

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, siga os passos listados abaixo.

## Preparando o ambiente

Se você não tem ainda Java, Eclipse Java EE e Tomcat instalado, seguem os passos:

- 1) Preferencialmente, tenha o **Java 10** instalado. Instale também o **Eclipse Java EE**.
- 2) Baixe [aqui](#) o **Tomcat 9**, descompacte-o e configure-o como servidor no Eclipse.
- 3) Caso você não tenha o projeto do curso anterior, baixe-o [aqui](#) e importe-o no Eclipse.
- 4) No Eclipse, associe o projeto **gerenciador** ao Tomcat.

## Exercícios da aula

- 1) Crie um novo pacote, com o nome `br.com.alura.gerenciador.modelo`, e mova as classes `Banco` e `Empresa` para dentro dele.
- 2) Corrija as importações das classes `Banco` e `Empresa` no restante do projeto.
- 3) Crie uma nova servlet com o nome `UnicaEntradaServlet` que estende a classe `HttpServlet`, possui o mapeamento para `/entrada` e o método `service`:

```
@WebServlet("/entrada")
public class UnicaEntradaServlet extends HttpServlet {
    protected void service(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

    }
}
```

- 4) Nessa servlet, dentro do método `service` lê o parâmetro `acao` e cria um if-else para cada ação:

```
String paramAcao = request.getParameter("acao");

if(paramAcao.equals("ListaEmpresas")) {
    ListaEmpresas acao = new ListaEmpresas();
    acao.executa(request, response);
} else if(paramAcao.equals("RemoveEmpresa")) {
    RemoveEmpresa acao = new RemoveEmpresa();
    acao.executa(request, response);
} else if(paramAcao.equals("MostraEmpresa")) {
    MostraEmpresa acao = new MostraEmpresa();
    acao.executa(request, response);
} else if(paramAcao.equals("AlteraEmpresa")) {
    AlteraEmpresa acao = new AlteraEmpresa();
    acao.executa(request, response);
} else if(paramAcao.equals("NovaEmpresa")) {
    NovaEmpresa acao = new NovaEmpresa();
    acao.executa(request, response);
}
```

O código não compila ainda pois falta criar as ações!

- 5) Crie mais um pacote com o nome `br.com.alura.gerenciador.acao` e dentro dele, crie as classes `AlteraEmpresa`, `ListaEmpresas`, `MostraEmpresa`, `NovaEmpresa` e `RemoveEmpresa`. Em todas as classes, crie o método `executa`, que recebe como parâmetro um `HttpServletRequest` e um `HttpServletResponse`.
- 6) No método `executa` da classe `AlteraEmpresa`, utilize o código do método `doGet` da servlet `AlteraEmpresaServlet`. Faça o mesmo para todas as classes criadas no passo anterior, copiando os códigos dos métodos `doGet` / `doPost` das servlets de mesmo nome para dentro dos métodos `executa` das suas respectivas classes.
- 7) Em `NovaEmpresa`, `AlteraEmpresa` e `RemoveEmpresa`, faça o redirecionamento para a `entrada`, enviando a ação a ser executada, que é a listagem de empresas:

```
response.sendRedirect("entrada?acao=ListaEmpresas");
```

- 8) Em `listaEmpresas.jsp`, utilize o novo link, apontando para a `entrada` e passando o parâmetro de ação, além do `id` da empresa:

```
<a href="/gerenciador/entrada?acao=MostraEmpresa&id=${empresa.id }">edita
<a href="/gerenciador/entrada?acao=RemoveEmpresa&id=${empresa.id }">remov
```

- 9) Em `formAlteraEmpresa.jsp`, modifique o `c:url` para apontar para a `entrada` e altere o link, para ficar mais claro:

```
<c:url value="/entrada" var="linkEntradaServlet"/>
```

- 10) E no formulário, modifique o `action` e envie a ação a ser executada, através de um `input` escondido:

```
<form action="${linkEntradaServlet }" method="post">
    Nome: <input type="text" name="nome" value="${empresa.nome }" />
    Data Abertura: <input type="text" name="data"
        value="<fmt:formatDate value="${empresa.dataAbertura }"
        pattern="dd/MM/yyyy"/>" />
    <input type="hidden" name="id" value="${empresa.id }">
    <input type="hidden" name="acao" value="AlteraEmpresa">
    <input type="submit" />
</form>
```

- 11) Faça o mesmo em `formNovaEmpresa.jsp`, modificando o `c:url` para apontar para a `entrada` e alterando o link, para ficar mais claro:

```
<c:url value="/entrada" var="linkEntradaServlet"/>
```

- 12) No formulário, modifique o `action` e envie a ação a ser executada, através de um `input` escondido:

```
<form action="${linkEntradaServlet }" method="post">
    Nome: <input type="text" name="nome" />
    Data Abertura: <input type="text" name="data" />
    <input type="hidden" name="acao" value="NovaEmpresa">
    <input type="submit" />
</form>
```

- 13) Por fim, apague todas as servlets, deixando apenas a `OiMundoServlet` e a `UnicaEntradaServlet`.

## Opinião do instrutor

Para sua comparação a classe `UnicaEntradaServlet` ficará assim:

```
@WebServlet("/entrada")
public class UnicaEntradaServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

        String paramAcao = request.getParameter("acao");

        if(paramAcao.equals("ListaEmpresas")) {
            ListaEmpresas acao = new ListaEmpresas();
            acao.executa(request, response);
        } else if(paramAcao.equals("RemoveEmpresa")) {
            RemoveEmpresa acao = new RemoveEmpresa();
            acao.executa(request, response);
        } else if(paramAcao.equals("MostraEmpresa")) {
            MostraEmpresa acao = new MostraEmpresa();
            acao.executa(request, response);
        } else if(paramAcao.equals("AlteraEmpresa")) {
            AlteraEmpresa acao = new AlteraEmpresa();
            acao.executa(request, response);
        } else if(paramAcao.equals("NovaEmpresa")) {
            NovaEmpresa acao = new NovaEmpresa();
            acao.executa(request, response);
        }
    }
}
```

Continue com os seus estudos, e se houver dúvidas, não hesite em recorrer ao nosso fórum!