

Cargo:	Docente Especialista		
Nombre:	M.Sc Elvis Pachacama		
Asignatura:	Lenguajes de Programación 2		
Carrera:	Desarrollo de Software	Nivel:	Tercer nivel
Estudiante:	Gabriel Q. Pazmiño		

ACTIVIDAD PRÁCTICO EXPERIMENTAL EN EL ENTORNO ACADÉMICO

PROCESO DE FORMULARIO

Objetivos

"Establecerá los objetivos de aprendizaje esperados a alcanzar con la actividad." Ejemplo:

Antecedentes/Escenario

"Proporcionará información relevante o necesaria para comprender el contexto de la actividad."

Recursos necesarios

Planteamiento del problema

"Describirá el problema o desafío práctico a resolver."

Ejemplo:

deben ser ingresados por teclado y al elegir cada uno de las opciones debe implementarse un menú.

Pasos por realizar

"Detallará los pasos o procedimientos a seguir durante la actividad." Ejemplo:

Desarrollo

"Guiará el proceso de ejecución de la actividad, incluyendo observaciones, resultados obtenidos, y análisis de estos" Ejemplo:

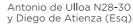
Título: Gestión de Datos de Usuario con Formulario jsp y Servlet en Java

Contexto: Eres parte del equipo de desarrollo de una aplicación web educativa. Uno de los módulos requiere recolectar y procesar información básica de los usuarios mediante un formulario HTML. Los datos ingresados deben ser procesados en el servidor usando Servlets en un entorno Jakarta EE (Tomcat o similar).

Objetivo del ejercicio: Desarrollar una aplicación web que permita al usuario llenar un formulario HTML con su información personal (nombre, correo electrónico, edad y











carrera), enviar estos datos al servidor mediante un Servlet, y mostrar un mensaje de confirmación con los datos recibidos.

Requisitos:

- Crear un archivo jsp (index.jsp) que incluya: Un formulario con los campos: Como se muestra en el código en la parte inferior
- Botón de envío.
- Método POST.

•

- Crear un Servlet llamado FormularioServlet.java que: Reciba los datos enviados desde el formulario.
- Verifique que los datos no estén vacíos.
- Imprima los datos recibidos en pantalla mediante una respuesta HTML.

•

1. Configurar correctamente el archivo web.xml o usar anotaciones @WebServlet para mapear el servlet.

Escenario: El usuario accede al formulario desde su navegador, llena los datos requeridos y hace clic en "Enviar". La aplicación procesa los datos con el Servlet y devuelve una página con un mensaje como:

Tecnologías sugeridas:

- JSP
- Java Servlet (Jakarta EE)
- Tomcat o servidor compatible
- IDE: IntelliJ IDEA o Eclipse

INDEX.JSP

<%--

Created by IntelliJ IDEA.

Desarrollador: Gabriel Quijije Pazmiño

Fecha: 10/05/2025

Descripción: Página JSP que muestra un formulario HTML con estilos CSS.

Los datos del formulario se envían a un Servlet llamado FormularioServlet

--%>

<%@ page contentType="text/html;charset=UTF-8" language="java" %>

<%-- Esta directiva indica que la página usará UTF-8 como codificación de caracteres y lenguaje Java --%>

<!DOCTYPE html>

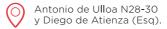
<html>

<head>

<meta charset="UTF-8">





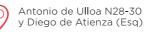






```
<%-- Define la codificación de caracteres del documento HTML como UTF-8 --%>
<title>Formulario de Registro</title>
<%-- Título que aparece en la pestaña del navegador --%>
<style>
  /* Estilos CSS para mejorar el diseño visual del formulario */
  body {
    background-color: #f0f0f5; /* Color de fondo claro para toda la página */
    font-family: Arial, sans-serif; /* Fuente utilizada en la página */
    padding: 40px; /* Espaciado interior alrededor del contenido */
  }
  h1 {
    color: #333366; /* Color del título principal */
  }
  form {
    background: #ffffff; /* Fondo blanco para el formulario */
    padding: 40px; /* Espaciado interior dentro del formulario */
    border-radius: 10px; /* Bordes redondeados */
    box-shadow: 0 0 10px #ccccc; /* Sombra alrededor del formulario */
    width: 420px; /* Ancho fijo del formulario */
    margin: auto; /* Centra el formulario horizontalmente */
  }
  label {
    display: block; /* Hace que cada etiqueta aparezca en su propia línea */
    margin-bottom: 8px; /* Espacio debajo de cada etiqueta */
    margin-top: 15px; /* Espacio encima de cada etiqueta */
  }
  input, select {
    width: 100%; /* Campos ocupan el ancho del contenedor */
    padding: 10px; /* Espacio interno en los campos */
    margin-top: 5px; /* Separación entre etiqueta y campo */
  }
  button {
    margin-top: 20px; /* Espacio superior antes del botón */
    background-color: #333366; /* Color de fondo del botón */
    color: white; /* Texto en color blanco */
    padding: 10px; /* Espaciado interno del botón */
    border: none; /* Sin borde alrededor del botón */
    width: 100%; /* Ocupa el ancho del formulario */
    cursor: pointer; /* Cambia el cursor al pasar sobre el botón */
    border-radius: 5px; /* Bordes redondeados del botón */
```





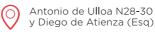




```
}
    button:hover {
      background-color: #444477; /* Color más claro al pasar el mouse por encima */
  </style>
</head>
<body>
<h1>Formulario de Registro</h1>
<%-- Título principal visible para el usuario --%>
<form action="FormularioServlet" method="post">
  <%--
    Formulario que se enviará al Servlet llamado "FormularioServlet" usando el
metodo POST.
    POST se usa para enviar los datos de forma segura sin que aparezcan en la URL.
  --%>
  <label for="nombre">Nombre:</label>
  <input type="text" name="nombre" id="nombre" required>
  <%-- Campo de texto para el nombre. El atributo "required" obliga al usuario a
llenarlo --%>
  <label for="correo">Correo Electrónico:</label>
  <input type="email" name="correo" id="correo" required>
  <%-- Campo de tipo email que valida el formato de correo automáticamente --%>
  <label for="edad">Edad:</label>
  <input type="number" name="edad" id="edad" required>
  <%-- Campo numérico para que el usuario ingrese su edad --%>
  <label for="carrera">Carrera:</label>
  <select name="carrera" id="carrera" required>
    <%-- Menú desplegable para seleccionar una carrera. También es obligatorio --%>
    <option value="">Seleccione una opción</option>
    <%-- Opción por defecto para obligar a seleccionar una carrera válida --%>
    <option value="Ingeniería en Sistemas">Ingeniería en Sistemas
    <option value="Medicina">Medicina</option>
    <option value="Arquitectura">Arquitectura
    <option value="Derecho">Derecho</option>
  </select>
  <button type="submit">Enviar</button>
  <%-- Botón que envía los datos del formulario al servlet --%>
</form>
```











```
</body>
</html>
```

FORMULARIOSERVLET.JAVA

```
// Paguete donde se encuentra esta clase Java
package com.Formulario;
// Importación de clases necesarias para trabajar con Servlets
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;
// Anotación que indica la URL a la que responderá este servlet
@WebServlet("/FormularioServlet")
public class FormularioServlet extends HttpServlet {
  // Metodo que maneja las peticiones HTTP POST (cuando se envía el formulario)
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    // Obtener los datos enviados desde el formulario usando los nombres de los
campos
    String nombre = request.getParameter("nombre");
    String correo = request.getParameter("correo");
    String edad = request.getParameter("edad");
    String carrera = request.getParameter("carrera");
    // Verificar si alguno de los campos está vacío o no fue enviado
    if (nombre == null | | nombre.isEmpty() | |
        correo == null || correo.isEmpty() ||
        edad == null || edad.isEmpty() ||
        carrera == null || carrera.isEmpty()) {
      // Si algún campo está vacío, se responde con error 400 (Bad Request) y un
mensaje
      response.sendError(HttpServletResponse.SC_BAD_REQUEST, "Todos los campos
son obligatorios.");
      return; // Finaliza la ejecución del metodo
    }
    // Establecer el tipo de contenido de la respuesta como HTML con codificación
```







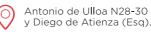


UTF-8

```
response.setContentType("text/html;charset=UTF-8");
    // Obtener el objeto PrintWriter para escribir la respuesta HTML al cliente
    PrintWriter out = response.getWriter();
    // Generar la respuesta HTML mejorada con estilos visuales embebidos
    out.println("<!DOCTYPE html>");
    out.println("<html>");
    out.println("<head>");
    out.println("<meta charset='UTF-8'>");
    out.println("<title>Confirmación</title>");
    // Estilos CSS embebidos para mejorar la presentación
    out.println("<style>");
    out.println("body { background-color: #f0f0f5; font-family: Arial, sans-serif;
padding: 40px; }");
    out.println("h2 { color: #333366; }");
    out.println(".container { background: #ffffff; padding: 25px; border-radius: 10px; "
        + "box-shadow: 0 0 10px #cccccc; width: 500px; margin: auto; }");
    out.println("p { font-size: 1.1em; margin-bottom: 10px; }");
    out.println("a { display: inline-block; margin-top: 20px; text-decoration: none; "
        + "background-color: #333366; color: white; padding: 10px 15px; border-
radius: 5px; }");
    out.println("a:hover { background-color: #444477; }");
    out.println("</style>");
    out.println("</head>");
    out.println("<body>");
    // Contenedor visual para centrar y organizar el contenido
    out.println("<div class='container'>");
    out.println("<h2>Se han registrado los datos correctamente:</h2>");
    // Mostrar los datos ingresados
    out.println("<strong>Nombre:</strong> " + nombre + "");
    out.println("<strong>Correo:</strong> " + correo + "");
    out.println("<strong>Edad:</strong> " + edad + "");
    out.println("<strong>Carrera:</strong> " + carrera + "");
    // Enlace para regresar al formulario
    out.println("<a href='index.jsp'>Volver al formulario</a>");
    out.println("</div>");
    out.println("</body>");
    out.println("</html>");
```



admisiones@ita.edu.ec

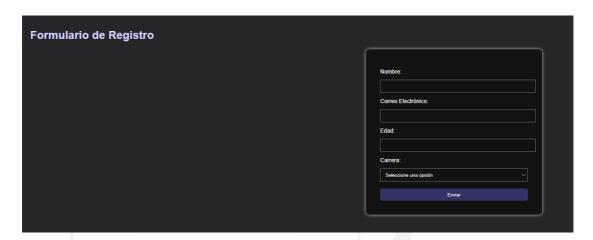


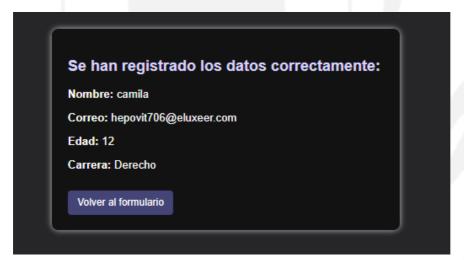




} }

CAPTURAS





GITHUB

https://github.com/Vanship77/DEBER-FORMULARIO-10-05-25.git



