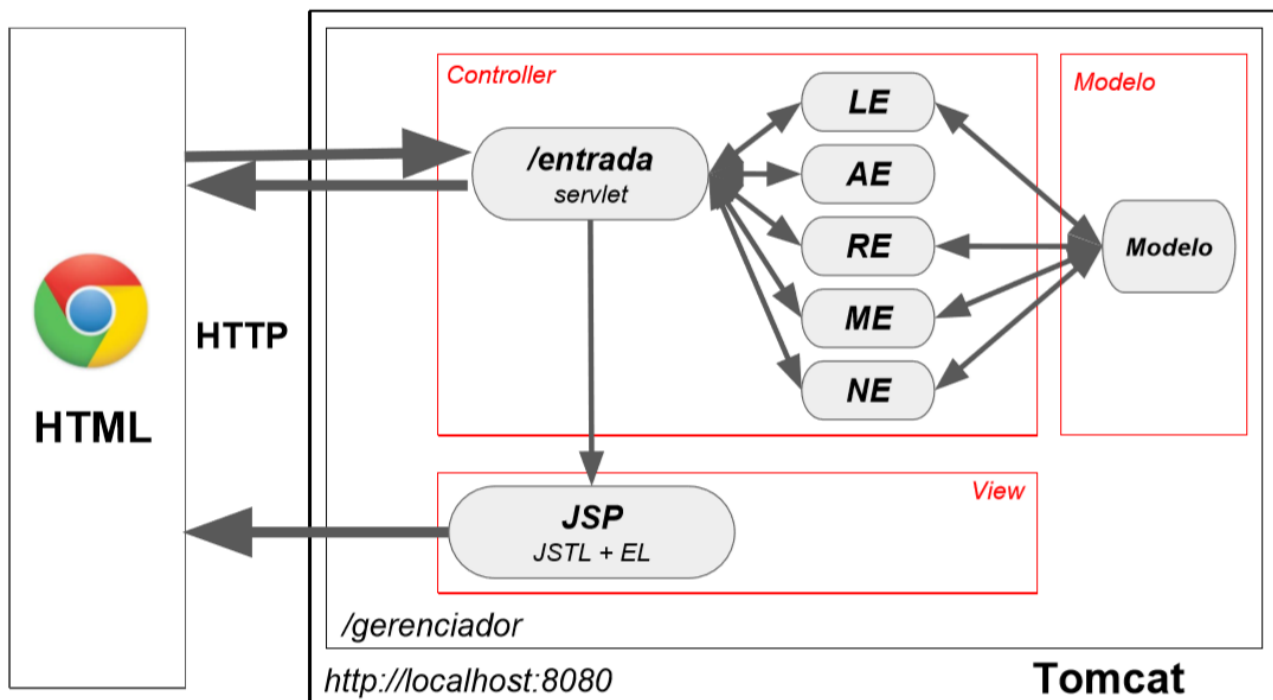


Resumo

Transcrição

Olá, tudo bom?

Vamos recapitular o ponto em que estamos no nosso projeto?



Temos um Servlet que recebe todas as requisições da nossa aplicação. Qualquer link ou formulário que utilizarmos deverá chegar no nosso `/entrada`. A grosso modo nosso código já está funcionando assim, apesar de terem sobrado alguns ajustes.

Agora, esse Servlet decide qual das nossas solicitações (também chamadas de ações ou *actions*) deve ser executada:

- `ListaEmpresas`

- AlteraEmpresa
- RemoveEmpresa
- MostraEmpresa
- NovaEmpresa

Nosso Servlet recebe um parâmetro dentro da requisição para decidir qual ação deve ser executada, controlando o fluxo. Por isso, ele também pode ser chamado de **controlador**. Agora já estamos nos aproximando bastante daquilo que existe em projetos reais.

O nosso controlador decide a ação que deve ser tomada, por exemplo `listaEmpresas`, e usa o nosso modelo para executá-la, certo? Na verdade não é bem assim...

Repare, por exemplo, na nossa classe `ListaEmpresas`:

```
public class ListaEmpresas {  
  
    public void executa(HttpServletRequest request, HttpServletResponse response)  
    {  
  
        System.out.println("listando empresas");  
  
        Banco banco = new Banco();  
        List<Empresa> lista = banco.getEmpresas();  
  
        request.setAttribute("empresas", lista);  
  
        RequestDispatcher rd = request.getRequestDispatcher("/");  
        rd.forward(request, response);  
    }  
}
```

[COPIAR CÓDIGO](#)

Essa classe usa o *Request Dispatcher* para chamar o `/listaEmpresas.jsp`, assim como tínhamos feito no `ListaEmpresaServlet`. Nós somente copiamos o código do Servlet e colamos nessa classe, que é mais simples do que um Servlet.

Porém, quem está chamando o *Request Dispatcher* é a nossa ação, e a ideia é que o nosso Servlet central chame não só a ação, mas também o `.jsp` de maneira central, controlando o fluxo.

Trabalharemos isso no próximo vídeo. Até lá!