

Desenvolvimento de Aplicações Web

Braully Rocha da Silva

SENAC-GO

2019

Sumário

Primeiros passos

- Abordagem top-down e bottom-up

- JavaServer Faces

- Cadastro Cliente

- Tela de Cadastro de Cliente

- Modelo para o Cadastro de Cliente

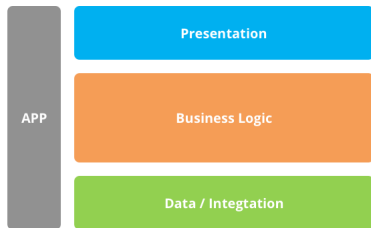
- Controle para o Cadastro de Cliente

- Ligar a Tela ao Controle – JSF

- Commit e Push para o GitHub

Primeiros passos

Abordagem top-down e bottom-up



Primeiros passos

JavaServer Faces

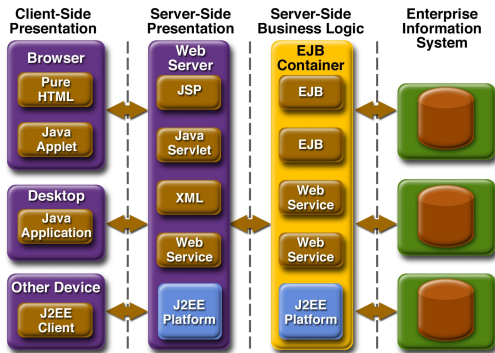
1. Especificação J2EE -Java Enterprise Edition
2. Manipulação de eventos
3. Regras de Navegação
4. Validação
5. Gerenciamento de Estado
6. Componentes visuais
7. Vasta quantidade de componentes de terceiros

Primeiros passos

Especificação J2EE -Java Enterprise Edition

Especificação Java Padrão
para Aplicações
Corporativas:

1. Retrocompatibilidade
2. Longevidade
3. Definições e Especificações
4. Implementação Padrão



Primeiros passos

Manipulação de eventos

1. 'Escutar' eventos e ações em componentes

```
<h:selectOneMenu  
    value="#{selectOneMenuValue}"  
    valueChangeListener="#{testBe  
    <f:selectItem itemLabel="Valo  
    <f:selectItem itemLabel="Valo  
    <f:selectItem itemLabel="Valo  
    <f:ajax/>  
</h:selectOneMenu>
```

Primeiros passos

Validação e Gerenciamento de Estados

1. Validação dos elementos da pagina
2. Manutenção do estado da pagina e aplicação



The screenshot shows a web browser window with the address bar displaying `localhost:8080/Jsf2validation/productentry.jspx`. The page content includes four red error messages:

- Product id is required!
- Product Name is required!
- Purchase amount must be at least 0.10!
- Product quantity must greater than 0!

Below the errors are four input fields and a button:

Product ID:

Product Name:

Purchase Price:

Quantity:

Primeiros passos

Componentes visuais

1. Componentes visuais padrões e simples
2. Construção de componentes próprios
3. Vasta quantidade de componentes de terceiros: Primefaces, RichFaces, IceFaces, Tomahawk

Primeiros passos

Componentes visuais



Primeiros passos

Frameworks e Ferramentas

1. Java 11
2. Netbeans 9
3. Git
4. Maven 3
5. SpringBoot 2
6. JSF 2.3 e Primefaces 6
7. JPA2 com Hibernate
8. IReport e Jasper
9. NPM e JavaScript*

Primeiros passos

...

Executar Netbeans Criar um projeto do arquétipo Clonar Projeto
Base Repositorio de Exemplos: <https://github.com/spring-guides>

Primeiros passos I

Cadastro Cliente

Primeiros passos II

Cadastro Cliente

Resultado Final:

- ▶ Cadastro básico de clientes
- ▶ Busca de clientes
- ▶ Persistir dados no banco
- ▶ Layout básico
- ▶ Hospedado na web

Primeiros passos III

Cadastro Cliente

Começando:

1. Entrar no seu GitHub
2. Fazer uma copia (Fork) do projeto base: `https://github.com/braully/desenvolvimento-web-senac`
3. Executar o Netbeans
4. Baixar sua propria copia do projeto: `https://github.com/.../desenvolvimento-web-senac.git`
5. Compilar o projeto: Build
6. Executar o projeto: Run
7. Abrir o Hello World: `https://localhost:8080/`

Primeiros passos IV

Cadastro Cliente

1º Requisito: Cadastro Básico de de Clientes, contendo as informações de Nome, CPF, Data de Nascimento, Endereço e se é um Cliente que está ativo.

Primeiros passos I

Tela de Cadastro de Cliente

Abordagem Top-Down, então primeiro a Tela em HTML.

Primeiros passos II

Tela de Cadastro de Cliente

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Cadastro Cliente</title>
  </head>
  <body>
    <form>
      <label>Nome</label><input type="text" />

      <label>CPF</label><input type="text" />

      <label>Data nascimento</label><input type="text" />

      <label>Endereço</label><input type="text" />

      <label>Cidade</label><input type="text" />

      <label>Estado</label><select size="1"></select>

      <label>Ativo</label><input type="checkbox" />

      <button type="submit" value="Salvar" />
    </form>
  </body>
</html>
```

Primeiros passos III

Tela de Cadastro de Cliente

Tela HTML de Cadastro de Cliente: `http://localhost:8080/cliente/cadastro-cliente-0.html`

Primeiros passos IV

Tela de Cadastro de Cliente

Melhorias no visual:

- ▶ Frameworks CSS: Bootstrap, JQuery UI, Foundation, Materialize CSS e etc
- ▶ Bootstrap 4: <https://getbootstrap.com/>
 - ▶ Maduro
 - ▶ Layout profissional
 - ▶ Responsividade
 - ▶ Componentes visuais comuns
 - ▶ Farta documentação, exemplos e casos de uso

Primeiros passos V

Tela de Cadastro de Cliente

Primeiros passos com o Bootstrap 4.0:

- ▶ Instalar o Bootstrap: "`<link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">`"
- ▶ Aplicar classes de estilo ao nosso formulário: <https://getbootstrap.com/docs/4.0/components/forms/>

Primeiros passos I

Tela de Cadastro de Cliente

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Cadastro Cliente</title>
  <meta charset="UTF-8">
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    rel="stylesheet" />
</head>
<body>
  <div class="container-fluid">
    <form>
      <div class="form-group"><label>Nome</label><input class="form-control" /></div>

      <div class="form-group"><label>CPF</label><input class="form-control" /></div>

      <div class="form-group"><label>Data nascimento</label><input class="form-control" /></div>

      <div class="form-group"><label>Endereço</label><input class="form-control" /></div>

      <div class="form-group"><label>Cidade</label><input class="form-control" /></div>

      <div class="form-group"><label>Estado</label>
      <select class="form-control" size="1"><select></div>

      <div class="form-check"><input class="form-check-input" type="checkbox" />
        <label class="form-check-label">Ativo</label>
      </div>

      <button type="submit" class="btn btn-primary" value="Salvar" />
    </form>
  </div>
</body>
</html>
```

Primeiros passos II

Tela de Cadastro de Cliente

Tela HTML de Cadastro de Clientes com Layout Bootstrap.

`http://localhost:`

`8080/cliente/cadastro-cliente-0-css.html`

Interdisciplinaridade com Design: Cores e Fontes pra um Layout Harmônico e Profissional.

Primeiros passos I

Tela de Cadastro de Cliente

Do HTML para o JSF:

- ▶ Converter do HTML para o XHTML
- ▶ Importar a Biblioteca de Componentes Básicos HTML-JSF:
`xmlns:h="http://java.sun.com/jsf/html"`
- ▶ Substituir o componente HTML pelo equivalente componente JSF: Ligado a camada de Controle Java

Primeiros passos II

Tela de Cadastro de Cliente

Componente	Tag Html	Tag Html-Jsf
Formulário	form	h:form
Entrada	input	h:input
Seleção	select	h:selectOneMenu
Botão	button	h:commandButton

Primeiros passos III

Tela de Cadastro de Cliente

```
<?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 lang="pt-br"
    xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html">
<head>
<title>Cadastro Cliente</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        rel="stylesheet" />
</head>
<body>
    <div class="container-fluid">
        <h:form>
            <div class="form-group"><label>Nome</label><h:inputText class="form-control" /></div>

            <div class="form-group"><label>CPF</label><h:inputText class="form-control" /></div>

            <div class="form-group"><label>Data nascimento</label><h:inputText class="form-control" /></div>

            <div class="form-group"><label>Endere o</label><h:inputText class="form-control" /></div>

            <div class="form-group"><label>Cidade</label><h:inputText class="form-control" /></div>

            <div class="form-group"><label>Estado</label>
            <h:selectOneMenu class="form-control"></h:selectOneMenu></div>

            <div class="form-check"><h:selectBooleanCheckbox class="form-check-input" />
                <label class="form-check-label">Ativo</label>
            </div>

            <h:commandButton class="btn btn-primary" value="Salvar" />
        </h:form>
    </div>
</body>
</html>
```

Primeiros passos IV

Tela de Cadastro de Cliente

Tela HTML de Cadastro de Clientes com Layout Bootstrap <http://localhost:8080/cliente/cadastro-cliente-0-css.html>

Primeiros passos V

Tela de Cadastro de Cliente

Documentação oficial:

- ▶ Todos os componentes HTML-JSF: <https://docs.oracle.com/javaee/7/javaxserver-faces-2-2/vldocs-facelets/h/tld-summary.html>
- ▶ Todas as bibliotecas de componentes padrão JSF: <https://docs.oracle.com/javaee/7/javaxserver-faces-2-2/vldocs-facelets/toc.htm>

Primeiros passos I

Modelo para o Cadastro de Cliente

Cliente
nome cpf dataNascimento endereco cidade estado ativo

Primeiros passos II

Modelo para o Cadastro de Cliente

Cliente
nome : String cpf : String dataNascimento : Date endereco : String cidade : String estado : String ativo : Boolean

Primeiros passos III

Modelo para o Cadastro de Cliente

Criar um pacote chamado 'modelo' e nele a classe Cliente.java:

```
package com.github.braully.dws.modelo;
```

```
import java.util.Date;
```

```
public class Cliente {  
    String nome;  
    String cpf;  
    Date dataNascimento;  
    String endereco;  
    String cidade;  
    String estado;  
    Boolean ativo;  
}
```

Primeiros passos IV

Modelo para o Cadastro de Cliente

Pela especificação Java EE, os componentes da aplicação devem respeitar o Padrão JavaBeans:

- ▶ Tutorial: <https://docs.oracle.com/javase/tutorial/javabeans/>
- ▶ Especificação:
<https://www.oracle.com/technetwork/java/javase/documentation/sp136004.html>
- ▶ Possuir um construtor sem argumento
- ▶ Métodos get/set para acesso as suas propriedades

Primeiros passos V

Modelo para o Cadastro de Cliente

Cliente
nome : String ... ativo : Boolean
+ Cliente() + setNome(nome : String) : void + getNome() : String + setCpf(cpf : String) : void + getCpf() : String + setDataNascimento(dataNascimento : Date) : void + getDataNascimento() : Date + setEndereco(endereco : String) : void + getEndereco() : String + setCidade(cidade : String) : void + getCidade() : String + setEstado(estado : String) : void + getEstado() : String + setAtivo(estado : Boolean) : void + getAtivo() : Boolean

Primeiros passos VI

Modelo para o Cadastro de Cliente

Criar o construtor padrão e os métodos de acesso get/set para cada propriedade, para respeitar o Padrão JavaBeans.

Primeiros passos VII

Modelo para o Cadastro de Cliente

```
/* ... */  
public Cliente() {}  
  
public String getNome() {  
    return nome;  
}  
  
public void setNome(String nome) {  
    this.nome = nome;  
}  
  
/* ... */  
public Boolean getAtivo() {  
    return ativo;  
}  
  
public void setAtivo(Boolean ativo) {  
    this.ativo = ativo;  
}
```

Primeiros passos I

Controle para o Cadastro de Cliente

Precisamos Ligar a Tela ao Modelo, para tanto precisaremos do Controle, onde as ações serão executadas.

Primeiros passos II

Controle para o Cadastro de Cliente

Criar um pacote 'controle' e nela a classe ClienteControle.java:

```
package com.github.braully.dws.controle;

import com.github.braully.dws.modelo.Cliente;
import org.springframework.stereotype.Component;

@Component
public class ClienteControle {
    Cliente cliente;

    public ClienteControle() {
        novoCliente();
    }

    public Cliente getCliente() {
        return cliente;
    }

    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }

    public void novoCliente() {
        this.cliente = new Cliente();
    }

    public void salvarCliente() {
        String mensagem = "Cliente Salvo:" + cliente;
        System.out.println(mensagem);
    }
}
```

Primeiros passos I

Ligar a Tela ao Controle – JSF

Linguagem de Expressão (EL) JSF:

<https://docs.oracle.com/javaee/7/tutorial/jsf-el.htm>

- ▶ Possui operadores lógicos e aritméticos: +, -, *, or, and, == e etc
- ▶ As expressões devem estar dentro do ambiente JSF: `#{ }`
- ▶ Referenciar objetos, atributos, coleções e métodos do controle: `#{customer.name}`, `#{customer.address["street"]}`, `#{customer.toString()}` e `#{customer.age + 20}`

Primeiros passos II

Ligar a Tela ao Controle – JSF

Ligar o valor dos componentes da tela, ao respectivo modelo e ações no controle. Exceto a Data de nascimento.

Primeiros passos III

Ligar a Tela ao Controle – JSF

```
<?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 lang="pt-br"
    xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html">
<head>
    <title>Cadastro Cliente</title>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
        rel="stylesheet" />
</head>
<body>
    <div class="container-fluid">
        <h:form>
            <div class="form-group" value="#{clienteControle.cliente.nome}">
                <label>Nome</label><h:inputText class="form-control" />
            </div>

            <!-- ... -->

            <div class="form-check">
                <h:selectBooleanCheckbox class="form-check-input" />
                <label class="form-check-label">Ativo</label>
            </div>

            <h:commandButton class="btn btn-primary" value="Salvar"
                action="#{clienteControle.salvarCliente}"/>
        </h:form>
    </div>
</body>
</html>
```

Primeiros passos IV

Ligar a Tela ao Controle – JSF

Construir o projeto e executar a tela e realizar um "Cadastro".

Primeiros passos V

Ligar a Tela ao Controle – JSF

Adicionar um componente JSF para receber mensagens do Controle

Primeiros passos VI

Ligar a Tela ao Controle – JSF

```
<!-- ... -->
<body>
  <div class="container-fluid">
    <h:form>
      <h:messages showSummary="true" />
      <div class="form-group" value="#{clienteControle.cliente.nome}">
        <label>Nome </label><h:inputText class="form-control" />
      </div>

      <!-- ... -->
    </h:form>
  </div>
</body>
</html>
```

Primeiros passos VII

Ligar a Tela ao Controle – JSF

Alterar o Modelo do Cliente e colocar um Método toString() com saída mais amigável

Primeiros passos VIII

Ligar a Tela ao Controle – JSF

Cliente.java:

```
/* ... */  
public class Cliente {  
    /* ... */  
    public String toString() {  
        return "Cliente{" + "nome=" + nome  
            + ",_cpf=" + cpf  
            + ",_dataNascimento=" + dataNascimento  
            + ",_endereco=" + endereco  
            + ",_cidade=" + cidade  
            + ",_estado=" + estado  
            + ",_ativo=" + ativo + '}';  
    }  
}
```

Primeiros passos IX

Ligar a Tela ao Controle – JSF

Alterar o Controle para Enviar uma Mensagem JSF para Tela

Primeiros passos X

Ligar a Tela ao Controle – JSF

ClienteControle.java:

```
/* ... */
public class ClienteControle {
    /* ... */
    public void salvarCliente() {
        String mensagem = "Cliente Salvo:" + cliente;
        FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(mensagem));
    }
}
```

Primeiros passos XI

Ligar a Tela ao Controle – JSF

Construir o projeto novamente e executar a tela e realizar um "Cadastro".

Primeiros passos I

Commit e Push para o GitHub

Primeiros passos II

Commit e Push para o GitHub

- ▶ Salvar todas as alterações
- ▶ Fazer o Commit no repositório local
- ▶ Fazer o Push para sua conta no GitHub

Primeiros passos III

Commit e Push para o GitHub

Subir a aplicação web pra uma nuvem gratuita.

Primeiros passos IV

Commit e Push para o GitHub

Heroku: <https://www.heroku.com/>

- ▶ Fazer uma conta gratuita de estudante
- ▶ Criar uma aplicação gratuita Java
- ▶ Ligar a conta com sua conta do GitHub
- ▶ Fazer a implantação da sua aplicação