

 14

## Criando a ação de gravar livro e associando à página

Vamos agora começar a criar as ações que devem ser executadas pelos componentes da página de cadastro de livro do nosso projeto:

1. Para operacionalizar nossos componentes da página `livro.xhtml`, precisamos agora implementar os *ManagedBeans* que posteriormente serão conectados a esses componentes.
2. Vá em **Java Resources**, clique com o botão direito do mouse na pasta `src` e selecione **New > Class**.
3. Dê o nome `LivroBean` e diga que ficará no *Package* `br.com.caelum.livraria.bean`. Clique em **Finish**.
4. Já na classe, criaremos o método `gravar` que, por enquanto, apenas imprimirá uma mensagem no console.

```
public void gravar(){  
    System.out.println("Gravando livro!");  
}
```

1. Também precisamos indicar que a classe será gerenciada pelo JSF, por isso devemos anotá-la com `@ManagedBean` do pacote `javax.faces.bean`

```
import javax.faces.bean.ManagedBean;  
  
@ManagedBean  
public class LivroBean {  
    // métodos omitidos  
}
```

1. Voltando à página `livro.xhtml`, podemos agora fazer a conexão (binding) do componente `h:commandButton` com o método `gravar()` do *ManagedBean* usando o parâmetro `action`. Para isso usaremos uma linguagem de expressões (*Expression Language*) que começa sempre com `#{}.`
2. No parâmetro `action` do `h:commandButton` colocaremos dentro dos parênteses da *Expression Language* o caminho completo até o método.

ATENÇÃO: Usando *Expression Language*, a primeira letra do nome da classe *ManagedBean* deve ser minúscula. Ex: LivroBean ficará `#[livroBean]`

```
<h:commandButton value="Gravar" action="#{livroBean.gravar}" />
```

1. Salve todos os arquivos do projeto e "restarte" o servidor Tomcat (botão verde de *play*);
2. Abra a página `livro.xhtml` no navegador, clique no botão **Gravar** e observe na *View Console* do Eclipse se o método `gravar()` do *ManagedBean* foi executado.
3. Cole aqui o código do seu `LivroBean`:

### Responda

INSERIR CÓDIGO

FORMATAÇÃO