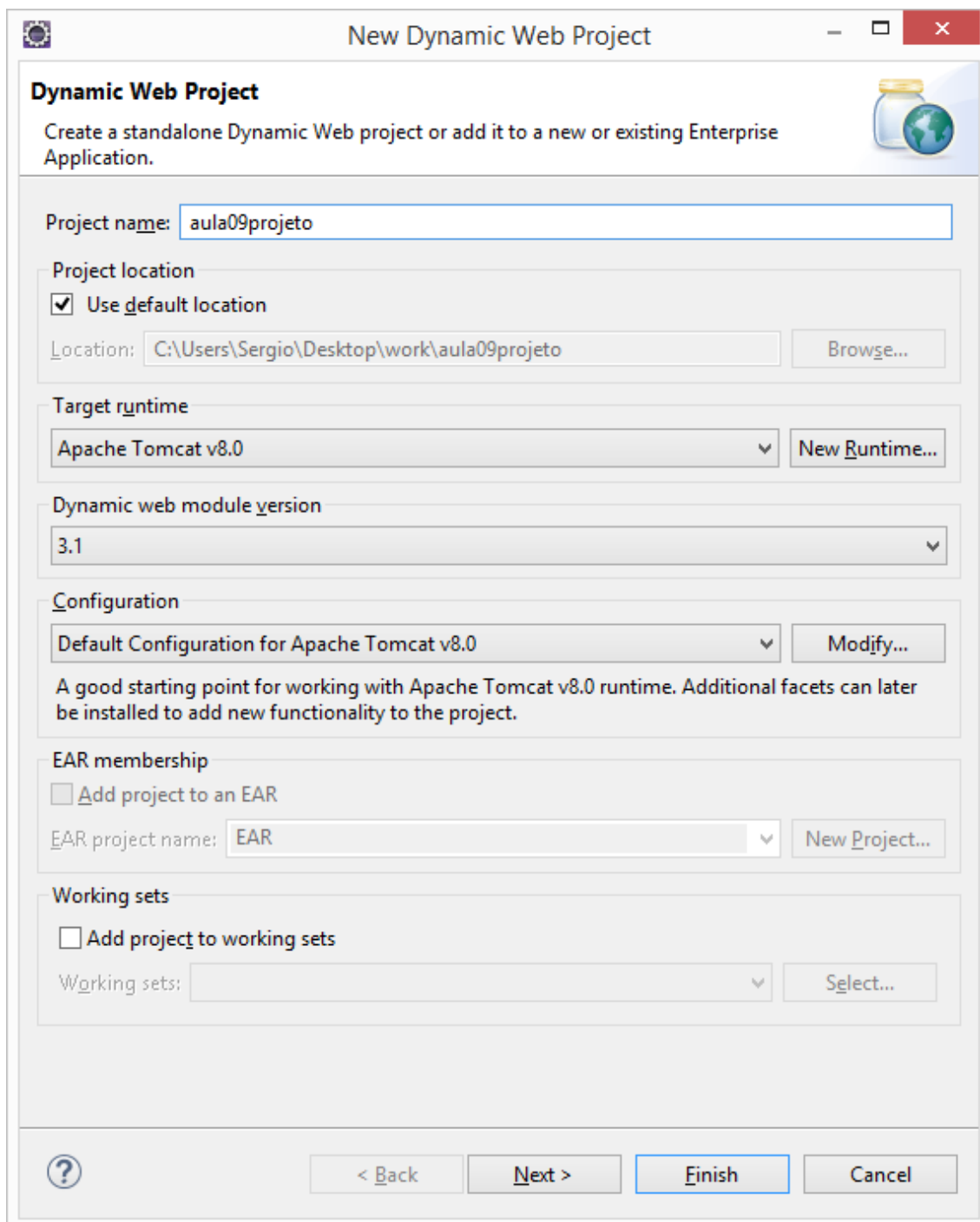


Criando o projeto:

## File > New > Dynamic Web Project



The screenshot shows the 'New Dynamic Web Project' dialog box in the Eclipse IDE. The dialog is titled 'New Dynamic Web Project' and has a close button (X) in the top right corner. It contains several sections for configuring the project:

- Dynamic Web Project**: A section with a description: 'Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.' and a small icon of a globe and a jar.
- Project name:** A text field containing 'aula09projeto'.
- Project location**: A section with a checkbox 'Use default location' which is checked. Below it is a text field for 'Location' containing 'C:\Users\Sergio\Desktop\work\aula09projeto' and a 'Browse...' button.
- Target runtime**: A dropdown menu showing 'Apache Tomcat v8.0' and a 'New Runtime...' button.
- Dynamic web module version**: A dropdown menu showing '3.1'.
- Configuration**: A dropdown menu showing 'Default Configuration for Apache Tomcat v8.0' and a 'Modify...' button. Below this is a descriptive text: 'A good starting point for working with Apache Tomcat v8.0 runtime. Additional facets can later be installed to add new functionality to the project.'
- EAR membership**: A section with a checkbox 'Add project to an EAR' which is unchecked. Below it is a text field for 'EAR project name' containing 'EAR' and a 'New Project...' button.
- Working sets**: A section with a checkbox 'Add project to working sets' which is unchecked. Below it is a text field for 'Working sets' and a 'Select...' button.

At the bottom of the dialog, there is a help icon (question mark), and four buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

### Criando a base de dados: **Modelagem 1 para Muitos**

```
drop database if exists aula09;
create database aula09;
use aula09;

create table fornecedor(
    idfornecedor      integer      auto_increment,
    nome              varchar(50)  not null,
    descricao         varchar(250) not null,
    primary key(idfornecedor));

create table produto(
    idproduto         integer      auto_increment,
    nome              varchar(50)  not null,
    preco             double       not null,
    quantidade        integer      not null,
    datacadastro      date         not null,
    idfornecedor      integer      not null,
    primary key(idproduto),
    foreign key(idfornecedor) references fornecedor(idfornecedor));

--view para obter dados de produtos e fornecedor
create view vwprodutofornecedor
as
    select
        p.idproduto,
        p.nome as nomeproduto,
        p.preco,
        p.quantidade,
        f.idfornecedor,
        f.nome as nomefornecedor,
        f.descricao
    from produto as p
    inner join fornecedor as f
    on f.idfornecedor = p.idfornecedor;

desc vwprodutofornecedor;
```

---

### Criando as Classes de entidade:

```
package br.com.brq.entities;

import java.util.Date;

public class Produto {

    private Integer idProduto;
    private String nome;
    private Double preco;
```

```
private Integer quantidade;
private Date dataCadastro;

// Relacionamento de Produto com 1 Fornecedor..
private Fornecedor fornecedor; // TEM 1

// construtor default..
public Produto() {
    // TODO Auto-generated constructor stub
}

public Produto(Integer idProduto, String nome, Double preco,
    Integer quantidade, Date dataCadastro) {
    super();
    this.idProduto = idProduto;
    this.nome = nome;
    this.preco = preco;
    this.quantidade = quantidade;
    this.dataCadastro = dataCadastro;
}

public Produto(Integer idProduto, String nome, Double preco,
    Integer quantidade, Date dataCadastro,
    Fornecedor fornecedor) {
    this(idProduto, nome, preco, quantidade, dataCadastro);
    this.fornecedor = fornecedor;
}

public Integer getIdProduto() {
    return idProduto;
}

public void setIdProduto(Integer idProduto) {
    this.idProduto = idProduto;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public Double getPreco() {
    return preco;
}

public void setPreco(Double preco) {
    this.preco = preco;
}

public Integer getQuantidade() {
    return quantidade;
}
```

```
public void setQuantidade(Integer quantidade) {
    this.quantidade = quantidade;
}

public Date getDataCadastro() {
    return dataCadastro;
}

public void setDataCadastro(Date dataCadastro) {
    this.dataCadastro = dataCadastro;
}

public Fornecedor getFornecedor() {
    return fornecedor;
}

public void setFornecedor(Fornecedor fornecedor) {
    this.fornecedor = fornecedor;
}

@Override
public String toString() {
    return "Produto [idProduto=" + idProduto + ", nome=" + nome
        + ", preco=" + preco + ", quantidade=" + quantidade
        + ", dataCadastro=" + dataCadastro + "]\n";
}
}

package br.com.brq.entities;

import java.util.Collection;

public class Fornecedor {

    private Integer idFornecedor;
    private String nome;
    private String descricao;

    // Relacionamento (Cada Fornecedor TEM Muitos Produtos)
    private Collection<Produto> produtos;

    public Fornecedor() {
        // TODO Auto-generated constructor stub
    }

    public Fornecedor(Integer idFornecedor, String nome, String descricao)
    {
        super();
        this.idFornecedor = idFornecedor;
        this.nome = nome;
        this.descricao = descricao;
    }
}
```

```
public Fornecedor(Integer idFornecedor, String nome, String descricao,
Collection<Produto> produtos) {
    this(idFornecedor, nome, descricao);
    this.produtos = produtos;
}

public Integer getIdFornecedor() {
    return idFornecedor;
}

public void setIdFornecedor(Integer idFornecedor) {
    this.idFornecedor = idFornecedor;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getDescricao() {
    return descricao;
}

public void setDescricao(String descricao) {
    this.descricao = descricao;
}

public Collection<Produto> getProdutos() {
    return produtos;
}

public void setProdutos(Collection<Produto> produtos) {
    this.produtos = produtos;
}

@Override
public String toString() {
    return "Fornecedor [idFornecedor=" + idFornecedor
        + ", nome=" + nome + ", descricao=" + descricao + "];"
}
}
```

---

### Criando a Classe DAO

Data Access Object

```
package br.com.brq.persistence;
```

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;

public class DAO {

    private static final String DRIVER = "com.mysql.jdbc.Driver";
    private static final String URL = "jdbc:mysql://localhost:3306/aula09";
    private static final String USER = "root";
    private static final String PASSWORD = "brqbrq";

    protected Connection con;
    protected PreparedStatement stmt;
    protected CallableStatement call;
    protected ResultSet rs;

    protected void openConnection() throws Exception{
        Class.forName(DRIVER);
        con = DriverManager.getConnection(URL, USER, PASSWORD);
    }

    protected void closeConnection() throws Exception{
        if(con != null){
            con.close();
        }
    }
}
```

---

### **Criando a Classe de persistencia para Fornecedor:**

DAOFornecedor.java

```
package br.com.brq.persistence;

import java.util.ArrayList;
import java.util.List;

import br.com.brq.entities.Fornecedor;

public class DAOFornecedor extends DAO{

    //método para cadastrar um fornecedor na base de dados..
    public void insert(Fornecedor f) throws Exception{

        String query = "insert into fornecedor(nome, descricao) "
            + "values(?, ?)";

        openConnection();
        stmt = con.prepareStatement(query);
        stmt.setString(1, f.getNome());
        stmt.setString(2, f.getDescricao());
        stmt.execute();
        stmt.close();

        closeConnection();
    }
}
```

```
//método para listar os fornecedores da base de dados..
public List<Fornecedor> findAll() throws Exception{

    String query = "select * from fornecedor";

    openConnection();
    stmt = con.prepareStatement(query);
    rs = stmt.executeQuery();

    List<Fornecedor> lista = new ArrayList<Fornecedor>();

    while(rs.next()){ //enquanto houver registros..
        Fornecedor f = new Fornecedor();
        f.setIdFornecedor(rs.getInt("idfornecedor"));
        f.setNome(rs.getString("nome"));
        f.setDescricao(rs.getString("descricao"));

        lista.add(f); //adicionar na lista..
    }

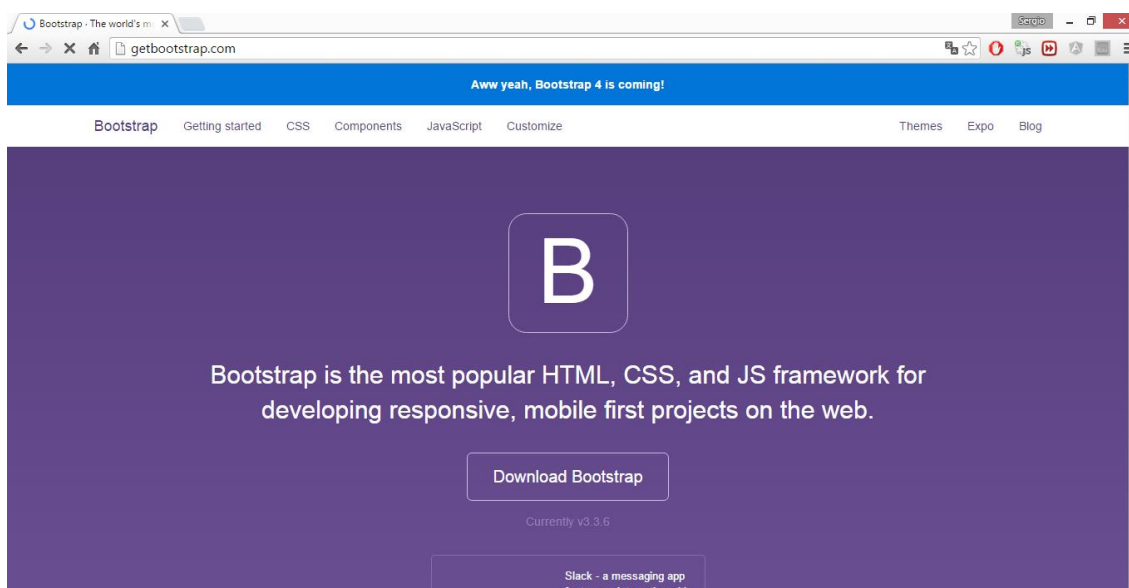
    stmt.close();
    closeConnection();

    return lista;
}
```

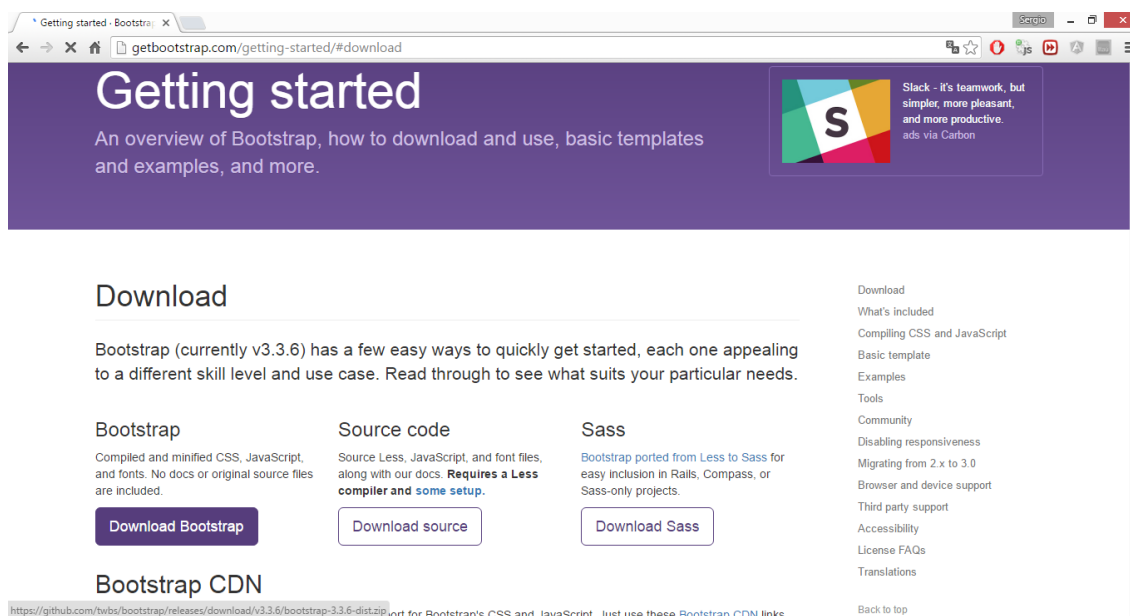
## Bootstrap (www.getbootstrap.com)

Biblioteca CSS para frontend de aplicações web:

### - Obtendo o bootstrap:



<http://getbootstrap.com/getting-started/#download>



### Criando a página index.jsp com bootstrap:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Treinamento Java BRQ</title>

    <!-- Arquivos CSS do bootstrap -->
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css"/>
    <link rel="stylesheet" type="text/css" href="css/bootstrap-theme.css"/>
</head>
<body class="container">
    <!-- Cabeçalho da página -->
    <div class="well well-sm">
        <h2>Projeto Controle de Fornecedores e Produtos</h2>
        Treinamento BRQ COTI Informatica - Java Developer
    </div>
    <!-- Conteúdo principal da página -->
    <div class="col-md-12">

        <h3>Bem vindo ao projeto</h3>
        <hr/>
        <p>
            Selecione a operação desejada:
        </p>
        <br/><br/>
        <div class="col-md-3 panel text-center">
            <div class="glyphicon glyphicon-blackboard">
```

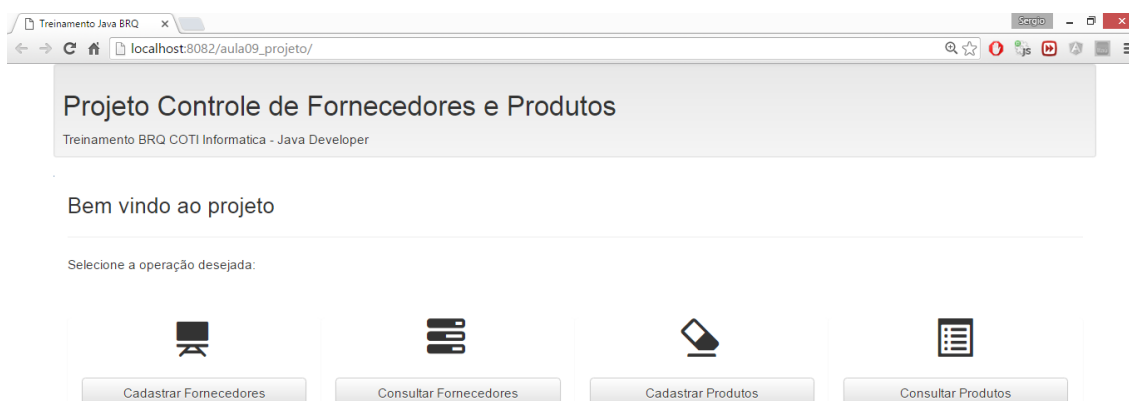


```

        style="font-size: 3em;"></div>
    <br/><br/>
    <a href="cadastroFornecedor.jsp"
        class="btn btn-default btn-block">
        Cadastrar Fornecedores
    </a>
</div>
<div class="col-md-3 panel text-center">
    <div class="glyphicon glyphicon-tasks"
        style="font-size: 3em;"></div>
    <br/><br/>
    <a href="consultaFornecedor.jsp"
        class="btn btn-default btn-block">
        Consultar Fornecedores
    </a>
</div>
<div class="col-md-3 panel text-center">
    <div class="glyphicon glyphicon-erase"
        style="font-size: 3em;"></div>
    <br/><br/>
    <a href="cadastroProduto.jsp"
        class="btn btn-default btn-block">
        Cadastrar Produtos
    </a>
</div>
<div class="col-md-3 panel text-center">
    <div class="glyphicon glyphicon-list-alt"
        style="font-size: 3em;"></div>
    <br/><br/>
    <a href="consultaProduto.jsp"
        class="btn btn-default btn-block">
        Consultar Produtos
    </a>
</div>
</div>
</body>
</html>

```

### Resultado:



Criando a página de cadastro de fornecedor:

### **cadastroFornecedor.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>Treinamento Java BRQ</title>

    <!-- Arquivos CSS do bootstrap -->
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css"/>
    <link rel="stylesheet" type="text/css" href="css/bootstrap-theme.css"/>

</head>
<body class="container">

    <!-- Cabeçalho da página -->
    <div class="well well-sm">
        <h2>Projeto Controle de Fornecedores e Produtos</h2>
        Treinamento BRQ COTI Informatica - Java Developer
    </div>

    <!-- Conteudo principal da página -->
    <div class="col-md-12">

        <h3>Cadastro de Fornecedores</h3>
        <a href="index.jsp">Voltar</a> para a página inicial.
        <hr/>

        <div class="col-md-4">

            <form name="formulario" method="post"
                action="ControleFornecedor?acao=cadastrar">

                <label>Nome do Fornecedor:</label>
                <input type="text" name="nome" id="nome"
                    class="form-control"
                    placeholder="Digite aqui"/>
                <br/>

                <label>Descrição:</label>
                <textarea name="descricao" id="descricao"
                    class="form-control"></textarea>
                <br/>

                <input type="submit" value="Cadastrar Fornecedor"
                    class="btn btn-success"/>

                <!-- EL (Expression Language) -->
                <h4>${mensagem}</h4>
            </form>
        </div>
    </div>
</body>
</html>
```

### Criando o Servlet para cadastrar o Fornecedor:

ControleFornecedor.java

```
package br.com.brq.control;  
  
import java.io.IOException;  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
import br.com.brq.entities.Fornecedor;  
import br.com.brq.persistence.DAOFornecedor;  
  
@WebServlet("/ControleFornecedor")  
public class ControleFornecedor extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    public ControleFornecedor() {  
        super();  
    }  
  
    //método para centralizar as requisições recebidas pelos métodos doGet e doPost  
    protected void execute(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        //resgatar o comando enviado pela página..  
        String acao = request.getParameter("acao");  
  
        if(acao != null){ //se o valor foi resgatado..  
            if(acao.equalsIgnoreCase("cadastrar")){  
  
                try{  
  
                    //resgatando os valores dos campos do formulário..  
                    Fornecedor f = new Fornecedor(); //entidade..  
                    f.setNome(request.getParameter("nome"));  
                    //valor do campo nome..  
                    f.setDescricao(request.getParameter("descricao"));  
                    //valor do campo descricao..  
  
                    //gravar na base de dados..  
                    DAOFornecedor d = new DAOFornecedor();  
                    //persistencia..  
                    d.insert(f); //gravando o fornecedor..  
  
                    //enviando mensagem para a página..
```

```
        request.setAttribute("mensagem", "Fornecedor "
                                + f.getNome() + ", cadastrado com sucesso.");
    }
    catch(Exception e){
        //enviando mensagem de erro.
        request.setAttribute("mensagem",
                                "Ocorreu um Erro: " + e.getMessage());
    }
    finally{ //sempre é executado..
        //redirecionar de volta para a página
        //de cadastro do fornecedor...
        request.getRequestDispatcher
            ("cadastroFornecedor.jsp")
            .forward(request, response);
    }
}

}

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    execute(request, response); //encaminhando para o execute..
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    execute(request, response); //encaminhando para o execute..
}

}
```

Continua...