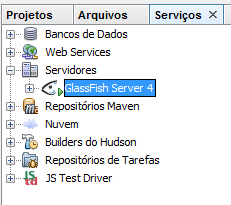
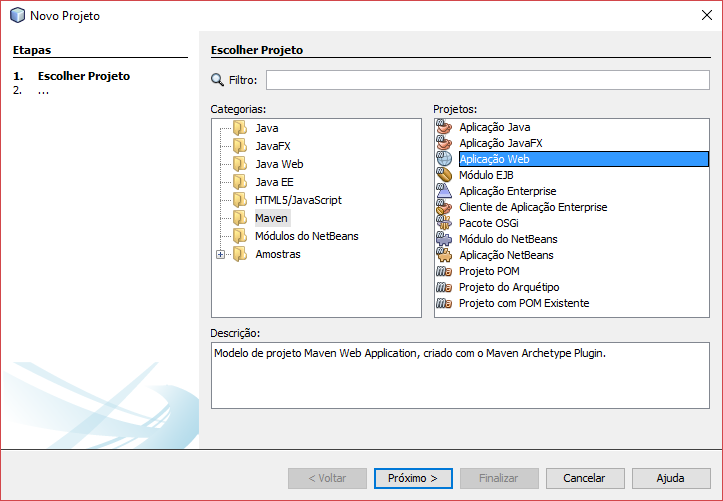
**Tutorial – Criando Aplicativos JSF**

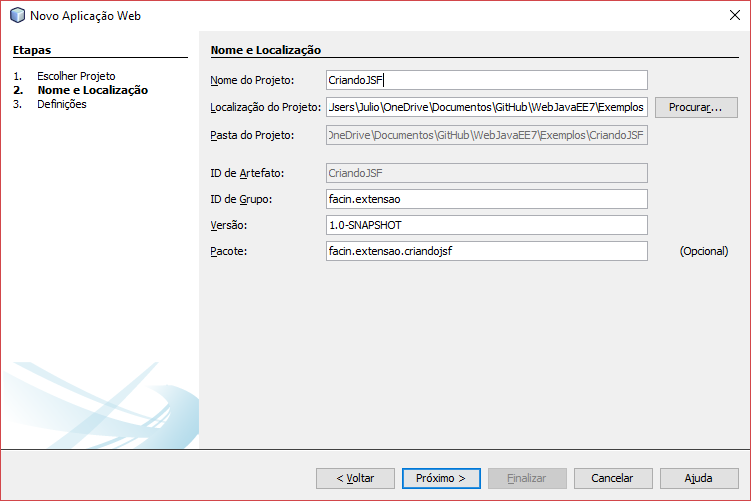
1. Inicie o NetBeans.

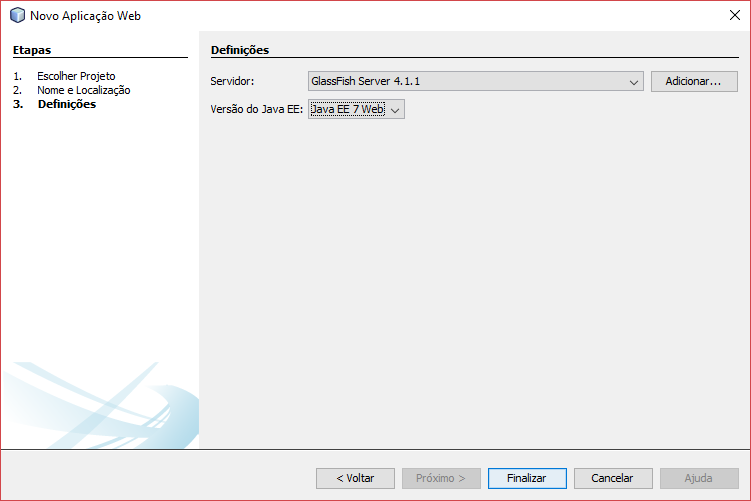
2. Inicie o servidor Glassfish local. Para tal, localize a aba “Serviços”, abra o nodo “Servidores” e, com o botão direito sobre o Glassfish, clique no menu “Iniciar”. Essa ação irá iniciar tanto o servidor Glassfish quanto o servidor de banco de dados local JavaDB.



3. Crie um novo projeto através do menu “Arquivo > Novo projeto”. Selecione a categoria “Maven” e, na lista de projetos disponíveis, selecione “Aplicação Web”. Clique “Próximo”. Dê o nome “CriandoJSF” para o projeto. Clique “Próximo”. Selecione o servidor “Glassfish” e “JavaEE 7 para Web”. Clique “Finalizar”.

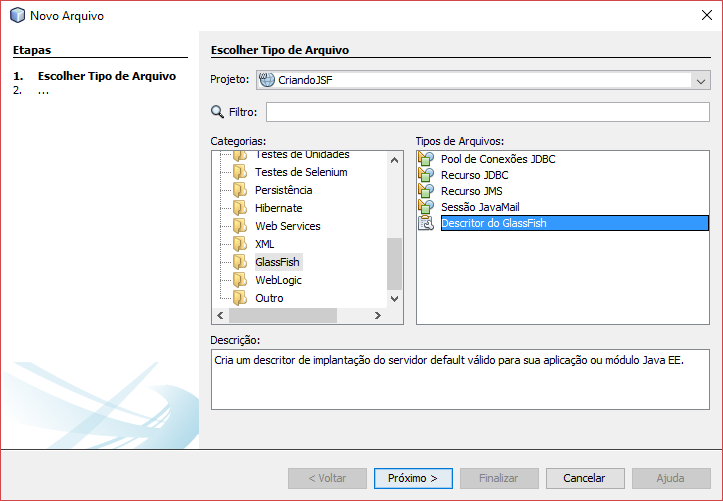






4. Localize e remova o arquivo “index.html” do projeto.

5. Acrescente um novo descritor de configuração do GlassFish para configurar o uso de Unicode UTF-8 como codificação de caracteres. Clique com o botão direito sobre o projeto e selecione o menu “Novo>Outros”. Localize a categoria “GlassFish” e selecione “Descritor do GlassFish”. Clique “Próximo”. Clique “Finalizar”.



6. Edite o arquivo “glassfish-web.xml” para que ele se pareça com o seguinte (mude para a visão adequada clicando no botão “XML” da janela de edição):

<?xml version="1.0" encoding="UTF-8"?>

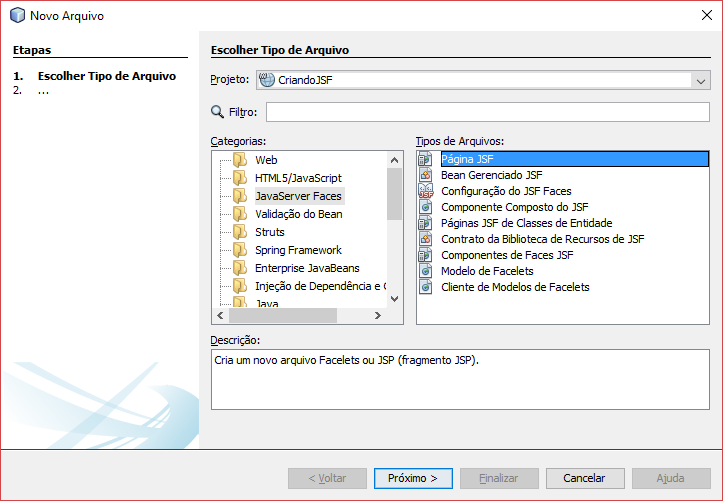
<!DOCTYPE glassfish-web-app PUBLIC "-//GlassFish.org//DTD GlassFish Application Server 3.1 Servlet 3.0//EN" "http://glassfish.org/dtds/glassfish-web-app\_3\_0-1.dtd">

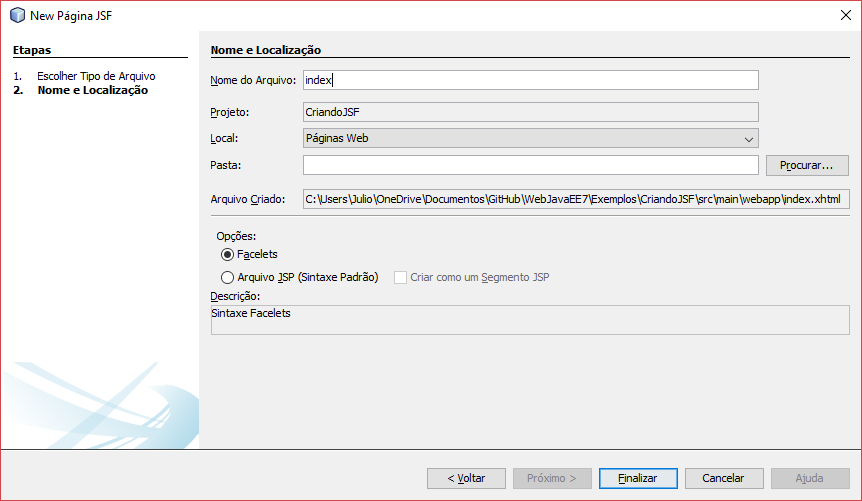
<glassfish-web-app error-url="">

<parameter-encoding default-charset="UTF-8"/>

</glassfish-web-app>

7. Adicione um novo arquivo *facelet* para a *view*. Clique com o botão direito sobre o projeto e selecione o menu “Novo>Outros”. Localize a categoria “JavaServer Faces” e selecione “Página JSF”. Clique “Próximo”. Utilize “index” para o nome do arquivo. Clique “Finalizar”.





8. Observe que foi criado mais um arquivo descritor “web.xml” com o mapeamento do *servlet* responsável pelo JSF.

9. Abra o arquivo “index.xhtml” e edite a importação de espaço de nome do documento XML para incluir a biblioteca de marcações do JSF Core. Utilize xmlns:f="http://xmlns.jcp.org/jsf/core" dentro da marcação <html>.

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://xmlns.jcp.org/jsf/html"

xmlns:f="http://xmlns.jcp.org/jsf/core">

10. Edite o corpo do documento (dentro da marcação *<h:body>*) para que ele apresente um formulário com uma caixa de texto (*inputText* com id “txtNome”) e uma caixa de seleção (*selectOneListBox* com id “lbEstadoCivil”). A página deve ser parecer com o código a seguir dentro do corpo do documento.

<h:form id="form1">

<h:inputText id="txtNome"/>

<br />

<h:selectOneListbox id="lbEstadoCivil" size="1">

<f:selectItem itemValue="Solteiro"/>

<f:selectItem itemValue="Casado"/>

<f:selectItem itemValue="Viuvo"/>

<f:selectItem itemValue="Separado"/>

</h:selectOneListbox>

</h:form>

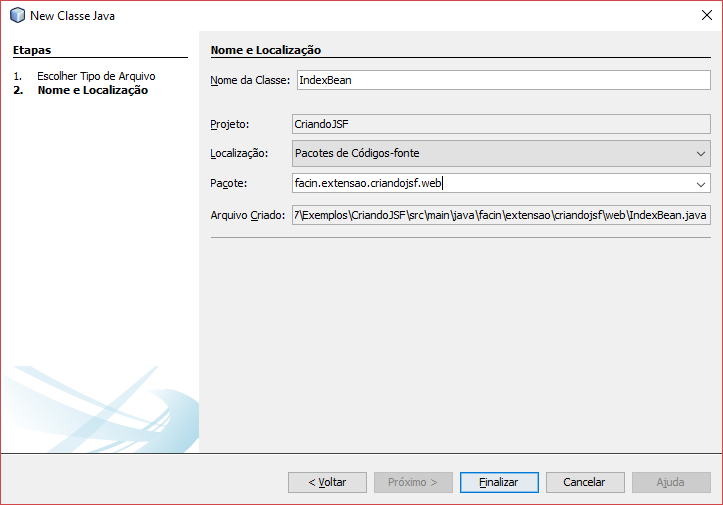
11. Adicione um botão (*commandButton* com id “btOk”) para submissão do formulário logo após a caixa de seleção. Cuidado para manter as marcações dentro do formulário.

<h:commandButton id="btOk" value="Ok"/>

12. Adicione um bloco de texto (*outputText* com id “txtResultado”) para apresentar o resultado do processamento do formulário.

<h:outputText id="txtResultado"/>

13. Adicione uma classe Java que funcionará como *managed bean* (um objeto “model”) para a página JSF, fornecendo propriedades para os valores dos respectivos componentes visuais da *view*. Clique com o botão direito sobre o projeto e selecione o menu “Novo > Classe Java”. Na janela de configuração, utilize “IndexBean” para o nome da classe e “facin.extensao.criandojsf.web” para o nome do pacote. Clique “Finalizar”.



14. Abra o código fonte da classe “IndexBean.java” e adicione as seguintes anotações à classe:

import javax.enterprise.context.RequestScoped;

import javax.inject.Named;

@Named

@RequestScoped

public class IndexBean {

}

15. Abra o código fonte da classe “IndexBean.java” e adicione propriedades para os valores dos componentes visuais utilizados na página JSF:

private String nome;

private String estadoCivil;

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getEstadoCivil() {

return estadoCivil;

}

public void setEstadoCivil(String estadoCivil) {

this.estadoCivil = estadoCivil;

}

public String getResultado() {

if((nome != null) && (estadoCivil != null))

return nome + " está " + estadoCivil;

else

return "";

}

16. Abra a página “index.xhtml”. Edite os atributos “value” dos componentes visuais com uma expressão EL correspondente à propriedade do *managed bean*:

txtNome - #{indexBean.nome}

lbEstadoCivil - #{indexBean.estadoCivil}

txtResultado - #{indexBean.resultado}

17. Compile, implante e execute a aplicação. Use as opções do menu flutuante clicando com o botão direito sobre o projeto.

