Java para Web: Java Server Faces

Exercícios Propostos

Conversores e Validadores

Carlos Tosin

Histórico

Data	Observação
29/12/2011	Criação do documento

Sobre a Softblue

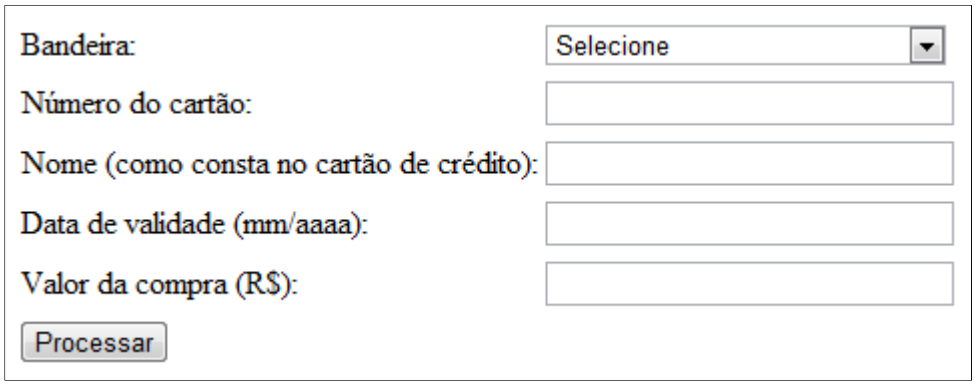
Fundada em 2003 na cidade de Curitiba-PR por André Milani e Carlos Tosin, inicialmente para dedicar-se à construção de soluções para web, a **Softblue** rapidamente expandiu seus negócios para a realização de cursos e treinamentos, devido ao fato de seus sócios serem altamente especializados em determinadas áreas de TI, certificados nas tecnologias em que atuam e autores de livros de informática sobre os mesmos temas.

Atualmente, a **Softblue** disponibiliza cursos e treinamentos on-line, permitindo que pessoas do Brasil inteiro possam realizá-los, estudando e aprendendo novas tecnologias, aprimorando seus conhecimentos para o mercado de trabalho, de acordo com sua disponibilidade de horários.

Exercícios Propostos

Exercício 1

Crie um sistema que simula pagamentos via cartão de crédito. A tela de entrada de dados deve ser semelhante à figura abaixo.



Para trabalhar com os conversores do JSF, considere o seguinte:

- A bandeira é um *enum* com as opções *VISA*, *MASTERCARD*, *AMEX* e *DINERS*. O valor selecionado deve ser mapeado para um atributo no bean cujo tipo é do *enum* criado.
- A data de validade deve ser mapeada para um objeto do tipo *Date* no bean.
- O valor da compra deve ser mapeado para um objeto do tipo *Double* no bean.
- O número de cartão e o nome são mapeados para *Strings*.

Caso ocorram erros de conversão, as mensagens devem aparecer ao lado do componente onde o erro foi originado. As mensagens exibidas devem ser lidas de um arquivo de recursos próprio da aplicação, que sobrescreve as mensagens relevantes do arquivo de recursos *Messages_xx_XX.properties* do JSF.

Exercício 2

Utilize o mesmo sistema criado no *Exercício 1* e adicione algumas regras de validação ao formulário de pagamento. As regras são as seguintes:

- Todos os campos são de preenchimento obrigatório.
- O número do cartão deve ser composto por 16 dígitos (**Dica**: esta validação pode ser facilmente realizada por uma expressão regular).
- O nome deve ter, no máximo, 50 caracteres.
- A data de validade do cartão não pode ser anterior à data atual (*mm/aaaa*). **Dica**: um validador customizado pode ser utilizado para fazer esta validação
- A valor da compra não pode ser um número negativo.

Assim como no *Exercício 1*, caso ocorram erros de validação, as mensagens devem aparecer ao lado do componente onde o erro foi originado. As mensagens exibidas devem ser lidas de um arquivo de recursos próprio da aplicação, que sobrescreve as mensagens relevantes do arquivo de recursos *Messages_xx_XX.properties* do JSF.

Desafio: Tente fazer com que o validador customizado também leia a mensagem a partir do arquivo de recursos que você criou para a aplicação. Para isso, pesquise sobre os métodos getApplication() da classe *FacesContext* e também *getMessageBundle()* da classe *Application*.