Desenvolvimento de Aplicações Web

Braully Rocha da Silva

SENAC-GO

2019

Sumário

Primeiros passos I

Continuação: Subir alterações e publicação

...

Primeiros passos I

Continuação: Subir alterações e publicação

Commit e Push para o GitHub

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

Primeiros passos II

Continuação: Subir alterações e publicação

Subir a aplicação web pra uma nuvem gratuita.

Primeiros passos III

Continuação: Subir alterações e publicação

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

Primeiros passos I

Conversor e Validadores de Tipo

Operação de requisição web (HTTP Request):

- Sempre são strings
- Precisam ser convertidos
- Precisam ser validados

Primeiros passos II

Conversor e Validadores de Tipo

Ligar o atributo 'Data nascimento' ao seu respectivo valor no modelo

Primeiros passos III

Conversor e Validadores de Tipo

```
<?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>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>Data nascimento</label>
            <h:inputText value="#{clienteControle.cliente.dataNascimento}"
            class="form-control" />
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos IV

Conversor e Validadores de Tipo

Tentar cadastrar um novo cliente:

- Erro de conversão
- Utilizar os conversores padrões no "Core do JSF"
- Incluir o "pacote" Core do JSF
- Utilizar o Conversor de Tempo e Data

Primeiros passos V

Conversor e Validadores de Tipo

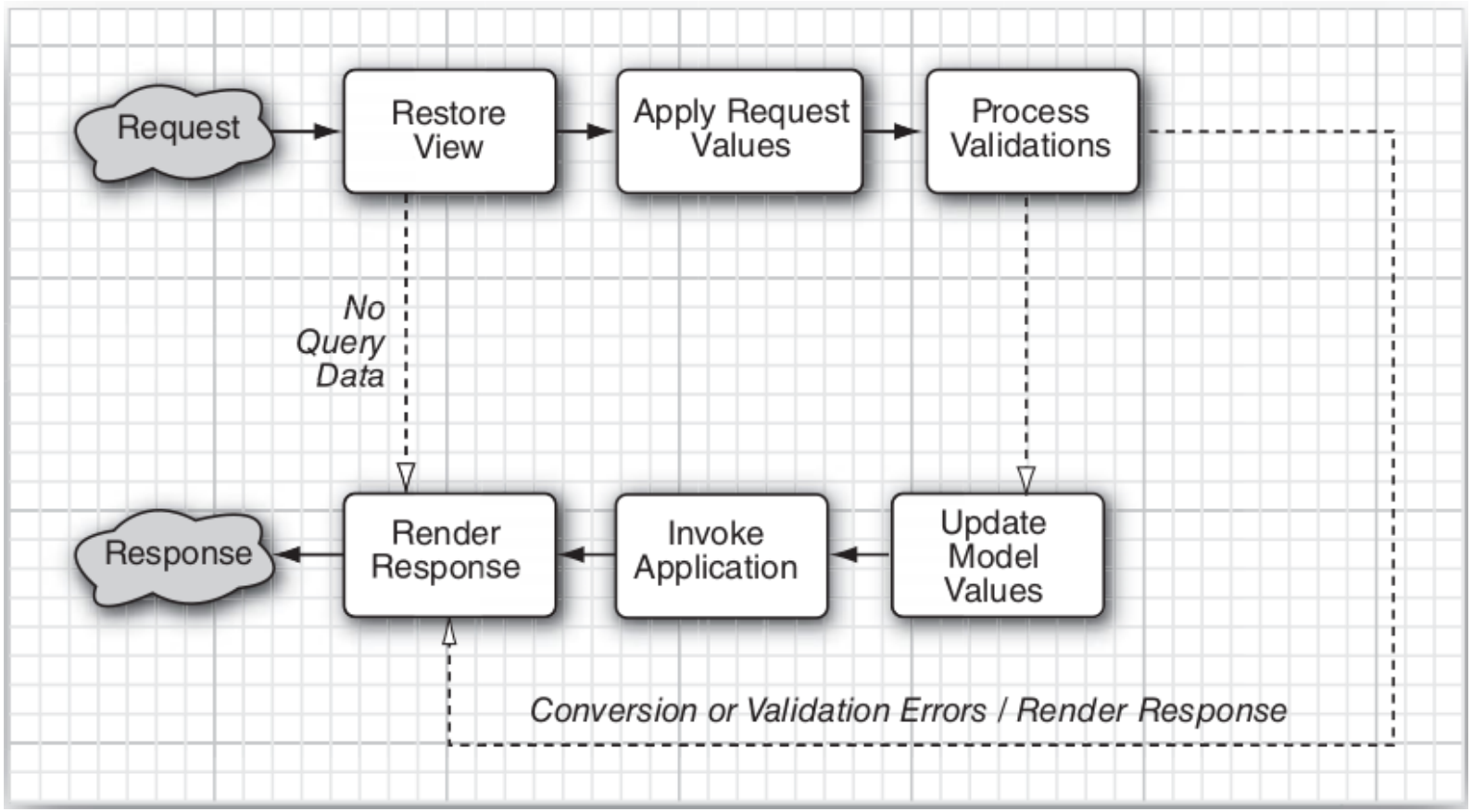

Figura: Ciclo de vida JSF

Primeiros passos VI

Conversor e Validadores de Tipo

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>Data nascimento</label>
            <h:inputText value="#{clienteControle.cliente.dataNascimento}"
            class="form-control">
                <f:convertDateTime />
            </h:inputText>
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos VII

Conversor e Validadores de Tipo

Converter no formato Brasileiro de Data (dia/mês/ano)

Primeiros passos VIII

Conversor e Validadores de Tipo

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>Data nascimento</label>
            <h:inputText value="#{clienteControle.cliente.dataNascimento}"
            class="form-control">
                <f:convertDateTime pattern="dd/MM/yyyy"/>
            </h:inputText>
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos IX

Conversor e Validadores de Tipo

Cadastrar novo cliente: Tudo Ok. Conversores Padrões do JSF: `https://docs.oracle.com/javaee/7/api/javax/faces/convert/package-summary.html`

Primeiros passos X

Conversor e Validadores de Tipo

Clientes precisam

- Obrigatoriamente informar o Nome
- Campo CPF deve estar no formato correto e somente digitos.

Primeiros passos XI

Conversor e Validadores de Tipo

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>Nome</label>
            <h:inputText value="#{clienteControle.cliente.nome}"
            class="form-control">
                <f:validateRequired  />
            </h:inputText>
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos XII

Conversor e Validadores de Tipo

Mensagem amigável de erro

Primeiros passos XIII

Conversor e Validadores de Tipo

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>Nome</label>
            <h:inputText value="#{clienteControle.cliente.nome}"
            class="form-control"
                         validatorMessage="Nome␣é␣necessário">
                <f:validateRequired  />
            </h:inputText>
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos XIV

Conversor e Validadores de Tipo

CPF verificar se numero no formato do CPF. Usar o validator de expressão regular "validateRegex"

Primeiros passos XV

Conversor e Validadores de Tipo

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>CPF</label>
            <h:inputText value="#{clienteControle.cliente.cpf}"
            class="form-control"
                 validatorMessage="CPF␣deve␣conter␣11␣n meros">
                <f:validateRegex pattern="^[0-9]{3}\.?[0-9]{3}\.?[0-9]{3}\-?[0-9]{2}$" />
            </h:inputText>
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos XVI

Conversor e Validadores de Tipo

Um excelente profissional de TI deve dominar expressões regulares.
Material recomendado: `https://aurelio.net/regex/`

Primeiros passos I

Itens de valores para seleção

As seleções podem ser de diversos tipos:

- Limitadas na própria tela
- Uma lista de strings
- Uma lisa de objetos (exige conversão)

Primeiros passos II

Itens de valores para seleção

Limitada na própria tela:

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>Estado</label>
            <h:selectOneMenu value="#{clienteControle.cliente.estado}"
            class="form-control">
                <f:selectItem itemValue="" />
                <f:selectItem itemValue="Goiás" />
            </h:selectOneMenu>
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos III

Itens de valores para seleção

Lista de Strings:

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->
        <div class="form-group">
            <label>Estado</label>
            <h:selectOneMenu value="#{clienteControle.cliente.estado}"
            class="form-control">
                <f:selectItem itemValue="" />
                <f:selectItems value="#{clienteControle.listaEstados}" />
            </h:selectOneMenu>
        </div>
        <!-- ... -->
</body>
</html>
```

Primeiros passos IV

Itens de valores para seleção

ClienteControle.java:

```
/* ... */
public class ClienteControle {
    /* ... */
    List<String> listaEstados = new ArrayList<>();

    {
        listaEstados.add("Goiás");
        listaEstados.add("Distrito␣Federal");
        listaEstados.add("São␣Paulo");
        /* ... */
    }

    public List<String> getListaEstados() {
        return listaEstados;
    }
}
```

Primeiros passos V

Itens de valores para seleção

Lista de Objetos.

Primeiros passos VI

Itens de valores para seleção

Usar uma enumeração para representar o Estado.

Primeiros passos VII

Itens de valores para seleção

Estado.java:

```
public enum Estado {
    GO("Goiás"), DF("Distrito Federal"), AC("Acre"), AL("Alagoas"), AM("Amazonas"),
    AP("Amapá"), BA("Bahia"), CE("Ceará"), ES("Espírito Santo"), MA("Maranhão"),
    MG("Minas Gerais"), MS("Mato Grosso do Sull"), MT("Mato Grosso"), PA("Pará"),
    PB("Paraíba"), PE("Pernambuco"), PI("Piauí"), PR("Paraná"), RJ("Rio de Janeiro"),
    RN("Rio Grande do Norte"), RO("Rond nia"), RR("Roraima"), RS("Rio Grande do Sul"),
    SC("Santa Catarina"), SE("Sergipe"), SP("São Paulo"), TO("Tocantins");

    private String nome;

    private Estado(String nome) {
        this.nome = nome;
    }

    public String getSigla() {
        return this.name();
    }

    public String getNome() {
        return nome;
    }
}
```

Primeiros passos VIII

Itens de valores para seleção

Trocar o tipo do estado em Cliente de String para a enumeração Estado.

Primeiros passos IX

Itens de valores para seleção

Cliente.java:

```
/* ... */
public class Cliente {
    /* ... */
    Estado estado;

    public Estado getEstado() {
        return estado;
    }

    public void setEstado(Estado estado) {
        this.estado = estado;
    }
}
```

Primeiros passos X

Itens de valores para seleção

Trocar a Lista de valores de estado no ClienteControle

Primeiros passos XI

Itens de valores para seleção

ClienteControle.java:

```
/* ... */
public class ClienteControle {
    /* ... */
    public Estado[] getListaEstados() {
        return Estado.values();
    }
}
```

Primeiros passos I

Listagem de Clientes

- "Persistir" os clientes
- Listar os clientes cadastrados

Primeiros passos II

Listagem de Clientes

Salvar o cliente em um ”repositorio”

Primeiros passos III

Listagem de Clientes

ClienteControle.java:

```
/* ... */
public class ClienteControle {
    /* ... */
    List<Cliente> clientes = new ArrayList<>();

    public List<Cliente> getClientes() {
        return clientes;
    }

    public void salvarCliente() {
        String mensagem = "Cliente Salvo: " + cliente;
        System.out.println(mensagem);
        FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(mensagem));
        clientes.add(cliente);
        novoCliente();
    }
}
```

Primeiros passos IV

Listagem de Clientes

Adicionar uma Tabela JSF: `https://www.tutorialspoint.com/jsf/jsf_display_datatable.htm`

Primeiros passos V

Listagem de Clientes

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
<head>
    <!-- ... -->
</head>
<body>
        <!-- ... -->

            <h:dataTable value="#{clienteControle.clientes}" var="clint">
                <h:column>
                    <f:facet name="header">Nome</f:facet>
                    <h:outputText value="#{clint.nome}"/>
                </h:column>
                <h:column>
                    <f:facet name="header">Cpf</f:facet>
                    <h:outputText value="#{clint.cpf}"/>
                </h:column>
                <h:column>
                    <f:facet name="header">Date de Nascimento</f:facet>
                    <h:outputText value="#{clint.dataNascimento}"/>
                </h:column>
            </h:dataTable>

        <!-- ... -->
</body>
</html>
```

Primeiros passos VI

Listagem de Clientes

Sem Layout. Aplicar estilo de tabela Bootstrap:

https://getbootstrap.com/docs/4.0/content/tables/

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core">
        <!-- ... -->
            <h:dataTable class="table␣table-striped" value="#{clienteControle.clientes}" var="clint">
        <!-- ... -->
<\html>
```

Primeiros passos VII

Listagem de Clientes

Problemas da DataTable Padrão JSF:

- Componente complexo
- Visual simples e difícil personalização
- Oferece pouco a mais que o HTML

Primeiros passos VIII

Listagem de Clientes

Alternativa 1: fazer em HTML com o minimo de 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"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets">
        <!-- ... -->
        <table class="table table-striped">
            <thead>
                <tr>
                    <th scope="col">Nome</th>
                    <th scope="col">CPF</th>
                    <th scope="col">Data de Nascimento</th>
                </tr>
            </thead>
            <tbody>
                <ui:repeat value="#{clienteControle.clientes}" var="clint">
                    <tr>
                        <td>#{clint.nome}</td>
                        <td>#{clint.cpf}</td>
                        <td>#{clint.dataNascimento}</td>
                    </tr>
                </ui:repeat>
            </tbody>
        </table>
        <!-- ... -->
<\html>
```

Primeiros passos IX

Listagem de Clientes

Alternativa 2: Buscar componentes mais ricos de datatables em bibliotecas de terceiros. Atributos:

- Paginação
- Filtro
- Selecionar

Primeiros passos X

Listagem de Clientes

Biblioteca de Componentes JSF Primefaces:

`https://www.primefaces.org/downloads/`

Instalar no projeto, pom.xml:

```
<dependency>
    <groupId>org.primefaces</groupId>
    <artifactId>primefaces</artifactId>
    <version>6.2</version>
</dependency>
<dependency>
    <groupId>org.primefaces.themes</groupId>
    <artifactId>all-themes</artifactId>
    <version>1.0.10</version>
</dependency>
```

Primeiros passos XI

Listagem de Clientes

```
<?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"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:p="http://primefaces.org/ui">

      <h:head>
        <title>Cadastro Cliente</title>
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" />
      </h:head>

        <!-- ... -->
           <p:dataTable value="#{clienteControle.clientes}" var="clint">
                <p:column headerText="Nome">
                    #{clint.nome}
                </p:column>
                <p:column headerText="CPF">
                    #{clint.cpf}
                </p:column>
                <p:column headerText="Data␣de␣Nascimento">
                    #{clint.dataNascimento}
                </p:column>
            </p:dataTable>
        <!-- ... -->
<\html>
```

Primeiros passos XII

Listagem de Clientes

Explorar as propriedades do componente: https://www.primefaces.org/showcase/ui/data/datatable/basic.xhtml

```
<!-- ... -->
    <p:dataTable value="#{clienteControle.clientes}"
                 var="clint"
                 rows="10" paginator="true"
                 emptyMessage="Nenhum cliente encontrado"
                 resizableColumns="true"
                 draggableColumns="true">
<!-- ... -->
```

Primeiros passos XIII

Listagem de Clientes

Selecionar um Cliente da Listagem, para realizar edições dos dados.

Primeiros passos XIV

Listagem de Clientes

Botão que carrega um dos Clientes da lista para a Edição

```
<!-- ... -->
    <p:dataTable value="#{clienteControle.clientes}"
                 var="clint"
                 rows="10" paginator="true"
                 emptyMessage="Nenhum cliente encontrado"
                 resizableColumns="true"
                 draggableColumns="true">
<!-- ... -->
```

Primeiros passos XV

Listagem de Clientes

Botão que carrega um dos Clientes da lista para a Edição

```
<h:form>
    <p:dataTable value="#{clienteControle.clientes}" var="clint"
        <!-- ... -->
        <p:column>
            <h:commandButton value="Editar" title="Editar"
                             actionListener="#{clienteControle.setCliente(clint)}" />
        </p:column>
   </p:dataTable>
</h:form>
```

Primeiros passos XVI

Listagem de Clientes

Fim do Cadastro e Listagem do Cliente. Próximo Encontro:

- Template em JSF
- Refatoração de Telas