separar-los logicamente.
src/main/webapp/WEB-INF	Arquivos descritores (de comportamento) da aplicação. Ex: web.xml, struts-config.xml.
src/test/java	Código fonte para testes dos componentes da aplicação. Ex: classes de testes unitários.
src/test/resources	Arquivos complementares utilizados em testes. Ex: arquivos contendo massa de dados utilizadas pelas classes de testes.

Para organização física dos arquivos referentes a documentação da aplicação, verificar o seguinte guia presente na Metodologia CELEPAR: *“Padrão de Versionamento de Aplicações e Organização Física de Arquivos”*.

## 2.1 Nomenclatura de páginas Web

Para a nomenclatura desses arquivos, deve-se utilizar a concatenação do tipo de página que está sendo construída, com as três primeiras letras da função da página, seguida do nome que identifica a sua funcionalidade. Não utilizar acentuação e descrever tudo em minúsculo na composição do nome.

A nomenclatura dos arquivos deverá seguir o padrão abaixo:

<b>Tipo de página</b>	<b>Função</b>	<b>Exemplo</b>
-----------------------	---------------	----------------

Páginas de formulário	Visualização e exclusão.	view_[nome-da-página].jsp
	Listagem	list_[nome-da-página].jsp
	Inclusão e Alteração	edit_[nome-da-página].jsp
	Inclusão	inc_[nome-da-página].jsp
	Alteração	alt_[nome-da-página].jsp
	Exclusão	exc_[nome-da-página].jsp
	Pesquisa com filtro ou Listagem com pesquisa	con_[nome-da-página].jsp  Esta nomenclatura será utilizada para telas que possuam filtro de pesquisa e visualização dos dados pesquisados em outra tela. Como é o caso das telas de seleção de relatório ou telas de listagem com filtro de pesquisa onde a listagem será exibida em outra tela.
Páginas de relatório	Relatório	rel_[nome-do-relatório].jsp
Páginas de controle	---	ctrl_[nome-do-controle].jsp
Páginas estáticas	Informação	[nome-da-página].html

**Nota1:** Nomenclatura de página [nome\_da\_página] - utilizar primeira letra minúscula e para cada palavra interna utilizar primeira letra maiúscula e demais letras minúsculas.

Ex: edit\_matriculaAluno.jsp

**Nota 2:** Listagem é um formulário onde é possível a interatividade com o conteúdo apresentado. Já no relatório o conteúdo é apresentado de forma estática, com o intuito de visualização e/ou impressão.

Quando houver uma funcionalidade que necessite ser executada em várias etapas utilizar a seguinte nomenclatura:

Tipo de página	Função	Exemplo
Páginas de formulário	Inclusão	inc_[nome_da_página][Nome_subfunção].jsp
		inc_[nome_da_página][Nome_subfunção].jsp
	Exclusão	exc_[nome_da_página][Nome_subfunção].jsp
		exc_[nome_da_página][Nome_subfunção].jsp

Exemplos:

inc\_matriculaAluno.jsp

inc\_matriculaTurma.jsp

Obs: se não for possível agrupar em subfunção utilizar numeração sequencial.