

Java na Web

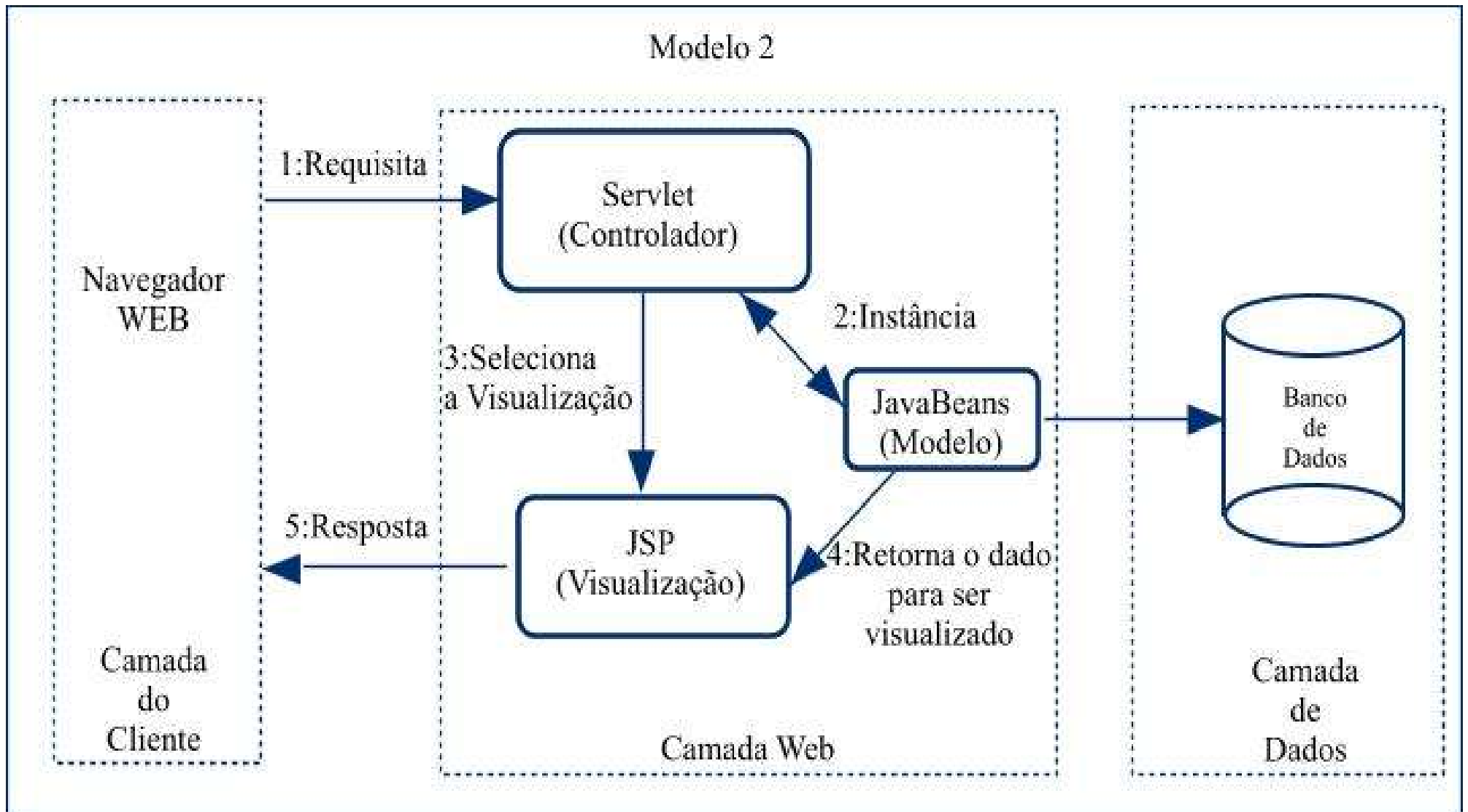


Aplicações Web

- **Modelo-Visualização-Controle;**
- **JavaBeans;**
- **Exemplo prático: Agenda.**

Java na Web

Padrão MVC



Java na Web

Padrão MVC



Aplicações Web MVC

O ganho para o desenvolvimento é separar melhor as responsabilidades, diminuir código JSP e reusabilidade. Neste padrão MVC, é utilizada a arquitetura de implementação com servlet, JavaBeans e JSP. O Controle fica a cargo de um servlet que gerencia o fluxo da aplicação Web, o Modelo é delegado para um JavaBeans e a Visualização é de responsabilidade de uma JSP, que produz uma saída em HTML para o navegador Web do cliente.

Java na Web



Aplicações Web

Iniciando HSQLDB

Execute o arquivo *.bat que tem estes comandos:

```
java -classpath ./lib/hsqldb.jar org.hsqldb.WebServer -database.0 dbmvc  
-dbname.0 mvc
```

- Agora no NetBeans abra o projeto MVC;
- Vá até o diretório “Source Packages”, e crie um pacote denominado:

-

“br.javanoroeste.mvc”

- Agora crie um Servlet denominado “ControllerAgenda”;

Java na Web



Aplicações Web

Montando o ambiente com HSQLDB

Explicação

Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

Add information to deployment descriptor (web.xml)

Class Name:

Servlet Name:

URL Mapping(s):

Init Parameters:

Name	Value
------	-------

< Back Next > Finish Cancel Help

Java na Web



Aplicações Web

Montando o ambiente com HSQLDB

Explicação

```
index.jsp x ControllerAgenda.java x web.xml * x
Overview Servlets Filters Pages XML
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <servlet>
    <servlet-name>ControllerAgenda</servlet-name>
    <servlet-class>br.javanoroeste.mvc.ControllerAgenda</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ControllerAgenda</servlet-name>
    <url-pattern>/ControllerAgenda</url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>
      30
    </session-timeout>
  </session-config>
  <welcome-file-list>
    <welcome-file>
      index.jsp
    </welcome-file>
    <welcome-file>
      index.html
    </welcome-file>
    <welcome-file>
      index.htm
    </welcome-file>
  </welcome-file-list>
</web-app>
```

Java na Web



Aplicações Web

Cole o seguinte código no servlet:

```
protected void processRequest(HttpServletRequest request,  
HttpServletResponse response)  
throws ServletException, IOException {
```

```
    response.setContentType("text/html");
```

```
    PrintWriter saida = response.getWriter();  
    saida.println("< html> ");  
    saida.println("Olá Javanoroeste!");  
    saida.println("< /html> ");
```

```
    saida.close();
```

```
}
```

É só executar...

Java na Web



Aplicações Web

Agora vamos alterar o código.

Inclua no início da classe as linhas abaixo:

```
import java.util.*;  
import java.sql.*;
```

Substitua o código que está marcado

```
response.setContentType("text/html");
```

```
PrintWriter saida = response.getWriter();
```

```
saida.println("<html> ");
```

```
saida.println("Olá Javanoroeste!");
```

```
saida.println("</html> ");
```

```
saida.close();
```

```
}
```

Pelo código a seguir:

Java na Web



Aplicações Web

```
String driver = "org.hsqldb.jdbcDriver";
String url = "jdbc:hsqldb:http://localhost/mvc";
String usuario = "sa";
String senha = "";
try
{
    Class.forName(driver);
}
catch (ClassNotFoundException e)
{
    System.out.println("Erro ao carregar o driver JDBC: " + e.getMessage());
    System.exit(0);
}
```

continua....

Java na Web



Aplicações Web

```
try
{
    Connection con = DriverManager.getConnection(url, usuario,
senha);
    Statement st = con.createStatement();
    String sql = "SELECT nome, email FROM AGENDA";
    ResultSet res = st.executeQuery(sql);

    saida.println("< html> ");
    saida.println("< head> ");
    saida.println("  < title> HSQLDB< / title> ");
    saida.println("< / head> ");
    saida.println("< body> ");
    saida.println("< h1> HSQLDB< / h1> ");
    saida.println("< table> ");
```

continua....

Java na Web



Aplicações Web

```
while (res.next()) {  
    saida.println("<tr><td> " + res.getString(1) + "</td> <td> " +  
res.getString(2) + "</td> </tr>");  
  
    }  
    saida.println("</table>");  
    saida.println("</body>");  
    saida.println("</html>");  
  
    saida.close(); res.close(); st.close(); con.close();  
} catch (SQLException e)  
{  
    System.out.println("Erro ao se conectar com a URL fornecida - " +  
e.getMessage() + " - error code: " + e.getErrorCode());  
    System.exit(0);  
}
```

Java na Web

Aplicações Web

Execute o programa...



Java na Web



Aplicações Web

JavaBeans

JavaBeans são objetos escritos em Java, cujas implementações estão de acordo com um conjunto de convenções projetadas para promover modularidade e reusabilidade. *JavaBeans* usa um estilo de programação que conduz a partes de código de programa independentes, pelo qual encapsulam comportamento, funcionalidade ou dados relacionados, podendo ser usados e reutilizados em múltiplos contextos sem que seja preciso conhecer detalhes de sua operação interna.

Java na Web



Aplicações Web

JavaBeans

Componente é um objeto independente, representando uma coleção de propriedades e comportamento. Pelo fato dessas propriedades e comportamentos poderem ser acessados, sem levar em consideração a implementação básica, é possível descrever as capacidades de um componente, independentemente da linguagem de programação em que este foi originalmente escrito. No caso da JSP, os componentes escritos como *JavaBeans* são acessados por meio de uma sintaxe alternativa e não através de uma linguagem de programação.

Java na Web

Aplicações Web



Crie um novo pacote:

br.javanoroeste.beans

New Java Package

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Package Name:

Project:

Location:

Created Folder:

< Back Next > Finish Cancel Help

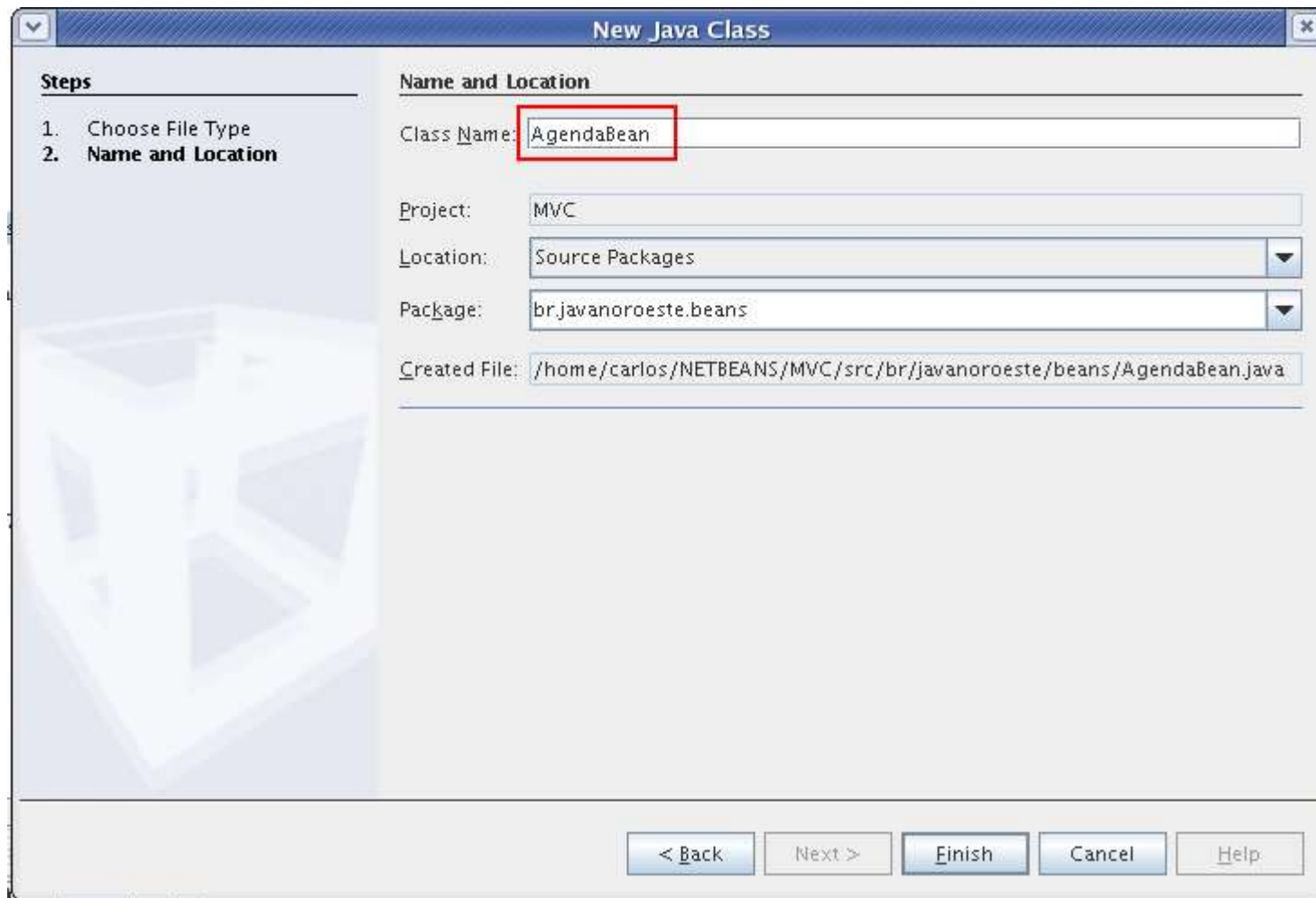
Java na Web

Aplicações Web



Crie uma nova classe:

AgendaBean



Java na Web



Aplicações Web

Escreva o código para o **AgendaBean** a seguir:

```
public class AgendaBean {  
  
    private String nome;  
    private String email;  
  
    public void setNome(String nome){  
        this.nome= nome;  
    }  
  
    public void setEmail(String email){  
        this.email= email;  
    }  
  
    public String getNome(){  
        return nome;  
    }  
    public String geEmail(){  
        return email;  
    }  
}
```

Java na Web



Aplicações Web

Agora altere o código do servlet ControllerAgenda:

Inclua a linha abaixo nas declarações de import:

```
import br.javanoroeste.mvc.bean.AgendaBean;
```

Dentro do método processRequest acrescente o código:

```
Vector listaAgenda = new Vector();
```

Apague o trecho de código abaixo:

```
saida.println("<html> ");  
saida.println("<head> ");  
saida.println("  <title> HSQLDB</title> ");  
saida.println("</head> ");  
saida.println("<body> ");  
saida.println("<h1> HSQLDB</h1> ");  
saida.println("<table> ");
```

Java na Web



Aplicações Web

No trecho que aparece o código abaixo:

```
while (res.next()) {
```

```
    saida.println("<tr><td> " + res.getString(1) + "</td> <td> " +  
res.getString(2) + "</td> </tr> ");
```

```
}
```

Substitua por:

```
while (res.next()) {
```

```
    AgendaBean agenda = new AgendaBean();  
    agenda.setNome(res.getString(1));  
    agenda.setEmail(res.getString(2));  
    listaAgenda.add(agenda);
```

```
}
```

Java na Web



Aplicações Web

Apague o trecho de código a seguir.:

```
saida.println("</table> ");  
saida.println("</body> ");  
saida.println("</html> ");  
saida.close();
```

Acrescentar no fim do código da classe estas linhas.:

```
request.setAttribute("listaagenda", listaAgenda);  
String dispatch = "/viewlista.jsp" ;  
RequestDispatcher dispatcher = request.getRequestDispatcher  
(dispatch);  
  
dispatcher.forward(request, response);
```