

Disciplina Regular 3

Desenvolvimento Web com Java EE

Graduação em Engenharia de Software - 2020

Etapa 1 Aula 1

Servlets

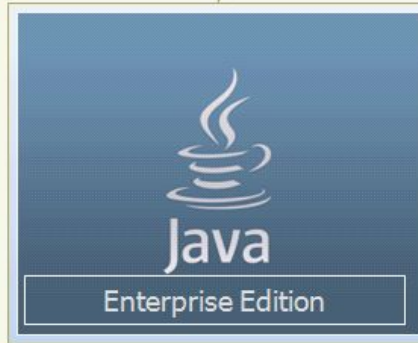
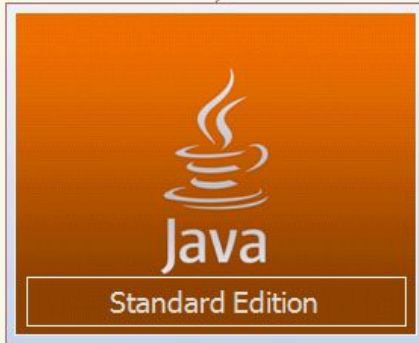
Para acompanhar pelo Moodle você
deve estudar as etapas 1 e 2

Competências Trabalhadas Nesta Etapa

- Escrever interfaces simples com HTML e CSS.
- Construir aplicativos Java utilizando Servlets.

Java Enterprise Edition

Plataforma Java e suas Edições





Enterprise Edition



Usaremos o
Spring Framework
no lugar desse
recurso

Specification	Java EE 6 ^[11]	Java EE 7 ^[3]	Java EE 8 ^[5]
Servlet	3.0	3.1	4.0
JavaServer Pages (JSP)	2.2	2.3	2.3
Unified Expression Language (EL)	2.2	3.0	3.0
Debugging Support for Other Languages (JSR-45)	1.0	1.0	1.0
JavaServer Pages Standard Tag Library (JSTL)	1.2	1.2	1.2
JavaServer Faces (JSF)	2.0	2.2	2.3
Java API for RESTful Web Services (JAX-RS)	1.1	2.0	2.1
Java API for WebSocket (WebSocket)	n/a	1.0	1.1
Java API for JSON Processing (JSON-P)	n/a	1.0	1.1
Common Annotations for the Java Platform (JSR-250)	1.1	1.2	1.3
Enterprise JavaBeans (EJB)	3.1 Lite	3.2 Lite	3.2
Java Transaction API (JTA)	1.1	1.2	1.2
Java Persistence API (JPA)	2.0	2.1	2.2
Bean Validation	1.0	1.1	2.0
Managed Beans	1.0	1.0	1.0
Interceptors	1.1	1.2	1.2
Contexts and Dependency Injection for the Java EE Platform	1.0	1.1	2.0
Dependency Injection for Java	1.0	1.0	1.0

O Que é JEE?

- JEE – foco no desenvolvimento de Sistemas Corporativos:
 - Sistemas corporativos têm maior escala de processamento.
 - Requer a implementação de disponibilidade, segurança, escalabilidade etc.
- “Camadas” JEE:
 - **JEE Web Tier** composta por JavaServer Pages (JSP) e Servlets.
 - **JEE Business Tier** composta por Enterprise Java Beans (EJB).
- As “camadas” JEE são implementadas através de softwares chamados “servidores de aplicação”.

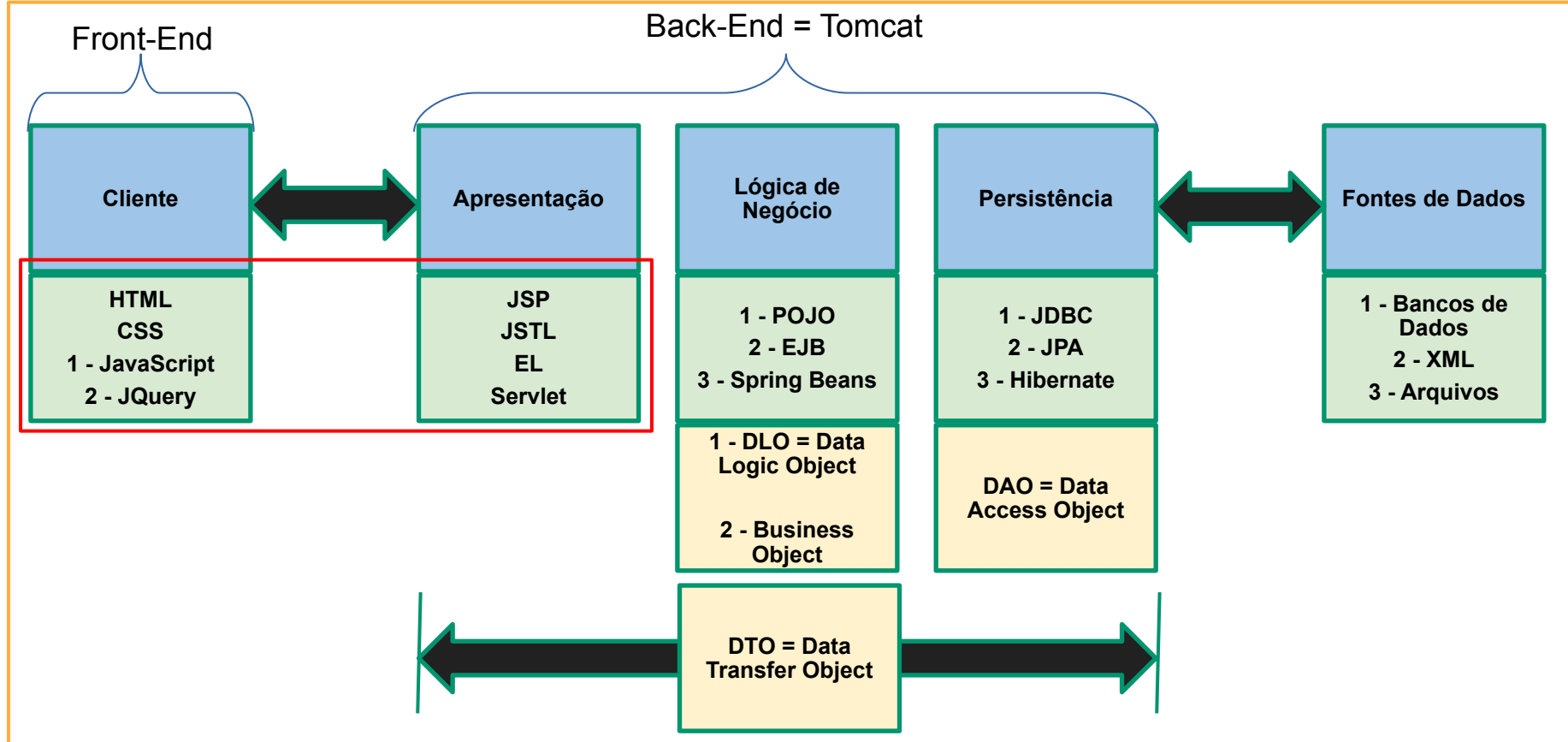
O Que é JEE?

- Os servidores de aplicação são distribuídos por diversos fornecedores.
- Alguns são *open-source* e outros são pagos.
- Destacam-se três servidores *open-source*:
 - JEE Web Tier:
 - **Tomcat**, da Apache Foudation.
 - JEE Business Tier:
 - **Jboss**, da Red Hat.
 - **Glassfish**, da Oracle.



O endereço <http://www.theserverside.com/matrix> contém um catálogo desses servidores.

Modelo de Referência de Arquitetura



Servlets

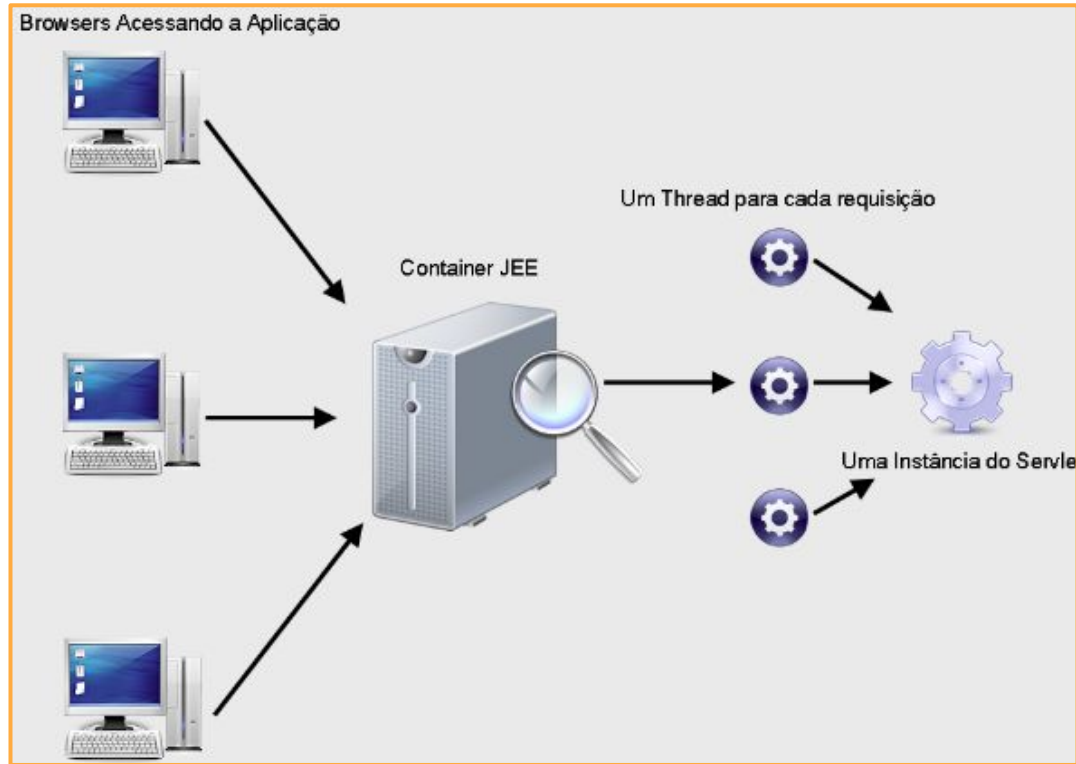
O que são Servlets?

- Servlet é uma classe Java executada em um **servlet container** ou **servidor de aplicação**.
- Um servlet se integra a um servidor de aplicação, implementando as APIs da plataforma JEE.
- Recebe requisições de clientes, processa dados e envia HTML, XML ou texto como resposta.
- **Sua principal função é controlar o recebimento de dados de formulários, processar esses dados e redirecioná-los para páginas de resultado.**

O que são Servlets?

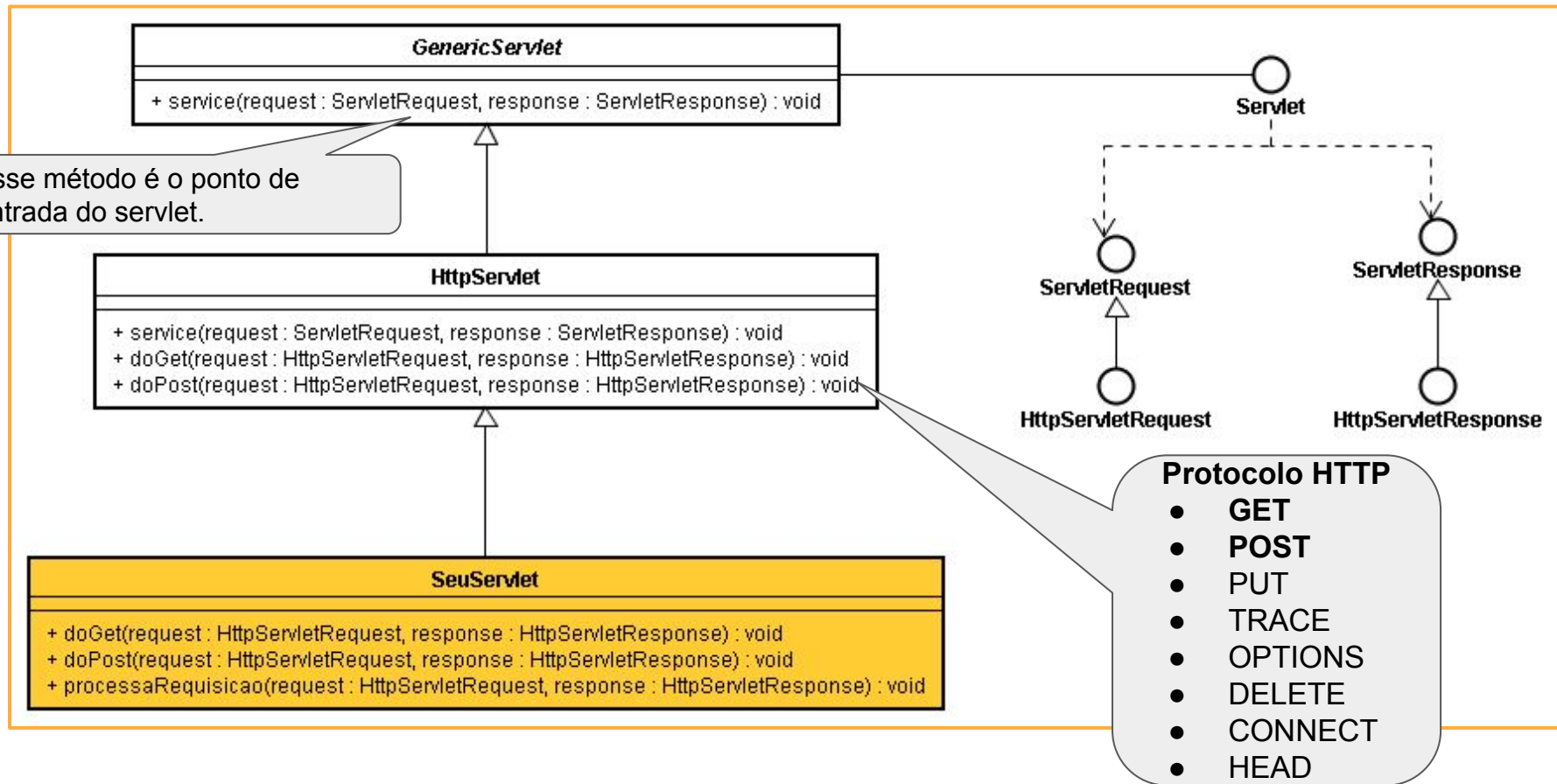
- **Servidor de Aplicação** é um software que roda em um computador ligado à rede e tem a função de receber requisições, executar servlets e responder aos browsers.
- Ele controla todo o processo de recebimento da requisição e envio das respostas, criando os objetos necessários e repassando as requisições para os servlets.
- Você sobrescreve os métodos que desejar e o container os chama automaticamente. Dessa forma, o servlet é como uma extensão do container.

Execução de Servlets



Uma aplicação web tem pelo menos duas partes: a parte cliente (browser) e os componentes que rodam no servidor (servlets).

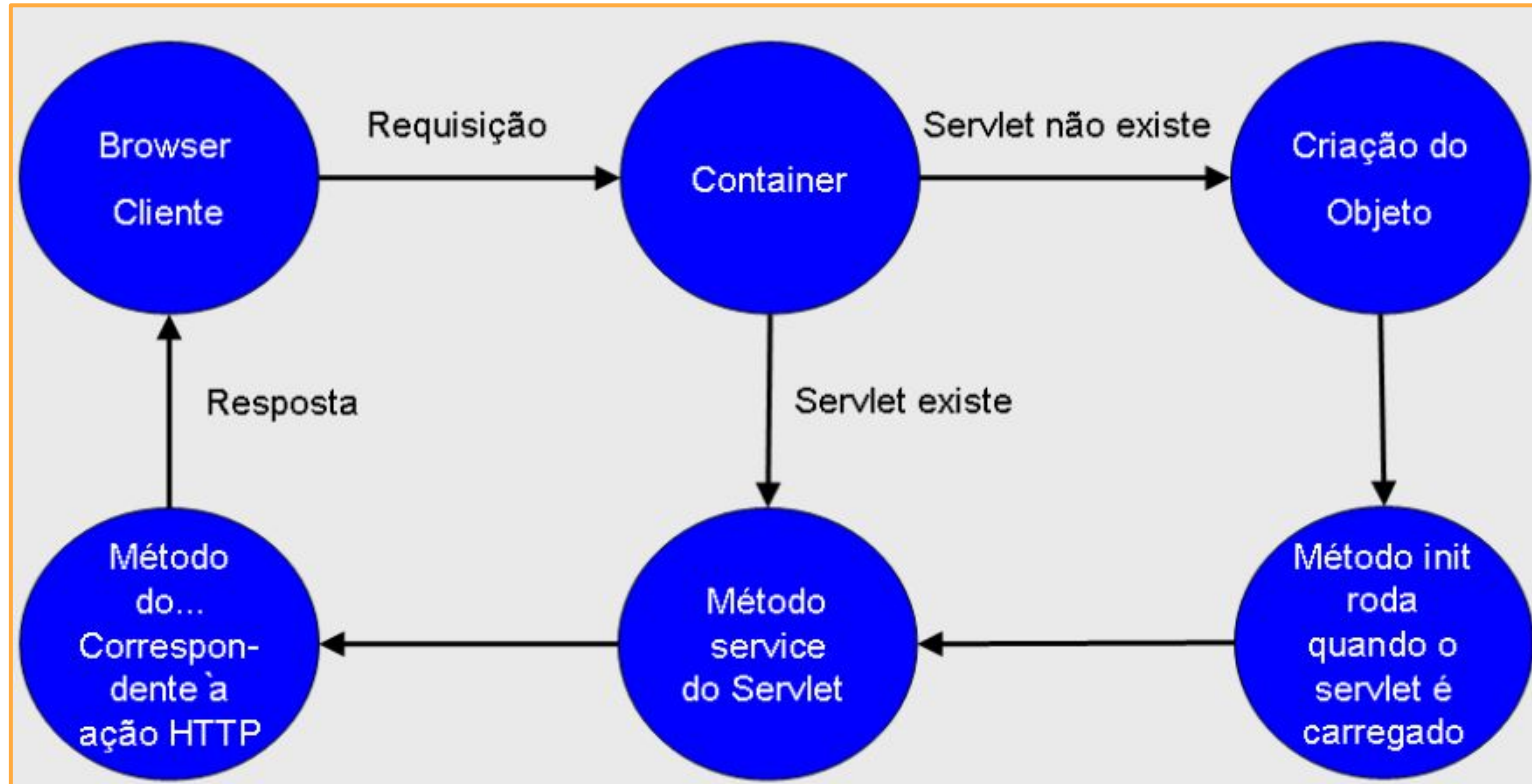
Hierarquia de Classes



Hierarquia de Classes

- Devemos criar uma subclasse de `HttpServlet` para implementar um servlet voltado às nossas aplicações.
- Sobrescrevemos os métodos desejados de acordo com o nosso objetivo.
- Em geral, os métodos **doGet** e **doPost** são os que vamos sobrescrever. Os outros métodos normalmente não são utilizados em uma aplicação web.
- **Jamais devemos sobrescrever o método `service`**, pois ele contém o código que verifica o tipo de requisição HTTP feita para chamar o método “do---” correto.

Ciclo de Vida dos Servlets



O container carrega e descarrega os servlets conforme a demanda.

Ciclo de Vida dos Servlets

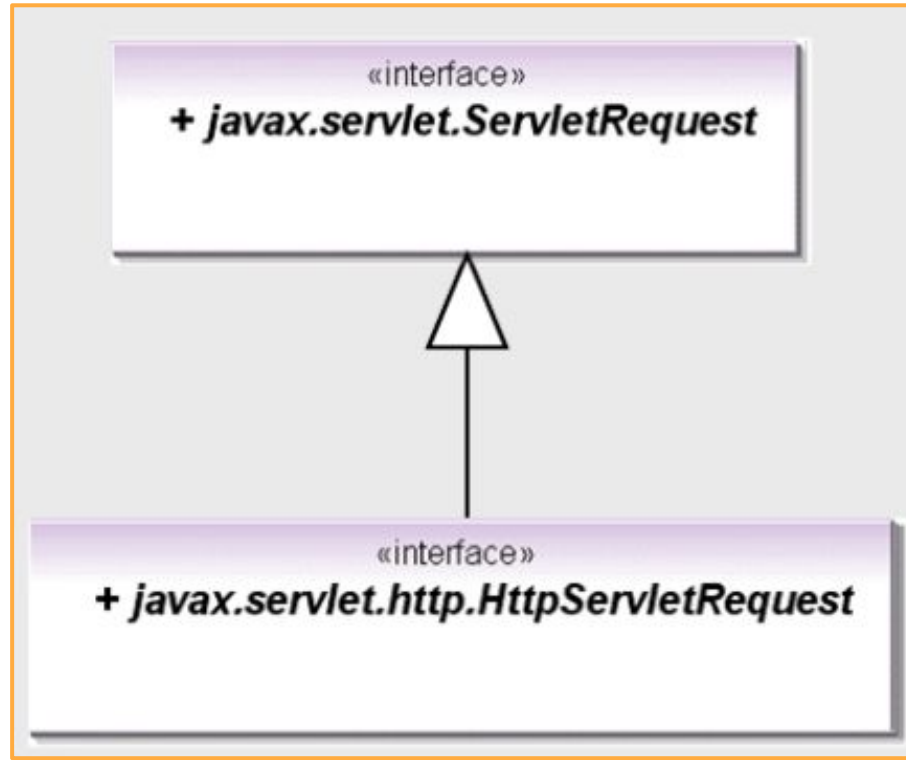
- Para criar um servlet, você deve:
 - Criar uma classe que estenda **HttpServlet**.
 - Sobrescrever os métodos **doGet** e **doPost**, chamando um método **processarRequisicao** ou **processRequest**.
 - Implementar o método **processarRequisicao**, executando o código desejado.
 - Você pode sobrescrever apenas **doGet** ou **doPost**, se assim desejar.

Exercício

- Crie um projeto Web e implemente um servlet de exemplo.

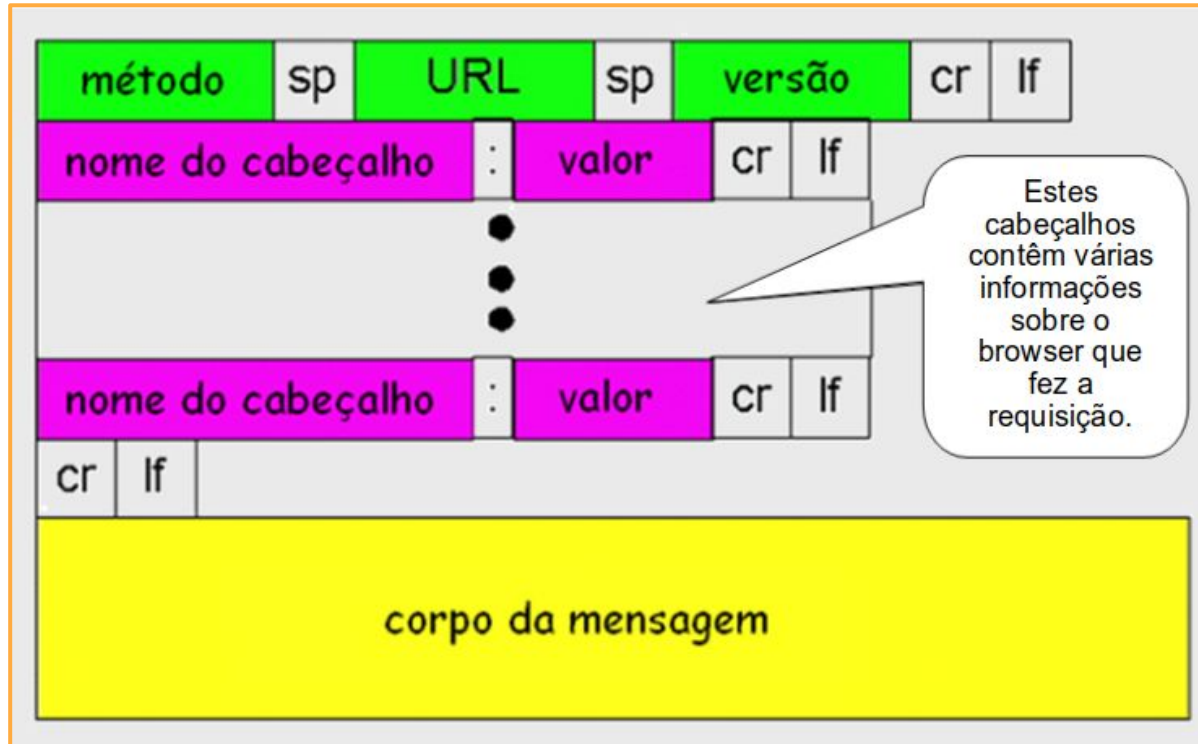
Requisição

Interface HttpServletRequest



A classe que implementa essa interface deriva objetos que contêm os dados transmitidos pelo browser.

Estrutura da Requisição HTTP



Uma requisição é um bloco organizado de bytes.

Estrutura da Requisição HTTP

GET /servlet/com.infnet.Exemplo HTTP/1.1

Accept: image/gif, image/jpeg, application/pdf

Accept-Language: pt-BR

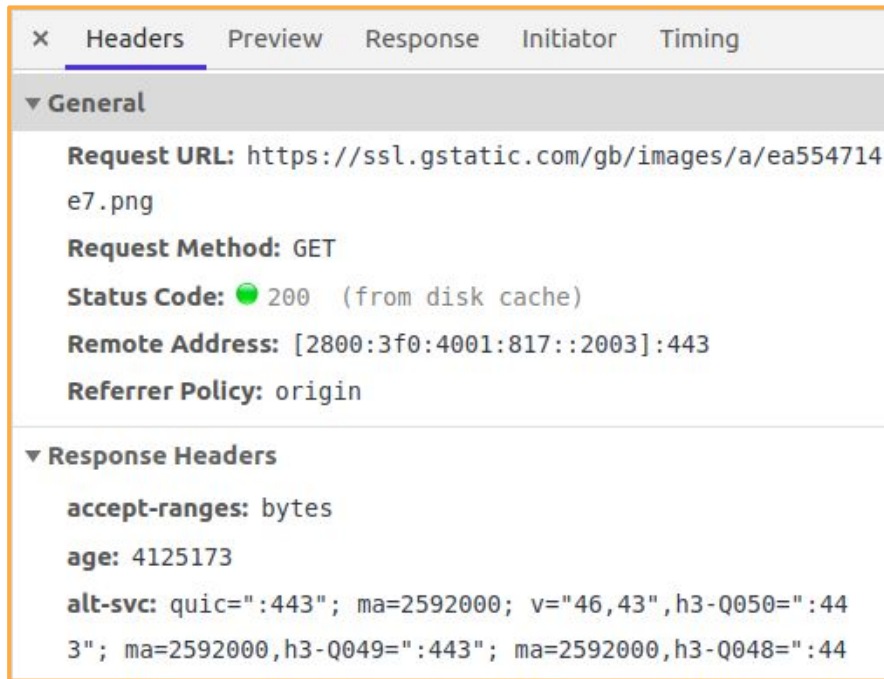
Accept-Charset: iso-8859-1

Connection: keep-alive

Host: www.infnet.com.br

Referer: http://www.infnet.com.br/index.html

User-Agent: Mozilla/4.7 [en] (win98; U)



The screenshot shows the 'Headers' tab of a web browser's developer tools. The 'General' section displays the following information:

- Request URL:** https://ssl.gstatic.com/gb/images/a/ea554714e7.png
- Request Method:** GET
- Status Code:** 200 (from disk cache)
- Remote Address:** [2800:3f0:4001:817::2003]:443
- Referrer Policy:** origin

The 'Response Headers' section displays the following information:

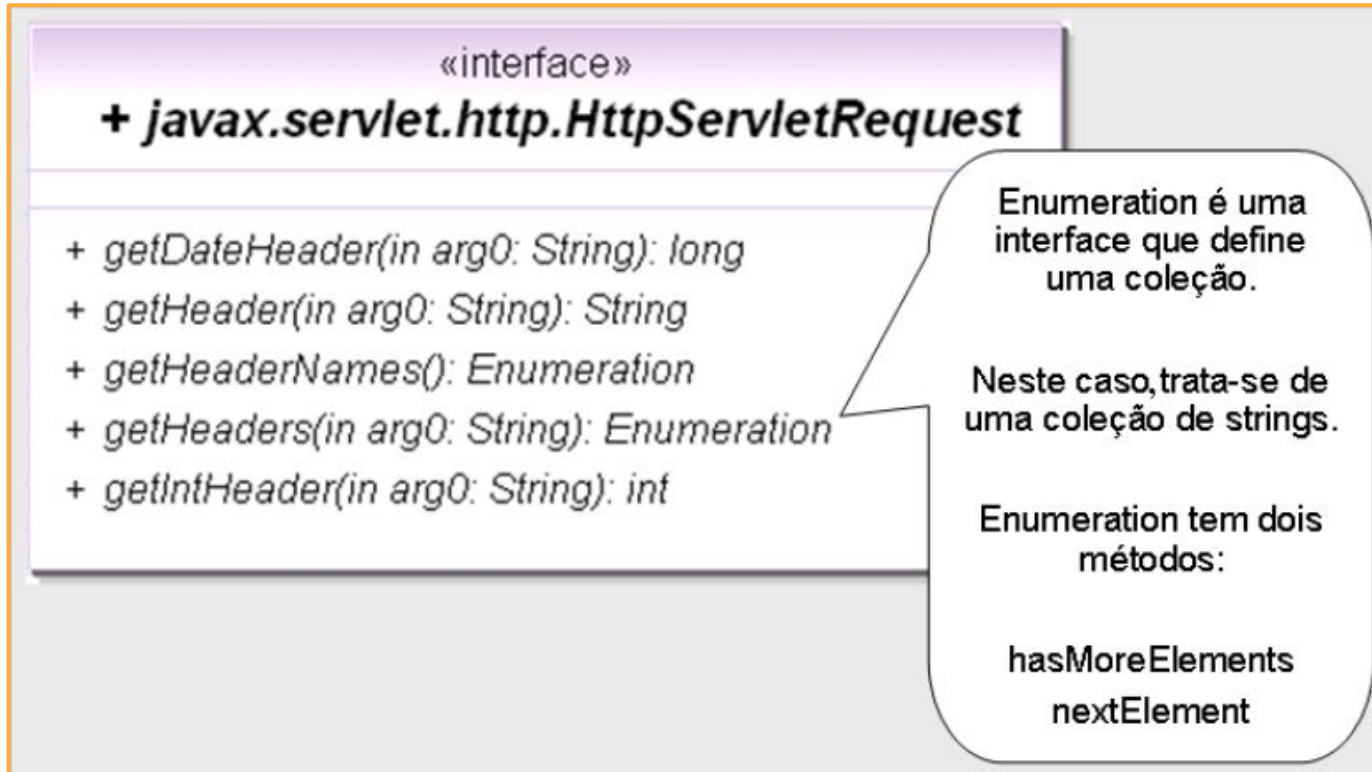
- accept-ranges:** bytes
- age:** 4125173
- alt-svc:** quic=":443"; ma=2592000; v="46,43",h3-Q050=":443"; ma=2592000,h3-Q049=":443"; ma=2592000,h3-Q048=":443"

Observe os dados do browser representados pelos cabeçalhos da requisição.

Métodos HTTP

- GET
 - Usado quando um usuário clica em um hyperlink de uma página HTML ou quando o endereço é digitado no browser.
- POST
 - Encapsulado pelo método doPost() - usado quando a informação é passada por formulários HTML.
- HEAD.
- PUT.
- DELETE.
- OPTIONS.
- TRACE.

Cabeçalhos HTTP de Requisição



Como obter as informações de TODOS os cabeçalhos de uma requisição?

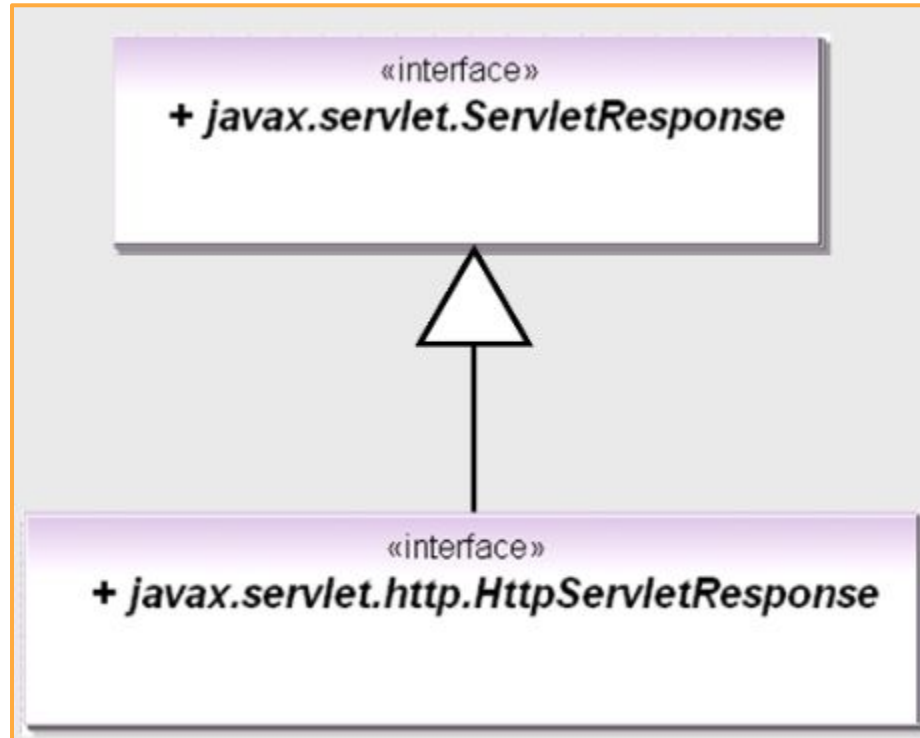
Exercício

- Crie um projeto Web e implemente um servlet que liste todos os cabeçalhos da requisição.
 - Utilize o método `getHeaderNames` para obter a lista de nomes de cabeçalhos e o método `getHeader` para obter os seus valores.
 - Pesquise o significado de cada um dos cabeçalhos.

Se for preciso, consulte a API de Enumeration (JavaDoc).

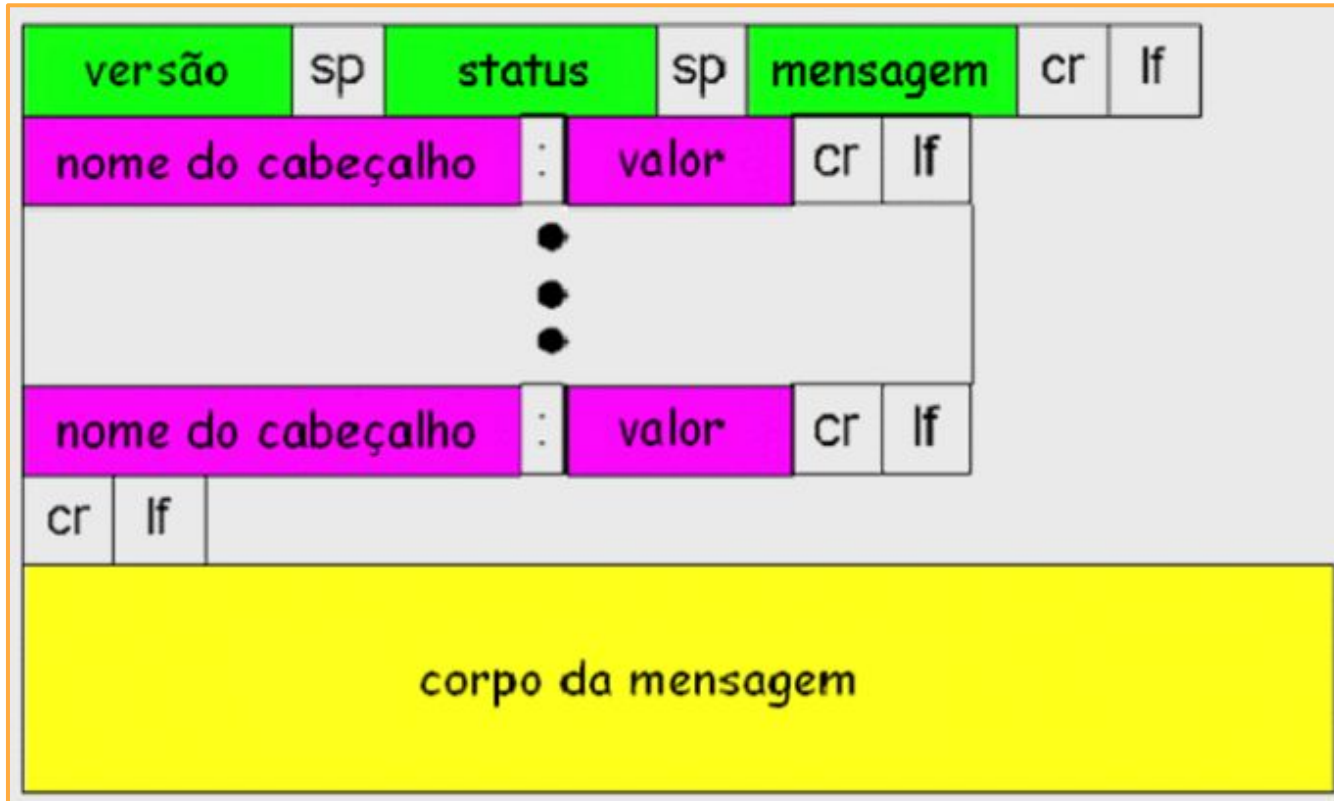
Resposta

Interface HttpServletResponse



A classe que implementa essa interface deriva objetos com os dados que serão transmitidos para o browser.

Estrutura de Resposta HTTP



A resposta é semelhante à requisição: um bloco organizado de bytes.

Estrutura de Resposta HTTP

HTTP/1.0 200 OK

Content-Type:text/html

Date: Sun, 14 Oct 2001 17:32:37 GMT

Server: Apache Tomcat/5.5 (HTTP/1.1 Connector)

Connection: close

<html><head>

<h3>Java, Servlets, JSP e Struts!</h3>

</head></html>

HTTP Status Codes



Observe que o corpo da resposta é o HTML que será interpretado pelo browser.

Tipos MIME

- Vários tipos de arquivo podem ser gerados por um servlet, não apenas texto.
- Esses tipos são especificados no padrão MIME (*Multipurpose Internet Mail Extensions*)
- Parte do padrão MIME corresponde à definição do tipo de conteúdo (descrição dos tipo de resposta).
 - Existem definições para PDF, Excel, imagens, som etc.

Métodos Úteis

- A interface HttpServletResponse tem alguns métodos interessantes:

```
// Solicita ao browser que carregue a página em 5s
```

```
response.setHeader("Refresh", "5");
```

```
// Pega da resposta um objeto para envio dos dados ao cliente
```

```
PrintWriter out = response.getWriter();
```

```
// Gera Html
```

```
out.println("<html>");
```

Exercício

- Criar um projeto de notícias que mostre manchetes diferentes a cada 5 segundos.
 1. Crie o projeto.
 2. Crie o servlet NoticiasServlet no pacote br.edu.infnet.noticias.
 3. Declare um atributo de instância (propriedade da classe) chamado notícias. Esse atributo é um array de strings.
 4. Inicie o atributo com três manchetes.
 5. Crie outro atributo de instância chamado “atual”, do tipo inteiro.
 6. No método processRequest, retorne apenas a notícia atual, indicada pelo atributo “atual”.
 7. Incremente o atributo.
 8. Se o valor for maior que o tamanho do array, volte para a primeira notícia.
 9. Defina o cabeçalho de resposta “Refresh” com o valor de 5 (cinco) segundos.