

**UNIP**  
UNIVERSIDADE PAULISTA

## Aplicação Linguagem de Programação Orientada a Objeto

### ALPOO – 04.1 - Arquitetura web - Fundamentos

Prof. Msc Wanderley Gonçalves Freitas

Prof. Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 1 de 14

**UNIP**  
UNIVERSIDADE PAULISTA

## Plataforma Web

**Importante (1):**

- Baseada em HTTP (RFC 2068)
  - Protocolo simples de transferência de arquivos
  - Sem estado (não mantém sessão aberta)
- Funcionamento (simplificado):

Máquina www.xyz.com

Soquete de Serviço: 80

Servidor HTTP

index.html

Cliente HTTP (browser)

Abre conexão para www.xyz.com:80

Uma requisição: GET /index.html HTTP/1.1 ...

Uma resposta: HTTP/1.1 200 OK ...

Fecha conexão

Prof. Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 14

**UNIP**  
UNIVERSIDADE PAULISTA

## Cliente e servidor HTTP

**Importante (1):**

- Servidor HTTP
  - Gerencia sistema virtual de arquivos e diretórios
  - Mapeia pastas do sistema de arquivos local (ex: c:\htdocs) a diretórios virtuais (ex: /) acessíveis remotamente (notação de URI)
- Papel do servidor HTTP
  - Interpretar requisições HTTP do cliente (métodos GET, POST, ...)
  - Devolver resposta HTTP à saída padrão (código de resposta 200, 404, etc., cabeçalho RFC 822\* e dados)
- Papel do cliente HTTP
  - Enviar requisições HTTP (GET, POST, HEAD, ...) a um servidor. Requisições contêm URI do recurso remoto, cabeçalhos RFC 822 e opcionalmente, dados (se método HTTP for POST)
  - Processar respostas HTTP recebidas (interpretar cabeçalhos, identificar tipo de dados, interpretar dados ou repassá-los).

Prof. Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 3 de 14

**UNIP**  
UNIVERSIDADE PAULISTA

## Principais métodos HTTP (requisição)

**Importante (2):**

- GET - pede ao servidor um arquivo (informado sua URI) absoluta (relativa à raiz do servidor)
 

```
GET <uri> <protocolo>/<versão>
<Cabeçalhos HTTP>: <valores> (RFC 822)
<linha em branco>
```

  - GET pode enviar dados através da URI (tamanho limitado)
 

```
<uri>?dados
```
  - Método HEAD é idêntico ao GET mas servidor não devolve página (devolve apenas o cabeçalho)
- POST - envia dados ao servidor (como fluxo de bytes)
 

```
POST <uri> <protocolo>/<versão>
<Cabeçalhos HTTP>: <valores>
<linha em branco>
<dados>
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 4 de 14

**UNIP**  
UNIVERSIDADE PAULISTA

## Cabeçalhos HTTP

- Na **requisição**, passam informações do cliente ao servidor
  - Fabricante e nome do browser, data da cópia em cache, cookies válidos para o domínio e caminho da URL da requisição, etc.
- Exemplos:
 

```
User-Agent: Mozilla 5.5 (Compatible; MSIE 6.0; MacOS X)
If-Modified-Since: Thu, 23-Jun-1999 00:34:25 GMT
Cookies: id=344; user=Jack; flv=yes; mis=no
```
- Na **resposta**, passam informações do servidor ao cliente
  - Tipo de dados do conteúdo (text/xml, image/gif) e tamanho, cookies que devem ser criados, endereço para redirecionamento, etc.
- Exemplos:
 

```
Content-type: text/html; charset-iso-8859-1
Refresh: 15; url=/pags/novaPag.html
Content-length: 246
Set-Cookie: nome=valor; expires=Mon, 12-03-2001 13:03:00 GMT
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 5 de 14

**UNIP**  
UNIVERSIDADE PAULISTA

## Comunicação HTTP: detalhes

- Página HTML
 

```

```

Interpreta HTML → Gera requisição GET
- Requisição: browser solicita imagem
 

```
GET tomcat.gif HTTP/1.1
User-Agent: Mozilla 6.0 [en] (Windows 95; I)
Cookies: querty=uiop; SessionID=D236S11943245
```

Linha em branco termina cabeçalhos
- Resposta: servidor devolve cabeçalho + stream
 

```
HTTP 1.1 200 OK
Server: Apache 1.32
Date: Friday, August 13, 2003 03:12:56 GMT-03
Content-type: image/gif
Content-length: 23779
```



tomcat.gif

```
!#GIF89~| 7
.55.a||6rU4 ...
```

Prof. Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 6 de 14

## Tomcat

- O Apache Jakarta Tomcat é a **implementação de referência** para aplicações Web
  - Tomcat 3.x - I.R. para servlets 2.2 e JSP 1.1
  - Tomcat 4.x - I.R. para servlets 2.3 e JSP 1.2
- Em produção, geralmente é acoplado a um servidor de páginas estáticas eficiente (Apache, ou outro)
- Em desenvolvimento, pode-se usar o servidor distribuído com o Tomcat
- Instale o Tomcat
  - Use `c:\tomcat-4.0` ou `/usr/jakarta/tomcat` e defina as variáveis `CATALINA_HOME` e `TOMCAT_HOME` apontando para o mesmo lugar

Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 7 de 14

## Estrutura do Tomcat

**Importante (3):**

<ul style="list-style-type: none"> <li>bin</li> <li>webapps</li> <li>ROOT</li> <li>common           <ul style="list-style-type: none"> <li>classes</li> <li>lib</li> </ul> </li> <li>classes</li> <li>lib</li> <li>server</li> <li>conf</li> <li>logs</li> <li>work</li> <li>temp</li> </ul>	<p>Executáveis. <b>Para iniciar o Tomcat, rode <code>startup.bat</code></b></p> <p>Contém pastas de contextos (aplicações Web)</p> <p>Contexto raiz (coloque suas páginas Web aqui)</p> <p>Classpaths do Tomcat (valem para todas as aplicações)</p> <p>Classpath (coloque classes aqui)</p> <p>Classpath de JARs (o <code>servlet.jar</code> está aqui)</p> <p>Classpath do servidor (não use)</p> <p>Classpath do servidor para JARs (não use)</p> <p>Executáveis do Tomcat</p> <p>Arquivos de <b>configuração</b> (<code>server.xml</code> e outros)</p> <p><b>Logs</b> para todas as aplicações</p> <p>Contém servlets gerados a partir de JSPs</p> <p>Diretório temporário</p>
--	---

- `$TOMCAT_HOME/conf/server.xml`: configuração do servidor (onde se pode configurar novos contextos)
- `$TOMCAT_HOME/common/lib/*.jar`: Classpath para todas as aplicações que rodam no container (use com cuidado para evitar conflitos)

Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 8 de 14

## Como iniciar e parar o Tomcat

- No diretório `bin/` há vários arquivos executáveis
- Para iniciar o Tomcat, use, a partir do `bin/`
  - `./startup.sh` ou
  - `startup.bat`
- Para encerrar o Tomcat, use
  - `./shutdown.sh` ou
  - `shutdown.bat`
- Crie atalhos na sua área de trabalho para esses arquivos ou para o diretório `bin` (se eles já não existirem). Eles serão usados frequentemente
- Refira-se aos arquivos no diretório `logs/` para realizar depuração de suas aplicações. Crie um atalho para ele também

Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 9 de 14

## Como implantar uma aplicação no Tomcat

- Há três maneiras
  - Transferir os arquivos da aplicação (JSP, servlets) para **contextos** já reconhecidos pelo servidor
  - Configurar o servidor para que reconheça um **novo contexto** onde os arquivos da aplicação residem (`server.xml`)
  - Implantar a aplicação como um WebArchive (WAR)
- Contextos são diretórios devidamente configurados que o Tomcat reconhece como aplicações Web
- O contexto raiz chama-se **ROOT**.
  - Arquivos copiados para `$TOMCAT_HOME/webapps/ROOT/` podem ser acessados via `http://servidor:8080/`
  - Servlets em `webapps/ROOT/WEB-INF/classes` podem ser acessados via `http://servidor:8080/servlet/`

Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 10 de 14

## Referências bibliográficas

FAEMAN, Julio. Design Patterns Aplicados. **Java Magazine**, Rio de Janeiro, v. 3, n.20, p. 52 a 58, 2005.

FOWLER, Martin. **UML essencial : um breve guia para a linguagem** – padrão de modelagem de objetos. Tradução Vera Pezerico e Christian Thomas Prices. 3. ed. Porto Alegre : Bookman, 2005.

GRADY, Booch; RUMBAUGH, James; JACOBSON, Ivar. **UML, guia do usuário** . tradução de Fábio Freitas da Silva. Rio de Janeiro: Editora Campus. 2000.

JOHN, Metsker, Steven. **Padrões de projeto em java**. Porto Alegre: Editora Bookman. 2004.

KAZUO, Jorge. Catálogo de Soluções : Padrões de projeto na arquitetura JE22. **Java Magazine**, Rio de Janeiro, v. 3, n.20, p. 12 a 13, 2005.

LOZANO, Fernando. Padrões de Projeto e Classes de Negócio. **Java Magazine**, Rio de Janeiro, v. 3, n.20, p. 10 a 11, 2005. Cateína.

MENDES, Antonio. **Arquitetura de software : desenvolvimento orientado para arquitetura**. Rio de Janeiro: Editora Campus. 2002.

QUATRANI, Terry. **Modelagem visual racional rose 2000 e UML**. Rio de Janeiro: Editora Ciência Moderna Ltda. 2001.

[Scott W. Ambler. The Elements of Java Style](http://www.amblysoft.com/books/elementsJavaStyle.html). Disponível em: <http://www.amblysoft.com/books/elementsJavaStyle.html> />. Acesso em : 01/12/2008

[Writing Robust Java Code](http://www.amblysoft.com/downloads/javaCodingStandards.pdf/)  
<<http://www.amblysoft.com/downloads/javaCodingStandards.pdf/>> Acesso em : 01/12/2008

Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 11 de 14

## Referências Bibliográficas

BRUCE, Eckel. **Thinking in Java**. 4. ed. Massachusetts: Editora Prentice hall: 2006.

FOWLER, Martin. **UML essencial : um breve guia para a linguagem** – padrão de modelagem de objetos. Tradução Vera Pezerico e Christian Thomas Prices. 3. ed. Porto Alegre : Bookman, 2005.

FURLAN, Jose Davi. **Modelagem de Objetos Através da UML** São Paulo: Editora Makron Books, 2000, ISBN 8534609241,

GRADY, Booch; RUMBAUGH, James; JACOBSON, Ivar. **UML, guia do usuário** . tradução de Fábio Freitas da Silva. Rio de Janeiro: Editora Campus. 2000.

SIERRA, kathy; BATES, Bert. **Certificação Sun para programadores e desenvolvedor Java 2**. Rio de Janeiro: Editora Altas Books. 2003.

SIERRA, kathy; BATES, Bert. **Head first java 5**. 2. ed. Sebastopol: Editora O' Reilly Media. Books. 2005.

TOM, Pender. **UML, a bíblia**. Rio de Janeiro: Elsevier, 2004.

Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 12 de 14

**UNIP**  
UNIVERSIDADE PAULISTA

## Perguntas



Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 13 de 14

**UNIP**  
UNIVERSIDADE PAULISTA

## Obrigado

Wanderley

[Wanderley.unip@gmail.com](mailto:Wanderley.unip@gmail.com)

[www.wg.pro.br](http://www.wg.pro.br)

Prof: Msc Wanderley Gonçalves Freitas ALPOO 04 – Arquitetura Web 14 de 14