ELE2 – LPG3

Eletiva 2
Linguagem de Programação 3 (JAVA)
Aplicações WEB - Servlets

JavaEE (Java Enterprise Edition)

- Conhecido no passado como J2EE
- Especificação para construção de servidores WEB de aplicações Java
- API:
  - Servlets, são utilizados para o desenvolvimento de aplicações Web com conteúdo dinâmico.
    - Contém uma API que abstrai e disponibiliza os recursos do servidor Web de maneira simplificada para o programador;
    - Classe Java que responde solicitações HTTP e que aceita codificação HTML em seu conteúdo.

JavaEE (Java Enterprise Edition)

- API:
  - JSP (Java Server Pages), uma especialização do servlet que permite que conteúdo dinâmico seja facilmente desenvolvido.
    - Basicamente é uma página HTML onde podem ser inseridos códigos JAVA.

JavaEE (Java Enterprise Edition)

- API:
  - EJBs (Enterprise Java Beans), utilizados no desenvolvimento de componentes de *software*.
    - Permitem que o programador se concentre nas necessidades do negócio do cliente, enquanto questões de infra-estrutura, segurança, disponibilidade e escalabilidade são responsabilidade do servidor de aplicações.

Servidores JavaEE

- No mercado, existem diversos servidores Java, muitos gratuitos e openSource
- Os mais conhecidos são:
  - TOMCAT, container JAVA do APACHE
  - JBoss
  - WebSphere, da IBM
  - GlassFish, da Sun
  - WebLogic, da Oracle
  - Jetty
  - ...entre outros

Servlet - Definição

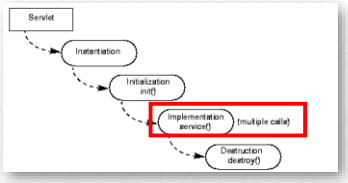
- É uma classe JAVA
  - Serve para estender a capacidade de um servidor de aplicações a uma classe comum
  - Oferece mecanismos para tratamento de requisição e resposta
- Comumente utilizada para programar uma resposta a uma requisição HTTP

API

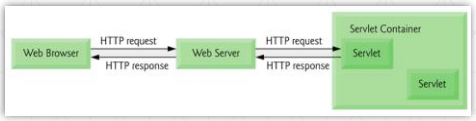
- Package javax.servlet
- Package javax.servlet.http
- Suporte para:
  - Gerenciamento de ciclo de vida
  - Acesso ao contexto
  - Utilidades
  - Classes de suporte específico ao HTTP

Ciclo de Vida

- Servlets são executados no servidor
- O Servidor é responsável por inicializar, invocar e destruir a instância do servlet
- Os métodos herdados pela classe são:
  - init()
  - service()
  - destroy()

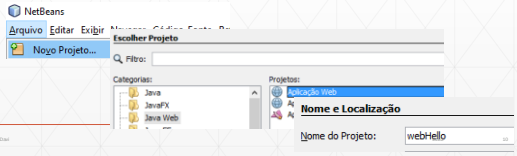


Ciclo de Requisição



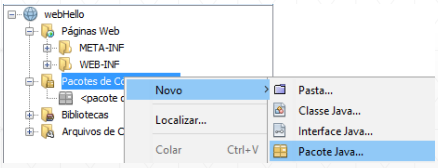
Criando um Servlet

- No NetBeans, crie um projeto
- Escolha a categoria , categoria “Java WEB” e o modelo de projeto “Aplicação Web”
- Chame o projeto de “webHello”
- Escolha qualquer servidor e finalize o wizard
- Inicialmente é criado um arquivo chamado “index.html”. Exclua-o!



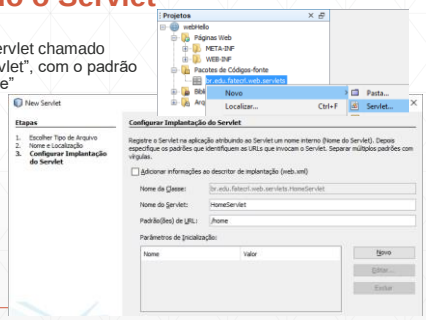
Criando o Pacote

- Em pacotes do código-fonte, crie um pacote chamado “br.edu.fatecrl.web.servlets”

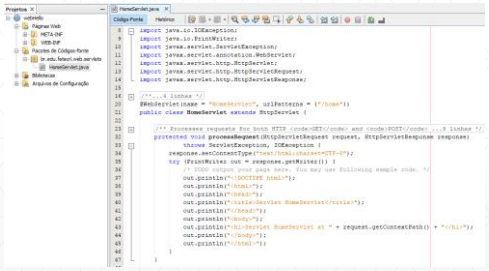


Criando o Servlet

- Crie um servlet chamado “HomeServlet”, com o padrão web “/home”



Analisando o modelo

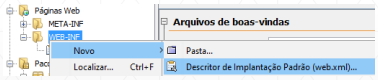


Escrevendo a saída HTML do método processRequest()

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Meu primeiro Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Olá, mundo do JavaEE!!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

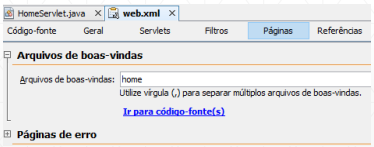
Configurando o servlet como página inicial

- Na pasta “WEB-INF”, o arquivo “Descritor de Implantação Padrão”, chamado “web.xml”. Ele é utilizado para diversas configurações no projeto.



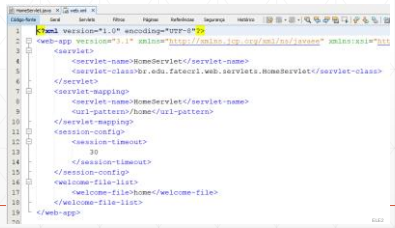
Configurando o servlet como página inicial

- Abra esse arquivo e escolha a aba “Páginas”.
- Nesta aba, você pode escolher a página inicial (de boas-vindas) e as páginas de erros.



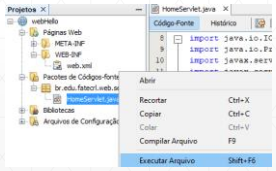
Gerenciando o web.xml

- Essa interface gráfica apresentada pelo NetBeans é uma interface para o arquivo web.xml, que pode ser alterado manualmente através da aba “Código-fonte”

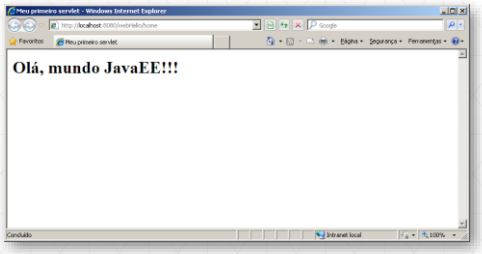


Executando o Servlet

- Pode ser feito de três formas:
  - Executando o projeto
  - Pelas teclas de atalho <F6> ou <Shift>+<F6>
  - Executando diretamente o arquivo no menu de contexto



Resultado



Depuração

- Um servlet pode ser depurado como qualquer classe JAVA

Contador de requisições

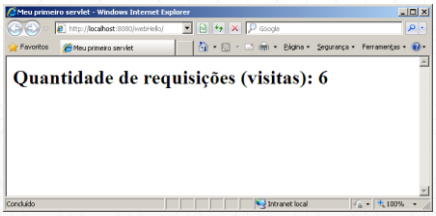
- Podemos utilizar um atributo de instância do Servlet para controlar a quantidade de visitas à página

```
public class HomeServlet extends HttpServlet {

    int contador = 0;

    /**...*/
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Meu primeiro servlet</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Quantidade de requisições (visitas): " + contador++ + "</h1>");
            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
        }
    }
```

Resultado



Parâmetros

- Uma requisição web pode conter parâmetros oriundos de HTML forms ou da própria URL
- Podemos capturar esses parâmetros através do objeto request, parâmetro do método processRequest()

```
    protected void processRequest(HttpServletRequest request,
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Meu primeiro servlet</title>");
            out.println("</head>");
            out.println("<body>");
            request.getParameter
            out.p  getParameter(String name)              String
            out.p  getParameterMap()                      Map
            out.p  getParameterNames()                    Enumeration
        } finally  getParameterValues(String name) String[]
```

Capturando parâmetros da query

- São parâmetros enviados pela URL da página acrescida de '?' Com os parâmetros separados por '&'

```
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Meu primeiro servlet</title>");
            out.println("</head>");
            out.println("<body>");
            String par1 = request.getParameter("parametro1");
            String par2 = request.getParameter("parametro2");
            out.println("<h1>Recebido os parâmetros: " + par1 + " e " + par2 + "</h1>");
            out.println("</body>");
```