

## Colocando em prática: Criando a tabela de Negociações

1. Use **ctrl + N** HTML para criar um novo arquivo na pasta WebContent chamado `index.xhtml`. Como já fizemos antes, clique em *Next* e, na tela seguinte, escolha o template *xhtml 1.0 transitional*.

O Eclipse vai gerar um arquivo com um pouco de informações a mais, mas ainda muito parecido com o seguinte, onde mudamos o title:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Argentum Web</title>
  </head>
  <body>

  </body>
</html>
```

- 2- Além disso, para que os componentes consigam incluir seu CSS à nossa página, altere as tags head e body de forma a usar suas versões gerenciadas pelo JSF:

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:f="http://java.sun.com/jsf/core"
      xmlns:p="http://primefaces.org/ui">
  <h:head>
    <title>Argentum Web</title>
  </h:head>
  <h:body>

  </h:body>
</html>
```

- 3- Agora, **dentro do h:body**, vamos começar a montar nossa tabela de negociações. O componente que usaremos para isso é o `p:datatable`, do Primefaces. Ele precisará da lista de negociações e, assim como um `foreach`, uma variável para que cada coluna seja preenchida.

```
<p:datatable var="negociacao" value="#{argentumBean.negociacoes}">
</p:datatable>
```

- 3- Esse código acima diz que o componente `dataTable` do Primefaces chamará o método `getNegociacoes()` da classe `ArgentumBean` e, para cada linha da tabela, disponibilizará a negociação da vez na variável `negociacao`.

O problema é que o managed bean `ArgentumBean` ainda não existe e, claro, nem o método `getNegociacoes()` dela. E como cada vez que a página `index.xhtml` for requisitada ela fará algumas chamadas ao `getNegociacoes`, faremos a chamada ao *webservice* no construtor e, a cada chamada ao getter, apenas devolveremos a referência à mesma lista.

- Crie a classe `ArgentumBean` com `ctrl + N Class`, no pacote `br.com.caelum.argentum.bean` e anote ela com `@ManagedBean` .

```
@ManagedBean
public class ArgentumBean {
```

}

- Adicione o construtor que faça a chamada ao webservice através do `ClienteWebservice` , guarde a lista em um atributo e crie o getter que o componente chamará.

```
@ManagedBean
public class ArgentumBean {

    private List<Negociacao> negociacoes;

    public ArgentumBean() {
        negociacoes = new ClienteWebService().getNegociacoes();
    }

    public List<Negociacao> getNegociacoes() {
        return negociacoes;
    }
}
```

4- Agora, nossa página já não dá erro, mas nada é mostrado na tela, quando a acessamos. Falta indicarmos quais colunas queremos na nossa tabela -- no nosso caso: preço, quantidade, volume e data. Em cada coluna, adicionaremos um título e cada uma delas também mostrará o valor de texto.

Para criar a coluna com o título, usaremos o componente `p:column` e, como já fizemos antes, para mostrar o valor necessário, usaremos a `h:outputText` .

```
<p:dataTable var="negociacao" value="#{argentumBean.negociacoes}">
    <p:column headerText="Preço">
        <h:outputText value="#{negociacao.preco}" />
    </p:column>
    <p:column headerText="Quantidade">
        <h:outputText value="#{negociacao.quantidade}" />
    </p:column>
    <p:column headerText="Volume">
        <h:outputText value="#{negociacao.volume}" />
    </p:column>
    <p:column headerText="Data">
        <h:outputText value="#{negociacao.data}" />
    </p:column>
</p:dataTable>
```

5- Reinicie o Tomcat e acesse em seu navegador o endereço `http://localhost:8080/argentum/index.xhtml` . O resultado deve ser algo parecido com:

Preço	Quantidade	Volume	Data
224.35	13	2916.549999999997	2016-05-01T00:00
365.03	21	7665.629999999999	2016-05-01T00:00
354.0	20	7080.0	2016-05-01T00:00
253.93	14	3555.02	2016-05-01T00:00
352.43	20	7048.6	2016-05-02T00:00
379.88	22	8357.36	2016-05-02T00:00
227.11	13	2952.4300000000003	2016-05-02T00:00
342.39	20	6847.799999999999	2016-05-02T00:00
440.67	25	11016.75	2016-05-03T00:00
361.4	21	7589.4	2016-05-03T00:00
396.64	23	9122.72	2016-05-03T00:00
346.12	20	6922.4	2016-05-03T00:00
437.66	25	10941.5	2016-05-04T00:00
238.64	14	3340.96	2016-05-04T00:00
307.48	18	5534.64	2016-05-04T00:00