

## Curso Profissional: Programação de Informática

PSD – 11.º ano: UFCD 10791 – Desenvolvimento de aplicações web em JAVA

Ficha de Trabalho 6

Ano letivo 22/23

**Java para a web** é uma tecnologia usada para desenvolvimento de aplicações web baseadas em Java, que podem ser executadas num servidor web para fornecer serviços e recursos para utilizadores finais, através de um navegador da web (browser).

Uma aplicação que é executada do lado do servidor e que cria uma página dinâmica é chamada de aplicação para a web.

### Principais tecnologias usadas em Java para desenvolvimento de aplicações web

Containers e API *	Definição
Servlets	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. São componentes Java executados no servidor que tem por objetivo gerar conteúdo (HTML e XML) dinâmico para a web.
Java Server Pages (JSP)	Uma especialização do servlet que permite que conteúdos dinâmicos sejam facilmente desenvolvidos e serem mais robustos.
JavaServer Faces (JSF)	Especificação Java para a construção de interfaces de utilizador baseadas em componentes (como calendários, tabelas e formulários) para aplicações web. Segue o padrão MVC (Model, View e Controller).
Java Persistence API (JPA)	API que padroniza o acesso a bases de dados através de mapeamento Objeto/Relacional dos Enterprise Java Beans. Consegue-se com esta API desenvolver aplicações que trabalham com bases de dados sem escrever nenhuma linha SQL.
Enterprise JavaBeans (EJB)	Utilizados no desenvolvimento de componentes de software. Permitem que o programador se concentre nas necessidades do negócio do cliente, enquanto que as questões de infraestrutura, segurança, disponibilidade e escalabilidade são da responsabilidade do servidor de aplicações.
Java Message Service (JMS)	API que padroniza o acesso a bases de dados através de mapeamento Objeto/Relacional dos Enterprise Java Beans.
Java API for RESTful Web Services **(JAX-RS).	JAX-RS é uma especificação Java para construir aplicações da web que usa a arquitetura REST (Representational State Transfer) e uma das principais tecnologias usadas para criar serviços web RESTful em Java. Com JAX-RS podem-se criar serviços web que são acessíveis através de URLs e que retornam dados em formato JSON ou XML.

\* uma **API - Application Programming Interface** é uma interface de comunicação que permite que diferentes sistemas troquem informações, um **container** é um ambiente de execução de aplicações que fornece uma camada de abstração entre o sistema operativo e as aplicações que estão a ser executadas (exemplos: GlassFish, TomCat, JBossEAP).

\*\* **WebServices** São serviços/métodos que as aplicações disponibilizam para serem consumidos/executados por outras aplicações.

## A utilização de Java para o desenvolvimento de aplicações Web inclui os seguintes benefícios:

Portabilidade, escalabilidade, segurança e a disponibilização de um vasto conjunto de recursos e de ferramentas.

**Java EE (Java Enterprise Edition)**, agora conhecido como Jakarta EE, é uma plataforma para desenvolvimento de aplicações corporativas em Java. Fornece-nos um conjunto de APIs (Application Programming Interfaces) e de especificações para desenvolver aplicações que se executam em servidores de aplicações Java, e é amplamente utilizada para criar aplicações corporativas em larga escala.

O Java EE inclui várias tecnologias e API (as mais relevantes mencionadas acima). A plataforma é baseada no Java SE (Java Standard Edition), que fornece a base para a maioria das aplicações Java, mas inclui recursos adicionais para aplicativos corporativos.

**Servlet** são classes Java, desenvolvidas de acordo com uma estrutura bem definida que quando instaladas e configuradas num servidor que implemente um **Servlet Container**, podem tratar requisições recebidas de clientes Web, como por exemplo os Browsers (Internet Explorer® e Mozilla Firefox®).

**Servlet** é uma tecnologia usada para processar solicitações e fornecer respostas numa aplicação web Java.

O nome “servlet” vem do inglês e dá uma ideia de servidor pequeno. Esta tecnologia, basicamente, funciona como um controlador que recebe requisições HTTP, processa-as e responde ao cliente (função proxy). Essa resposta pode ser um HTML, uma imagem, etc. Pode, então, ser utilizada para gerar dinamicamente uma página HTML.

Em Java, temos o **Tomcat**, um Web container (componente servlet) que engloba outros componentes (Servlet) para dar real funcionalidade a tudo.

### Exemplo de um Servlet:

```
package org.computer.br;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class EnderecoIpRemoto extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet EnderecoIpRemoto</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Endereco ip Remoto:" + request.getRemoteAddr() +
            "</h1>");
        out.println("</body>");
        out.println("</html>");
    }
    out.close();
}
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
    processRequest(request, response);
}
protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
    processRequest(request, response);
}
public String getServletInfo() {
    return "Short description";
}
}
```

### Código -html gerado pelo servlet Java

```
<html>
<head>
    <title>Servlet EnderecoIpRemoto</title>
</head>
<body>
    <h1>Endereco ip Remoto:127.0.0.1</h1>
</body>
</html>
```

## Java Server Pages (JSP)

**Java Server Pages (JSP)** é uma tecnologia usada para criar páginas da web dinâmicas que podem ser executadas num servidor web. O JSP possibilita a escrita de páginas da web usando a sintaxe Java padrão, permitindo a criação de aplicações dinâmicas e interativas.

**A tecnologia JSP é a extensão da tecnologia Servlet.** Podemos usar todos os recursos do Servlet em JSP. Além disso, podemos usar objetos implícitos, tags pré-definidas, linguagem de expressão e tags personalizadas em JSP, o que facilita o desenvolvimento de JSP.

Um JSP é um documento de texto que contém dois tipos de texto: dados estáticos e dados dinâmicos. Os dados estáticos podem ser expressos em qualquer formato baseado em texto (como HTML, XML, SVG e WML), e o conteúdo dinâmico pode ser expresso por elementos JSP.

### Exemplo de uma JSP:

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
</head>
<body>
    <h1>O seu endereço ip: <%= request.getRemoteAddr() %></h1>
</body>
</html>
```

## Diferença entre Servlet e JSP

SERVLET	JSP
Servlet é um código java.	JSP é um código baseado em html.
Escrever código para servlet é mais difícil do que JSP, pois é html em java.	JSP é fácil de codificar, pois é java em html.
O servlet desempenha um papel controlador na abordagem MVC.	JSP é a visão na abordagem MVC para mostrar a saída.
Servlet é mais rápido que JSP.	O JSP é mais lento do que o Servlet porque a primeira etapa no ciclo de vida do JSP é a tradução do JSP para o código Java e depois a compilação.
O servlet pode aceitar todos os pedidos de protocolo.	JSP só aceita requests http.
No Servlet, podemos substituir o método service().	Em JSP, não podemos substituir seu método service().
No Servlet, por padrão, o gerenciamento de sessão não está habilitado, o usuário deve habilitá-lo explicitamente.	Em JSP, o gerenciamento de sessão é ativado automaticamente.
Em Servlet, temos que implementar tudo como lógica de negócios e lógica de apresentação em apenas um arquivo de servlet.	Em JSP, a lógica de negócios é separada da lógica de apresentação usando javaBeans.
A modificação no Servlet é uma tarefa demorada porque inclui recarregar, recompilar e reiniciar o servidor.	A modificação JSP é rápida, basta clicar no botão Atualizar.

**JavaBeans** é um componente de software reutilizável que segue algumas convenções definidas pela plataforma Java. Essas convenções incluem a implementação de um construtor padrão sem argumentos, a disponibilização de métodos de acesso e modificação de propriedades e a implementação da interface `Serializable` (`java.io.Serializable`) para suportar a serialização de objetos.

JavaBeans são usados principalmente em interfaces gráficas de usuário (GUIs) e em tecnologias de persistência de dados, como JDBC e JPA.

Os JavaBeans simples são frequentemente utilizados nas JSPs, nas quais ativam o transporte de valores de formulários nas páginas da Web.

**Enterprise JavaBeans (EJB)** é uma especificação da plataforma Java que define um modelo de componente de software para sistemas distribuídos e empresariais. EJBs fornecem serviços de negócios, como transações, segurança e gestão de estado, e são executados num servidor de aplicação Java EE. Além disso, o EJB oferece serviços de persistência de dados, controle de concorrência, envio de mensagens, serviço de agendamento, chamadas a métodos remotos e a web services.

Ao utilizar EJB é possível construir aplicações distribuídas, totalmente seguras e com um eficaz tratamento de transações.

### Bibliografia

[https://pt.wikipedia.org/wiki/Java\\_Platform\\_Enterprise\\_Edition](https://pt.wikipedia.org/wiki/Java_Platform_Enterprise_Edition)  
<https://www.javatpoint.com/java-tutorial>  
<https://acervolima.com/diferenca-entre-servlet-e-jsp/>  
<https://www.devmedia.com.br/desenvolvendo-web-services-restful-utilizando-a-api-jax-rs-2-0-e-jersey/27141>  
<https://docs.oracle.com/javaee/7/tutorial/servlets.htm>  
<https://www.devmedia.com.br/introducao-a-servlets-em-java/25285>  
<https://www.javatpoint.com/jsp-tutorial>  
<https://pt.stackoverflow.com/questions/91620/o-que-%C3%A9-um-servlet-e-para-que-serve>  
<https://www.hostgator.com.br/blog/containers-o-que-sao-no-mundo-da-programacao/>  
<https://www.devmedia.com.br/web-services-com-net-framework/22013>  
[http://walderson.com/IBM/RUP7/LargeProjects/tech.j2ee/guidances/concepts/javabean\\_D488CF3B.html](http://walderson.com/IBM/RUP7/LargeProjects/tech.j2ee/guidances/concepts/javabean_D488CF3B.html)  
<https://docs.oracle.com/javaee/7/tutorial/ejb-intro.htm>  
<https://www.devmedia.com.br/enterprise-javabeans/26402>  
[JSF PrimeFaces Hibernate.pdf](#)