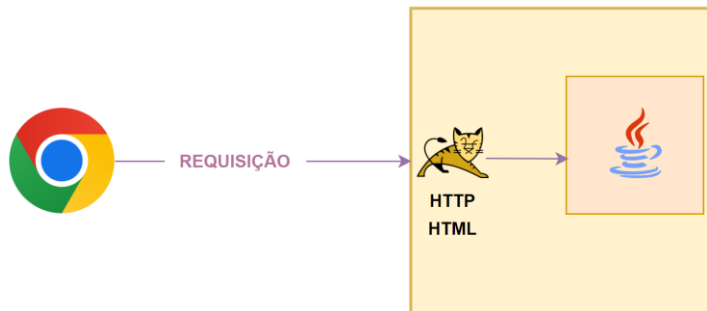




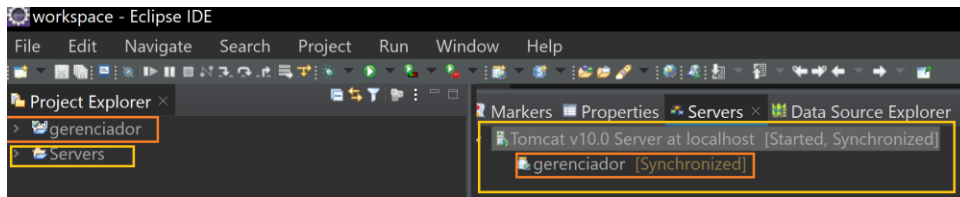
O gato Tom

O **Tomcat** é um servidor para **aplicações Java** que vão trabalhar na WEB, com "**trabalhar na WEB**" eu to querendo dizer que minha aplicação vai funcionar **recebendo requisições HTTP** e vai precisar lidar com **renderizações de HTML**.

Eu posso dizer de uma forma bem "**grosseira**" que o **tomcat** é um servidor que **hospeda** uma **aplicação java**, então o **tomcat** sabe como direcionar **requests para a aplicação** e como **devolver respostas ou mostrar conteúdos dinâmicos em HTML**:



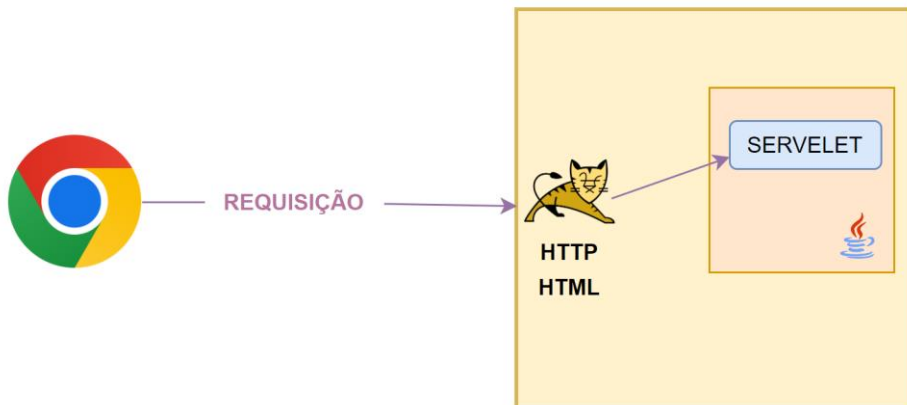
O IntelliJ community não lida bem com o tomcat e projetos java EE, então pra não usar o ultimate eu usei o eclipse:



Java – Servlet

Beleza, o **tomcat** é o servidor web em que a **aplicação java** vai ficar, ele sabe como passar requisições HTTP e sabe como deve ser devolvida respostas até com **HTML**. Mas onde entra o **SERVELET** nessa brincadeira?

O **Servlet** é um objeto especial que fica na nossa **aplicação java**, ele é **especial porque permite ser acionado por protocolo HTTP**, então fica um fluxo: **servidor tomcat** recebe requisição, chama o **servelet**, **servelet** processa a requisição e faz algo:



Para que uma classe **JAVA** seja um **Servelet** ela precisa **estender de Servlet** (com o protocolo específico, nesse caso **HTTP**), a **URL de acesso** desse **servelet** pode ser configurada com uma **anotação** (ou XML), dessa forma o **Tomcat** vai saber como **redirecionar um request específico para a classe de Servlet específica**. Outro ponto aqui é que a classe mãe de **HttpServlet** tem um **método para tratar requisição e resposta** que precisa ser sobrescrito:

```
@WebServlet(urlPatterns = "/oi")
public class OiMundo extends HttpServlet {

    @Override
    protected void service(HttpServletRequest request, HttpServletResponse response) {

    }

}
```

Esse **objeto de request** quem fornece é o próprio **tomcat** ao receber uma **requisição**, já que a gente pode manipular a resposta (e queremos na verdade), podemos usar o **objeto de response**.

Importante dizer que a **resposta** é um fluxo de dados de saída, podemos usar tanto o **outputStream** (binários) ou o **writer** (saída de mais alto nível utilizando caracteres):

```
@WebServlet(urlPatterns = "/oi")
public class OiMundo extends HttpServlet {

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

        PrintWriter out = response.getWriter();

        out.print("<html>");
        out.print("<body>");
        out.print("<h1>Escrevi um Servlet</h1>");
        out.print("</body>");
        out.print("</html>");

    }

}
```

← → ↺ 🏠 ⓘ localhost:8080/gerenciador/oi

Escrevi um Servlet