

# **Programação Web III**

## **EJB - Enterprise Java Beans**

### **JSP – Java Server Pages**

#### **Aula 2: Exemplos Básicos usando o NetBeans e GlassFish**

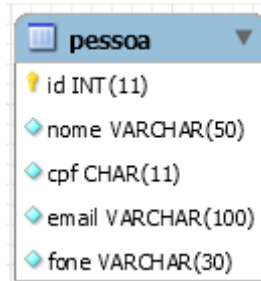
#### **Sumário**

1. Usando a base de dados da Aula 1 .....	2
2. Criando o Enterprise Java Bean .....	2
3. Criando a interface com JSP para acessar o EJB - Enterprise Java Bean.....	13
4. Criando uma aplicação Java Desktop para acessar o EJB - Enterprise Java Bean.....	19

## 1. Usando a base de dados da Aula 1

Vamos usar a base de dados criada na Aula 1 para realizar o cadastro de Pessoas, utilizando um Java Bean, que será acessado tanto por uma página JSP quanto por uma aplicação Desktop Java.

Vamos usar o usuário **iffar** com senha **iffar**, para acessar a base de dados com o nome **iffar**, utilizando a tabela:



## 2. Criando o Enterprise Java Bean

Enterprise JavaBeans (EJB) é um componente da plataforma Java EE que roda em um container de um servidor de aplicação (GlassFish, Tomcat ou JBoss). Seu principal objetivo consiste em fornecer um desenvolvimento rápido e simplificado de aplicações Java, com base em componentes distribuídos, transacionais, seguros e portáteis.

Vamos criar um projeto para desenvolver nosso EJB:

1. No Netbeans, crie um Novo Projeto => Java EE => Módulo EJB.

2. Dê o nome do projeto: **EJBAulaWeb**

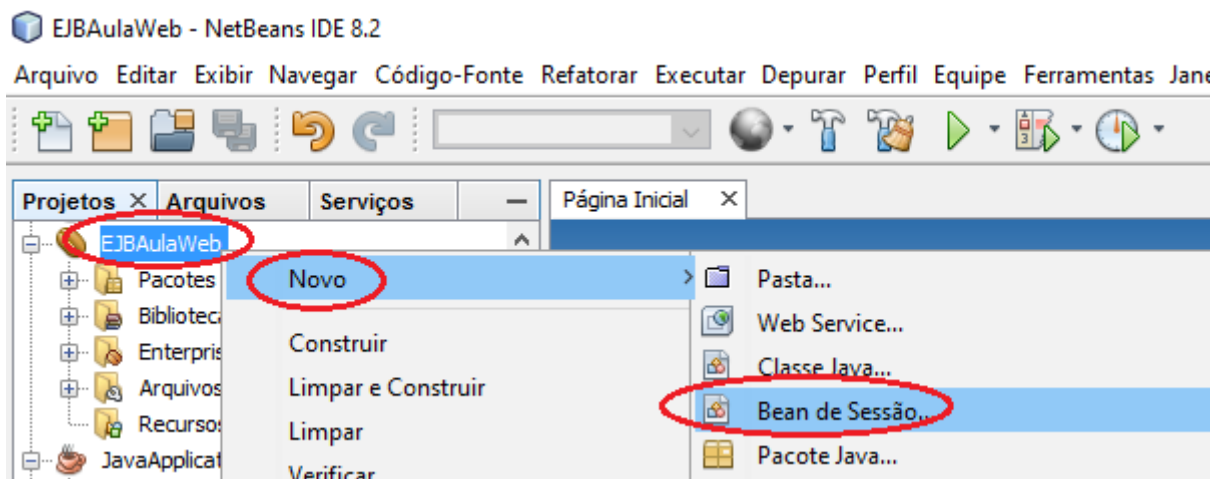
Precisamos criar também um projeto com a biblioteca para a interface do módulo EJB poder ser utilizada em outros projetos, como uma aplicação desktop, por exemplo.

3. Crie um Novo Projeto => Java = Biblioteca de Classe Java.

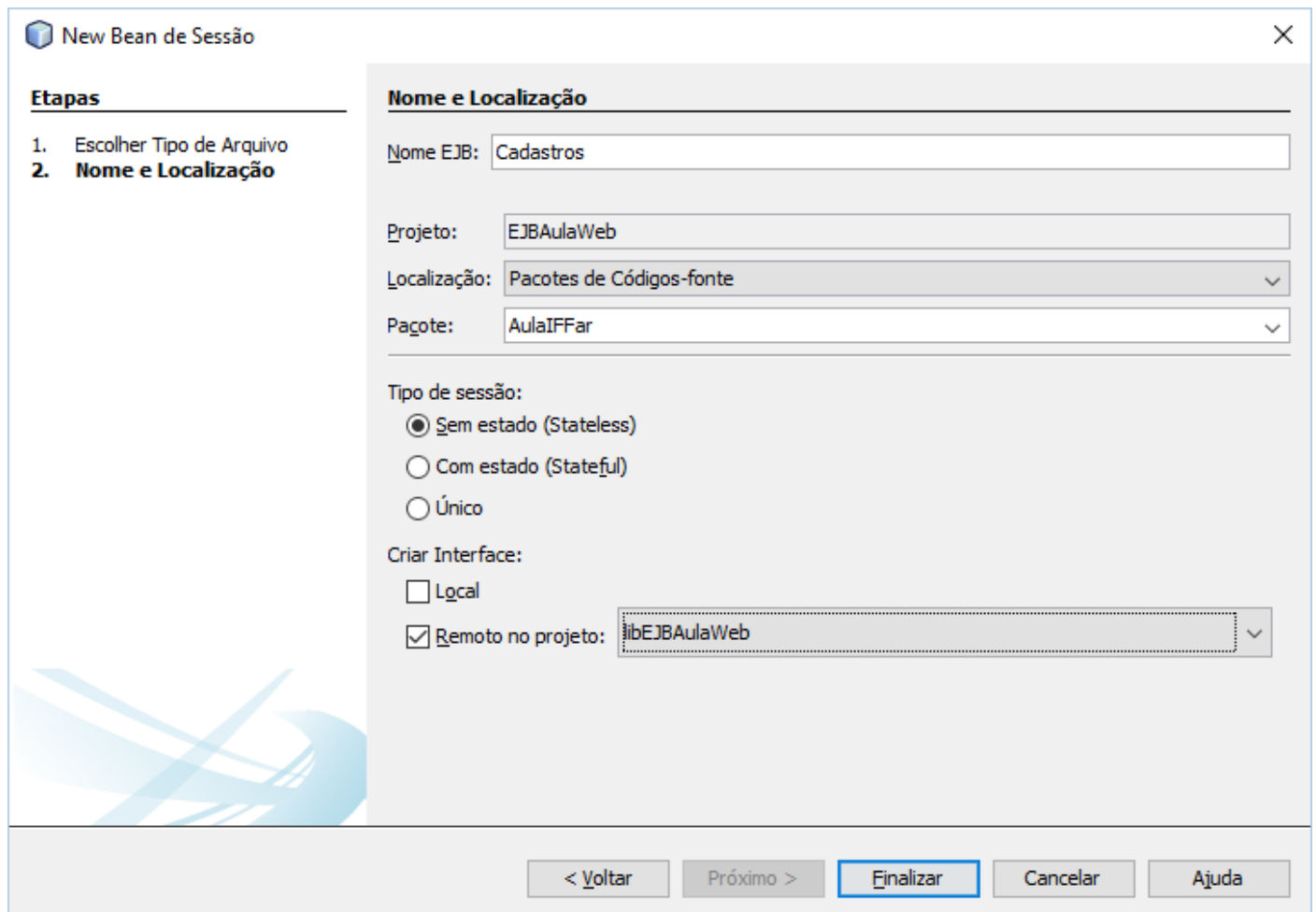
4. Dê o nome do projeto de **libEJBAulaWeb**

Acesse o projeto EJBAulaWeb e vamos criar nosso Java Bean.

5. Clique sobre o projeto EJBAulaWeb => Novo => Bean de Sessão

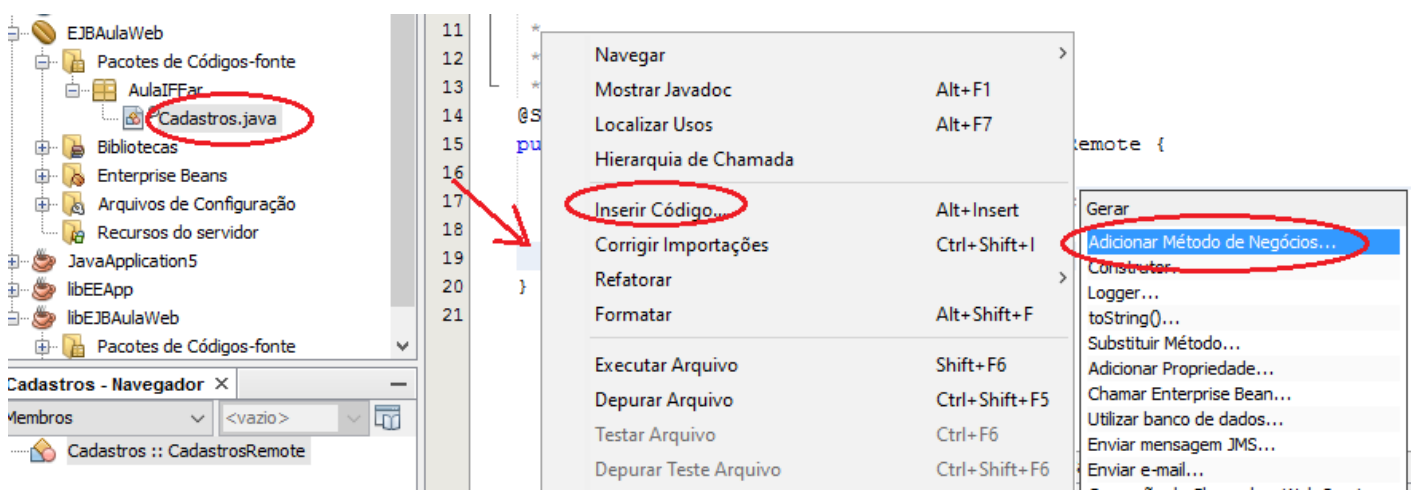


6. Dê o nome **Cadastros**, nome do pacote **AulaIFFar**, Tipo de Sessão: **Sem estado**, marque Criar Interface: Remoto no projeto **libEJBAulaWeb**



Agora precisamos criar as regras de negócio, com as operações que o EJB vai realizar.

7. Acesse o código do Cadastros.java e clique com o botão direito do mouse sobre o código escolha: Inserir Código => Adicionar Método de Negócios



8. Vamos criar os 5 métodos de negócio, utilizando sempre o Assistente:

- String cadastrarPessoa(String nome, String cpf, String email, String fone);

Adicionar Método de Negócios...

Nome: cadastrarPessoa

Tipo de Retorno: java.lang.String

Procurar...

Parâmetros

Exceções

Nome	Tipo	Final
nome	java.lang.String	<input type="checkbox"/>
cpf	java.lang.String	<input type="checkbox"/>
email	java.lang.String	<input type="checkbox"/>
fone	java.lang.String	<input type="checkbox"/>

Adicionar

Remover

Para Cima

Para baixo

Uso na interface:

☐ Local
☒ Remoto
☐ Ambos

OK

Cancelar

- String alterarPessoa(String id, String nome, String cpf, String email, String fone);

Adicionar Método de Negócios...

Nome: alterarPessoa

Tipo de Retorno: java.lang.String

Procurar...

Parâmetros

Exceções

Nome	Tipo	Final
id	java.lang.String	<input type="checkbox"/>
nome	java.lang.String	<input type="checkbox"/>
cpf	java.lang.String	<input type="checkbox"/>
email	java.lang.String	<input type="checkbox"/>
fone	java.lang.String	<input type="checkbox"/>

Adicionar

Remover

Para Cima

Para baixo

Uso na interface:

☐ Local
☒ Remoto
☐ Ambos

OK

Cancelar

- String excluirPessoa(String id);

Adicionar Método de Negócios...

Nome:

Tipo de Retorno:

Parâmetros Exceções

Nome	Tipo	Final
id	java.lang.String	<input type="checkbox"/>

Uso na interface: ☐ Local ☒ Remoto ☐ Ambos

- String listarPessoa(String id);

Adicionar Método de Negócios...

Nome:

Tipo de Retorno:

Parâmetros Exceções

Nome	Tipo	Final
id	java.lang.String	<input type="checkbox"/>

Uso na interface: ☐ Local ☒ Remoto ☐ Ambos

- Adicionar Método de Negócios...**

Nome:

Tipo de Retorno:

☒ Parâmetros ☐ Exceções

Nome	Tipo	Final

Uso na interface: ☐ Local ☒ Remoto ☐ Ambos

## Cadastrors.java

6

```

@Override
public String listarTodos() {
    return null;
}
}

```

Observe, que esse código deve ter sido gerado pelo assistente e não digitado, pois também foi criada a interface do nosso EJB no projeto libEJBAulaWeb, que deve estar dessa forma:

### CadastrosRemote.java

```

package AulaIFFar;

import javax.ejb.Remote;

@Remote
public interface CadastrosRemote {

    String cadastrarPessoa(String nome, String cpf, String email, String fone);

    String alterarPessoa(String id, String nome, String cpf, String email, String fone);

    String excluirPessoa(String id);

    String listarPessoa(String id);

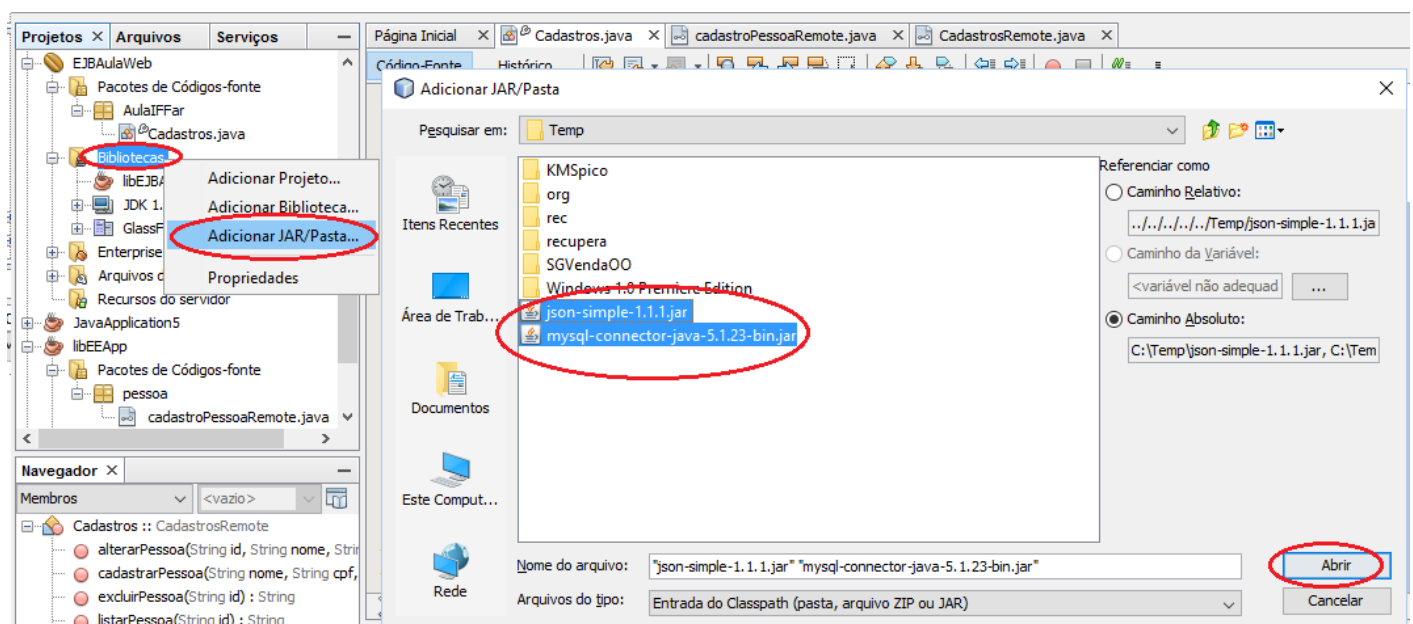
    String listarTodos();

}

```

Cada alteração no EJB deve refletir na biblioteca de interface.

9. Inclua as bibliotecas do JDBC do MySQL (mysql-connector-java-5.1.23-bin.jar) e do json-simple (json-simple-1.1.1.jar) no projeto. Aplicações Java EE/GlassFish as vezes se perdem na hora de encontrar os arquivos se os mesmos estiverem em pastas com espaço no nome ou caracteres especiais, como Arquivos de Programas, Program File, Área de Trabalho. Para evitar isso, copie o arquivo da biblioteca numa pasta sem espaços e sem caracteres especiais.



Agora vamos implementar os métodos do EJB com as operações que vão acessar a base de dados.

10. Acesse o arquivo Cadastros.java do EJB e altere o método cadastrarPessoa, para que fique com esse código:

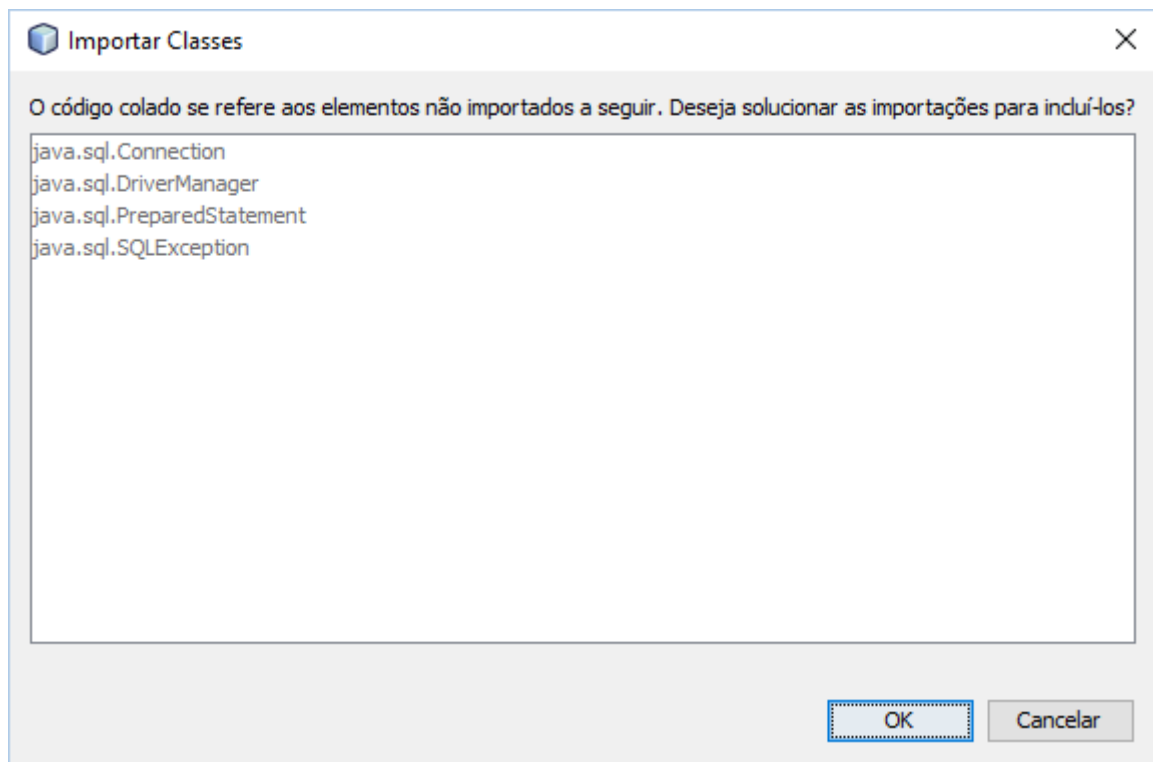
```
public String cadastrarPessoa(String nome, String cpf, String email, String fone) {  
  
    try {  
        Class.forName("com.mysql.jdbc.Driver");  
        Connection con =  
        DriverManager.getConnection("jdbc:mysql://localhost/iffar?user=iffar&password=iffar");  
        PreparedStatement pstmt = con.prepareStatement(" INSERT INTO pessoa (nome, cpf,  
email, fone) VALUES( ?, ?, ?, ?)");  
        pstmt.setString(1, nome);  
        pstmt.setString(2, cpf);  
        pstmt.setString(3, email);  
        pstmt.setString(4, fone);  
        pstmt.executeUpdate();  
    } catch (SQLException err) {  
        String r = err.getMessage();  
        String s1 = "" + (char) 10;  
        String s2 = "" + (char) 13;  
        r = r.replaceAll(s1, " ");  
        r = r.replaceAll(s2, " ");  
        return "ERRO: " + r;  
    } catch (ClassNotFoundException ex) {  
        String r = ex.getMessage();  
        String s1 = "" + (char) 10;  
        String s2 = "" + (char) 13;  
        r = r.replaceAll(s1, " ");  
        r = r.replaceAll(s2, " ");  
        return "ERRO: " + r;  
    }  
    return "OK";  
}
```

Observe que a última chave (em vermelho) faz parte da estrutura do programa e não deve ser copiada.

Observe também que o método pega os 4 parâmetros enviados (nome, cpf, email e fone) e insere na base de dados, devolvendo uma resposta “OK” ou uma mensagem com o ERRO que ocorreu.

Confirme as importações das classes necessárias:





11. Ainda no Cadastros.java do EJB, altere o método alterarPessoa, para que fique com esse código:

```
public String alterarPessoa(String id, String nome, String cpf, String email, String fone) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost/iffar?user=iffar&password=iffar");
        PreparedStatement pstmt = con.prepareStatement("update pessoa set nome=?, cpf=?,
        email=?, fone=? where id = ? ");
        pstmt.setString(1, nome);
        pstmt.setString(2, cpf);
        pstmt.setString(3, email);
        pstmt.setString(4, fone);
        pstmt.setString(5, id);
        System.out.println("Alterando: "+id+nome);
        pstmt.executeUpdate();
    } catch (SQLException err) {
        String r = err.getMessage();
        String s1 = "" + (char) 10;
        String s2 = "" + (char) 13;
        r = r.replaceAll(s1, " ");
        r = r.replaceAll(s2, " ");
        return "ERRO: " + r;
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Cadastros.class.getName()).log(Level.SEVERE, null, ex);
    } catch (ClassNotFoundException ex) {
        String r = ex.getMessage();
        String s1 = "" + (char) 10;
        String s2 = "" + (char) 13;
        r = r.replaceAll(s1, " ");
        r = r.replaceAll(s2, " ");
        return "ERRO: " + r;
    }
    return "OK";
}
```

12. Altere também o método excluirPessoa, para que fique com esse código:

```
public String excluirPessoa(String id) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
        DriverManager.getConnection("jdbc:mysql://localhost/iffar?user=iffar&password=iffar");
        PreparedStatement pstmt = con.prepareStatement(" delete from pessoa where id = ?");
        pstmt.setString(1, id);
        pstmt.executeUpdate();
    } catch (SQLException err) {
        String r = err.getMessage();
        String s1 = "" + (char) 10;
        String s2 = "" + (char) 13;
        r = r.replaceAll(s1, " ");
        r = r.replaceAll(s2, " ");
        return "ERRO: " + r;
    } catch (ClassNotFoundException ex) {
        String r = ex.getMessage();
        String s1 = "" + (char) 10;
        String s2 = "" + (char) 13;
        r = r.replaceAll(s1, " ");
        r = r.replaceAll(s2, " ");
        return "ERRO: " + r;
    }
    return "OK";
}
```

13. Altere também o método listarPessoa, para que fique com esse código:

```
public String listarPessoa(String id) {
    JSONObject obj = new JSONObject();

    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/iffar?user=iffar&password=iffar");
        PreparedStatement pstmt = con.prepareStatement("select * from pessoa where id = ? or
cpf = ?");
        pstmt.setString(1, id);
        pstmt.setString(2, id);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            obj.put("nome", rs.getString("nome"));
            obj.put("cpf", rs.getString("cpf"));
            obj.put("email", rs.getString("email"));
            obj.put("fone", rs.getString("fone"));
        }
    } catch (SQLException err) {
        return ("Erro");
    } catch (ClassNotFoundException ex) {
        String r = ex.getMessage();
        String s1 = "" + (char) 10;
        String s2 = "" + (char) 13;
        r = r.replaceAll(s1, " ");
        r = r.replaceAll(s2, " ");
        return "ERRO: " + r;
    }
    return obj.toJSONString();
}
```

Observe que o método retorna um objeto JSON.

14. Finalmente... altere o método listarTodos, para que fique com esse código:

```
public String listarTodos() {
    List l1 = new LinkedList();
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/iffar?user=iffar&password=iffar");
        PreparedStatement pstmt = con.prepareStatement("select * from pessoa order by
nome");
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            Map m1 = new LinkedHashMap();
            m1.put("id", rs.getString("id"));
            m1.put("nome", rs.getString("nome"));
            l1.add(m1);
        }
    } catch (SQLException err) {
        return ("Erro");
    } catch (ClassNotFoundException ex) {
        String r = ex.getMessage();
        String s1 = "" + (char) 10;
        String s2 = "" + (char) 13;
        r = r.replaceAll(s1, " ");
        r = r.replaceAll(s2, " ");
        return "ERRO: " + r;
    }
    String jsonString;
    jsonString = JSONValue.toJSONString(l1);
    return jsonString;
}
```

Observe que o método retorna uma string, com um objeto JSON contendo todos os registros.

Pronto! O EJB com a lógica do negócio está pronto para ser implantado, porém ele não tem interface. Vamos então criar a interface com JSP. Se quiser, pode implantar o EJB para garantir que ele está pronto. Clique sobre o projeto e escolha **Implantar**.

Aproveite e Construa a biblioteca do projeto **libEJBAulaWeb**, clicando sobre o projeto e escolhendo: **Construir**.

### 3. Criando a interface com JSP para acessar o EJB - Enterprise Java Bean

Vamos então criar a interface com JSP para acessar o EJB que tem a lógica do negócio.

1. Crie um Novo Projeto => Java Web => Aplicação Web -> Nome do Projeto: **AulaWeb**, sem nenhum Framework
2. Inclua nas bibliotecas o projeto **libEJBAulaWeb**
3. Inclua também a biblioteca **json-simple-1.1.1.jar**
4. No projeto AulaWeb, clique com o botão direito e escolha: Novo => JSP -> **cadPessoa**
5. No projeto AulaWeb, clique com o botão direito e escolha: Novo => JSP -> **alteraPessoa**
6. No projeto AulaWeb, clique com o botão direito e escolha: Novo => JSP -> **listaPessoa**
7. Altere os arquivos criados, para que fiquem com o seguinte código:

#### **cadPessoa.jsp**

```
<%@page import="AulaIFFar.CadastradosRemote"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Cadastro de Pessoa</title>
  </head>
  <body>
    <h1>Cadastro de Pessoa</h1>
    <div>
      <%@page language = "java" %>
      <%@page import = "java.util.*" %>
      <%@page import = "java.util.Properties" %>
      <%@page import = "java.util.logging.Level" %>
      <%@page import = "java.util.logging.Logger" %>
      <%@page import = "javax.naming.InitialContext" %>
      <%@page import = "javax.naming.NamingException" %>

      <%
        if (request.getParameter("nome") != null) {
          InitialContext ctx = null;
          CadastrosRemote cp;
          cp = null;
          try {
            ctx = new InitialContext(); // ser for aplicação local ou
            /* o código a seguir se for aplicação remota
              Properties p = new Properties();
              p.put("org.omg.CORBA.ORBInitialHost", "192.168.1.175");
              p.put("org.omg.CORBA.ORBInitialPort", "3700");
              ctx = new InitialContext(p); */
            } catch (NamingException ex) {
              out.println("<script>alert('Erro: " + ex + "');</script>");
            }
            try {
              cp = (CadastrosRemote) ctx.lookup("java:global/EJBAulaWeb/Cadastrados");
            } catch (NamingException ex) {
              out.println("<script>alert('Erro: " + ex + "');</script>");
            }
            if (cp != null) {
              String r = cp.cadastrarPessoa(request.getParameter("nome"),
              request.getParameter("cpf"), request.getParameter("email"), request.getParameter("fone"));
              if (r.equals("OK")) {
```

```

        out.println("<script>alert('" + request.getParameter("nome") + "
Cadastrado.');"</script>");
    }
    else {
        out.println("<script>alert(\"" + r + "\");</script>");
    }
    try {
        ctx.close();
    } catch (NamingException ex) {
        out.println("<script>alert('Erro: " + ex + "');"</script>");
    }
    } else {
        out.println("<script>alert('Erro acessando aplicação.');"</script>");
    }
}

%>
</div>
<div id=formPadrao> <!-- mostra um formulário para cadastrar pessoas -->
<form name="fCadPessoa" id="fCadPessoa" method="post" action="cadPessoa.jsp">
    <fieldset >
        <legend>Informe os dados da Pessoa:</legend>
        <label for="nome">Nome:</label><br />
        <input type="text" name="nome" id="nome" size="50" maxlength="50" required/><br
/>

        <label for="cpf">CPF:</label><br />
        <input type="text" name="cpf" id="cpf" size="50" maxlength="11" required/><br />
        <label for="email">E-Mail:</label><br />
        <input type="email" name="email" id="email" size="50" maxlength="100"
required/><br />

        <label for="fone">Fone:</label><br />
        <input type="text" name="fone" id="fone" size="50" maxlength="30" ><br />
        <br />
        <input type="submit" name="botao" id="botao" value="Cadastrar" />
    </fieldset>
</form>
</div>
<a href="listaPessoa.jsp">Listar Pessoas</a>
</body>
</html>

```

## alteraPessoa.jsp

```

<%--
    Document      : cadpessoa
    Created on    : 20/02/2017, 11:21:15
    Author       : Evandro
--%>

<%@page import="AulaIFFar.CadastradosRemote"%>
<%@page import="org.json.simple.parser.ParseException"%>
<%@page import="org.json.simple.JSONObject"%>
<%@page import="org.json.simple.parser.JSONParser"%>
<%@page import="javax.naming.NamingException"%>
<%@page import="javax.naming.InitialContext"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Cadastro de Pessoa</title>
    </head>
    <body>
        <h1>Cadastro de Pessoa</h1>
        <div id=formPadrao>

            <%
                InitialContext ctx = null;

```

```

        CadastrosRemote cp;
        cp = null;
        try {
            ctx = new InitialContext(); // ser for aplicação local ou
            /* o código a seguir se for aplicação remota
               Properties p = new Properties();
               p.put("org.omg.CORBA.ORBInitialHost", "192.168.1.175");
               p.put("org.omg.CORBA.ORBInitialPort", "3700");
               ctx = new InitialContext(p); */
        } catch (NamingException ex) {
            out.println("<script>alert('Erro: " + ex + "');</script>");
        }
        try {
            cp = (CadastrosRemote) ctx.lookup("java:global/EJBAulaWeb/Cadastros");
        } catch (NamingException ex) {
            out.println("<script>alert('Erro: " + ex + "');</script>");
        }

        if (request.getParameter("botao") != null &&
            request.getParameter("botao").equals("Salvar")) {
            if (cp != null) {
                String r = cp.alterarPessoa(request.getParameter("alterar"),
                    request.getParameter("nome"), request.getParameter("cpf"), request.getParameter("email"),
                    request.getParameter("fone"));
                if (r.equals("OK")) {
                    response.sendRedirect("listaPessoa.jsp");
                } else {
                    out.println("<script>alert(\"" + r + "\");</script>");
                }
                try {
                    ctx.close();
                } catch (NamingException ex) {
                    out.println("<script>alert('Erro: " + ex + "');</script>");
                }
            } else {
                out.println("<script>alert('Erro acessando aplicação.');"</script>");
            }
        }

        String nome = "";
        String cpf = "";
        String email = "";
        String fone = "";
        if (request.getParameter("alterar") != null) {
            if (cp != null) {
                String s = cp.listarPessoa(request.getParameter("alterar"));
                JSONParser parser = new JSONParser();
                try {
                    Object obj = parser.parse(s);
                    JSONObject obj2 = (JSONObject) obj;
                    nome = obj2.get("nome").toString();
                    cpf = obj2.get("cpf").toString();
                    email = obj2.get("email").toString();
                    fone = obj2.get("fone").toString();
                } catch (ParseException pe) {
                    System.out.println("position: " + pe.getPosition());
                    System.out.println(pe);
                }
            } else {
                out.println("<script>alert(\"Erro no acesso ao servidor de
aplicações.\");</script>");
            }
        }
        if (request.getParameter("botao") != null &&
            request.getParameter("botao").equals("Excluir")) {

            out.println("<script> if(confirm('Você deseja excluir definitivamente a
pessoa "
                + nome + "')) { location.href =
'alteraPessoa.jsp?acao=ConfirmaExcluir&id="
                + request.getParameter("alterar") + "';}else{ "

```

```

        + "alert('Registro não excluído!');";
        + "location.href = 'listaPessoa.jsp';";
        + "}    </script>");
    }

    if (request.getParameter("acao") != null &&
request.getParameter("acao").equals("ConfirmaExcluir")) {
        if (cp != null) {
            String r = cp.excluirPessoa(request.getParameter("id"));
            if (r.equals("OK")) {
                response.sendRedirect("listaPessoa.jsp");
            } else {
                out.println("<script>alert(\"Erro. Não é possível excluir esse
registro: \");</script>");
            }
            try {
                ctx.close();
            } catch (NamingException ex) {
                out.println("<script>alert('Erro: " + ex + "');</script>");
            }
        } else {
            out.println("<script>alert('Erro acessando aplicação.');"</script>");
        }
    }
}
%>
<div id=formPadrao>
    <form name="fCadPessoa" id="fCadPessoa" method="post" action="alteraPessoa.jsp">
        <fieldset >
            <legend>Alteração dos dados da Pessoa:</legend>
            <label for="nome">Nome:</label><br />
            <input type="text" name="nome" id="nome" size="50" maxlength="50"
value="<%out.println(nome);%>" required/><br />
            <label for="cpf">CPF:</label><br />
            <input type="text" name="cpf" id="cpf" size="50" maxlength="11"
value="<%out.println(cpf);%>" required/><br />
            <label for="email">E-Mail:</label><br />
            <input type="email" name="email" id="email" size="50" maxlength="100"
value="<%out.println(email);%>" required/><br />
            <label for="fone">Fone:</label><br />
            <input type="text" name="fone" id="fone" size="50" maxlength="30"
value="<%out.println(fone);%>" ><br />
            <br />
            <input type="submit" name="botao" id="botao" value="Salvar" />
            <input type="submit" name="botao" id="botao" value="Excluir" />
            <%
                if (request.getParameter("alterar") != null) {
                    out.println("<INPUT TYPE=\"hidden\" NAME=\"alterar\" VALUE=\"\" +
request.getParameter("alterar") + \">");
                }
            %>
        </fieldset>
    </form>
</div>
<br />
<a href="cadPessoa.jsp">Cadastrar Pessoas</a><br />
<a href="listaPessoa.jsp">Listar Pessoas</a>
</body>
</html>

```

## listaPessoa.jsp

```

<%--
    Document    : listaPessoa
    Created on  : 23/02/2017, 23:43:34
    Author      : Evandro
--%>

<%@page import="AulaIFFar.CadastrosRemote"%>

```



```

<%@page import="org.json.simple.JSONObject"%>
<%@page import="java.util.Iterator"%>
<%@page import="org.json.simple.JSONArray"%>
<%@page import="org.json.simple.parser.ParseException"%>
<%@page import="org.json.simple.parser.JSONParser"%>
<%@page import="javax.naming.NamingException"%>
<%@page import="javax.naming.InitialContext"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Cadastro de Pessoas</title>
    </head>
    <body>
        <h1>Cadastro de Pessoas</h1>
        <div id=formPadrao>
            <form name="fAlterarPessoa" id="fAlterarPessoa" method="post"
action="alteraPessoa.jsp">
                <fieldset >
                    <legend>Pessoas Cadastradas:</legend>
                    <table border=0><tr><th>Id</th><th>Nome</th><th></th></tr>
                        <%

                                InitialContext ctx = null;
                                CadastrosRemote cp;
                                cp = null;
                                try {
                                    ctx = new InitialContext();// ser for aplicação local ou
/* o codigo a seguir se for aplicação remota
Properties p = new Properties();
p.put("org.omg.CORBA.ORBInitialHost", "192.168.1.192");
p.put("org.omg.CORBA.ORBInitialPort", "3700");
ctx = new InitialContext(p); */
                                } catch (NamingException ex) {
                                    System.out.println("Erro: " + ex);
                                }
                                try {
                                    cp = (CadastrosRemote)
ctx.lookup("java:global/EJBAulaWeb/Cadastros");
                                } catch (NamingException ex) {
                                    System.out.println("Erro: " + ex);
                                }
                                if (cp != null) {
                                    String s = cp.listarTodos();
                                    JSONParser parser = new JSONParser();
                                    Object obj;
                                    obj = null;
                                    try {
                                        obj = parser.parse(s);
                                    } catch (ParseException ex) {
                                        System.out.println("Erro: " + ex);
                                    }
                                    JSONArray slideContent = (JSONArray) obj;
                                    Iterator i = slideContent.iterator();

                                    while (i.hasNext()) {
                                        String s2 = i.next().toString();
                                        try {
                                            Object objs = parser.parse(s2);
                                            JSONObject obj2 = (JSONObject) objs;
                                            out.println("<tr><td>" + obj2.get("id") +

"</td>"); //mostra o identificador

                                            out.println("<td>" + obj2.get("nome") +

"</td>"); //mostra o nome da pessoa

                                            out.println("<td><button type=\"submit\"
name=\"alterar\" id=\"alterar\" value=\"\" + obj2.get('id') + \">Alterar</button></td></tr>");
                                        } catch (ParseException pe) {
                                            System.out.println("position: " +

pe.getPosition());

                                            System.out.println(pe);
                                        }
                                    }
                                }
                        <%
                    </table>
                </fieldset>
            </form>
        </div>
    </body>
</html>

```

```

        }
    }
    %>
</table>
</fieldset>
</form>
</div>
<br />
<a href="cadPessoa.jsp">Cadastrar Pessoas</a>
</html>

```

## index.html

```

<html>
  <head>
    <title>Cadastro Pessoa</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <h1>Cadastro de Pessoa</h1>
    <a href="cadPessoa.jsp">Cadastrar Pessoas</a><br />
    <a href="listaPessoa.jsp">Listar Pessoas</a>
  </body>
</html>

```

Pronto!!! Agora é só implantar e testar a aplicação.

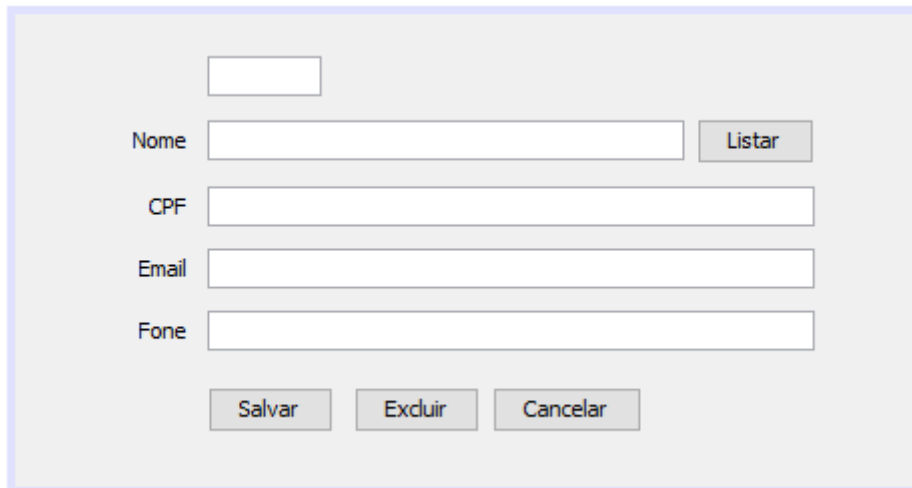
Observe que os arquivos JSP não tem acesso ao banco de dados e só acessam os métodos fornecidos pelo EJB.

## 4. Criando uma aplicação Java Desktop para acessar o EJB - Enterprise Java Bean

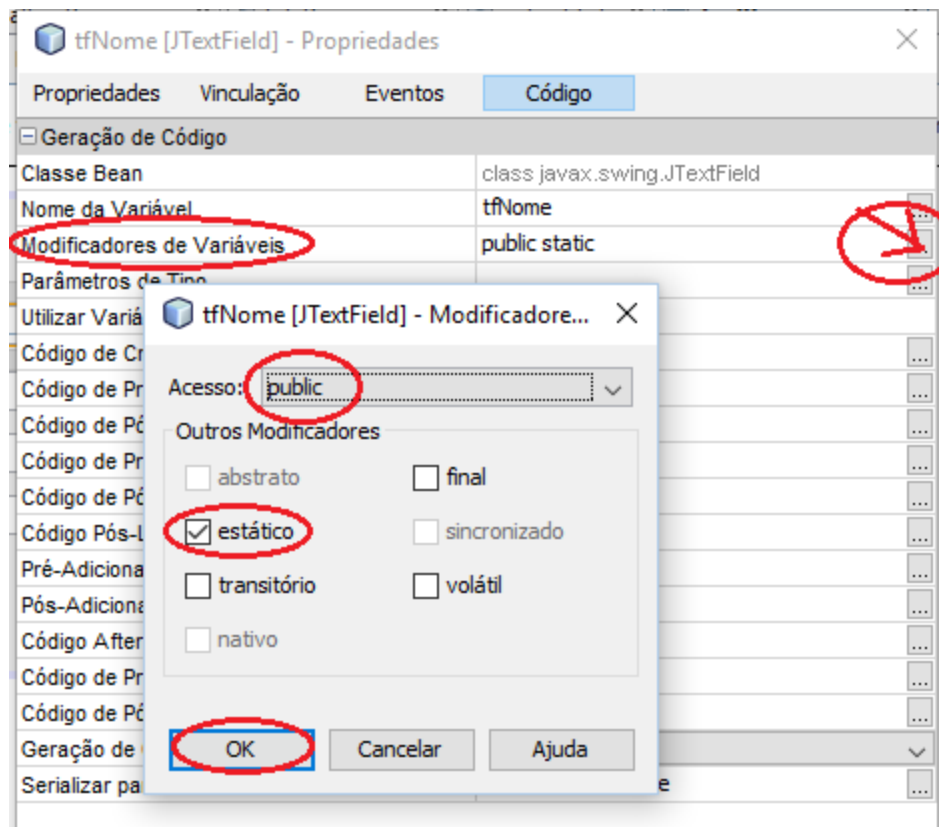
Vamos criar uma aplicação Java Desktop que fará acesso aos métodos do EJB. O EJB pode ser acessado a partir de uma aplicação Java Desktop convencional.

1. Para isso, crie um Novo Projeto => Java => Aplicação Java com o nome **Cadastros**, deixando desmarcada a opção criar Classe Principal
2. No projeto Cadastros, crie novo Form JFrame com o nome **CadPessoa**, no pacote **cadastros**.
3. Adicione da Paleta Controles Swing 4 Label (jLabel), 5 Campo de Texto (jTextField) e 4 Botão (jButton)
4. Clique com o botão direito sobre o componente e altere o seu texto, clicando em Editar Texto.
5. Altere os nomes das variáveis através da opção Alterar o Nome da Variável:
  - tfId
  - tfNome
  - tfCpf
  - tfEmail
  - tfFone
  - btListar
  - btSalvar
  - btExcluir
  - btCancelar

O Formulário JFrame deve ficar semelhante a esse:



6. Altere a propriedade do botão Listar, desmarcando a propriedade **focusable**.
7. Altere as propriedades dos jTextField tfId e tfNome para que sejam **public static**



Tenha certeza que o GlassFish esteja executando.

8. Nas propriedades do Projeto, adicione as seguintes bibliotecas:

- Adicione o projeto libEJBAulaWeb
- Adicione os arquivos appserv-rt.jar e javaee.jar da pasta **lib** do **GlassFish**
- Adicione também a biblioteca json-simple-1.1.1.jar

9. Declare as seguintes variáveis, bem no final da classe antes da última chave:

```
InitialContext ctx;
public static cadastroPessoaRemote cp;
```

Altere o método construtor para que fique com esse código (não altere o que está em vermelho):

```
public CadPessoa() {
    initComponents();
    tfId.setVisible(false);
    cp = null;
    ctx = null;
    try {
        ctx = new InitialContext();// ser for aplicação local ou
        /* o código a seguir se for aplicação remota
        Properties p = new Properties();
        p.put("org.omg.CORBA.ORBInitialHost", "192.168.1.192");
        p.put("org.omg.CORBA.ORBInitialPort", "3700");
        ctx = new InitialContext(p); */
    } catch (NamingException ex) {
        JOptionPane.showMessageDialog(this, "Erro: " + ex);
    }
}
```

```

    }
    try {
        cp = (CadastrosRemote) ctx.lookup("java:global/EJBAulaWeb/Cadastros");
    } catch (NamingException ex) {
        JOptionPane.showMessageDialog(this, "Erro: " + ex);
    }
}
}

```

10. Implemente os seguintes código nos eventos dos respectivos botões (actionPerformed) e no focusGained do tfNome:

```

private void btSalvarActionPerformed(java.awt.event.ActionEvent evt) {
    if (cp != null) {
        if (tfId.getText().length() == 0) {
            String res = cp.cadastrarPessoa(tfNome.getText(), tfCpf.getText(),
tfEmail.getText(), tfFone.getText());
            if (res.equals("OK")) {
                JOptionPane.showMessageDialog(this, tfNome.getText() + "
Cadastrado! .");
            } else {
                JOptionPane.showMessageDialog(this, res);
            }
        } else {
            String res = cp.alterarPessoa(tfId.getText(), tfNome.getText(),
tfCpf.getText(), tfEmail.getText(), tfFone.getText());
            if (res.equals("OK")) {
                JOptionPane.showMessageDialog(this, tfNome.getText() + " Alterado!
.");
            } else {
                JOptionPane.showMessageDialog(this, res);
            }
        }
        tfId.setText("");
        tfNome.grabFocus();
        btExcluir.setEnabled(false);
    } else {
        JOptionPane.showMessageDialog(this, "Erro acessando servidor de
aplicações.");
    }
}

private void tfNomeFocusGained(java.awt.event.FocusEvent evt) {
    tfNome.setText("");
    tfEmail.setText("");
    tfCpf.setText("");
    tfFone.setText("");
    if (tfId.getText().length() > 0) {
        String s = cp.listarPessoa(tfId.getText());
        if (s.length() > 3) {
            JSONParser parser = new JSONParser();
            try {
                Object obj = parser.parse(s);
                JSONObject obj2 = (JSONObject) obj;
                tfNome.setText(obj2.get("nome").toString());
                tfEmail.setText(obj2.get("email").toString());
                tfCpf.setText(obj2.get("cpf").toString());
                tfFone.setText(obj2.get("fone").toString());
            }

```

```

        btExcluir.setEnabled(true);
    } catch (ParseException pe) {
        System.out.println("position: " + pe.getPosition());
        System.out.println(pe);
    }
}

}

}

private void btExcluirActionPerformed(java.awt.event.ActionEvent evt) {

    if (JOptionPane.showConfirmDialog(null, "Deseja Excluir " + tfNome.getText() +
"?", "Confirmação", JOptionPane.OK_CANCEL_OPTION) == JOptionPane.OK_OPTION) {
        if (cp != null) {
            if (tfId.getText().length() > 0) {
                String res = cp.excluirPessoa(tfId.getText());
                if (res.equals("OK")) {
                    JOptionPane.showMessageDialog(this, tfNome.getText() + "
Excluido! .");
                } else {
                    JOptionPane.showMessageDialog(this, res);
                }
            }
            tfId.setText("");
            tfNome.grabFocus();
            btExcluir.setEnabled(false);
        } else {
            JOptionPane.showMessageDialog(this, "Erro acessando servidor de
aplicações.");
        }
    }
}

private void btListarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Consulta c = new Consulta();
    c.setVisible(true);
}

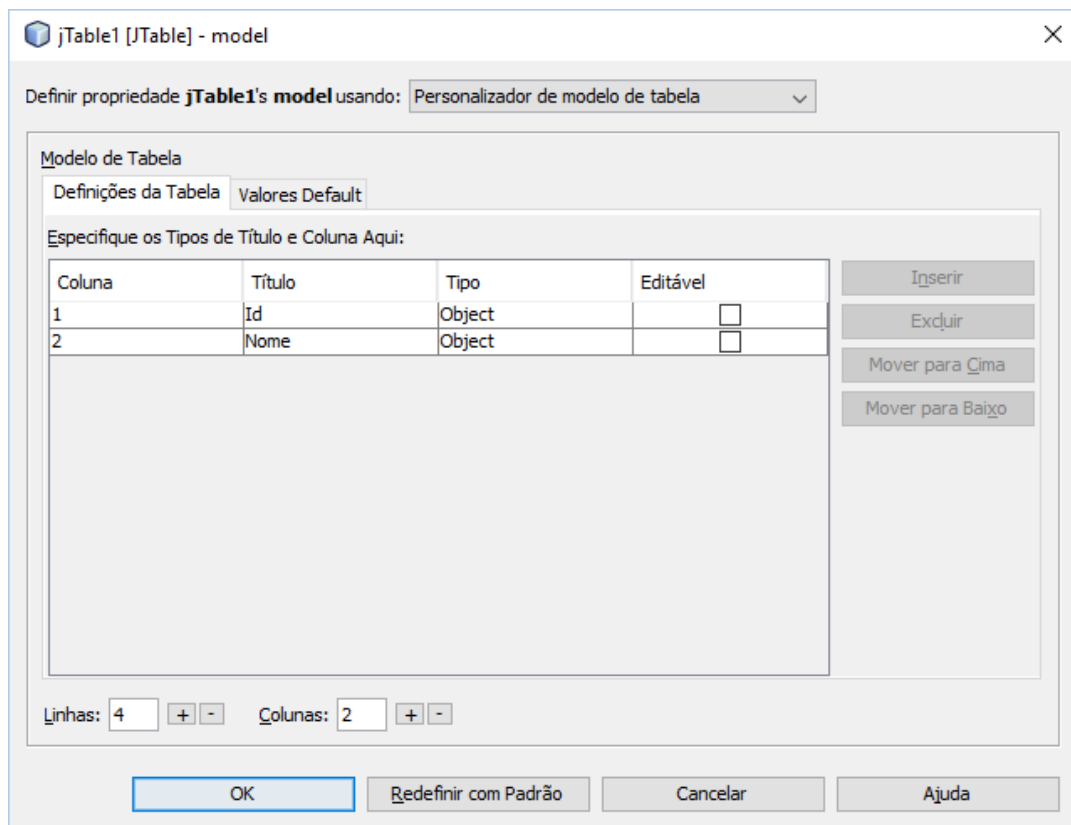
private void btCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    tfId.setText("");
    tfNome.grabFocus();
}
}

```

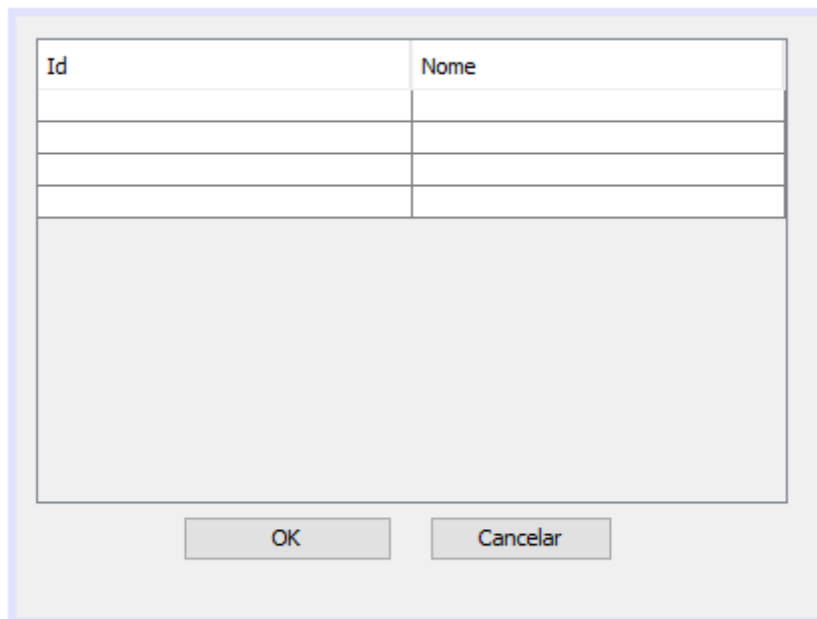
11. Crie um novo JFrame, com o nome Consulta

12. Acrescente um JTable e dois botões. Altere o nome dos botões para btOK e btCancelar

13. Altere a propriedade **Model** do **JTable**, deixando com 2 colunas e alterando o Título para **Id** e para **Nome** como está na figura a seguir:



Ele deve ficar com a aparência semelhante a essa:



14 .Declare a seguinte variável, bem no final da classe antes da última chave:

```
public String CampoRetorno;
```

15. Altere o método construtor, para que fique com o seguinte código:

```
public Consulta() {
    initComponents();
    jTable1.getColumnModel().getColumn(0).setPreferredWidth(50);
}
```

```

jTable1.getColumnModel().getColumn(1).setPreferredWidth(275);
CampoRetorno = "";
DefaultTableModel dtm = (DefaultTableModel) jTable1.getModel();
if (dtm.getRowCount() > 0) {
    for (int i = dtm.getRowCount() - 1; i >= 0; i--) {
        dtm.removeRow(i); // remove as linhas da tabela
    }
}
if (CadPessoa.cp != null) {
    String s = CadPessoa.cp.listarTodos();
    JSONParser parser = new JSONParser();
    Object obj;
    obj = null;
    try {
        obj = parser.parse(s);
    } catch (ParseException ex) {
        Logger.getLogger(CadPessoa.class.getName()).log(Level.SEVERE, null,
ex);
    }
    JSONArray slideContent = (JSONArray) obj;
    Iterator i = slideContent.iterator();
    while (i.hasNext()) {
        String s2 = i.next().toString();
        try {
            Object objs = parser.parse(s2);
            JSONObject obj2 = (JSONObject) objs;
            dtm.addRow(new Object[]{obj2.get("id").toString(),
obj2.get("nome").toString()});
        } catch (ParseException pe) {
            System.out.println("position: " + pe.getPosition());
            System.out.println(pe);
        }
    }
} else {
    System.out.println("Erro no cadastro.");
}
}

```

16. Implemente os seguintes eventos de actionPerformed dos botões e do MouseClicked da JTable:

```

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if (jTable1.getSelectedRow() >= 0) {
        CampoRetorno = jTable1.getValueAt(jTable1.getSelectedRow(), 0).toString();
    }
}

private void btOKActionPerformed(java.awt.event.ActionEvent evt) {
    CadPessoa.tfId.setText(CampoRetorno);
    CadPessoa.tfNome.grabFocus();
    dispose();
}

private void btCancelarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    CampoRetorno = "";
    CadPessoa.tfId.setText(CampoRetorno);
    CadPessoa.tfNome.grabFocus();
    dispose();
}

```



Pronto!!! É só testar!

O EJB foi acessado tanto de uma aplicação web quanto de uma aplicação Desktop.

Na próxima aula vamos criar um WebService baseado no EJB, com 3 cliques de mouse. Esse webservice poderá ser acessado tanto de uma aplicação PHP quanto de uma aplicação Android.

Assim, centralizamos a lógica do negócio num EJB e as aplicações servem como interface dos métodos disponibilizados. Se alterar a regra do negócio, mudamos apenas na implementação do EJB. Isso facilita a manutenção dos sistemas.

Com isso também evitamos o acesso ao Banco de Dados pelas aplicações, aumentando a segurança.