

## Mão à obra: remoção de livros

Chegou a hora de consolidar o que você aprendeu no capítulo, a remoção de livros. Seguem linhas gerais para guiá-lo nesta tarefa, mas claro, sem perder o teor de desafio.

1 - Em `livro.xhtml`, Vamos adicionar a funcionalidade de remoção e faremos isso diretamente na tabela. É claro, precisamos de uma nova coluna com a opção de remover:

```
<h:column>
    <f:facet name="header">Remover</f:facet>
    <h:commandLink value="Remover" />
</h:column>
```

2 - Você lembra que um `h:commandLink` fora de um `form` não funcionará como esperado? Envolva o `dataTable` dentro de um `h:form` utilizando o ID `formTabelaLivros`

3 - Ah! Lembre-se que quando gravamos um livro no botão `Gravar` que utiliza Ajax, o livro cadastrado deve aparecer na tabela de livros. Ainda lembra o que você deve fazer para que aquele "pedaço" da tela seja atualizado?

4 - Temos apenas um `h:commandLink` em nossa nova coluna, contudo ele nada faz. Vamos fazer um `binding` de `action`:

```
<h:column>
    <f:facet name="header">Remover</f:facet>
    <h:commandLink value="Remover" action="#{livroBean.remover(livro)}" />
</h:column>
```

5 - Não precisamos meditar muito para saber que o método `remover` que aceita um `Livro` ainda não existe. Você precisa criá-lo na classe `LivroBean`. Segue o método:

```
// na classe LivroBean

public void remover(Livro livro) {
    System.out.println("Removendo livro " + livro.getTitulo());
    new DAO<Livro>(Livro.class).remove(livro);
}
```

6 - Agora é só testar!