



Grupo **ibmec**

Java Servlets

Tópicos Especiais em Computação

Curso de Bacharelado em Ciências da Computação

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Prof. Cristiano Stüpp Nunes
cristiano.nunes@metrocamp.edu.br

Servlet

- É um programa java (utilizando pacotes específicos) que roda no lado servidor;
- Servlets recebem requisições via GET e POST, da camada de apresentação, e podem gerar respostas em HTML e/ou XML;
- Só rodam se hospedados em Servidores de Aplicações Web que suportam servlets.

Um servlet básico HelloWorld

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

@WebServlet("/HelloWorld")
public class HelloWorld
    extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter saida = response.getWriter();
        saida.println("Hello World!");
    }
}
```

Um servlet que envia resposta em HTML

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

@WebServlet("/HelloWorldInHTML")
public class HelloWorldInHTML extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter saida = response.getWriter();
        String html_str;
        html_str = "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 \" +
                    \"Transitional//EN\">\n" +
                    "<html>\n" +
                    "<head><title>Hello World em HTML</title></head>\n" +
                    "<body>\n" +
                    "<h1>Hello World (Usando HTML) !!!</h1>\n" +
                    "</body></html>";
        saida.println(html_str);
    }
}
```

Tratando campos de requisição GET com servlets

- Temos uma página HTML simples que envia um campo (parâmetro) para um servlet:

```
<html>
  <head>
    <title>Envia para um servlet que trata GET</title>
  </head>
  <body>
    <form action="tratarequisicoes" method="get">
      <input type="text" name="nome_usuario" />
      <input type="submit" value="Enviar" />
    </form>
  </body>
</html>
```

- `trataRequisicoes` é o (caminho e) nome do servlet
- `nome_usuario` é o nome do campo a ser enviado via método GET

Tratando campos de requisição GET com servlets

- E aqui se encontra o código do servlet `trataRequisicoes`:

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

@WebServlet("/tratarequisicoes")
public class TrataRequisicoes extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter saida = response.getWriter();
        String nome = request.getParameter("nome_usuario");
        saida.println("Bem vindo ao Tomcat " + nome + ". (Tratado via GET)!");
    }
}
```

- `doGet` é o método da classe `trataRequisicoes` que é executado sempre que esse servlet é invocado por uma requisição do tipo GET.
- `getParameter` é o método do objeto `request` que serve para pegar o valor dos campos enviados pela requisição (seja via GET ou POST)

Tratando campos de requisição POST com servlets

- Para tratar requisições POST basta incluir o método **doPost()**:

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

```
@WebServlet("/tratarequisicoes")
```

```
public class TrataRequisicoes extends HttpServlet {
```

```
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException { }
```

```
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
```

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

```
        String nome = request.getParameter("nome_usuario");
```

```
        saida.println("Bem vindo ao Tomcat " + nome + ". (Tratado via POST)!");
```

```
    }
}
```

- `doPost` é o método da classe `trataRequisicoes` que é executado sempre que esse servlet é invocado por uma requisição do tipo POST.
- `getParameter` é o método da classe `request` que serve para pegar o valor dos campos enviados pela requisição (seja via GET ou POST)

Quando requisições GET e POST devem executar o mesmo código

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

@WebServlet("/tratarequisicoes")
public class TrataRequisicoes extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter saida = response.getWriter();
        String nome = request.getParameter("nome_usuario");
        saida.println("Bem vindo ao Tomcat " + nome + ". (Tratado via GET)!");
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

- Se a requisição vem por GET, então executa o método `doGet`, se a requisição vem por POST então executa o método `doPost` e esse por sua vez invoca o método `doGet`.

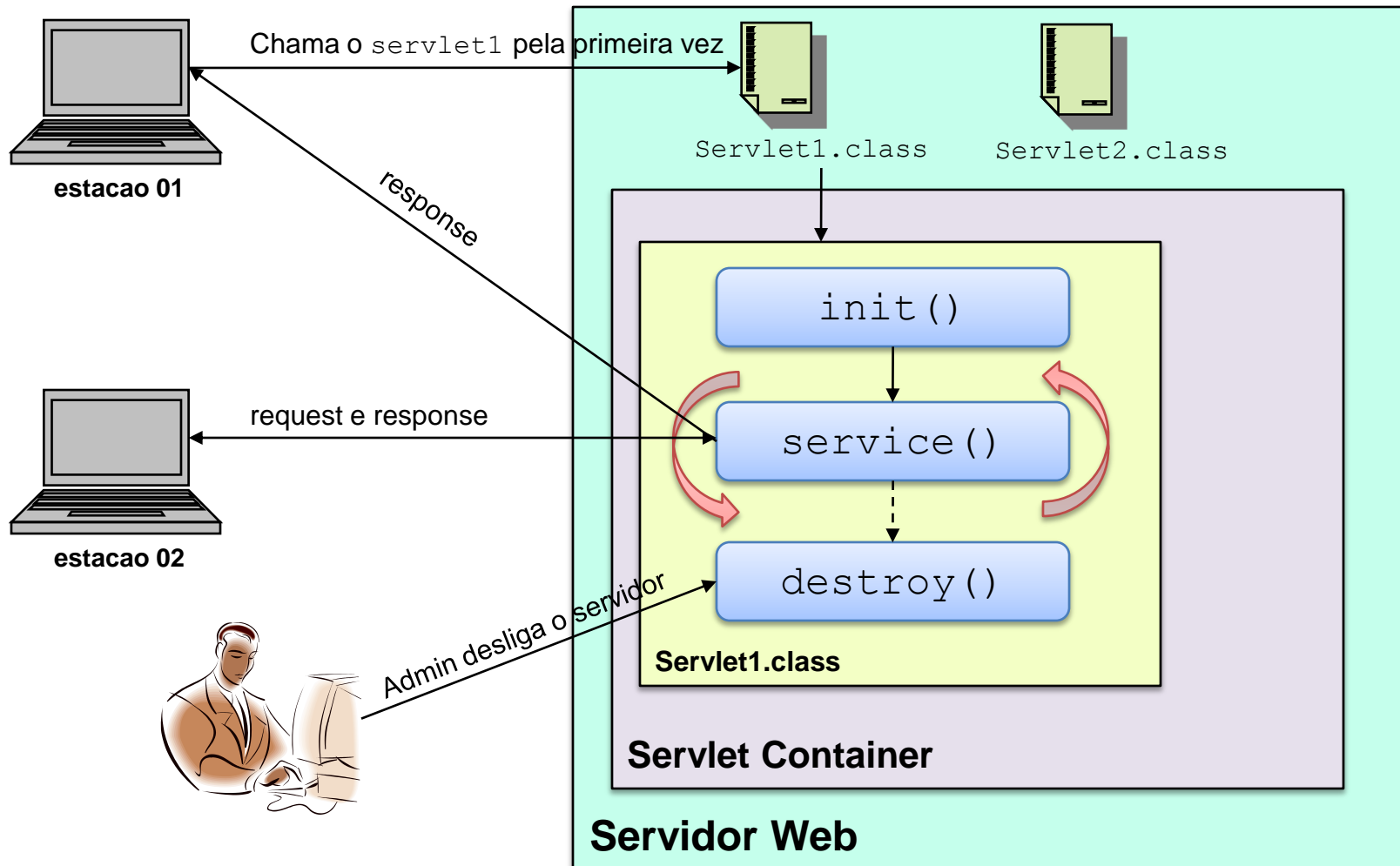
Ciclo de vida de um servlet

- Ao ser invocado pela primeira vez depois que o servidor web entrou no ar, o servlet é carregado no “contenedor de servlet” – *servlet container*;
- Ao ser carregado no container, o servlet executa automaticamente o método `init()` uma única vez e em seguida executa o método `service()`;
- A partir daí, as próximas requisições ao servlet, será executado apenas o método `service()`;
- O método `service()` é quem chama os métodos `doGet()` ou `doPost()` ou os outros `doXXX()`, dependendo do tipo de requisição enviada ao servidor web;

Ciclo de vida de um servlet

- Quando o servidor web é desligado, o servlet é descarregado do container e então executa o método `destroy()`;

Ciclo de vida de um servlet



Ciclo de vida de um servlet

```
import java.io.*;                //Servlet que ilustra o seu ciclo de vida
import javax.servlet.*;
import javax.servlet.http.*;

public class ServletLifecycleExample extends HttpServlet {

    private int count;

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
        getServletContext().log("metodo init() invocado");
        count=0;
    }

    protected void service(HttpServletRequest request,
                           HttpServletResponse response)
                           throws ServletException, IOException {
        getServletContext().log("metodo service() invocado");
        count++;
        response.getWriter().write("Servlet invocado: " +count+ " vezes");
    }

    public void destroy() {
        getServletContext().log("metodo destroy() invocado");
    }
}
```