

**Tema**

FORMULARIO WEB SIN MVC UTILIZANDO APACHE NETBEANS

**Tutor**

Ing. Eduardo Mauricio Campaña Ortega

MIS.MDU.CCNA.CCIA.

PhD. (c) Ingeniería de Software

PhD. (c) Seguridad Información

**Fecha**

07/11/2022

FORMULARIO WEB SIN MVC EN APACHE NETBEANS

[MARCO TEÓRICO 2](#_Toc118711100)

[JAVA WEB 2](#_Toc118711101)

[JAVA SERVER PAGES (JSP) 3](#_Toc118711102)

[SERVLETS 3](#_Toc118711103)

[JAVABEANS 3](#_Toc118711104)

[CREACIÓN DEL PROYECTO 4](#_Toc118711105)

[OBJETIVO 4](#_Toc118711106)

[CREAR UN SITIO WEB LLAMADO FORMULARIOWEBSINMVC 4](#_Toc118711107)

[\_APELLIDO\_ NOMBRE 4](#_Toc118711108)

[CODIFICACIÓN DE LOS ARCHIVOS .JSP 6](#_Toc118711109)

[**CODIFICACIÓN DEL ARCHIVO INDEX.JSP** 7](#_Toc118711110)

[**CODIFICACIÓN DEL ARCHIVO SALIDA.JSP** 11](#_Toc118711111)

[EJECUCIÓN DEL PROYECTO 15](#_Toc118711112)

[CONCLUSIONES 16](#_Toc118711113)

[RECOMENDACIONES 17](#_Toc118711114)

[BIBLIOGRAFÍA 17](#_Toc118711115)

**INDICE DE IMÁGENES**

[Figura 1. Logotipo identificativo de Java Enterprise Edition 3](#_Toc118713330)

[Figura 2. Logotipo identificativo de Java Server Pages 4](#_Toc118713331)

[Figura 3. Creación de carpetas para él proyecto. 5](#_Toc118713332)

[Figura 4. Logotipo identificativo de Apache NetBeans 15. 5](#_Toc118713333)

[Figura 5. Creación de un nuevo proyecto. 6](#_Toc118713334)

[Figura 6. Selección del tipo de aplicación a desarrollar 6](#_Toc118713335)

[Figura 7. Selección del nombre y directorio donde guardar el proyecto 6](#_Toc118713336)

[Figura 8. Selección del servidor y versión de Java EE. 7](#_Toc118713337)

[Figura 9. Selección del Frameworks y finalización de la creación del proyecto. 7](#_Toc118713338)

[Figura 10. Estructura del proyecto. 7](#_Toc118713339)

[Figura 11. Eliminación del archivo index.html. 8](#_Toc118713340)

[Figura 12. Selección de un archivo del tipo JSP 8](#_Toc118713341)

[Figura 13. Asignación del nombre al archivo a crear. 9](#_Toc118713342)

[Figura 14. Codificación del archivo index.jsp (Formulario) 9](#_Toc118713343)

[Figura 15. Selección de un archivo del tipo JSP 12](#_Toc118713344)

[Figura 16. Asignación del nombre al archivo a crear. 12](#_Toc118713345)

[Figura 17. Codificación del archivo salida.jsp 13](#_Toc118713346)

[Figura 18. Ejecución del proyecto 16](#_Toc118713347)

[Figura 19. Formulario del proyecto. 16](#_Toc118713348)

[Figura 20. Ingreso de Datos en el formulario index.jsp 17](#_Toc118713349)

[Figura 21. Despliegue del formulario salida.jsp 17](#_Toc118713350)

**ÍNDICE DE TABLAS**

[Tabla 1. Codificación del archivo index.jsp 10](#_Toc118713547)

[Tabla 2. Codificación del archivo salida.jsp 14](#_Toc118713548)

# MARCO TEÓRICO

## JAVA WEB



Figura 1. Logotipo identificativo de Java Enterprise Edition

Java Enterprise Edition es una plataforma que se encarga de definir las APIs y los requerimientos necesarios para poder ejecutar aplicaciones Java servidoras las cuales pueden ser: Cliente-Servidor, Multiusuarios, Transaccionalidad, Escalabilidad entre otros, donde tales características no son importantes o indispensables en aplicaciones de escritorio.

La plataforma Java Enterprise Edition se apoya en Java SE (Java Standard Edition) que a diferencia de Java EE se encarga de definir las APIs y los requerimientos necesarios para poder ejecutar aplicaciones de escritorio ya sea en ordenadores personales o portátiles como se requiera.

Java EE tiene un modelo la cual define una arquitectura para poder implementar servicios como aplicaciones multicapas donde se asegura la escalabilidad, accesibilidad, y la facilidad de gestión necesaria para el ámbito empresarial.

Su modelo divide la implementación en dos partes:

1. La lógica de presentación y de negocio a implementar por el desarrollador.
2. Los servicios estándar que ofrece la plataforma Java EE.

Además, Java EE utiliza un modelo de programación distribuido en distintas capas, donde la lógica de la aplicación se divide en distintos componentes los cuales dependen de su funcionalidad donde son desplegados en las distintas capas dependiendo a que capa pertenecen.

Componentes de Java Enterprise Edition

* Componentes clientes: son aplicaciones Java SE (AWT/Swing, Applets) o de un navegador web (Firefox, Chrome, IExplore) y se despliegan en la capa del cliente.
* Componentes web: son Java Servlets, JavaServer Pages (JSP) o JavaServer Faces (JSF) y se despliegan en la capa web.
* Componentes de negocio: Enterprise JavaBeans (EJB) y se despliegan en la capa de negocio.

## JAVA SERVER PAGES (JSP)

Una página JSP (JavaServer Page) es una página HTML a la que se le incrusta código de lenguaje de programación JAVA. El código JAVA se incrusta entre los siguientes indicadores <% y %>.

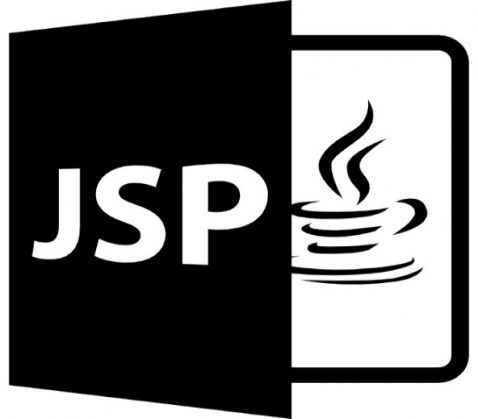


Figura 2. Logotipo identificativo de Java Server Pages

## SERVLETS

Un servlet es una clase Java la cual hereda de la clase (HttpServlet) corre en el servidor. Su palabra se deriva de applet, donde anteriormente un applet era un pequeño programa escrito en lenguaje de programación JAVA y el cuál corría en el navegador del cliente.

La función de un servlet es dar servicio a las peticiones de un navegador web donde recibe una petición, la procesa, y devuelve una respuesta al navegador, al ser escrito en lenguaje JAVA es portable como toda aplicación JAVA lo cual quiere decir que puede funcionar sin necesidad de hacer cambios en diferentes servidores.

Además, en un servlet no solo se puede tener código JAVA también se puede incrustar código HTML.

## JAVABEANS

Un JavaBean también conocido como Bean es un componente el cual se encuentra desarrollado con patrones de diseño bien definidos lo que hace que sea reutilizable y manipulado de manera visual por medio de una herramienta de programación en el lenguaje Java.

Características:

* Introspección: permite analizar a la herramienta de programación o IDE como trabaja en Bean.
* Customización: es la capacidad que tiene el programador de poder manipular y alterar la apariencia y la conducta del Bean.
* Events: se encarga de informar al IDE de aquellos sucesos que se pueden generar en respuesta en base a las acciones del usuario o del sistema. Además, de los sucesos que se pueden mejorar.
* Properties: permite cambiar los valores de las propiedades del Bean con el objetivo de personalizarlo.
* Persistence: tiene la capacidad de guardar el estado de los Beans los cuales han sido personalizados por el programador.

Reglas de los JavaBeans:

* Un Bean debe de tener un constructor por defecto (Constructor sin argumentos).
* Un Bean siempre debe de tener persistencia lo cual quiere decir que se debe de implementar una interfaz Serializable.

# CREACIÓN DEL PROYECTO

## OBJETIVO

Realizar un formulario WEB sin utilizar la arquitectura MVC (Modelo Vista Controlador) donde se apliquen los conceptos revisados previamente, utilizando como entorno de desarrollo Apache Netbeans 15.

## CREAR UN SITIO WEB LLAMADO FORMULARIOWEBSINMVC

## \_APELLIDO\_ NOMBRE

Para poder comenzar con el desarrollo del proyecto se crean las carpetas donde se almacenará el mismo el cual tiene la siguiente estructura en el **Disco Local C** se crea una carpeta llamada **FORMULARIOS APELLIDO** donde dentro de la carpeta se crea una nueva carpeta con el siguiente nombre **01\_WEB\_SIN\_MVC\_JAVA\_APELLIDO**, por último, se crea dos carpetas donde la primera se llamada **APLICATIVO** y **DOCUMENTACIÓN** respectivamente como se muestra en la siguiente imagen.

Texto

Descripción generada automáticamente

Figura 3. Creación de carpetas para él proyecto.

Una vez creados los directorios se procede a abrir el entorno de desarrollo el cual es Apache Netbeans 15.



Figura 4. Logotipo identificativo de Apache NetBeans 15.

Una vez abierto el entorno de desarrollo (Apache Netbeans 15) se deberá hacer clic en el apartado File donde seguido de esto se selecciona New Project como se muestra en la siguiente figura.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 5. Creación de un nuevo proyecto.

En la siguiente pantalla se selecciona Java Web seguido de eso Web Application como se muestra en la siguiente figura.

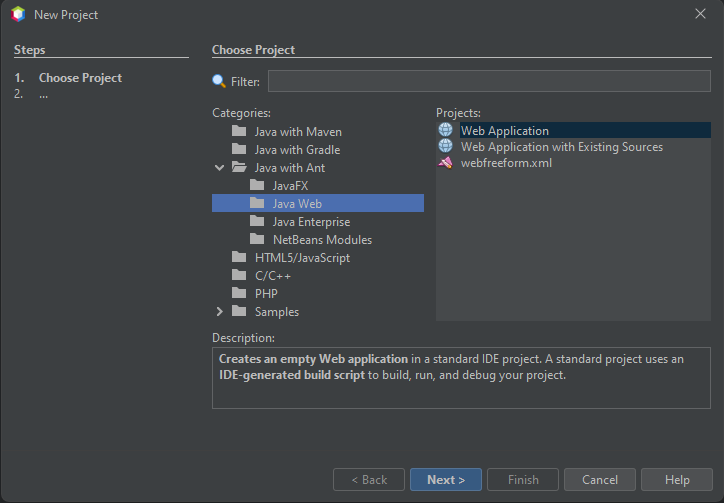


Figura 6. Selección del tipo de aplicación a desarrollar

Se procede a dar el siguiente nombre al proyecto WebSinMVCJavaApellidoNombre y se elige la ubicación que se creó anteriormente.

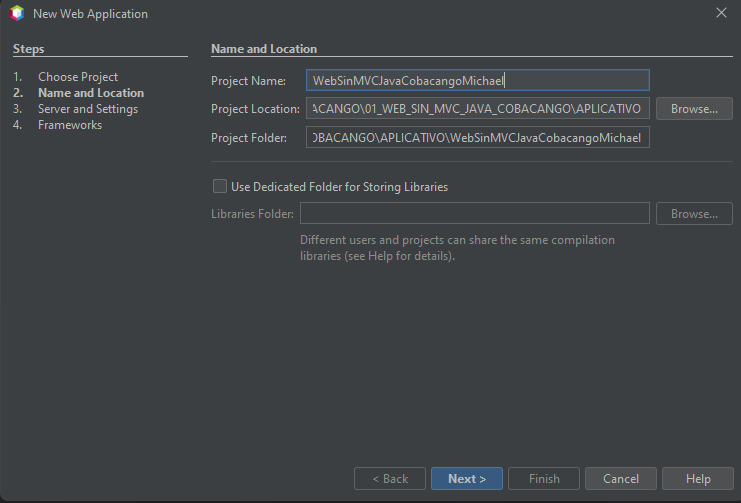


Figura 7. Selección del nombre y directorio donde guardar el proyecto

En la siguiente pantalla se selecciona el Servidor en este caso es Payara al igual que Java EE Version que es la última de Jakarta EE 8 Web como se muestra en la siguiente figura.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Figura 8. Selección del servidor y versión de Java EE.

A continuación, no se deberá seleccionar ningún Frameworks puesto que para la práctica no son requeridos. Finalmente se da clic en Finish.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 9. Selección del Frameworks y finalización de la creación del proyecto.

Una vez finalizada la creación del proyecto obtenemos la siguiente estructura

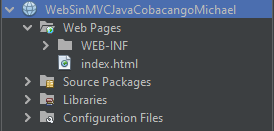


Figura 10. Estructura del proyecto.

## CODIFICACIÓN DE LOS ARCHIVOS .JSP

### **CODIFICACIÓN DEL ARCHIVO INDEX.JSP**

Para comenzar con el desarrollo y la codificación de los archivos se procede a eliminar el archivo index.html el cuál se crea de manera automática al momento de crear el proyecto

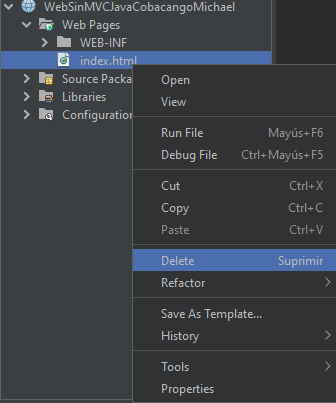


Figura 11. Eliminación del archivo index.html.

Seguido de eso se deberá dar clic derecho sobre el apartado Web Pages seleccionar New y elegir JSP como se muestra en la siguiente figura.

Captura de pantalla de computadora

Descripción generada automáticamente

Figura 12. Selección de un archivo del tipo JSP

Se indica el nombre del archivo el cual es index como se muestra en la siguiente figura.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura 13. Asignación del nombre al archivo a crear.

Una vez creado el archivo se procede a codificar el formulario que se mostrará al momento de ejecutar el proyecto.



Figura 14. Codificación del archivo index.jsp (Formulario)

Tabla 1. Codificación del archivo index.jsp

<%-- Document : index.jsp Created on : 31 oct. 2022, 17:51:48 Author : Michael

--%> <%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

    <title>Formulario Web Sin MVC Java Cobacango</title>

  </head>

  <body>

    <h1>LENGUAJES DE PROGRAMACIÓN</h1>

    <p>

      Indica los lenguajes de programación con los que Ud. esta famailiarizado

    </p>

    <form action="salida.jsp">

      <table border="1">

        <tbody>

          <tr>

            <td>Nombres</td>

            <td><input type="text" name="txtNombre" value="" /></td>

          </tr>

          <tr>

            <td>Apellidos</td>

            <td><input type="text" name="txtApellido" value="" /></td>

          </tr>

          <tr>

            <td>Java</td>

            <td><input type="checkbox" name="chkProgLeng" value="Java" /></td>

          </tr>

          <tr>

<tr>

            <td>PHP</td>

            <td><input type="checkbox" name="chkProgLeng" value="PHP" /></td>

          </tr>

          <tr>

            <td>Ruby</td>

            <td><input type="checkbox" name="chkProgLeng" value="Ruby" /></td>

          </tr>

          <tr>

            <td>Python</td>

            <td><input type="checkbox" name="chkProgLeng" value="Python" /></td>

          </tr>

          <tr>

            <td>

              <input type="file" name="imgProgramador" value="" width="250" />

            </td>

            <td><input type="submit" value="Enviar" /></td>

          </tr>

        </tbody>

      </table>

    </form>

  </body>

</html>

### **CODIFICACIÓN DEL ARCHIVO SALIDA.JSP**

Se deberá dar clic derecho sobre el apartado Web Pages seleccionar New y elegir JSP como se muestra en la siguiente figura.

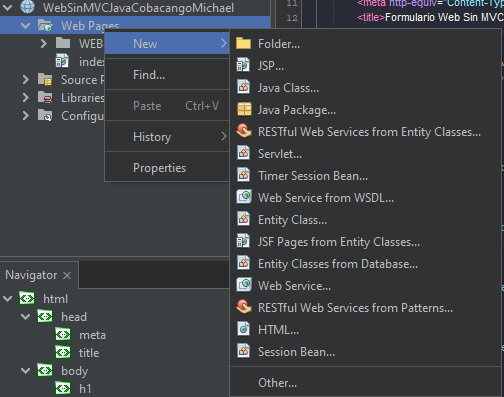


Figura 15. Selección de un archivo del tipo JSP

Se indica el nombre del archivo el cual es salida como se muestra en la siguiente figura.

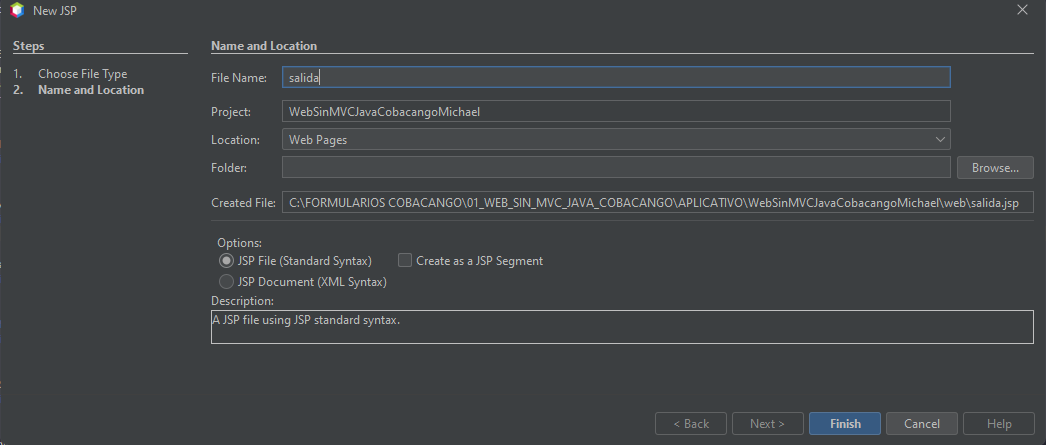
****

Figura 16. Asignación del nombre al archivo a crear.

Una vez creado el archivo se procede a codificar el archivo de salida el cuál recibirá los datos del formulario para poder mostrarlos al cliente.

Texto

Descripción generada automáticamente

Figura 17. Codificación del archivo salida.jsp

Tabla 2. Codificación del archivo salida.jsp

<%-- Document : salida Created on : 31 oct. 2022, 17:57:32 Author : Michael --%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

    <title>JSP Page</title>

  </head>

  <body>

    <h2>Gracias por llenar la encuesta</h2>

    <p>

      <%= request.getParameter("txtNombre") + " " +

      request.getParameter("txtApellido")%>, has respondido que estas

      familiarizado con los siguientes lenguajes de programación:

    </p>

    <ul>

      <% String[] lenguajesSeleccionados =

      request.getParameterValues("chkProgLeng"); if (lenguajesSeleccionados !=

      null) { for (int i = 0; i < lenguajesSeleccionados.length; i++) { %>

      <li><%= lenguajesSeleccionados[i]%></li>

      <%} } else { %>

      <script>

        alert("Falta de información");

        window.location.href = "index.jsp";

      </script>

<% } %> <% String fileName = (String)

      request.getParameter("imgProgramador"); if (fileName != null) { %><br />

      <img src="resources/<%=fileName%>" width="250" />

      <% } %>

    </ul>

  </body>

</html>

## EJECUCIÓN DEL PROYECTO

Para poder ejecutar el proyecto se deberá seleccionar la opción de Run Project como se muestra en la siguiente figura.

Texto

Descripción generada automáticamente

Figura 18. Ejecución del proyecto

Al ejecutar el proyecto se mostrará la siguiente aplicación donde se tiene el formulario que fue codificado en el archivo index.jsp

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Figura 19. Formulario del proyecto.

A continuación, se procede a llenar los campos solicitados y se hace clic en enviar.

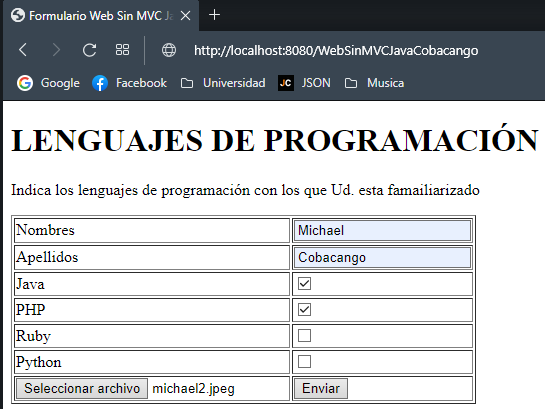


Figura 20. Ingreso de Datos en el formulario index.jsp

Una vez enviado los datos se mostrará la siguiente pantalla con la información ingresada.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura 21. Despliegue del formulario salida.jsp

## CONCLUSIONES

* Las aplicaciones Java EE se encuentran distribuidas en distintas capas lo que garantiza una mejor organización funcional donde las capas son: capa cliente (es la encargada de tener la interacción con el usuario), capa web (permite llevar el control de la aplicación y ciertas ocasiones permite la interacción con el usuario), capa de negocio (contiene la lógica del negocio), y la capa de datos (contiene la información del negocio).
* Se logró desarrollar un proyecto utilizando el entorno de desarrollo Apache Netbeans 15, donde se implementó un formulario y la salida de información ingresada aplicando los conceptos previos a la práctica.

## RECOMENDACIONES

* Es recomendable realizar una revisión previa de los conceptos como son Java EE, JSP, Servlets y tener un conocimiento básico de desarrollo web haciendo uso de tecnologías como HTML, CSS y JavaScript.
* Es recomendable tener conocimientos previos del lenguaje de programación Java ya que los archivos JSP nos permiten incrustar código Java para incrementar la funcionalidad de los proyectos que se encuentren en desarrollo.

## BIBLIOGRAFÍA

* Java EE Specifications <http://www.oracle.com/technetwork/java/javaee/tech/index.html>
* Building Web Apps in Java: Beginning & Intermediate Servlet & JSP Tutorials. <http://courses.coreservlets.com/Course-Materials/csajsp2.html>
* Bryan Basham, Kathy Sierra & Bert Bates: “Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam” O’Reilly, 2008. ISBN 0596516681.
* Richard Monson-Haefel: Enterprise JavaBeans, 3rd Edition. Ed. O’Reilly, September 2001.