

Introdução ao Java EE (Servlets e JSP)

Laércio Baldochi

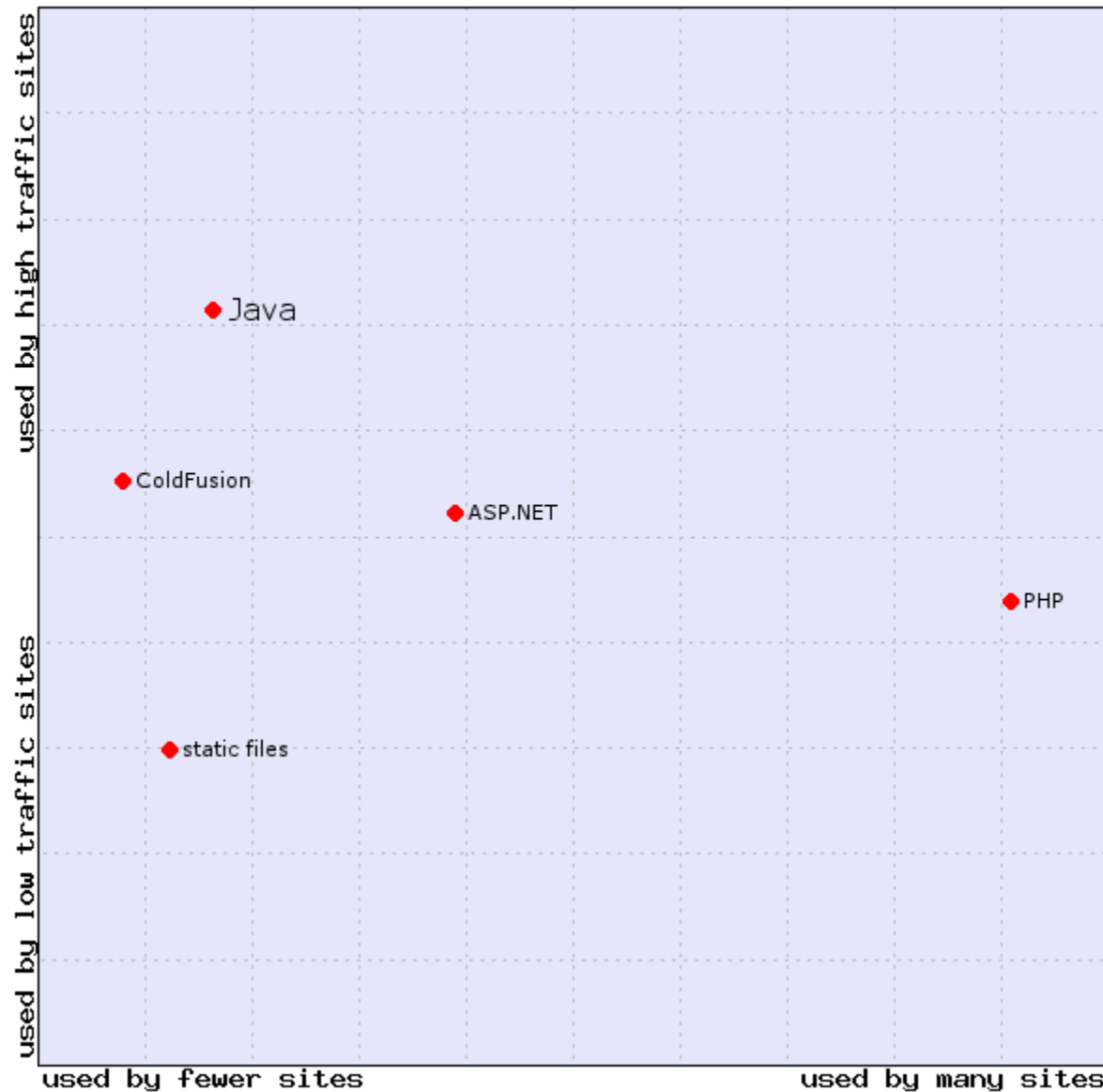
COM222
Aula 04

Conteúdo

- Introdução às aplicações Web
 - Protocolo HTTP
 - Containers Web
 - Servlets

Por que usar Java na Web?

Java Market Position, 14 Mar 2017, W3Techs.com



Por que usar Java na Web?

Informações adicionais:

<https://w3techs.com/technologies/details/pl-java/all/all>

https://w3techs.com/technologies/overview/programming_language/all

Introdução

- JEE permite a criação de aplicações Web dinâmicas
- Provê containers Web e duas APIs para desenvolvimento de aplicações
 - API Servlets
 - API Java Server Pages (JSP)
- Aplicações Web utilizam o browser como cliente universal
 - Páginas HTML
 - Protocolo HTTP

Protocolo HTTP

- Cliente inicia as ações com a requisição por uma página
 - GET e POST
- GET Request Method
 - Requisição de páginas estáticas ou dinâmicas
 - <http://www.dominio.com.br?discip=COM222>
 - Requisições com até 128 bytes
 - Requisição faz parte da URL

Protocolo HTTP

- POST Request Method
 - Requisição de páginas dinâmicas
 - Requisição faz parte do corpo da mensagem HTTP
 - Não há tamanho máximo
- Formulários em páginas Web podem usar tanto GET quanto POST
- Response HTTP com a página solicitada
 - Ou mensagem de erro

Aplicações e containers Web

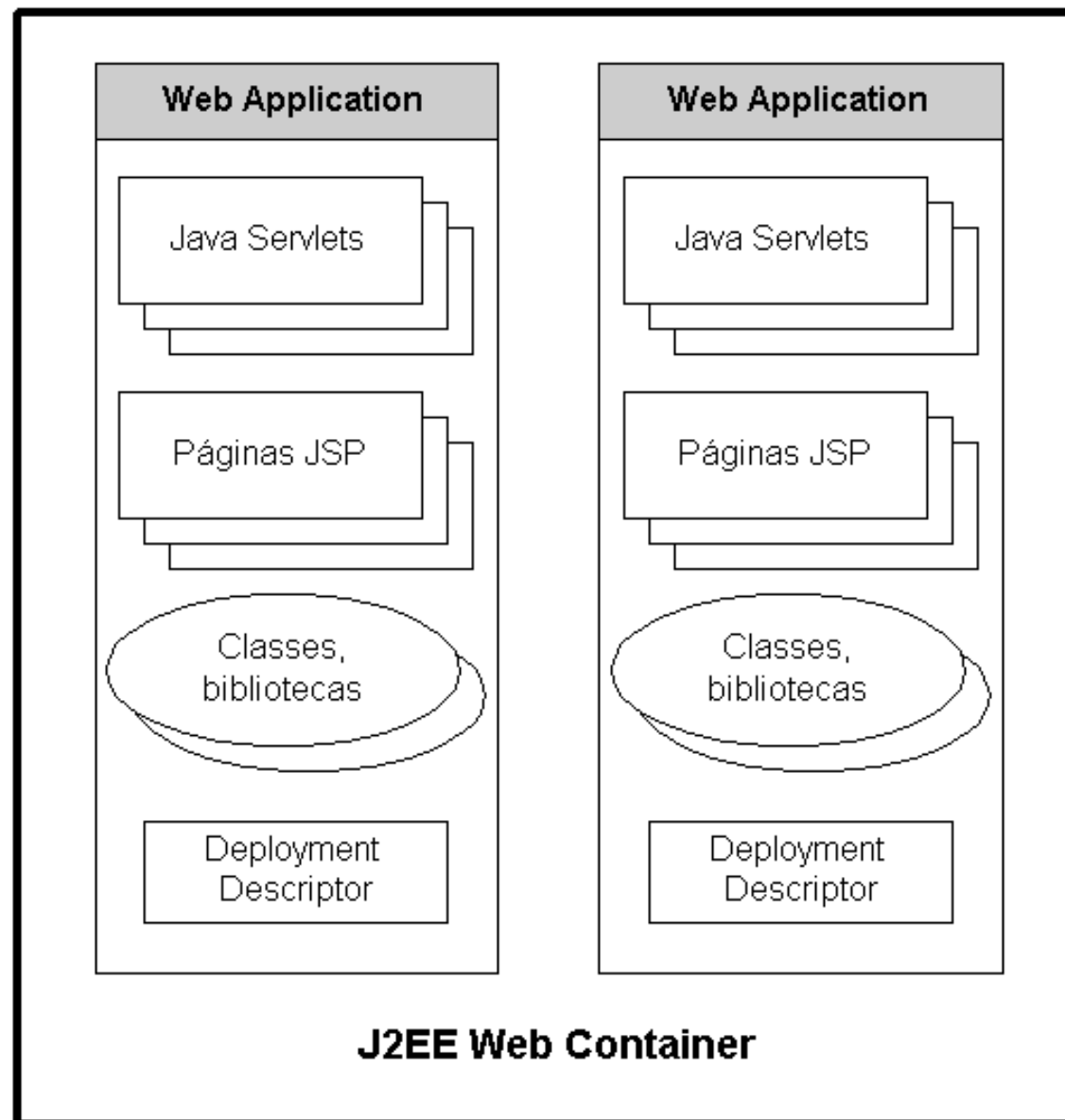
- Para rodar aplicações no servidor, é necessário
 - Um modelo de programação suportado por uma API
 - Suporte para execução de programas no servidor
 - Suporte de deployment
 - Para instalar aplicações no servidor
- JEE oferece
 - Componentes Web
 - Aplicações Web
 - Um container Web
 - Uma estrutura para empacotamento e deployment

Aplicações e containers Web

- Componentes Web
 - Java Servlets e JSP
- Aplicações Web
 - Coleção de servlets, páginas JSP, outras classes (beans) e recursos estáticos (páginas HTML, XML, etc)
- Container Web
 - Local para hospedar aplicações Web
 - JVM contendo uma API com suporte a servlets e JSP
 - É responsável pelo ciclo de vida dos componentes Web (inicialização, chamada, destruição, etc.)

Aplicações e containers Web

- Estrutura para empacotamento e deployment
 - Arquivos com formato especial para armazenar:
 - Web Component (WAR)
 - Web Application (EAR)
 - Deployment descriptor
 - Arquivo XML que permite customizar a aplicação no momento da sua instalação no container (deployment)

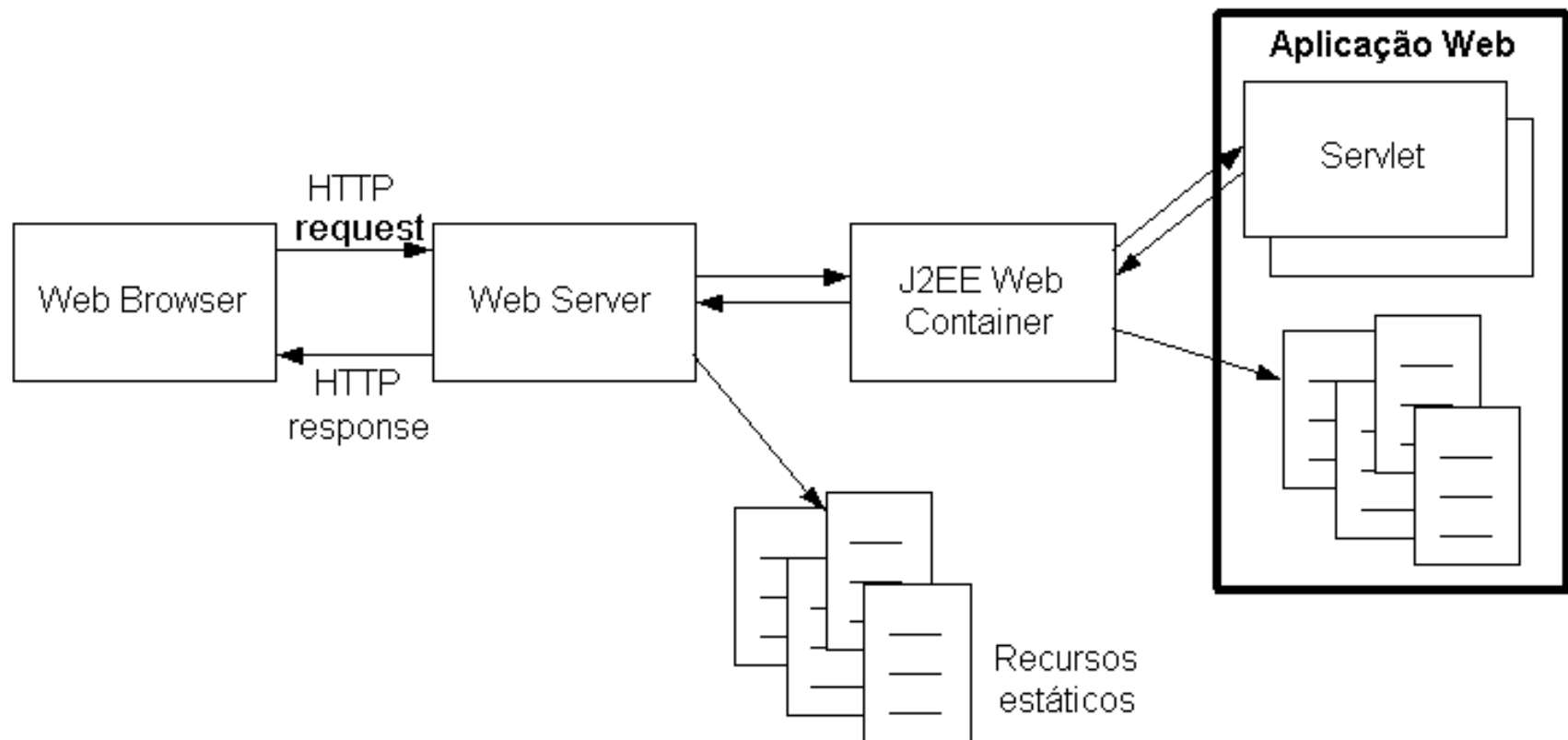


Servlets

- Permitem que a **lógica de aplicação** seja embutida no processo **request-reponse**
- Um servlet é um programa Java que roda no servidor e estende a funcionalidade do servidor Web
- A API de suporte a Servlets provê um **framework simples** para construir aplicações em servidores Web
- Existe um grande número de servidores Web com suporte a Servlets
 - Tomcat, WebLogic, Glassfish, WebSphere, JBoss, ...

Servlets

- Quando um servidor Web entende que uma URL deve ser atendida por um container, ele passa o controle para o container
 - Container decide qual aplicação Web deve ser executada
 - Se for um servlet, container controla sua execução
 - Utilizando a API, o servlet pode acessar a informação do Request, produzir e fornecer uma Response

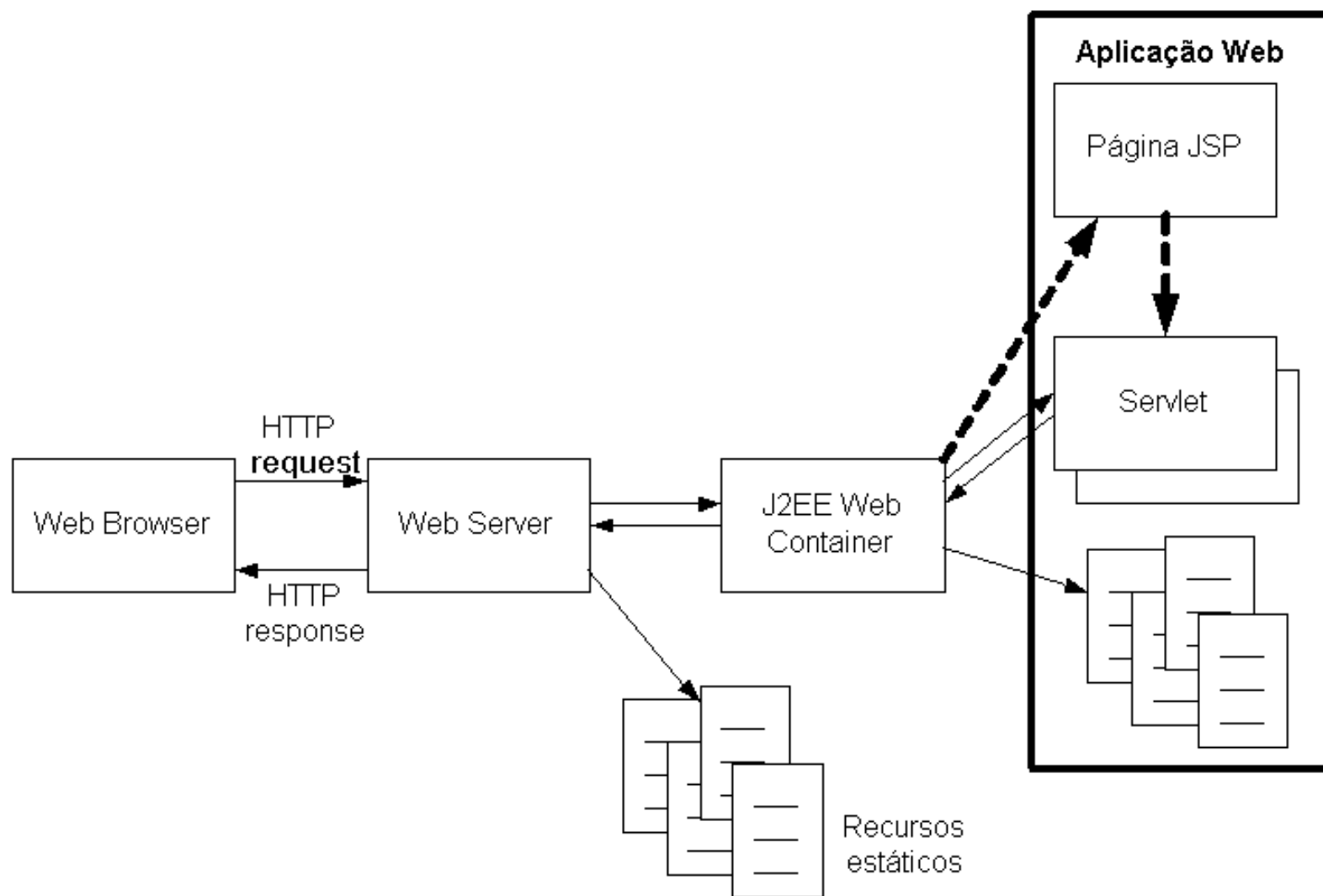


Java Server Pages (JSP)

- A geração de conteúdo dinâmico pode ser feita
 - Programaticamente (servlets)
 - Baseado no uso de templates (JSP)
- Templates
 - Páginas Web semi-prontas
 - Conteúdo estático pronto
 - Conteúdo dinâmico é gerado por scriplets
- Uso de templates permite que as páginas sejam criadas por Web Designers

Java Server Pages (JSP)

- JSP é uma extensão da tecnologia de Servlets
 - Página JSP mescla código em HTML (ou XML) com tags especiais contendo scriptlets
 - Páginas JSP são traduzidas para um servlet de maneira automática pelo servidor JEE
 - O servlet é compilado uma única vez
 - O servlet é executado para gerar o conteúdo dinâmico



Descritor de deployment

- Arquivo chamado web.xml que tem os seguintes propósitos:
 - Prover valores iniciais para parâmetros de aplicações
 - Definir servlets e JSPs
 - Cada servlet e página JSP usado pela aplicação deve ser descrito para permitir que seu deployment seja feito
 - Especificar mapeamentos
 - Definir como URLs que chegam do cliente serão mapeados para servlets e páginas JSP

Criação de aplicações JEE no NetBeans

- Com o lançamento das IDEs modernas (NetBeans e Eclipse JEE), passou a ser muito mais fácil criar aplicações Enterprise
 - IDE é capaz de rodar internamente o servidor
 - É possível compilar e executar aplicações dentro do ambiente da IDE
 - É possível gerar o arquivo war simplesmente exportando o projeto da IDE
- Utilizaremos o NetBeans na disciplina COM222

Rodando o HelloWorld no NetBeans

- Passos
 - Crie o projeto HW
 - Novo >> Java Web >> Aplicação Web
 - Crie o Servlet (HWServlet) dentro de HW
 - Novo >> Servlet
 - Coloque o código a seguir dentro do método `processRequest`

```
out.println("Olá, esta é a disciplina COM222!");  
out.close();
```
 - Salve o Servlet
 - Execute!

2ª Aplicação: Cadastro de email

- Faremos agora uma aplicação que inicia com uma página HTML convencional contendo um formulário
- Formulário submete dados que são capturados por um Servlet, o qual usa os dados capturados para produzir a página de resposta

index.html

```
<html>
<head> <title>Cadastro de Mail</title> </head>
<body>
<h1>Bem-vindo!</h1>
<form method="POST" action="/CadastroMail/cadastra">
<p>Seu nome <input type="text" name="nome" size="40"></p>
<p>Seu email <input type="text" name="email" size="40"></p>
<p><input type="submit" value="Submit">
<input type="reset" value="Reset">
</p> </form>
</body>
</html>
```

CadastroServlet.java

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class CadastroServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response)
        throws ServletException, IOException {

        // pega parâmetros do request
        String nome = request.getParameter("nome");
        String email = request.getParameter("email");

        GregorianCalendar calendário = new GregorianCalendar();
        String mensagem = calendário.get(Calendar.AM_PM) == Calendar.AM ?
            "Bom dia" : "Boa Tarde";
```

CadastroServlet.java

```
// acerta tipo MIME para a resposta
response.setContentType("text/html");

PrintWriter out = response.getWriter();
out.println("<HTML>");
out.println("<BODY>");
out.println("<P>" + mensagem + ", " + nome + "</P>");
out.println("<P>Obrigado por cadastrar seu email (" + email + ")
conosco.</P>");
out.println("</BODY>");
out.println("</HTML>");
out.close();
}
```


Exercício 1

- Elabore uma aplicação Web composta de 1 página HTML e um Servlet tal que:
 - Página HTML deve exibir um formulário com campos para entrada do nome e do ano de nascimento do usuário
 - O servlet, com base nos dados colhidos, deve montar uma página de resposta informando a idade do usuário

Exercício 2

- Elabore uma aplicação Web composta de 1 página HTML e um Servlet tal que:
 - Página HTML deve exibir um formulário com campos para entrada do nome e da data de nascimento do usuário
 - O servlet, com base nos dados colhidos, deve montar uma página de resposta informando a idade do usuário e seu signo astrológico

Exercício 3

- Elabore uma aplicação Web composta de 1 página HTML e um Servlet tal que:
 - Página HTML deve exibir um formulário com campos para entrada do nome de um usuário e a sigla de um curso da UNIFEI
 - Com base nos dados colhidos, o servlet deve montar uma página com a seguinte resposta:
 - “Fulano”, seu curso é integral/noturno e o tempo de integralização é X anos

Exercício 4

- Elabore uma aplicação Web composta de 1 página HTML e um Servlet tal que:
 - Página HTML deve exibir um formulário com campos para entrada do nome de um usuário e o nome de seu (suposto) time de coração
 - Com base nos dados colhidos, o servlet deve montar uma página com a seguinte resposta:
 - “Fulano”, se o campeonato terminasse hoje, seu time:
 - Seria campeão
 - Estaria na libertadores
 - Seria rebaixado
 - Estaria no meio da tabela