

✓ Trabalhando com Conversão, Validação e Arquivos de Propriedades no JSF

📌 1. Conversão de valores na Web (Strings para tipos Java)

Na web, todos os dados são inicialmente tratados como **Strings**. O JSF converte automaticamente esses valores para os tipos corretos com **conversores** (`<f:convertXXX>`).

🎯 Exemplo 1: Conversor de número com duas casas decimais

```
<h:inputText value="#{user.amount}">
  <f:convertNumber minFractionDigits="2"/>
</h:inputText>
```

📝 Entrada como "123,45" será convertida em 123.45 (dependendo da localidade).

🎯 Exemplo 2: Conversor de data no padrão brasileiro

```
<h:outputText value="#{user.date}">
  <f:convertDateTime pattern="dd/MM/yyyy"/>
</h:outputText>
```

📅 Exibe uma data como "12/05/2025".

📌 2. Validação de Dados

Você pode validar diretamente nos componentes. Exemplo de validação de tamanho mínimo:

```
<h:inputText id="cartao" value="#{user.cartao}">
  <f:validateLength minimum="4" />
</h:inputText>
```

Isso exige que o campo tenha pelo menos 4 caracteres.

3. Uso de Arquivos **.properties** (Internacionalização e Mensagens Personalizadas)

Passos para utilizar mensagens externas com JSF:

1. Criar o arquivo de propriedades

Exemplo: `messages.properties` (local:
`src/main/resources/com/corejsf/messages.properties`)

```
title=An Application to Test Data Conversion
enterPayment=Please enter the payment information
amount=Amount
creditCard=Credit Card
expirationDate=Expiration date (Month/Year)
process=Process
paymentInformation=Payment information
```

2. Referenciar o arquivo na sua página XHTML:

```
<f:loadBundle basename="com.corejsf.messages" var="msgs" />
```

Agora você pode acessar com `#{msgs.nomeDaChave}`.

3. Exemplo de uso no formulário:

```
<h:form>
  <h1><h:outputText value="#{msgs.enterPayment}" /></h1>

  <h:panelGrid columns="3">
    <h:outputText value="#{msgs.amount}" />
    <h:inputText id="amount" value="#{payment.amount}" label="#{msgs.amount}">
      <f:convertNumber minFractionDigits="2"/>
    </h:inputText>
    <h:message for="amount" />

    <h:outputText value="#{msgs.creditCard}" />
    <h:inputText id="card" value="#{payment.card}" label="#{msgs.creditCard}" />
  </h:panelGrid>

  <h:outputText value="#{msgs.expirationDate}" />
</h:form>
```

```
<h:inputText id="date" value="#{payment.date}" label="#{msgs.expirationDate}">
  <f:convertDateTime pattern="MM/yyyy" />
</h:inputText>
<h:message for="date" />
</h:panelGrid>

<h:commandButton value="#{msgs.process}" action="process" />
</h:form>
```

4. Registrar o **message-bundle** no **faces-config.xml**

```
<application>
  <message-bundle>com.corejsf.messages</message-bundle>
</application>
```

5. Internacionalização (i18n)

Você pode criar versões do mesmo arquivo de propriedades para diferentes idiomas:

Idioma	Nome do arquivo
Inglês (default)	<code>messages.properties</code>
Português (Brasil)	<code>messages_pt_BR.properties</code>
Espanhol	<code>messages_es.properties</code>

A escolha do idioma será feita com base no **locale** do navegador ou da aplicação.

Extras úteis

- Se quiser forçar a **locale**, você pode usar:

```
<f:view locale="pt_BR">
```

- Mensagens de erro de validação do JSF também podem ser sobrescritas nos arquivos `.properties`.

