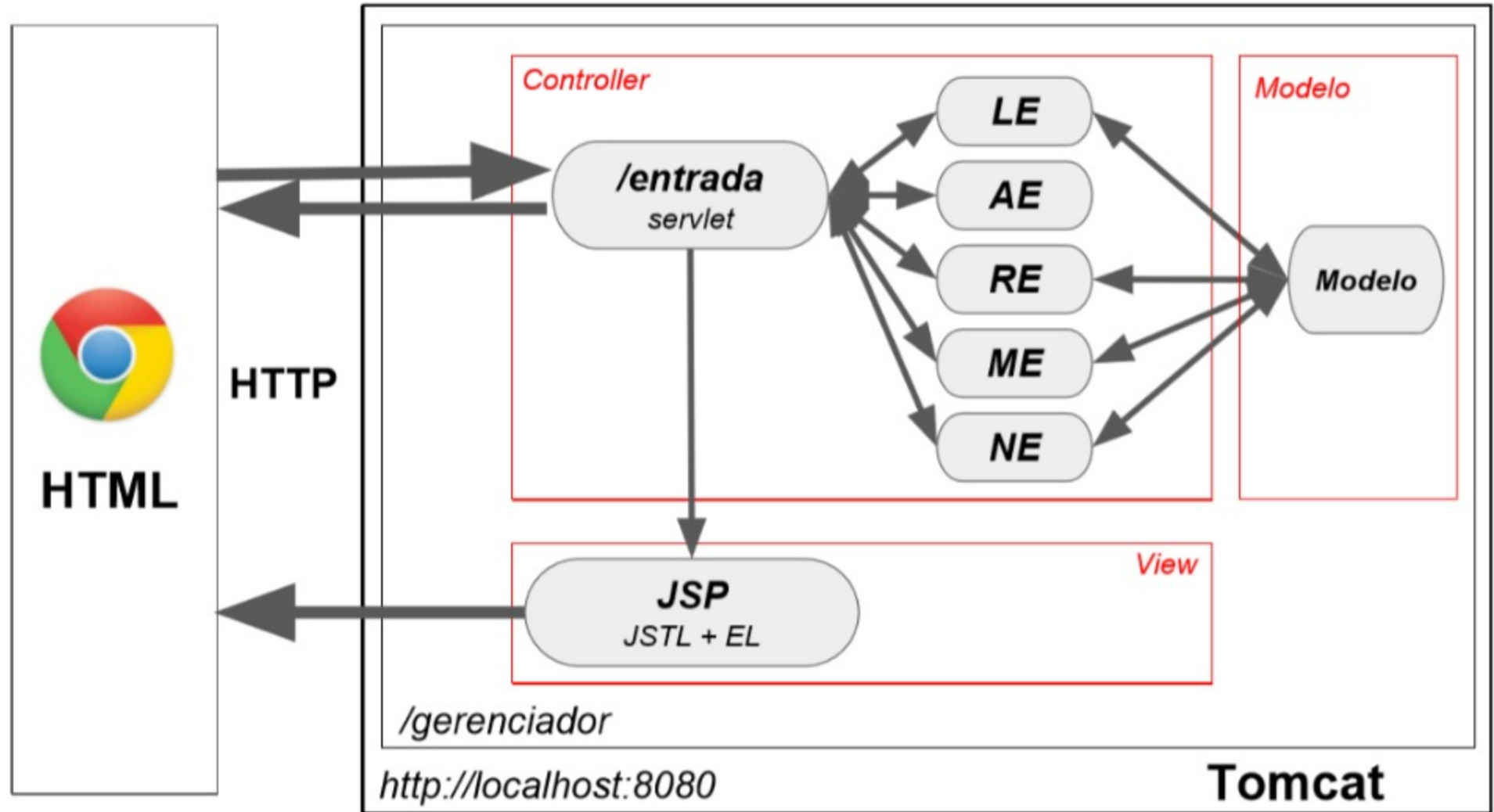


Bem vindo de volta! Nesse vídeo, faremos um pequeno resumo do que fizemos no nosso projeto até agora.



Nós implementamos um padrão arquitetural de controle e organizamos nossa aplicação em camadas. Isso é chamado de MVC - *Model View Controller*. Começamos construindo Servlets, e depois melhoramos para um controlador, e agora chegamos a um resultado apresentável.

No caso, temos um controlador (`UnicaEntradaServlet`) que recebe todas as requisições. Nunca vamos chamar um `.jsp` diretamente, pois eles estão escondidos. É o nosso controlador que recebe e analisa as requisições, para depois instanciar e chamar o método `executa()` das respectivas ações.

A tarefa do controlador é decidir qual ação será executada. Devemos nos atentar ao fato de que as ações também controlam o fluxo e fazem parte da camada **Controller** (controlador).

Por isso, ao invés de colocarmos todo o código em um único controlador, nós extraímos e separamos as ações. Essa camada também é chamada de "lógica" ou, mais comumente, de "*actions*" (ações).

Quando o controlador decide qual ação executar, a ação em si decide qual **Modelo** utilizar. Na verdade, essas ações também definem o nome da página **JSP**. Repare que não é a ação que faz o `redirect` ou o `forward` (chamar o **JSP**), pois isso é tarefa do controlador. Porém, quem decide se será feito um `redirect` ou um `forward` é a ação.

A camada do nosso **Modelo** ainda está um pouco fraca, pois não construímos um domínio muito sofisticado, está tudo em memória, ele não tem persistência e não trabalhamos com um banco de dados de verdade. Porém, foi o suficiente para entendermos como funcionam os Servlets em uma aplicação Web Java.

Futuramente você verá modelos mais sofisticados e, se ainda não sabe, aprenderá a construir a camada de persistência.

Já a camada **View** se preocupa em renderizar os dados. Note que o **View** não sabe de onde vem os dados e não acessa diretamente o banco de dados, apenas recebe e mostra os dados de visualização. Como nosso **View (JSP)** não funciona sozinho e depende que os dados sejam passados para ele, sempre será necessário passar pelas outras duas camadas para que ele funcione.

Por exemplo, nossa `listaEmpresas.jsp` só irá funcionar se chamarmos a ação `ListaEmpresas` antes.

Podemos encontrar esse tipo de estrutura de três camadas no **Spring MVC**, no **VRaptor** e até no **JavaServer Faces**, que apesar de ter alguns detalhes diferentes também é um **MVC**.

Essas três camadas também existem, com pequenas variações, em outras linguagens de programação, como **Ruby on Rails**, **Python** e **.NET**.

No momento não é necessário se preocupar com detalhes da implementação do nosso controlador, pois no futuro você usará outros controladores que já existem, como o próprio **Spring MVC**. É mais importante você se preocupar em criar as ações, o Modelo e o View.

Agora podemos continuar aprendendo mais recursos do nosso mundo Servlet, principalmente autenticação e autorização. Nós queremos que o usuário tenha que fazer login para listar as empresas ou cadastrar uma nova empresa, mas isso fica para a próxima aula. Até lá e obrigado!