

## **Tutorial Visual JSF**

O objetivo deste tutorial é introduzir o assunto Visual JSF, utilizando a IDE NetBeans 6.1. Será construída uma aplicação simples sobre duas tabelas relacionadas de 1 para muitos. É necessário ter JDK 1.6+, NetBeans 6+, MySQL 5+.

É necessário para desenvolvimento um computador com 1 GB de memória RAM, e um processador Pentium 4 ou superior.

Nesta tecnologia a vantagem é o desenvolvimento rápido para Web, com boa qualidade de interface e tratamento de código integrado pelo framework JSF. A maior desvantagem é em relação à performance, ainda baixa para as plataformas atuais.

### **Script de banco de dados**

Com o MySQL Query Browser aberto e logado como “root” iremos abrir uma janela de Script (File -> New Script Tab) e rodar o script a seguir:

```
CREATE DATABASE DBJSF;

USE DBJSF;

CREATE TABLE Pais(
  idPais INTEGER AUTO_INCREMENT,
  noPais VARCHAR(50) NOT NULL,
  CONSTRAINT pk_pais PRIMARY KEY (idPais)
)Engine=InnoDB;

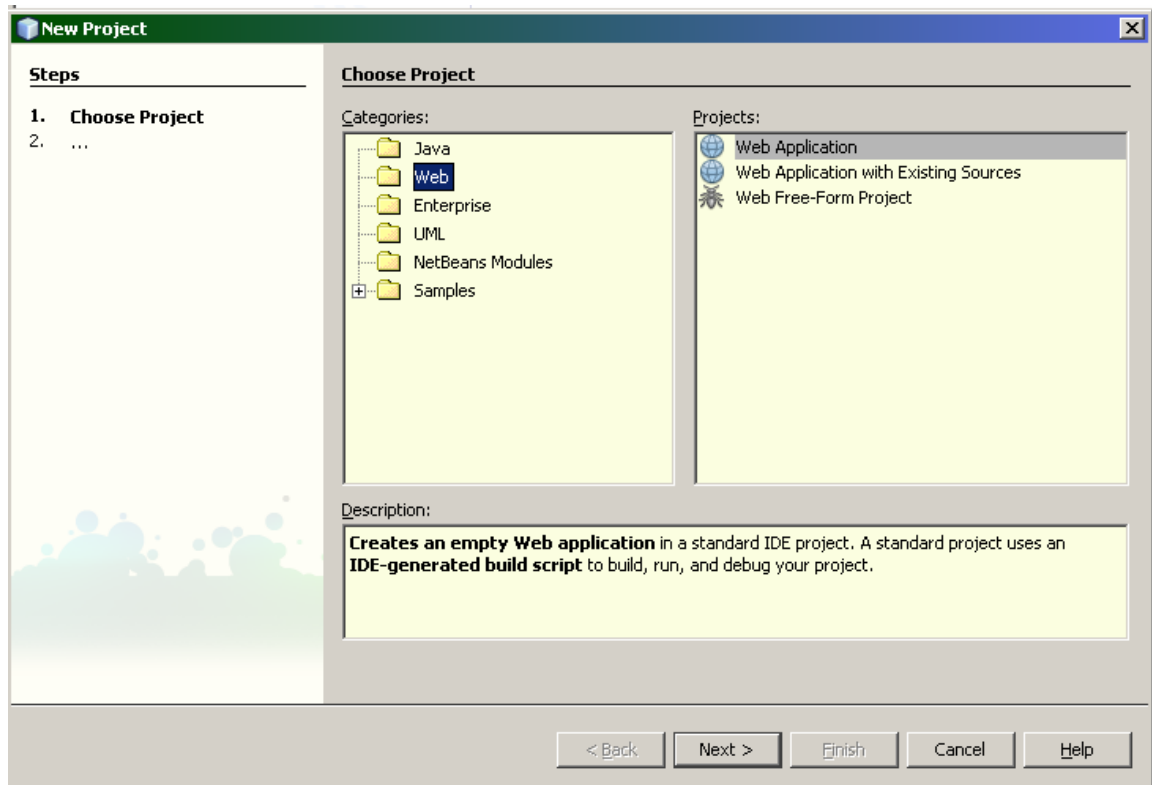
CREATE TABLE Estado(
  idEstado INTEGER AUTO_INCREMENT,
  cdPais INTEGER NOT NULL,
  noEstado VARCHAR(50) NOT NULL,
  nrPopulacao INTEGER NOT NULL,
  CONSTRAINT pk_estado PRIMARY KEY (idEstado),
  CONSTRAINT fk_pais_estado FOREIGN KEY (cdPais) REFERENCES
Pais(idPais)
)Engine=InnoDB;

INSERT INTO Pais (noPais) VALUES ('Brasil'), ('Argentina'),
('Uruguai');

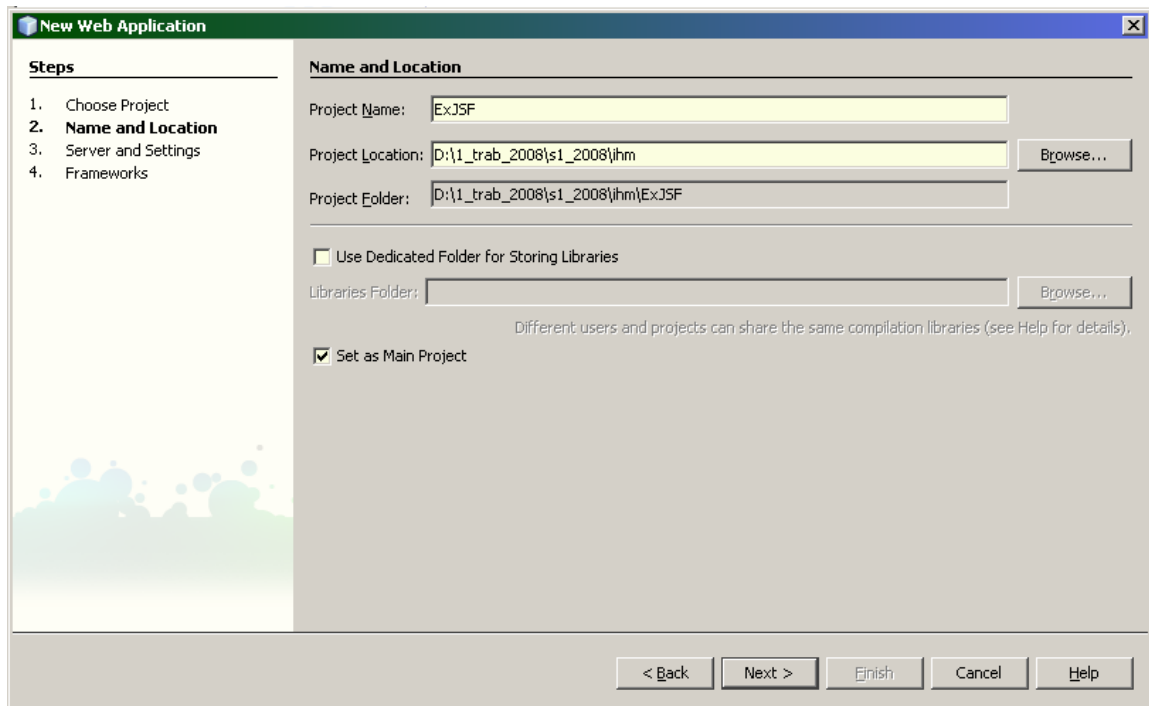
INSERT INTO Estado (noEstado, cdPais,nrPopulacao) VALUES
('Distrito Federal', 1, 2000000),
('Minas Gerais', 1, 16000000),
('Rio de Janeiro', 1, 10000000),
('Catamarca', 2, 5000000),
('Misiones', 3, 1000000);
```

## Criação de um projeto para JSF no NetBeans 6.1

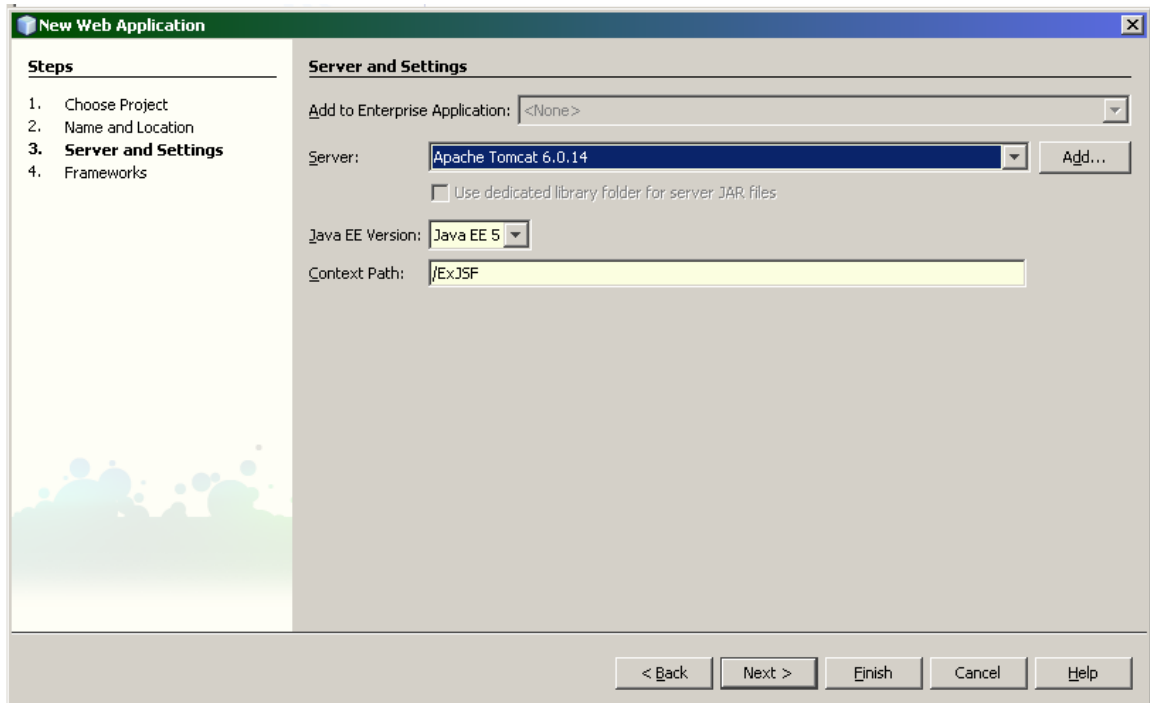
1) Acesse File -> New Project:



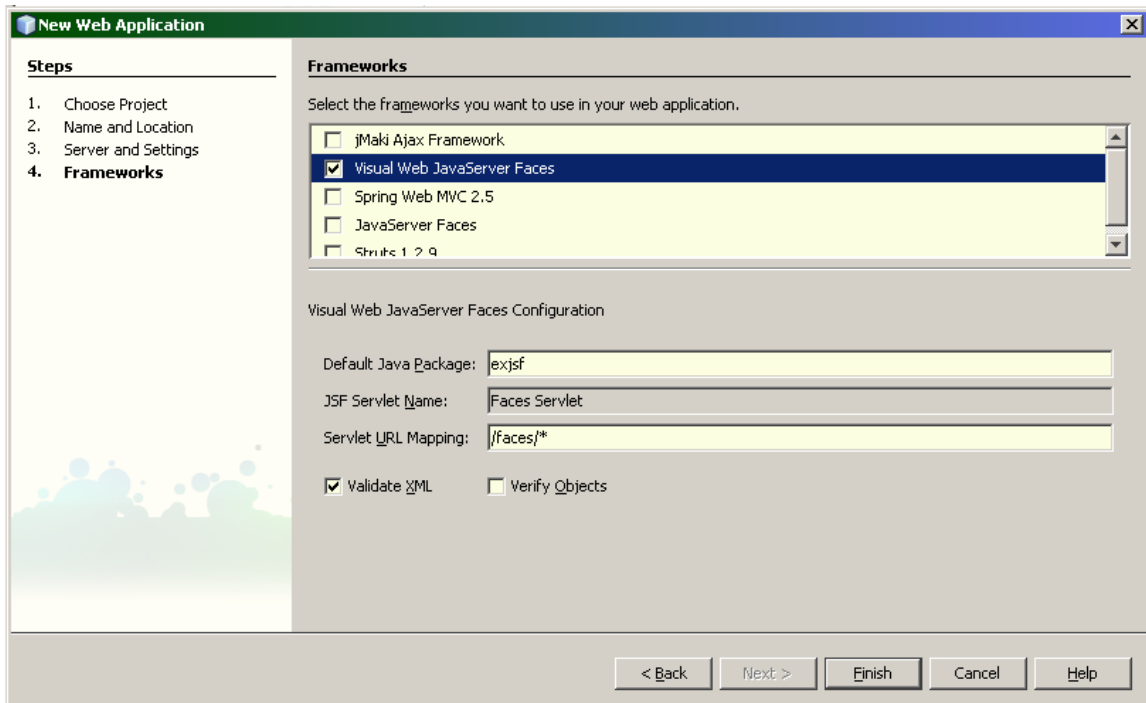
2) Escolha um nome e um local (path) para gravar o projeto:



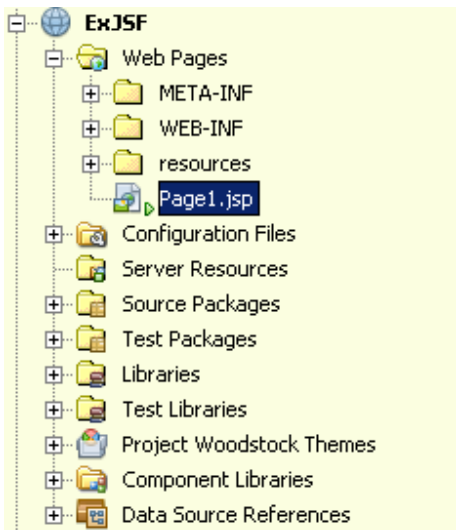
### 3) Escolha o servidor web Tomcat:



- 4) Marque o Framework “Visual Web JavaServer Faces” e finalize o wizard de criação de um projeto web que utiliza visual JSF:

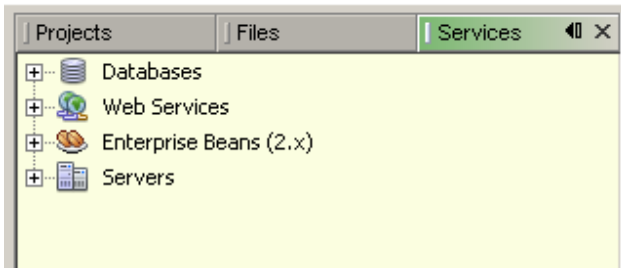


A árvore de diretório deverá se parecer com:

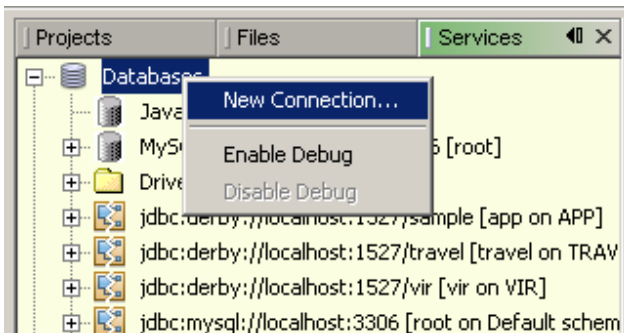


Acessando o banco de dados MySQL no NetBeans 6.1:

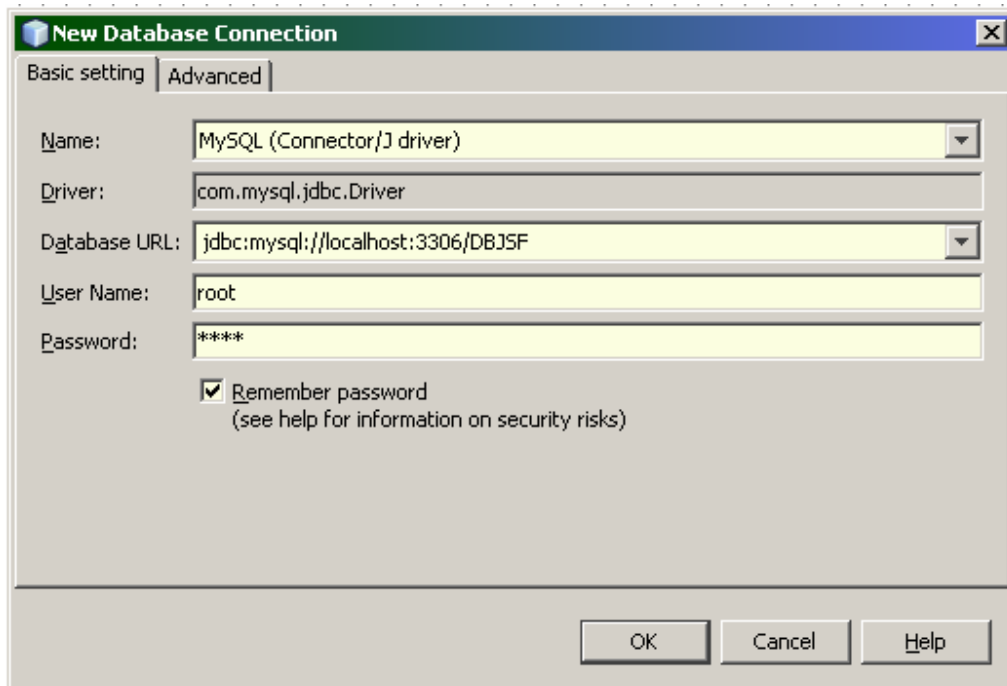
- 1) Acesse a Guia Services (no canto superior esquerdo) ou Window -> Services (Ctrl + 5):



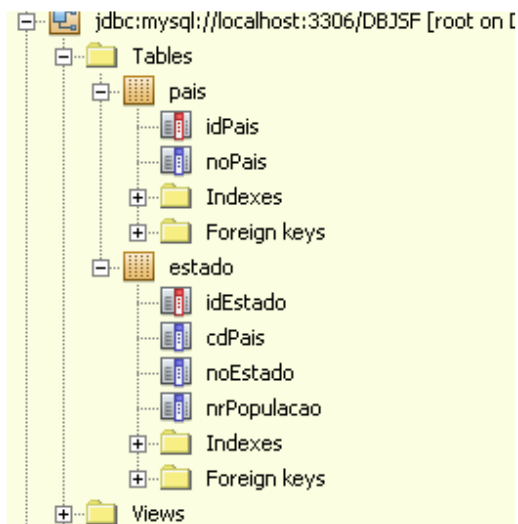
- 2) Abra o nó "Databases" e com botão inverso do mouse crie uma nova conexão:



- 3) Substitua os parâmetros do wizard pelos da sua configuração de ambiente. Como já fornecemos o Schema o segundo passo é desnecessário, só precisa clicar no OK mais uma vez:

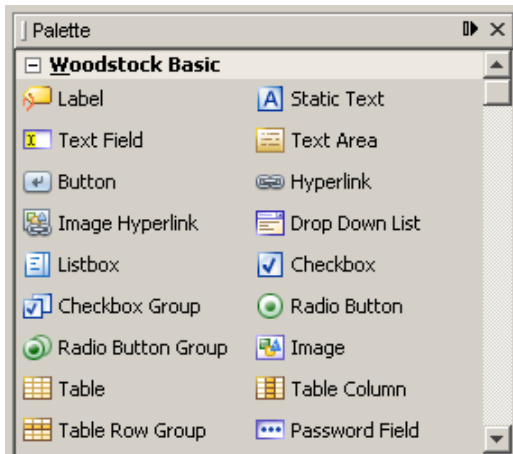


- 4) Abra o nó relativo ao schema “DBJSF” e verifique se as tabelas Pais e Estado estão disponíveis:

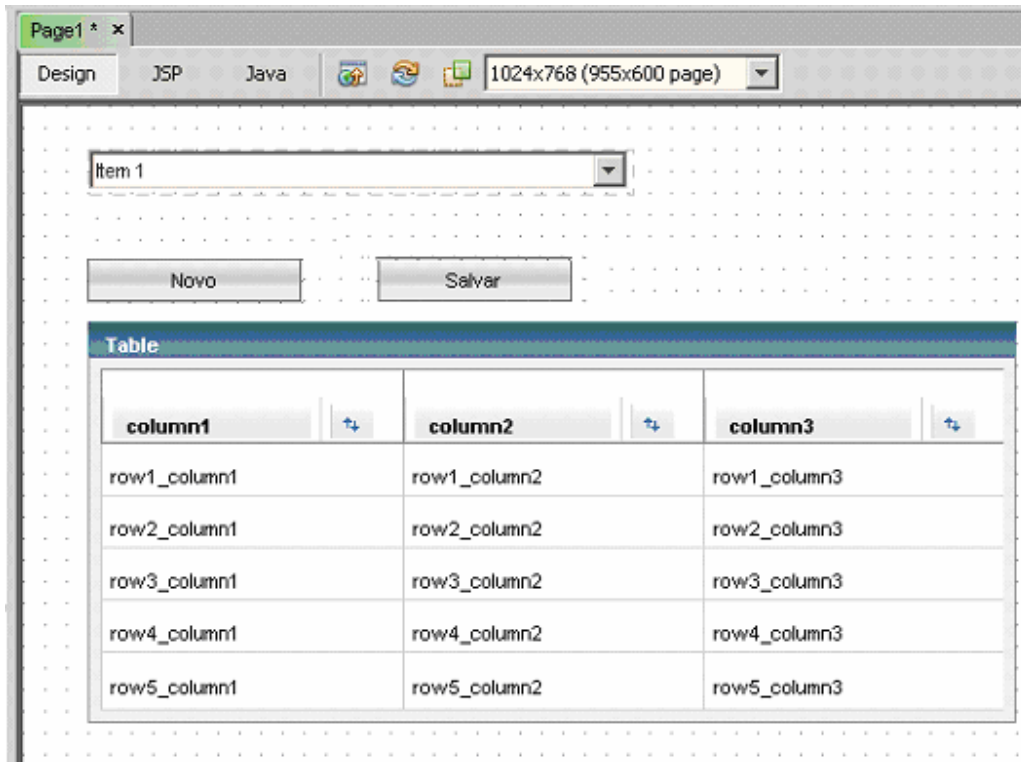


## Criação do layout de tela

Nesta etapa é demonstrado o poder do visual JSF no NetBeans 6.1, pois é possível a criação de páginas web funcionais no esquema de drag and drop. É disponibilizada uma série de componentes visuais na paleta (Window -> Palette) como na figura abaixo:

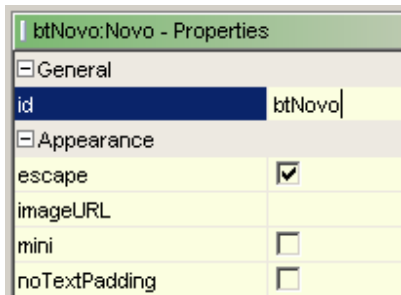


Utilizaremos a Page1.jsp na guia "Design" incluindo um "Drop Down List", uma "Table" e dois "Button" e uma "Message Group" como no desenho abaixo:



## Renomeando os componentes

Para renomear os componentes basta selecioná-lo, clicando sobre o mesmo uma vez e acessando a guia de Properties (Windows -> Properties) e alterando o nome na propriedade “id”, como na figura a seguir:



A relação de nomes é mostrada a seguir:

Drop Down List: ddPais  
Table: tbEstado  
Buttons: btSalavar, btApagar, btNovo

## Acoplado dados aos componentes binding

O componente ddPais será utilizado para mostrar o “noPais” e guardar internamente o “idPais”, isso será feito para que possamos consultar os dados de estado que pertencem ao país escolhido.

Inicialmente devemos acessar a guia “Services” e posteriormente arrastar a tabela “Pais” para dentro do desenho do “ddPais”, automaticamente será configurado para acesso aos dados da tabela “Pais”. Para verificar o resultado até o momento basta clicar em F6.

Faça o mesmo procedimento para o componente “table”, arrastando a tabela “Estado” para dentro do componente. Retire somente a coluna “cd\_pais”, pois ela já esta representada pelo Drop Down List. O resultado será próximo do seguinte:

abc

**Mensagens do sistema**  
 ■ [Lista de todos os resumos de mensagens](#)

Novo      Salvar

Tabela de Estados		
Identificador	Estado	População
123	abc	123
123	abc	123
123	abc	123

Os componentes ficaram com valores 123 quando o tipo for numérico e abc quando forem caracteres.

### Personalizando as colunas da tabela

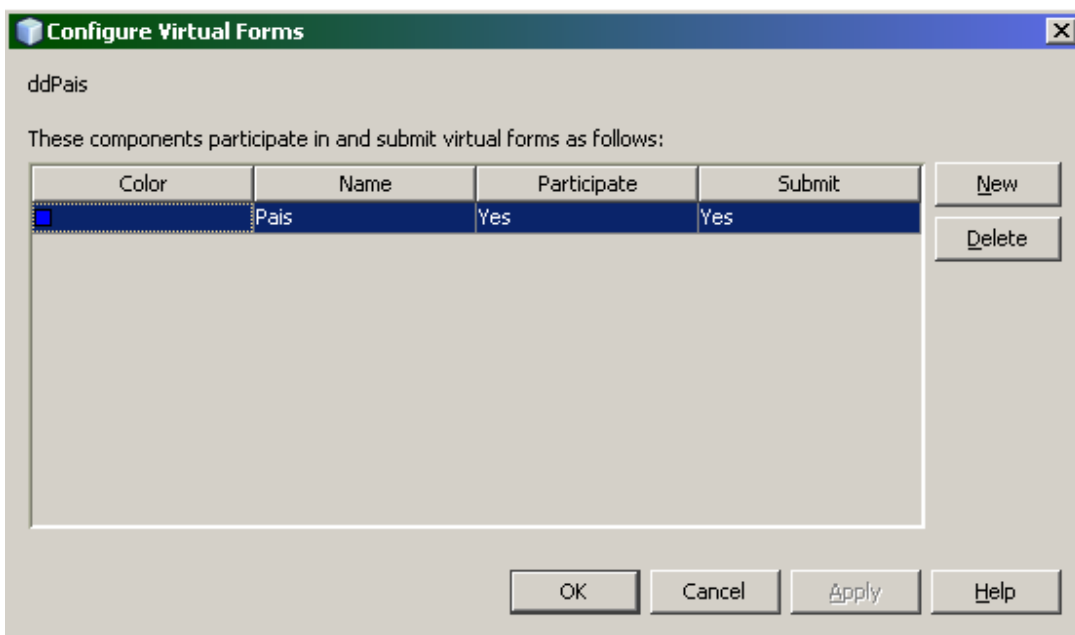
Para alterar os títulos das colunas é necessário selecionar a coluna, isso pode ser feito clicando uma vez sobre o label com o nome da coluna e acessando na guia de properties a propriedade headerText. Para trocar o título da tabela basta acessar a propriedade "title". O layout será parecido com:


Tabela de Estados		
Identificador	Estado	População
123	abc	123
123	abc	123
123	abc	123

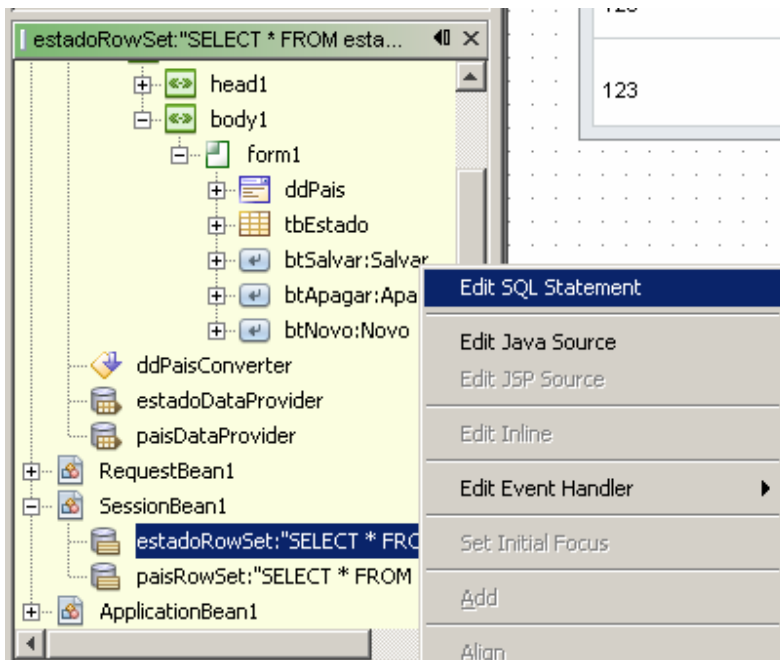
## Incluindo código para selecionar os estados somente do país que estiver marcado no Drop Down List

**Observação:** Na versão 6.1 é necessário selecionar o menu de contexto do Drop Down List e acessar a opção (Add Binding Attribute). Isto fará com que as propriedades do componente gráfico estejam disponíveis para acesso dentro do código Java. Para cada componente que for manipulado do lado do servidor (Java) é necessário fazer este procedimento.

- 1) Acessando o menu de contexto selecione “Auto-submit on Change”, isto fará com que quando for selecionado um novo país o formulário como um todo seja recarregado, ou seja, será processada alguma coisa no servidor, no caso, desejamos filtrar os estados pelo país selecionado.
- 2) Configure um formulário virtual para o Drop Down List acessando o menu de contexto e selecionando “Configure Virtual Forms”, inclua um novo com a seguinte parametrização:



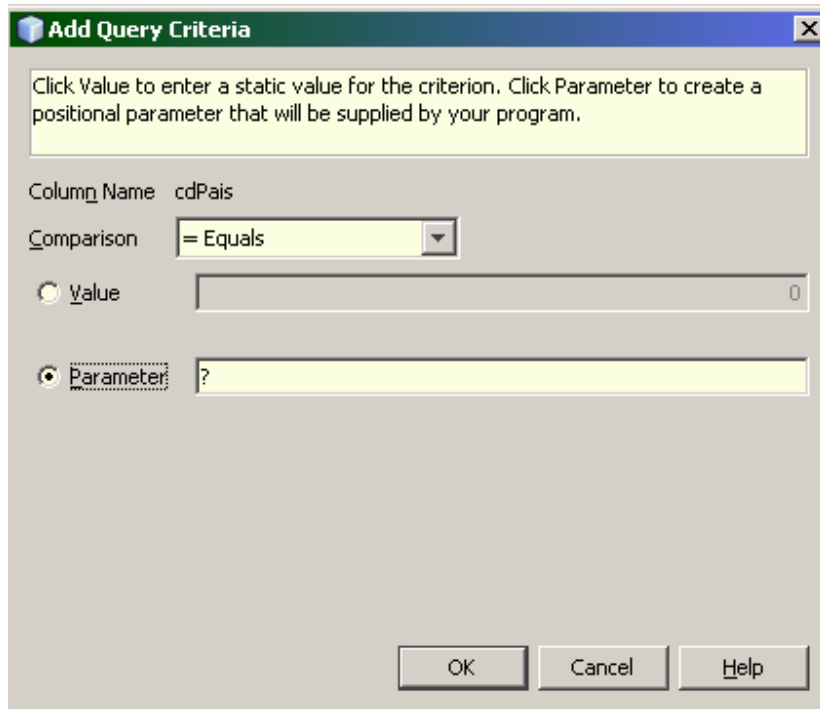
- 3) Para visualizar os formulários virtuais é necessário clicar no botão  que se encontra na barra superior da pagina editada.
- 4) Para adicionar um parâmetro a consulta que traz os estados é necessário ir até a janela de Navigator e acessar a SessionBean1. Embaixo do session temos o conjunto de linhas de estado que iremos “Edit SQL Statement”.



- 5) Inclua um critério de seleção para o campo “cdPais” que representa a chave estrangeira de Pais em Estado.

Column	Alias	Table	Output	Sort Type	Sort Order	Criteria	Order
idEstado		estado	<input checked="" type="checkbox"/>				
cdPais		estado	<input checked="" type="checkbox"/>				
noEstado		estado	<input checked="" type="checkbox"/>				
nrPopulacao		estado	<input checked="" type="checkbox"/>				

- 6) Marque o radio “Parameter”, que significa um parâmetro sendo usado em critério de igualdade para “cdPais”.



- 7) Note que apareceu a clausula WHERE no comando de busca.
- 8) Feche o editor de SQL.
- 9) Adicione código Java ao evento processValueChanged do Drop Down List clicando duas vezes sobre o componente. A seguir o código:

```

public void ddPais_processValueChanged(ValueChangeEvent event) {
    Object selectedPaisId = ddPais.getSelected();
    try {
        // Sincronizando o provedor de dados de pais com o que
aparece no
        // Drop Down List
        paisDataProvider.setCursorRow(
            paisDataProvider.findFirst("Pais.idPais",
                selectedPaisId));

        // Passando o parâmetro para filtrar somente os estados do
// país selecionado
        getSessionBean1().getEstadoRowSet().setObject(1,
selectedPaisId);
        estadoDataProvider.refresh();
    } catch (Exception ex) {
        error("Não posso selecionar o País: " + selectedPaisId);
    }
}

```

10) Adicione código Java ao método prerender que já existe. Este código servirá para mostrar os estados do primeiro país que aparece na lista Drop Down List:

```
public void prerender() {
    if (ddPais.getSelected() == null) {
        Object firstPaisId = null;
        try {
            // Setando estados para o primeiro país da lista Drop
Down
            paisDataProvider.cursorFirst();
            firstPaisId
            =
paisDataProvider.getValue("Pais.idPais");
            ddPais.setSelected(firstPaisId);
            getSessionBean1().getEstadoRowSet().setObject(
                1, firstPaisId);
            estadoDataProvider.refresh();
        } catch (Exception ex) {
            error("Não posso selecionar o País: " + firstPaisId);
        }
    }
}
```

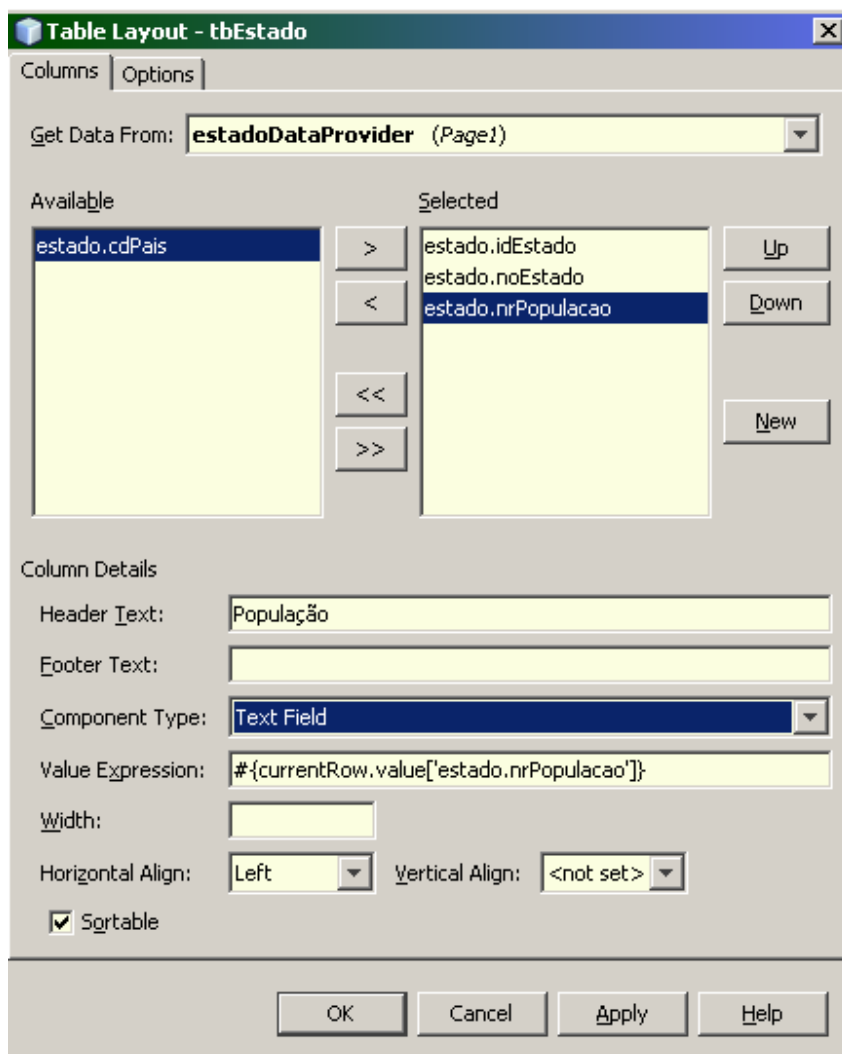
11) Teste a aplicação com F6 e veja se a tabela seleciona os estados do país que aparece no Drop Down List

### Modificando os componentes de Detail da Table

1) Selecione o menu de contexto Table Layout do componente table.



2) Troque a propriedade “Component Type” para TextField para as colunas de Estado “noestado” e população “nrPopulacao”.

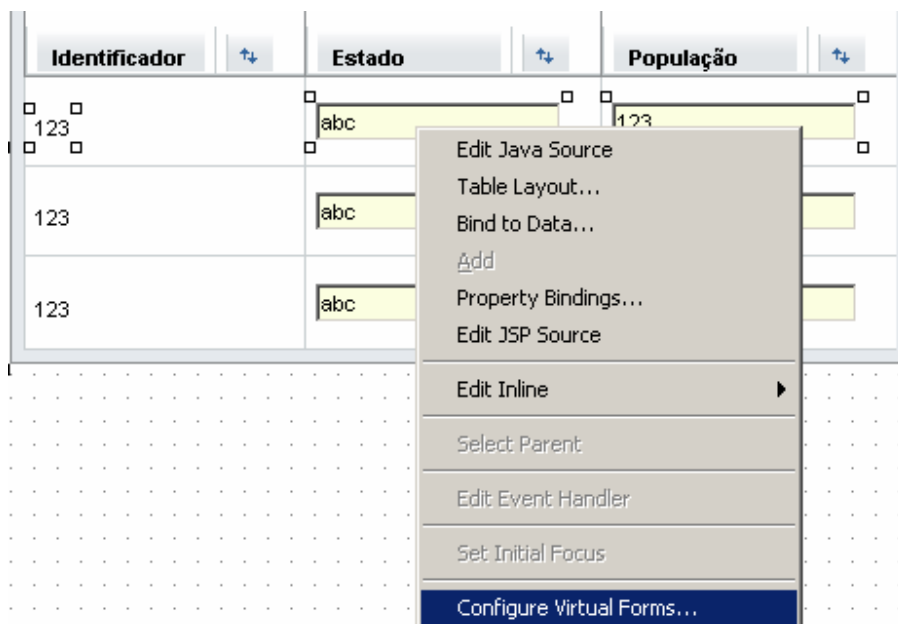


- 3) Teste a aplicação F6 e veja se os componentes das duas colunas foi alterado para TextField.

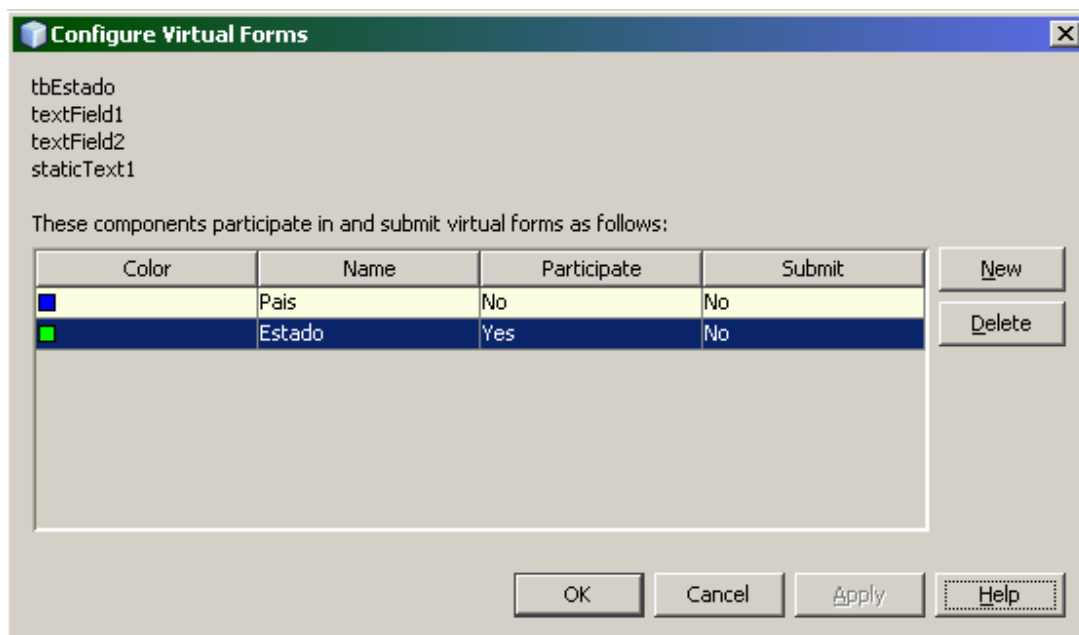
### **Configurando mais um formulário virtual**

Este formulário servirá para manipular os dados de estado. Será necessário para processarmos as operações vinculadas aos botões.

- 1) Selecione com CTRL + clique os componentes internos da primeira linha da tabela e clique com botão direito selecionando “Configure Virtual Forms”



2) Configure o novo formulário da seguinte forma.



### Incluindo código do botão novo

Inicialmente deve acrescentar um referência import:

```
import com.sun.data.provider.RowKey;
```

Depois no design da página clique duas vezes no botão de novo e insira o seguinte código:

```
public String btNovo_action() {  
    // Criando um novo registro para inserção de dados  
    // preenchida com dados vazios  
    try {  
        RowKey rk = estadoDataProvider.appendRow();  
        estadoDataProvider.setCursorRow(rk);  
  
        estadoDataProvider.setValue("Estado.idEstado",          new  
Integer(0));  
        estadoDataProvider.setValue("Estado.cdPais",  
ddPais.getSelected());  
        estadoDataProvider.setValue("Estado.noEstado", "");  
        estadoDataProvider.setValue("Estado.nrPopulacao",      new  
Integer(0));  
    } catch (Exception ex) {  
        error(ex.getMessage());  
    }  
    return null;  
}
```

### Incluindo código no botão salvar

Clique duas vezes no botão de salvar e insira o seguinte código:

```
public String btSalvar_action() {  
    try {  
        estadoDataProvider.commitChanges();  
        estadoDataProvider.refresh();  
        form1.discardSubmittedValues("Estado");  
    } catch (Exception ex) {  
        error("Erro :" + ex.getMessage());  
    }  
    return null;  
}
```