

Nombre: Freddy Edu Rivadeneira Leon

Curso: 4to de Ti

Materia: POO

## **Informe: Aplicación Web JSP para Registro de Usuarios**

### **1. Introducción**

En la actualidad, las aplicaciones web son una de las herramientas más utilizadas para la gestión de información, ya que permiten el acceso a datos desde cualquier dispositivo con conexión a internet. Java ofrece un conjunto de tecnologías que facilitan el desarrollo de este tipo de aplicaciones de manera estructurada y segura, entre ellas JSP (JavaServer Pages) y Servlets.

La presente práctica tiene como objetivo desarrollar una aplicación web básica utilizando JSP y Servlets, aplicando el método POST para el envío de datos y simulando una base de datos mediante estructuras de datos en memoria. A través de esta actividad, se refuerzan conceptos fundamentales como la arquitectura MVC, la separación de responsabilidades y la integración de Java con aplicaciones web.

### **2. Aplicación Web Java**

Una aplicación web Java es un sistema que se ejecuta en un servidor y responde a las solicitudes realizadas por los usuarios a través de un navegador web. Estas aplicaciones se caracterizan por separar la lógica del negocio, la presentación y el control del flujo de información, lo que permite un mejor mantenimiento y escalabilidad del sistema.

En esta práctica, la aplicación se ejecuta sobre el servidor Apache Tomcat, el cual es un contenedor de Servlets que permite procesar peticiones HTTP y generar respuestas dinámicas mediante JSP.

### **3. JSP y Servlet**

#### **3.1 JSP (JavaServer Pages)**

JSP es una tecnología que permite crear páginas web dinámicas combinando código HTML con fragmentos de código Java. Su principal función es la presentación de la información al usuario final. En esta práctica, los archivos JSP se encargan de mostrar el formulario de registro y la lista de estudiantes registrados.

El uso de JSP facilita la creación de interfaces dinámicas, ya que permite mostrar datos enviados desde el Servlet sin necesidad de escribir lógica compleja en la vista.

### 3.2 Servlet

Un Servlet es una clase Java que se ejecuta en el servidor y actúa como intermediario entre el navegador y la lógica de la aplicación. Su función principal es recibir las solicitudes del usuario, procesar los datos y decidir qué respuesta se debe enviar.

En esta práctica, el Servlet recibe los datos del formulario mediante el método POST, crea los objetos correspondientes y gestiona el acceso a la simulación de la base de datos. Además, se encarga de redirigir o reenviar la información a las páginas JSP según la acción seleccionada por el usuario.

### 4. Método POST

El método POST es un método HTTP utilizado para enviar datos desde el cliente al servidor de forma segura. A diferencia del método GET, los datos enviados por POST no son visibles en la barra de direcciones del navegador.

En la aplicación desarrollada, el método POST se utiliza para:

- Registrar los datos del estudiante.
- Solicitar la visualización de los registros almacenados.

Esto garantiza un mejor manejo de la información y cumple con las buenas prácticas en el desarrollo de aplicaciones web.

### 5. Arquitectura MVC

La aplicación sigue el patrón de diseño MVC (Modelo – Vista – Controlador), el cual permite organizar el código de forma clara y estructurada.

- **Modelo:** Representa los datos del sistema. En este caso, la clase Estudiante define la estructura de los datos del estudiante.
- **Vista:** Corresponde a las páginas JSP, encargadas de mostrar el formulario y la información registrada.
- **Controlador:** Es el Servlet, que recibe las solicitudes, procesa los datos y coordina la interacción entre el modelo y la vista.

El uso de MVC facilita el mantenimiento del sistema y permite que cada componente tenga una responsabilidad bien definida.

### 6. Simulación de Base de Datos en Memoria

Para esta práctica no se utiliza una base de datos real. En su lugar, se implementa una simulación de base de datos utilizando una estructura ArrayList, la cual almacena los objetos Estudiante en memoria.

Esta simulación permite:

- Registrar múltiples estudiantes.
- Mantener los datos disponibles mientras el servidor esté en ejecución.
- Comprender el funcionamiento básico del almacenamiento de información sin depender de un gestor de base de datos.

Aunque los datos se pierden al reiniciar el servidor, esta técnica es útil para prácticas académicas y pruebas iniciales.

## **7. Funcionamiento de la Aplicación**

El funcionamiento de la aplicación se puede resumir en los siguientes pasos:

1. El usuario accede a la página principal y visualiza el formulario de registro.
2. Ingresa los datos solicitados y presiona el botón correspondiente.
3. El formulario envía la información al Servlet utilizando el método POST.
4. El Servlet procesa los datos y los almacena en la lista que simula la base de datos.
5. Cuando el usuario solicita ver los registros, el Servlet envía la lista de estudiantes a la vista.
6. La página JSP muestra dinámicamente la información registrada.

Este flujo permite comprender claramente la interacción entre JSP, Servlets y estructuras de datos en Java.

## **8. Conclusiones**

La práctica permitió comprender de manera clara cómo se integran JSP y Servlets en una aplicación web Java. Además, se reforzó el uso del método POST, la organización del proyecto mediante el patrón MVC y la simulación de una base de datos en memoria utilizando estructuras Java.

Este tipo de ejercicios resulta fundamental para sentar las bases del desarrollo web con Java, ya que prepara al estudiante para trabajar con aplicaciones más complejas que involucren bases de datos reales, validaciones y mayor interacción con el usuario.

## Registro a Curso Virtual

|  |   |
|--|---|
| Nombre:  | <input type="text" value="Eida"/>                 |
| Correo:  | <input type="text" value="eidalz1516@gmail.com"/> |
| Curso:   | <input type="text" value="Java Básico"/>          |
| <input type="button" value="Guardar"/>           |   |
| <input type="button" value="Mostrar registros"/> |   |

## Estudiantes Registrados

**Nombre:** Freddy  
**Correo:** rivadeneirafreddyedu@gmail.com  
**Curso:** Programación Web

**Nombre:** Eida  
**Correo:** eidalz1516@gmail.com  
**Curso:** Bases de Datos

[Volver](#)