



PrimeFaces

AIRTON DA ROCHA BERNARDONI – SILVIA BERTAGNOLLI

Padrão MVC

- Model - Entidade
- View – Janela (visão)
- Controller - Controle

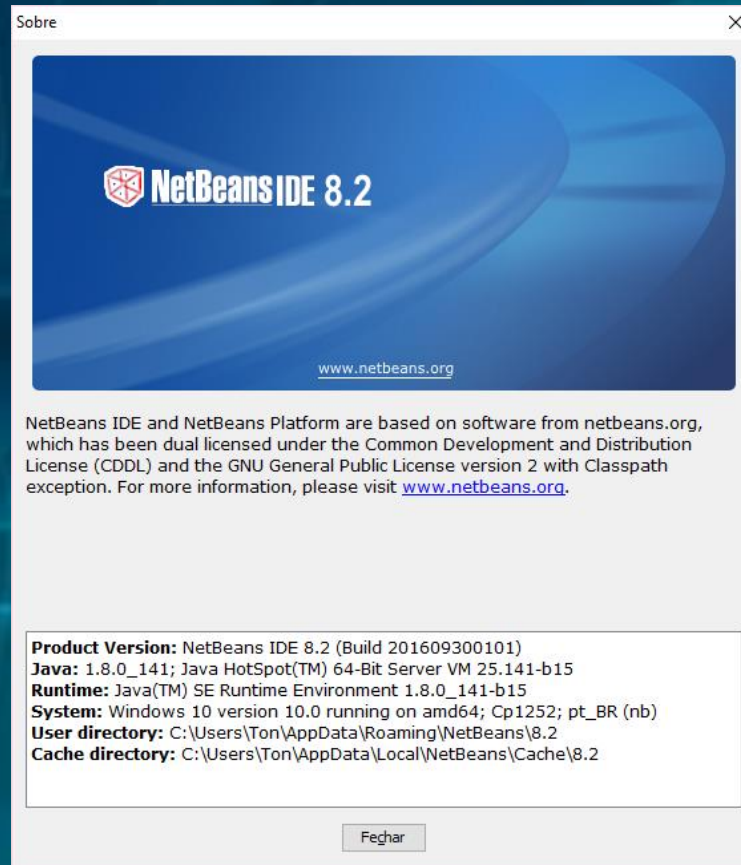
JSF

- Framework que implementa o MVC.
- Unifica operações de automação e personalização de frontend através de abstração do uso de JS e CSS.
- Contém um conjunto pré-fabricado de componentes para a utilização direta em Views (interface de usuário).
- Pode ser personalizado através de inserção de componentes de terceiros.
- É especificação para Java EE desde a versão 5
- IDEs já planejam seu uso para facilitar o trabalho de desenvolvedores.

PrimeFaces

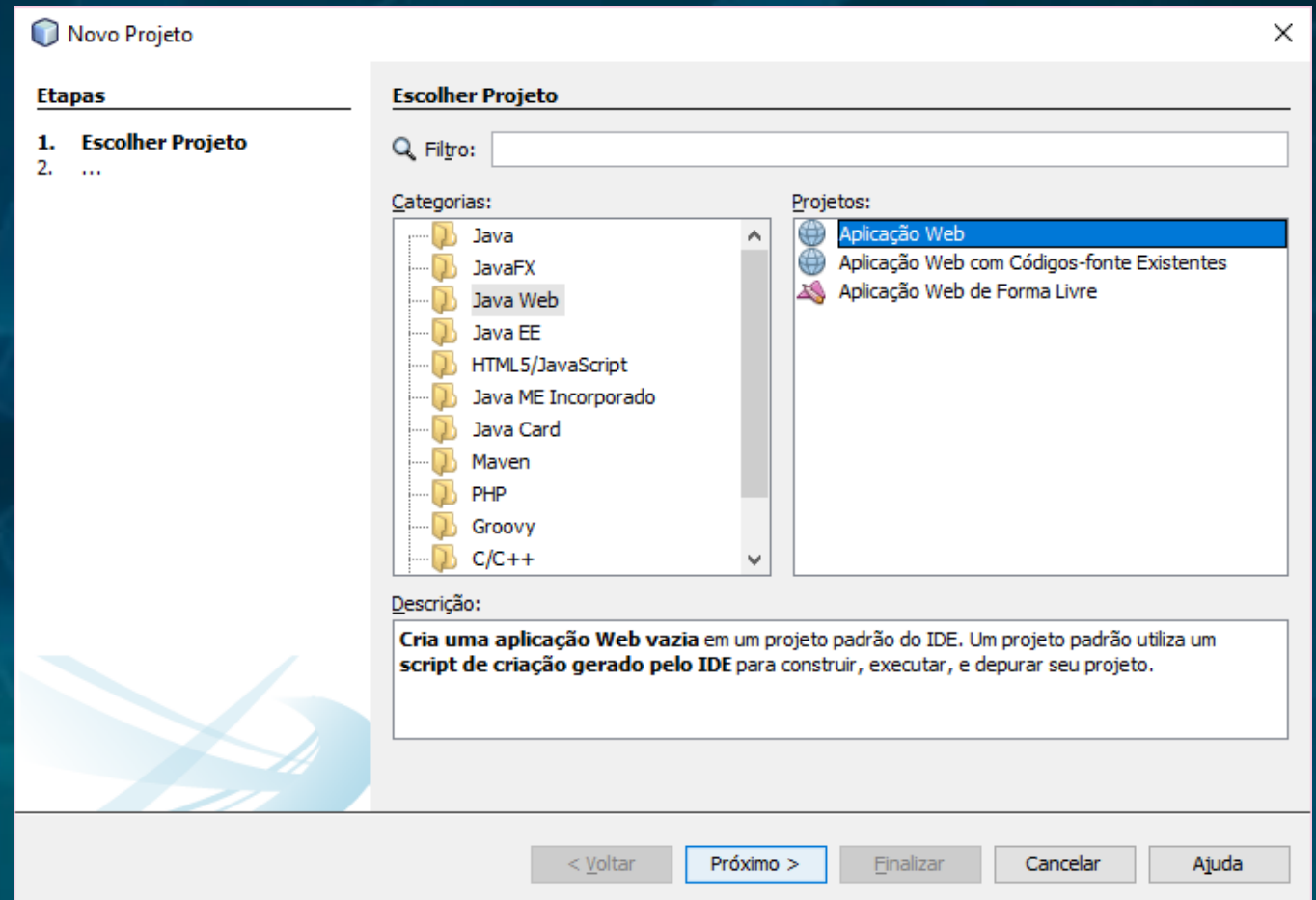
- Framework de JSF.
- Versão atual 6.14 (paga) e 6.1 (gratuita).

IDE utilizada – NetBean



Escolhendo projeto:

Aplicação Web



Definindo nome:

Novo Aplicação Web

Etapas

1. Escolher Projeto
- 2. Nome e Localização**
3. Servidor e Definições
4. Frameworks

Nome e Localização

Nome do Projeto:

Localização do Projeto: Procurar...

Pasta do Projeto:

☒ Usar pasta dedicada para armazenar bibliotecas

Pasta Bibliotecas: Procurar...

Usuários e projetos diferentes podem compartilhar as mesmas bibliotecas de compilação (consulte a Ajuda para obter detalhes).

< Voltar Próximo > Finalizar Cancelar Ajuda

Definindo Servidor:

Novo Aplicação Web

Etapas

1. Escolher Projeto
2. Nome e Localização
- 3. Servidor e Definições**
4. Frameworks

Servidor e Definições

Adicionar à aplicação corporativa: <Nenhum(a)>

Servidor: Apache Tomcat 8.0.27.0

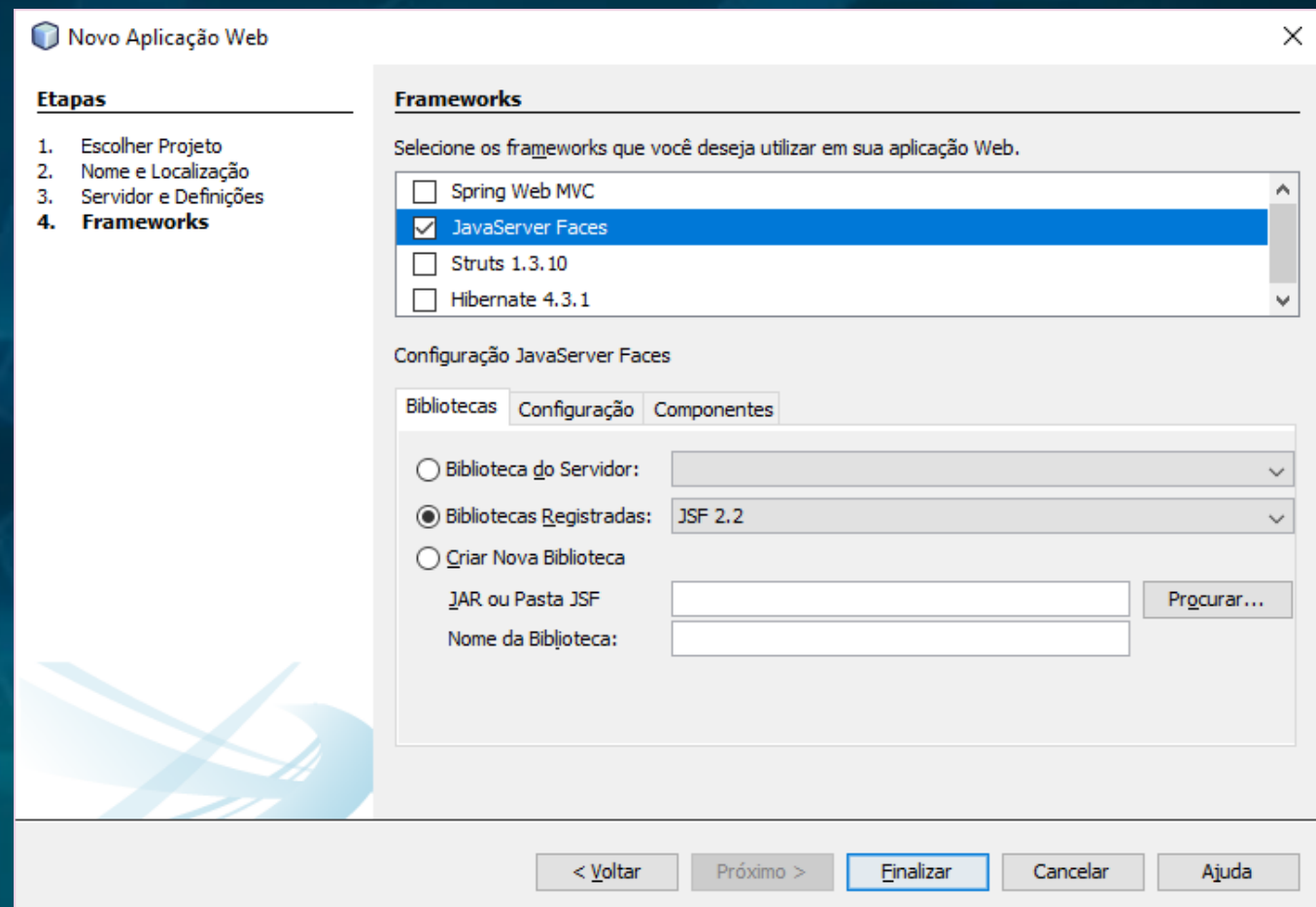
Versão do Java EE: Java EE 7 Web

Caminho do Contexto: /primefaces

< Voltar Próximo > Finalizar Cancelar Ajuda

Utilização do Framework

JavaServer Faces



Novo Aplicação Web

Etapas

1. Escolher Projeto
2. Nome e Localização
3. Servidor e Definições
- 4. Frameworks**

Frameworks

Selecione os frameworks que você deseja utilizar em sua aplicação Web.

- ☐ Spring Web MVC
- ☒ **JavaServer Faces**
- ☐ Struts 1.3.10
- ☐ Hibernate 4.3.1

Configuração JavaServer Faces

Bibliotecas | Configuração | Componentes

☐ Biblioteca do Servidor:

☒ Bibliotecas Registradas: JSF 2.2

☐ Criar Nova Biblioteca

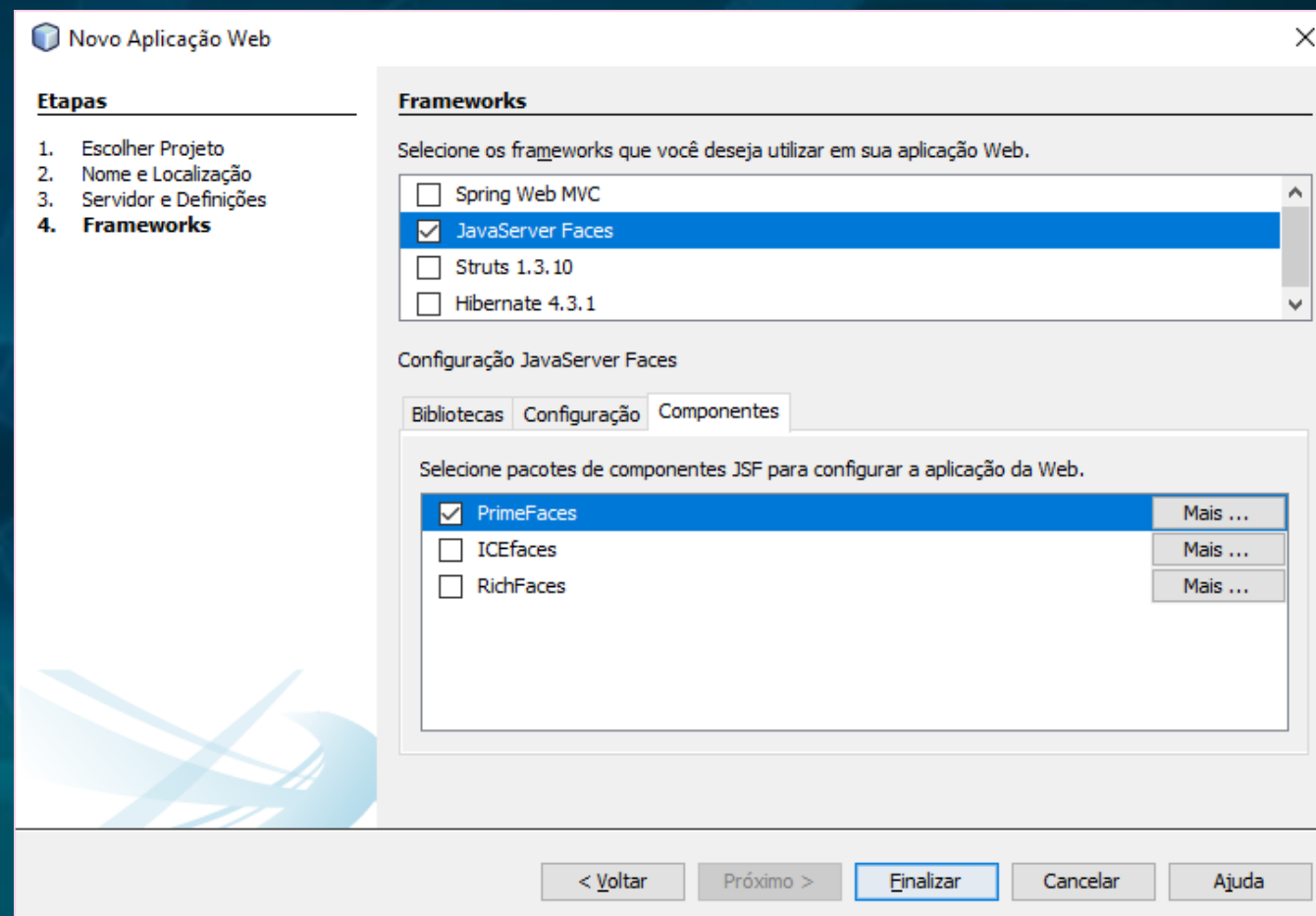
JAR ou Pasta JSF: Procurar...

Nome da Biblioteca:

< Voltar Próximo > **Finalizar** Cancelar Ajuda

Escolha do Componente

PrimeFaces



Novo Aplicação Web

Etapas

1. Escolher Projeto
2. Nome e Localização
3. Servidor e Definições
- 4. Frameworks**

Frameworks

Selecione os frameworks que você deseja utilizar em sua aplicação Web.

- ☐ Spring Web MVC
- ☒ JavaServer Faces
- ☐ Struts 1.3.10
- ☐ Hibernate 4.3.1

Configuração JavaServer Faces

Bibliotecas Configuração Componentes

Selecione pacotes de componentes JSF para configurar a aplicação da Web.

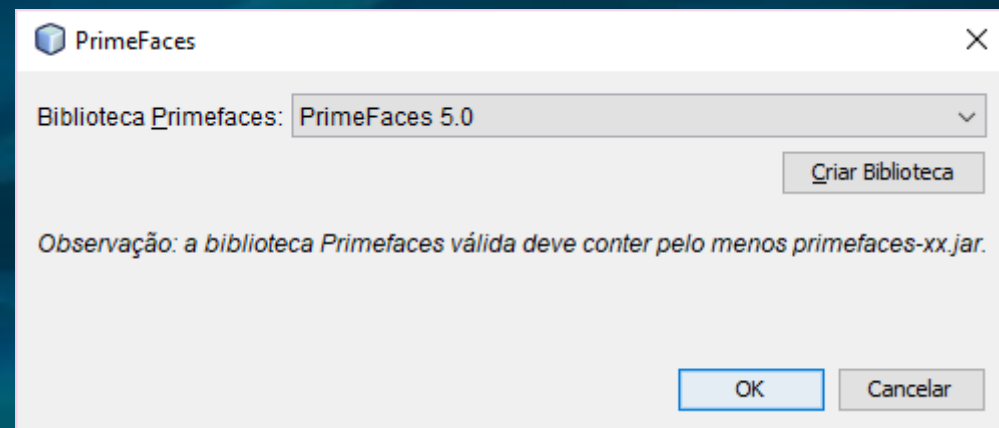
- ☒ PrimeFaces Mais ...
- ☐ ICEfaces Mais ...
- ☐ RichFaces Mais ...

< Voltar Próximo > Finalizar Cancelar Ajuda

Escolha da Biblioteca

Disponível no NetBeans: 5.0

Disponível no Site: 6.1



Conteúdo do web.xml

```
<context-param>
  <param-name>javax.faces.PROJECT_STAGE</param-name>
  <param-value>Development</param-value>
</context-param>
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>
    30
  </session-timeout>
</session-config>
<welcome-file-list>
  <welcome-file>faces/index.xhtml</welcome-file>
</welcome-file-list>
```

Estrutura do arquivo xhtml

Dica: Utilize tags de namespaces (não nativas do html) como se sua vida dependesse disso!!!

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html">
    <h:head>
        <title>Facelet Title</title>
    </h:head>
    <h:body>
        Hello from Facelets
        <br />
        <h:link outcome="welcomePrimefaces"
            value="Página de boas-vindas do Primefaces" />
    </h:body>
</html>
```

Namespaces

- `xmlns="http://www.w3.org/1999/xhtml"`
- `xmlns:h="http://java.sun.com/jsf/html"`
- `xmlns:f="http://java.sun.com/jsf/core"`
- `xmlns:ui="http://java.sun.com/jsf/facelets"`
- `xmlns:p="http://primefaces.org/ui"`

Exercícios

Estrutura MVC

- Classe POJO (Model) em pacote model ou modelo
- Classe Controle / Servico (Control) em pacote control ou controle
- Pasta com nome do item dentro da pasta raiz de Páginas Web.

Model

- public class Filme {
- private int ano;
- private String nome;
- private String descricao;
- // Construtores
- // Getters e Setters (obrigatório para uso em Beans)
- }

Control

- @ManagedBean
- @ViewScoped
- public class FilmeControle implements Serializable {
 - private List<Filme> lista;
 - private List<Filme> assistidos;
 - private DualListModel<Filme> listaODI;
 - private Filme filme;
 - ...

Control

...

@PostConstruct

public void init() {

 lista = new ArrayList();

 assistidos = new ArrayList();

 addLista(1960, "Psicose", "A obra-prima de Alfred Hitchcock.");

 addLista(1973, "O Exorcista", "Lançado em 1973.");

 addLista(1974, "O Massacre da Serra Elétrica", "É de se imaginar que pessoas que tiveram a oportunidade de assistir.");

 addLista(1976, "A Profecia", "Quando o remake de A Profecia foi lançado, a pré-estreia mundial aconteceu em seis de junho de 2006."); }

...

Control

...

```
public void addLista(int ano, String nome, String descricao) {  
    Filme filme = new Filme();  
    filme.setAno(ano);  
    filme.setNome(nome);  
    filme.setDescricao(descricao);  
    this.lista.add(filme);  
}
```

...

Control

```
...
```

```
// Getters e Setters
```

```
}
```

DICA

- Em Classes gerenciadas (Managed Beans) recomenda-se ter construtores vazios e getters and setters para todos os termos para evitar problemas durante o projeto.
- Não quer dizer que seja obrigatório que se ponha em todos, mas caso não haja, os atributos não poderão ser acessados pela camada view (XHTML).
- Pode-se definir nomes significativos para os atributos ou métodos através da anotação `@ManagedProperty`

PickList

```
<p:pickList  
    id="idHtml"  
    value="#{itemControle.listaODI}"  
    var="item"  
    itemLabel="#{item.nome}"  
    itemValue="#{item.nome}" />
```

PickList

- Dependências:
 - Atributo DualListModel<> no controle (ou bean);
 - Mínimo de duas listas para compor o item anterior;
 - Caso deseje reação a eventos, os Métodos:
 - onTransfer(TransferEvent event)
 - onSelect(SelectEvent event)
 - onUnselect(UnselectEvent event)
 - onReorder()

Carousel

```
<p:carousel  
    value="#{itemControle.lista}"  
    headerText="Custom"  
    var="item"  
    itemStyle="height:200px;text-align:center"  
    numVisible="1"  
    easing="easeOutBounce"  
    style="width:250px"  
    circular="true">
```

PanelGrid

```
<p:panelGrid  
    columns="2"  
    style="width:100%; margin:10px 0px"  
    columnClasses="label,value"  
    layout="grid"  
    styleClass="ui-panelgrid-blank">
```


Comand Button

```
<p:commandButton  
    value="Descrição"  
    actionListener="#{itemControle.feedBack()}"  
    update="message">  
    <p:confirm  
        header="O filme é bom?"  
        message="#{filme.descricao}" />  
    </p:commandButton>
```

TableEdit

```
<p:commandButton  
    value="Descrição"  
    actionListener="#{itemControle.feedBack()}"  
    update="message">  
    <p:confirm  
        header="O filme é bom?"  
        message="#{filme.descricao}" />  
    </p:commandButton>
```

Fonte de toda inspiração:

- <https://www.primefaces.org/showcase/>
- <https://www.primefaces.org/showcase/mobile/index.xhtml>