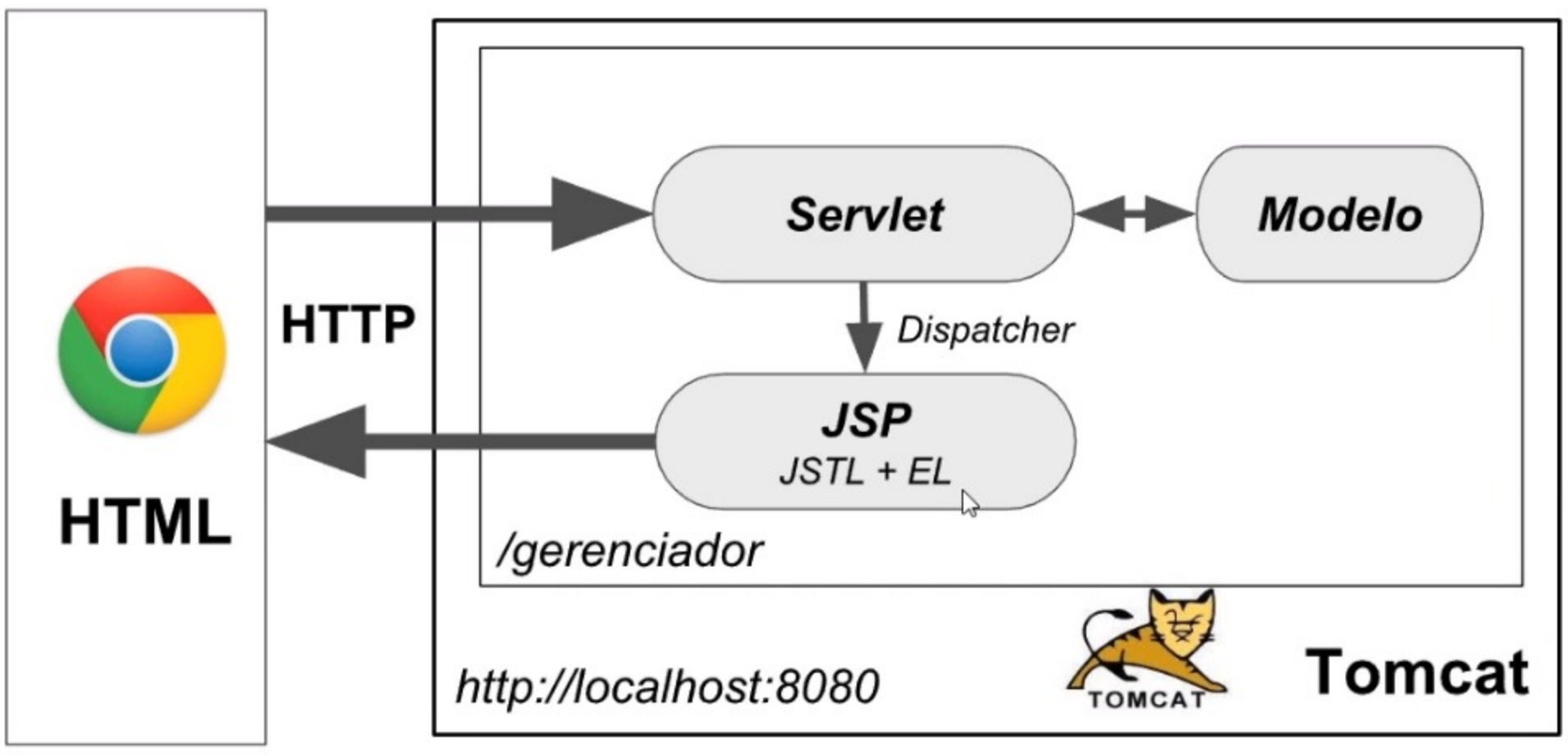
Transcrição

Vamos continuar com o nosso curso e falar um pouco sobre a organização do nosso código. No curso passado fizemos os nossos **Servlets**, o nosso **Modelo** e o nosso **view** JSP com JSTL e Expression Language.

Portanto, já temos uma separação, que chamamos de camadas - o nosso código Java (Servlet e Modelo) e o view JSP estão bem separados.



Porém, as camadas referentes ao Servlet e ao Modelo ainda não estão separadas, e o nosso primeiro passo será fazer essa separação para deixar claro, no nosso código, o que faz parte do Modelo e o que são os Servlets.

Para isso precisaremos criar um novo pacote, o que pode ser feito de várias maneiras. Uma delas é indicar para o Eclipse o que queremos fazer - por exemplo, queremos que a classe `Banco.java` fique no pacote Modelo:

```
package br.com.alura.gerenciador.modelo;
```

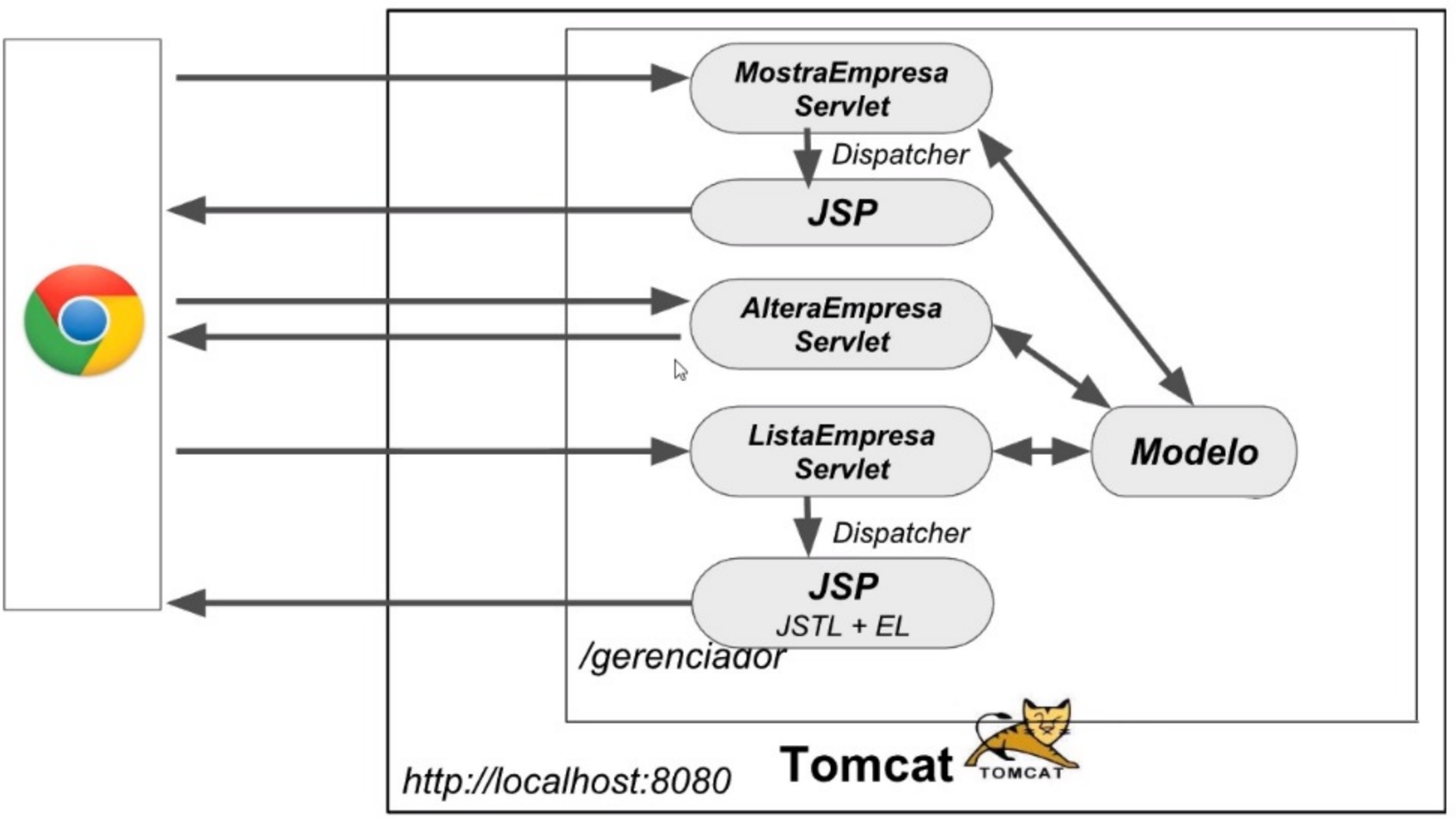
O Eclipse irá indicar um erro, mas na mesma linha irá aparecer um ícone de lâmpada. Clicando nele, temos a opção de mover essa classe para o pacote Modelo ("Move 'Banco.java' to package 'br.com.alura.gerenciador.modelo'"). Como esse pacote ainda não existia, o Eclipse irá criá-lo para nós.

Agora precisaremos mover a classe `Empresa.java` para esse pacote. Podemos fazer isso arrastando essa classe para dentro do pacote (ignorando os erros que aparecerem na tela).

Agora que `Banco.java` e `Empresa.java` estão dentro do pacote Modelo, as outras classes (`AlterarEmpresaServlet` , `ListaEmpresasServlet` , `MostraEmpresaServlet` , `NovaEmpresaServlet` , `OiMundoServlet` e `RemoveEmpresaServlet`) pararam de compilar, pois é necessário importar as classes que movemos.

Para isso, podemos utilizar o atalho "Ctrl + Shift + O" em cada uma das classes com erro, ou clicar sobre os ícones de lâmpada que apareceram e selecionar "Import 'Empresa'" e "Import 'Banco'" nos respectivos erros.

Agora temos pacotes diferentes para cada camada do nosso modelo, mas gostaríamos de ir além. No curso anterior, usamos o seguinte fluxograma para demonstrar como funcionará a alteração de uma empresa:



Nesse caso, precisaríamos de três Servlets:

- MostraEmpresa Servlet -> JSP, para mostrar os dados
- AlteraEmpresa Servlet, para alterar os dados
- ListaEmpresa Servlet -> JSP (JSTL + EL), para listar as empresas novamente

Aqui fica claro que conseguimos chamar cada Servlet através de uma requisição, que é basicamente no que consiste um Servlet - um objeto que podemos chamar através de uma requisição HTTP.

Porém, isso pode ser ruim, já que cada Servlet é uma porta de entrada para nossa aplicação, e cada porta tem o seu risco e precisa ser protegida.

No caso, gostaríamos de ter apenas uma porta para nossa aplicação - ou seja, apenas um Servlet que recebe todas as requisições e depois decide o que fazer. Além disso, escrever um Servlet é um pouco burocrático, pois precisamos extender o `HttpServlet` , temos que ter a configuração `@WebServlet` , precisamos criar um método para determinadas assinaturas, etc.

Antigamente não existiam anotações e precisávamos configurar cada Servlet no `.xml.` , o que motivava ainda mais esse processo. Porém, hoje isso não é mais necessário.

Dessa forma, ao invés de escrevermos oito linhas para cada Servlet, vamos escrever apenas um Servlet que seria a única entrada da nossa aplicação, e vamos reformular nossa aplicação. Até o próximo vídeo!