



# FRAMEWORK GERAÇÃO DE RELATÓRIOS “ JASPER REPORT ”



## INTEGRAÇÃO COM O JSF



# Equipe

---

- ❖ Carlos Maciel
- ❖ Eduardo Matheus
- ❖ Gustavo Costa
- ❖ Ramon Silva
- ❖ Sergio Roberto



# Links

---

Repositório no Github: <https://github.com/gustavocosta7/produtoJSF.git>

Vídeo do Tutorial no Youtube: <https://youtu.be/DGTgIvZoapY>

Script da Oficina: <http://donthpad.com/drajsf>

Primefaces: <http://search.maven.org/remotecontent?filepath=org/primefaces/primefaces/6.0/primefaces-6.0-sources.jar>

Donthpad: <http://donthpad.com/edu>

# Roadmap

---

- PARTE1: JasperReports

- O que é ?
- Funcionalidades
- Workflow
- Seções
- JRXML

- PARTE 2: JSF

- O que é?
- MVC
- Ciclo de Vida
- Componentes e Beans
- Primefaces

- PARTE 3: AMBIENTAÇÃO

- Softwares utilizados
- Configurações Necessárias

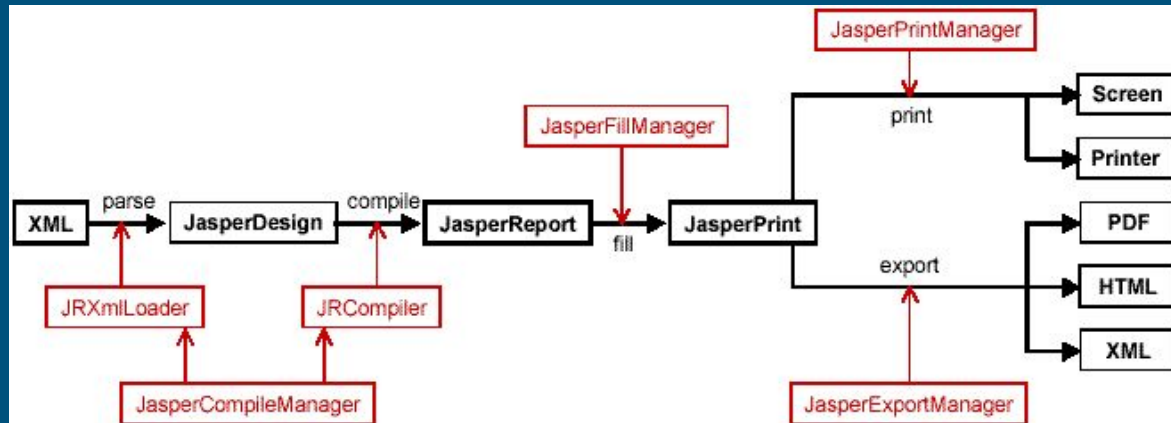
- PARTE 4: HANDS ON

- Apresentação do Minimundo
- Importação do Banco de Dados
- Geração de Back-End Java
- Geração de Front-End JSF (XHTML)
- Integração com JasperReports

# PARTE 1: JASPER REPORTS

O JasperReports é uma framework de geração de relatórios, feito em Java.

Ele gera automaticamente relatórios com diferentes formatos e informações facilmente editáveis, a partir de fontes de BD.



Formatos de Saida:

- Screen
- Printer
- PDF
- HTML
- Microsoft Excel
- RTF
- ODT
- arquivos XML.

# Funcionalidades

---

- Desenhar relatórios
- Construir relatórios a partir de um SGBD.
- Automatizar a criação de relatórios.
- Gerar diferentes saídas para os relatórios

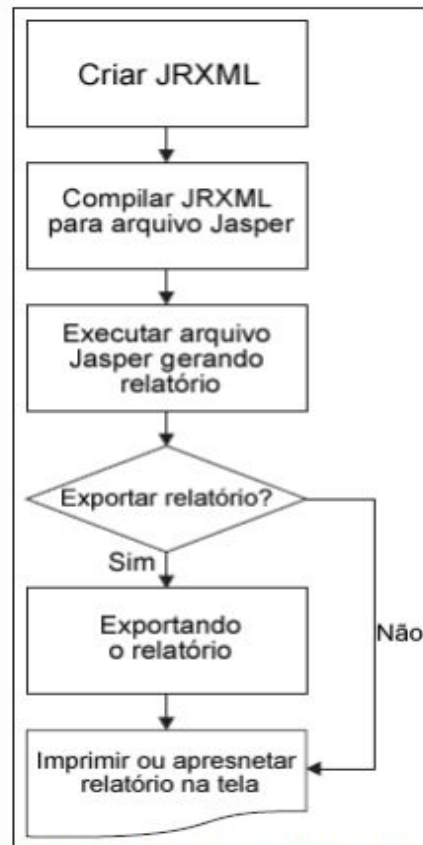


# Workflow

O JasperReports tem sempre os mesmos estágios de execução, incluindo a compilação de relatórios e renderização em um formato específico.

O primeiro estágio é o design do relatório. Você pode usar a ferramenta visual do Jaspersoft Studio ou ter um arquivo JRXML.

O próximo passo é compilar o JRXML em um arquivo Jasper. Esta fase de compilação não precisa acontecer todas as vezes; é necessário apenas se você alterar o JRXML.



**Workflow típico de relatório JasperReports**

Fonte: HEFFELFINGER, 2006

# Esquema do Layout com as Seções

	<b>Title</b>	
	<b>Page Header</b>	
	<b>Column Header</b>	
	<b>Detail</b>	
	<b>Column Footer</b>	
	<b>Page Footer</b>	



# Layout: Seções

---

1. **Title:** esta seção contém o título do relatório;

2. **Page Header:** esta seção aparece no topo de cada página para colocar datas, numeração de páginas, etc.;

3. **Column Header:** esta seção aparece no topo de cada coluna contém o título da coluna e as informações abaixo

4. **Detail:** nesta área serão colocadas as informações para cada item de registro. O JasperReports gera uma seção de detalhe para todo e cada registro processado;

5. **Column Footer:** esta seção aparece ao final de cada coluna;

6. **Page Footer:** esta seção aparece ao final de cada página;

7. **Last Page Footer:** esta seção aparece ao final da última página;

8. **Summary:** esta seção aparece ao término do relatório, logo após o último registro.

# JRXML

Esses arquivos são arquivos de layout contendo definições de projetos de relatórios salvos em XML.

As informações desses arquivos incluem são os elementos de design, campos de texto, gráficos, variáveis, layout do relatório, imagens e parâmetros.

Eles contribuem para a fácil e rápida criação de relatórios para os usuários, pois eles contêm dados necessários para a criação de relatórios.

# Elementos JRXML

**<staticText>**: define texto estático que não dependem de quaisquer fontes de dados, variáveis, parâmetros ou expressões de relatório.

**<textField>**: usado para exibir dados textuais dinâmicos nos relatórios.

**<textFieldExpression>**: define os dados que são exibido nos campos de texto.

**<field>**: Este elemento é usado para mapear dados de fontes de dados ou consultas em modelos de relatório. Os campos podem ser combinados em expressões de relatório para obter a saída necessária.

**<variable>**: expressões de relatório, usadas várias vezes em um relatório, pode ser atribuído a variáveis para simplifique o modelo.

**<parameter>**: este elemento é usado para definir o relatório parâmetros. Os valores dos parâmetros são fornecidos através de um `java.util.Map` chamando os métodos apropriados na API JasperReports.

**<group>**: este elemento é usado para agrupar registros consecutivos em um compartilhamento de fonte de dados algumas características comuns.

## PARTE 2: JSF - O Que é ?

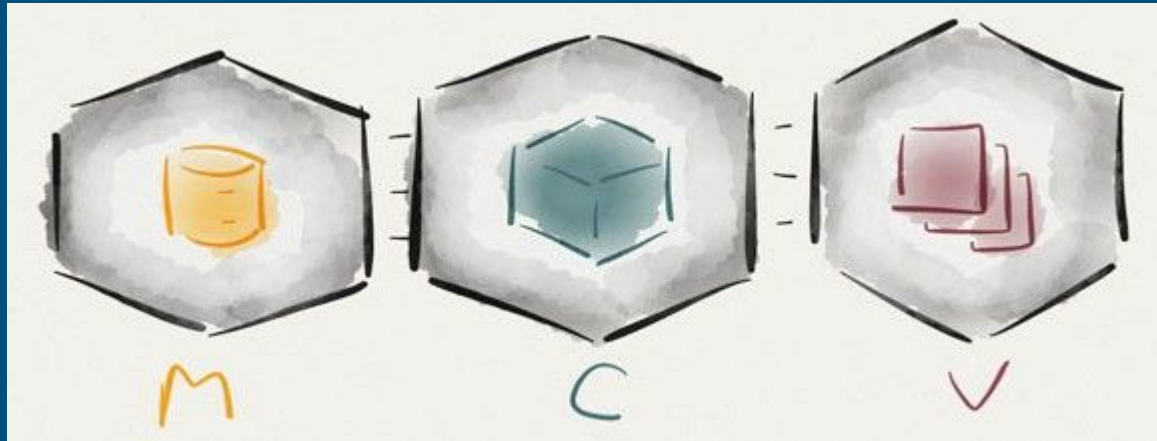
---

- ❖ É uma especificação Java para a construção de interfaces de usuário baseadas em componentes para aplicações web.
- ❖ Possui um modelo de programação dirigido a eventos, abstraindo os detalhes da manipulação dos eventos e organização dos componentes, permitindo que o programador se concentre na lógica da aplicação.



# PARTE 2: JSF - MVC

---



Entidades, Regras de  
Negócio e Objetos

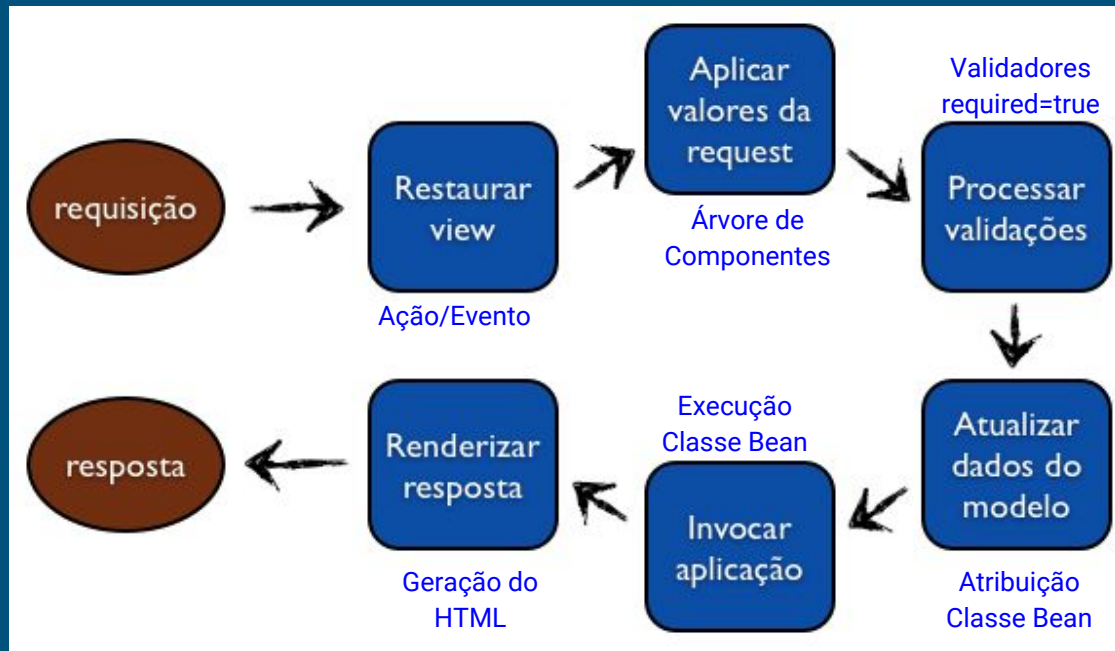
Faces Servlet

BackBean

Componentes UI

Kits Renderizadores

# PARTE 2: JSF - Ciclo de Vida



# PARTE 2: JSF - Componentes

A especificação do JSF fornece um conjunto de componentes visuais básicos em sua implementação de referência. Inclui duas bibliotecas de componentes básicos:

- ❖ Biblioteca “HTML” (sigla h): possui componentes que representam diversos elementos HTML.
- ❖ Biblioteca “Core” (sigla f): responsável por tarefas comuns no desenvolvimento de sistemas, como internacionalização, validação e conversão de dados de entrada.

```
<h:commandButton value="Botão"
  action="#{testeBT.teste}"
  actionListener="#{testeBT.testeMetodo}"
  styleClass="estiloBotao"
  disabled="true" />
```

```
<h:commandButton value="Botão"
  action="#{testeBT.teste}"
  actionListener="#{testeBT.testeVoid}"
  styleClass="estiloBotao"
  disabled="true">
  <f:ajax execute="@this" render="@form"/>
</h:commandButton>
```

# PARTE 2: JSF - Beans

É uma classe java que expõe propriedades e eventos a um framework como o JSF. Por sua vez, propriedade é um valor nomeado de um determinado tipo que pode ser lido e/ou escrito.

## Managed Beans

São beans que podem ser acessados a partir de uma página JSF.

### JAVA

```
@ManagedBean
public class ClientesBean implements Serializable {
    String nome;
    public String getNome(){
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

### JSP / XHTML

```
<h:form>
    <h:inputText Value="#{clientsBean.nome}" />
    <h:commandButton Action="resultado" value="enviar" />
</h:form>
```



## PARTE 2: JSF - Primefaces

---

É uma biblioteca de componentes de interface gráfica para as aplicações web baseadas em JSF.



# PARTE 3: AMBIENTAÇÃO

---

## Softwares utilizados

1. IDE – NetBeans
2. JSF – JavaSever Faces 2.x
3. Workbench Mysql
4. Servidor GlassFish
5. Java Persistence API (JPA) 2.0 - utilizando EclipseLink para gerar classes de entidade a partir do banco de dados e gerenciar transações.

# Configurações Necessárias

---

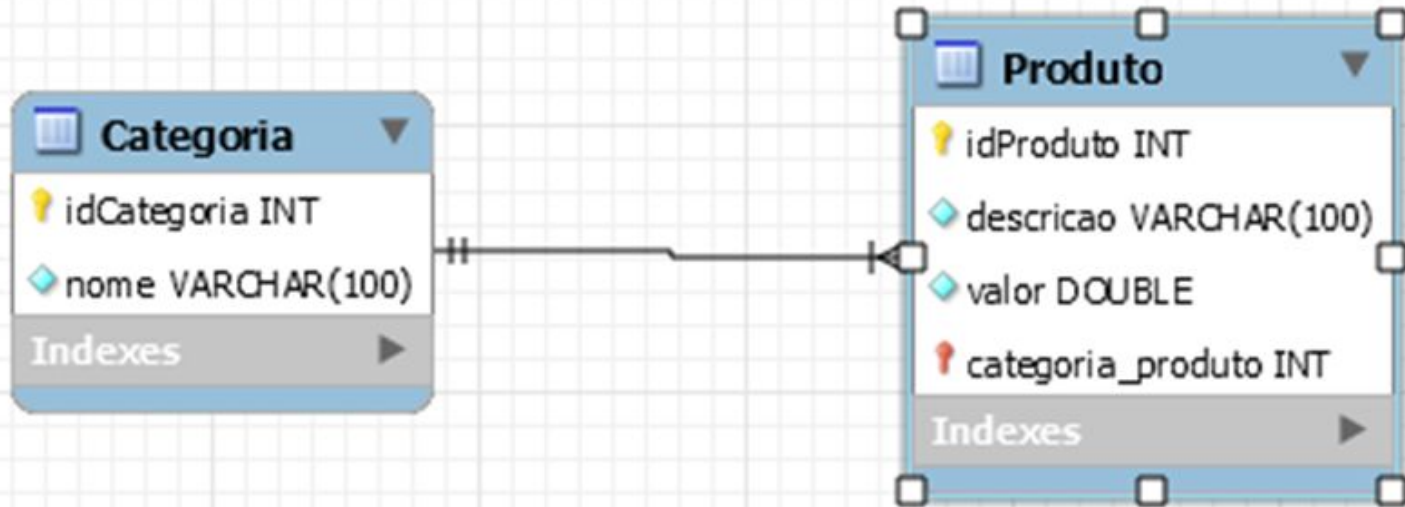


# Problematização

---

- Criar um Create, Read, Update, Delete, de forma rápida para um Sistema de Estoque.

# Introdução – Modelagem do Banco de Dados



# 1 – Passo – Criar projeto WEB

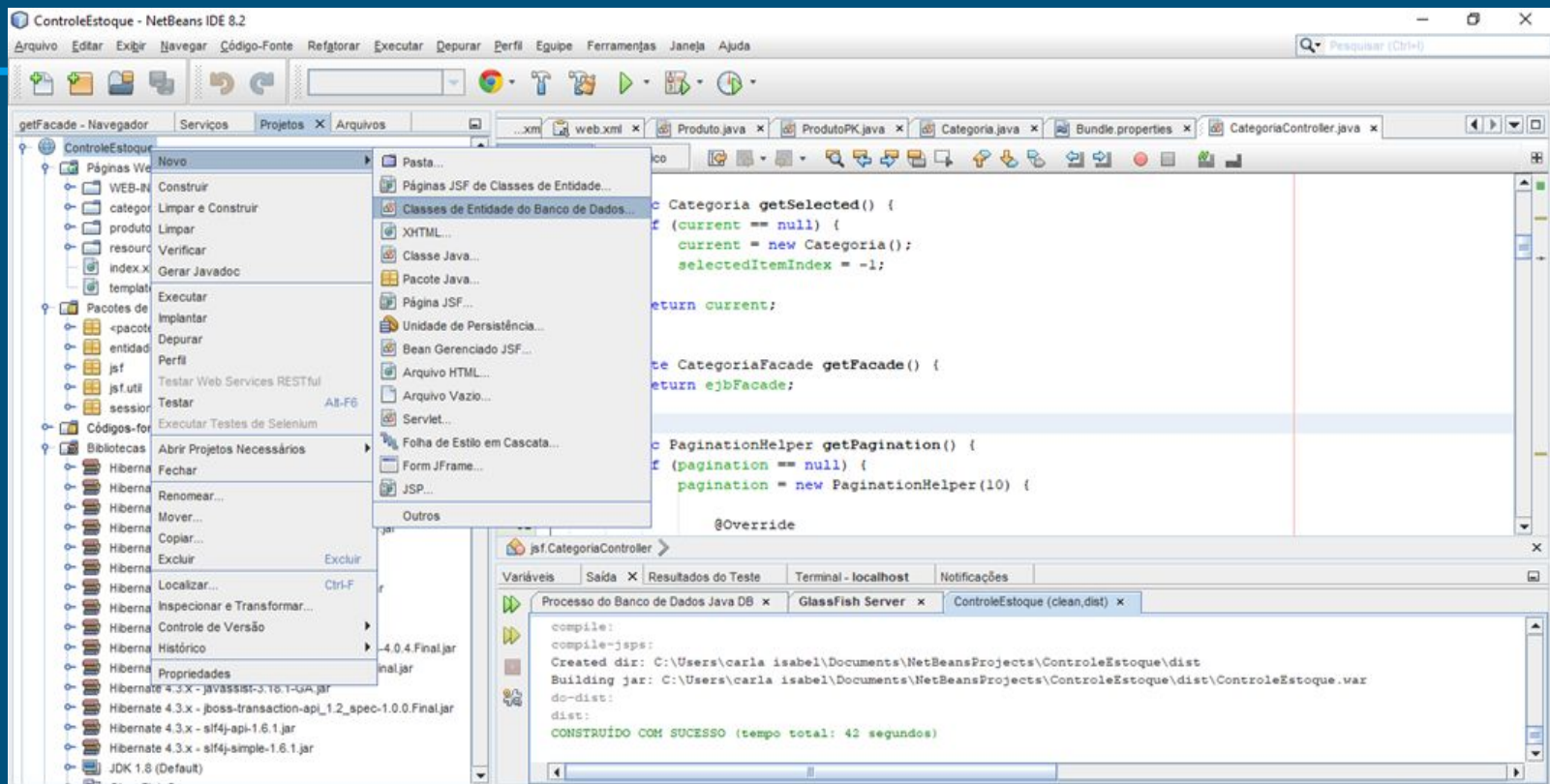
## Criando o Projeto de Aplicação Web

---

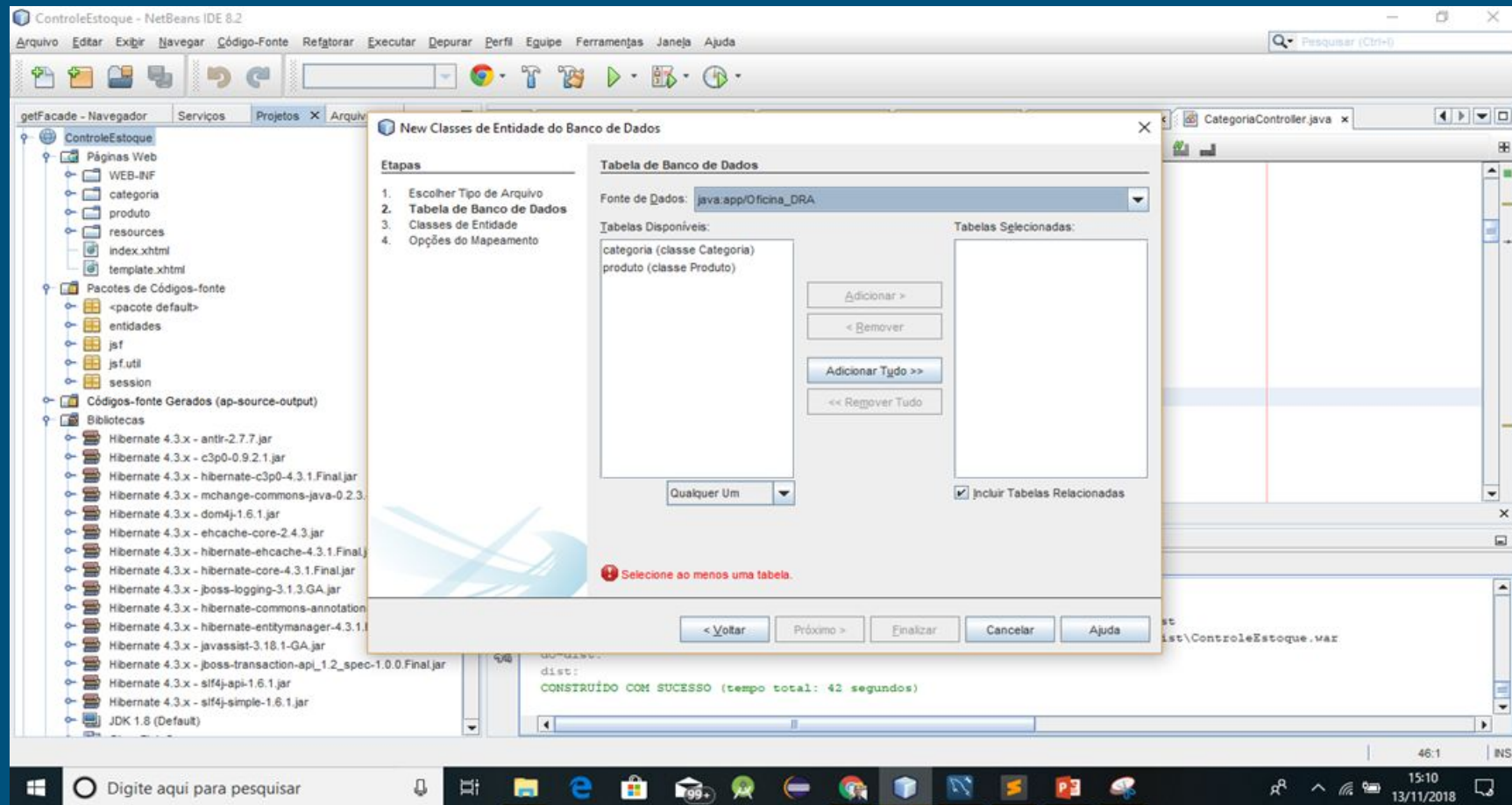
Neste exercício, você criará um projeto Web e adicionará o framework JavaServer Faces ao projeto. Ao criar o projeto, você irá selecionar JavaServer Faces no painel Frameworks do assistente de Novo Projeto.

1. Escolha Arquivo > Novo Projeto (Ctrl-Shift-N; &#8984-Shift-N no Mac) no menu principal.
2. Na categoria Java Web, selecione Aplicação Web. Clique em Próximo.
3. Digite  para o nome do projeto e defina a localização do projeto. Clique em Próximo.
4. Defina o servidor como GlassFish Server e defina a versão do Java EE para Java EE 6 Web ou Java EE 7 Web. Clique em Próximo.
5. No painel Frameworks, selecione a opção JavaServer Faces. Clique em Finalizar.

## 2 – Passo sincronização com o BD

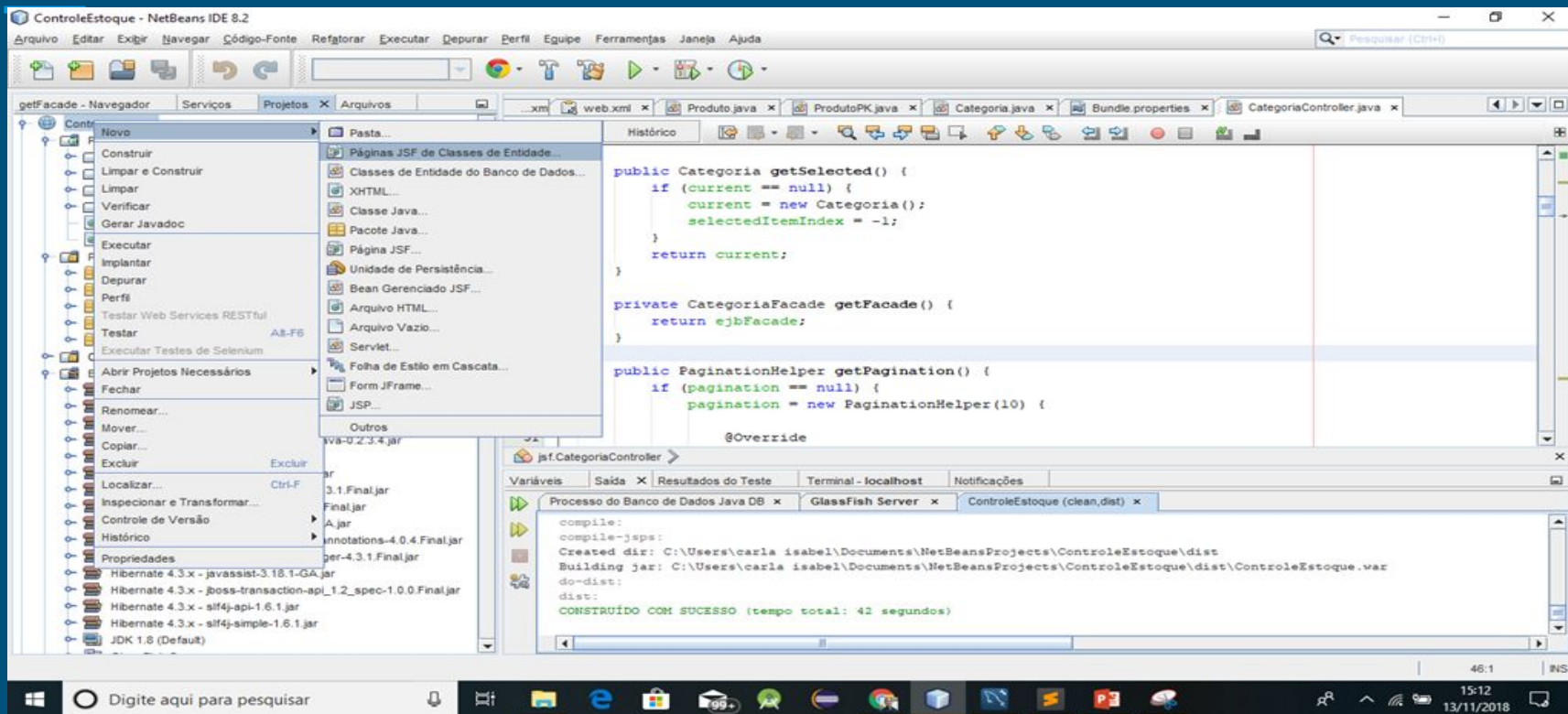


## 2 – Passo sincronização com o BD

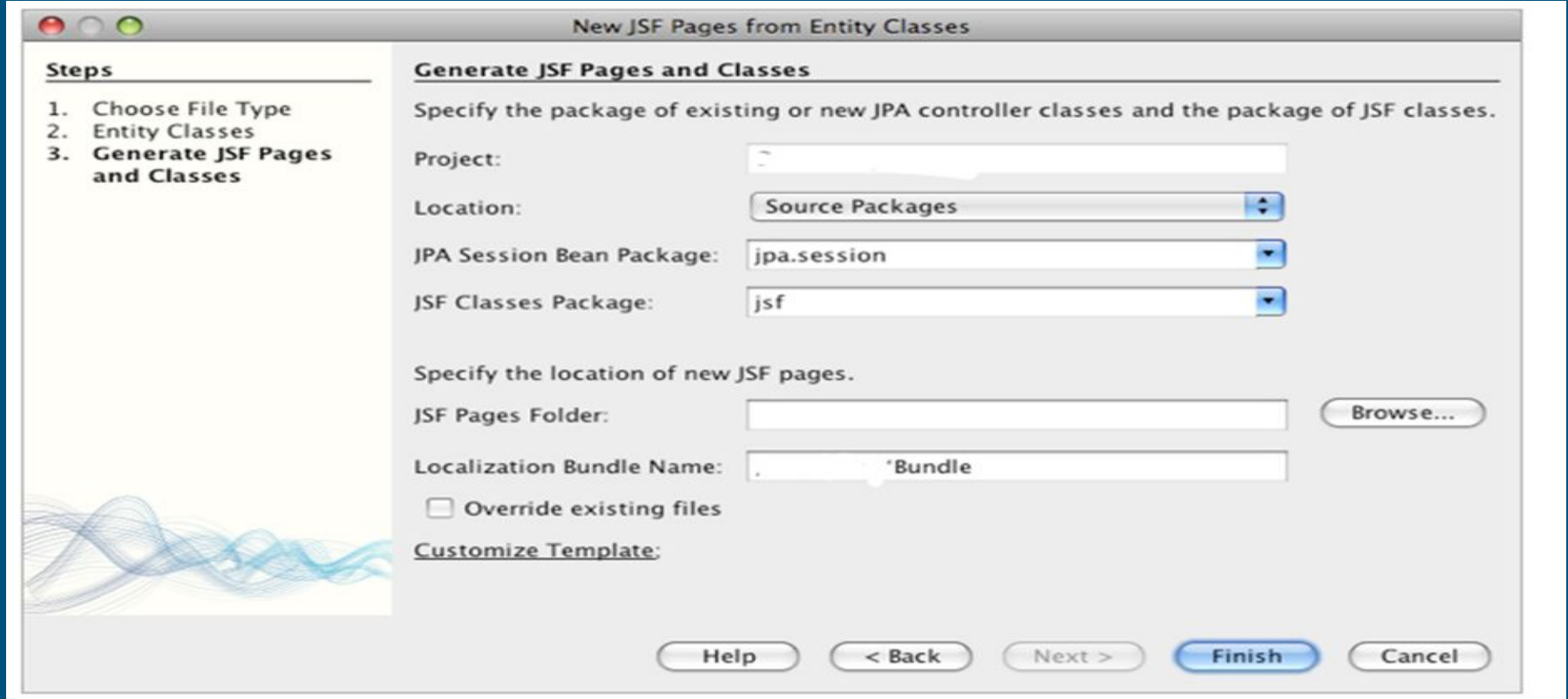




# 3 - Passo - Gerando Páginas JSF de Classes de Entidade.



## 3 Passo – Gerando Páginas JSF de Classes de Entidade



The screenshot shows a dialog box titled "New JSF Pages from Entity Classes". On the left, a "Steps" panel lists three steps: "1. Choose File Type", "2. Entity Classes", and "3. Generate JSF Pages and Classes", with the third step being the active one. The main area is titled "Generate JSF Pages and Classes" and contains the following fields and options:

- Specify the package of existing or new JPA controller classes and the package of JSF classes.**
  - Project:** A text field with a tilde (~) symbol.
  - Location:** A dropdown menu showing "Source Packages".
  - JPA Session Bean Package:** A dropdown menu showing "jpa.session".
  - JSF Classes Package:** A dropdown menu showing "jsf".
- Specify the location of new JSF pages.**
  - JSF Pages Folder:** A text field with a "Browse..." button to its right.
  - Localization Bundle Name:** A text field showing ".Bundle".
  - ☐ **Override existing files**
- Customize Template:** A section header with no visible options.

At the bottom of the dialog are five buttons: "Help", "< Back", "Next >", "Finish" (highlighted in blue), and "Cancel".

# REFERÊNCIAS BIBLIOGRÁFICAS

ORIO, Henrique. Protótipo de um conversor de relatórios Oracle Reports para jasper Reports. 2011. 125 f. Conclusão de Curso (Bacharel em Ciência da Computação)- Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul, Caxias do Sul, 2011.

DOCUMENT Views. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/documentviews/relatorios.htm>>. Acesso em: 05 nov. 2018.

DE OLIVEIRA SPINOLA, Gustavo. Java Reporting com JasperReports e iReport Open Source. Disponível em: <<https://www.devmedia.com.br/java-reporting-com-jasperreports-e-ireport-open-source-parte-i/1686>>. Acesso em: 09 nov. 2018.

DOCUMENTATION JasperSoft. Disponível em: <<https://community.jaspersoft.com/documentation>>. Acesso em: 06 nov. 2018.

DEVMEDIA. Guia de Referência JSF - JavaServer Faces. Disponível em: <<https://www.devmedia.com.br/guia/jsf-javascript-faces/38322>>. Acesso em: 28 de Outubro de 2018.

FÁBRICA DE SOFTWARE. SENAC. O que é JSF (Java Server Faces) ? Disponível em: <<http://fabrica.ms.senac.br/2013/06/o-que-e-jsf-java-server-faces>>. Acesso em: 27 de Outubro de 2018.

Documentação do NetBeans usado na Oficina:

GERANDO uma Aplicação CRUD JavaServer Faces 2.x Usando um Banco de Dados. Disponível em: <[https://netbeans.org/kb/docs/web/jsf20-crud\\_pt\\_BR.html](https://netbeans.org/kb/docs/web/jsf20-crud_pt_BR.html)>. Acesso em: 11 nov. 2018.