



Pontificia Universidad
Católica del Ecuador

SEDE
ESMERALDAS

ASIGNATURA:

Programación orientada a objetos

TÍTULO:

Práctica: Aplicación Web con JSP y Servlets

AUTORES:

Tandazo Rojas Johnny

CARRERA: TECNOLOGÍAS DE LA INFORMACIÓN

DOCENTE: Esteban Fabricio Gonzabay Jimenez

Esmeraldas, 2026

1. Introducción

En la presente investigación se desarrolló una aplicación web modular basada en la tecnología Java EE. De los varios objetivos marcados, el principal fue poder implementar el patrón de arquitectura Modelo-Vista-Controlador (MVC) para administrar el registro de estudiantes a un curso virtual, separando la lógica de la aplicación de la propia interfaz de usuario.

2. Definición de Conceptos Clave

- **JSP (JavaServer Pages):** Es la tecnología utilizada para la capa de la **Vista**. Permite mezclar código HTML con Java para generar contenido dinámico. En este proyecto, se usó para crear las interfaces visuales (index.jsp para el registro y listar.jsp para el reporte), permitiendo mostrar datos procesados por el servidor directamente en el navegador.
- **Servlet:** Es una clase Java que actúa como el **Controlador** de la aplicación. Su función es recibir las peticiones HTTP (requests) del usuario, procesar la información enviada desde el formulario y decidir a qué página redirigir (response). El archivo RegistroServlet.java gestiona la comunicación entre la vista y los datos.
- **Método POST:** Es el método HTTP utilizado para enviar información desde el cliente al servidor. A diferencia del método GET, POST envía los datos "ocultos" en el cuerpo de la petición y no en la URL. Se utilizó este método para garantizar un envío más limpio y adecuado para operaciones de guardado de datos.
- **Base de Datos Simulada:** Con la intención de satisfacer la necesidad de espacio de almacenamiento, sin contar con un motor para base de datos externo (por ejemplo, MySQL), se desarrolló una clase `BaseDatosSimulada` que se basa en una estructura de datos `ArrayList` estática, de manera que los datos queden guardados en la RAM del servidor mientras la aplicación se esté ejecutando.

3. Conclusión

La práctica permitió comprender el flujo de información en una aplicación web Java. Se logró implementar correctamente la comunicación entre el formulario (JSP) y el controlador (Servlet), así como la persistencia temporal de datos en memoria, cumpliendo con los requisitos de modularidad y funcionalidad solicitados.