

Práctica 5

JSP-Scriptles-Acciones Estándares-Sesiones-EL

Cree un proyecto Maven llamado **Practica5** repitiendo los pasos de la práctica anterior.

1. **Buen día / Buenas Tardes.** Cree una página JSP llamada **saludo.jsp** que muestre un mensaje diferente según la hora en que es accedida:

Si el horario es A.M. mostrará: "Buen día, hoy es:"+la fecha actual.

Si el horario es P.M. mostrará: "Buenas tardes, hoy es:"+la fecha actual.

Use la directiva `<%@ page ... %>` para importar la clase "java.util.Calendar", la cual provee un método para determinar si la fecha corresponde a A.M o P.M. y un método `getTime()` que devuelve la fecha.

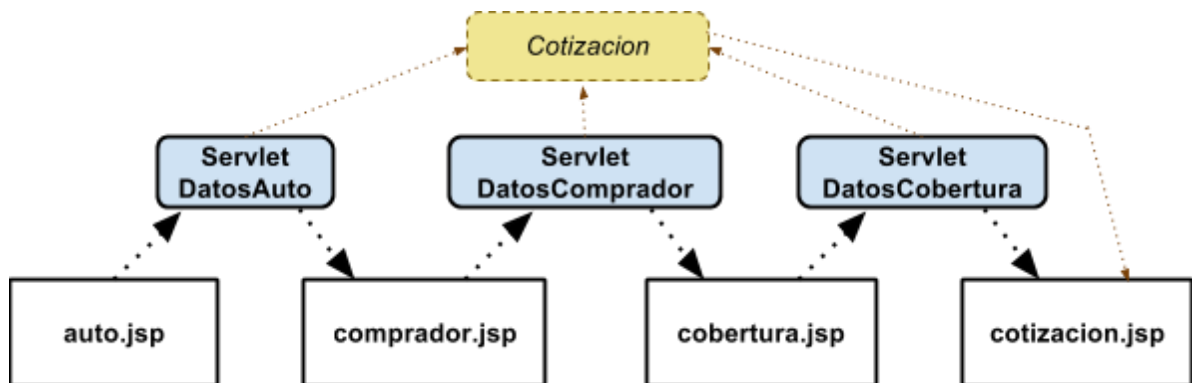
Puede usar el asistente de código para completar los atributos del tag.

- a. Ejecute la página JSP en el servidor Tomcat y **compruebe si funciona.**
- b. Abra el servlet generado desde su JSP y analícelo (busque dónde quedó la variable que definió, dónde se incrementa, objetos implícitos, etc.). También observe si hay definiciones de objetos implícitos.

Nota: Tomcat guarda las clases generadas a partir de la JSP por ejemplo en:

`<SuWorkspace>/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/work/Catalina/localhost/<NombreDeProyecto>...`

2. **Compañía de seguros.** Seguros S.A. requiere un sitio Web que permita cotizar online pólizas de seguros de vehículos. Idea general de la solución:



Para ello se desea construir una serie de páginas web en las cuales el visitante pueda ir personalizando la cotización y obtener finalmente el detalle y su costo. Cada página se enfocara en un tema específico, por ejemplo:

Página	Datos a mostrar en la página
Página 1 (auto.jsp)	Marca: (entrada de texto), Modelo: (entrada de texto), Año del vehículo: (ComboBox con valores de 1990 a 2016), Uso que da al vehículo en Km/Año: (ComboBox con valores múltiplos de 5000)
Página 2 (comprador.jsp)	Dueño: (entrada de texto), Sexo: (radiobutton femenino y masculino), Nacimiento: (date picker), Teléfono, Email

Página 3 (cobertura.jsp)	Tipo de cobertura: (ComboBox o radiobutton: Tercero Completo, Todo riesgo con franquicia, Todo riesgo total)
Página 4 (cotizacion.jsp)	Muestra los ítems elegidos y la cotización final. Use EL para acceder al bean cotización y sus propiedades

- Escriba un **JavaBean** llamado **Cotizacion** con las propiedades indicadas en las páginas 1 a 3 y una propiedad de tipo numérico para guardar la cotización. Agregue al bean un método llamado que simule el cálculo la cotización de acuerdo a los valores seleccionados por el usuario ya guardados en las propiedades del bean.
- Encabezado común a todas las páginas:** escriba una página JSP llamada **encabezado.jsp** que muestre como título y subtítulo los valores de los parámetros llamados titulo y subtítulo tomados desde el requerimiento HttpServletRequest. En el punto c indicaremos los valores de esos parámetros.
- Modifique las páginas (1 a 4) para que **incluyan:**
 - como título: Seguros S.A.
 - como subtítulo para las páginas 1 a 3: "Formulario de Cotización" y para la página 4: "Resultado de la Cotización".
- Responda: ¿Es indistinto utilizar la directiva **@include** o el tag **jsp:include**? **JUSTIFIQUE**

Cree un proyecto Maven llamado **BlogDeMensajes** repitiendo los pasos de la práctica anterior.

- Blog de Mensajes.** Se busca construir un sitio de publicación de mensajes que contenga las páginas que se describen a continuación. Cree los servlets que considere necesarios para que actúen como controllers de la aplicación.

Página	Descripción
visualizarMensajes.jsp	Muestra TODOS los mensajes publicados. Contiene un link a la página login.jsp si el usuario aún NO se autenticó en el sistema.
login.jsp	Solicita un usuario y contraseña y si los datos son válidos se muestra la página visualizarMensajes.jsp
agregarMensaje.jsp	Permite a los usuarios autenticados ingresar el texto de un mensaje a publicar. Invoca a un servlet que guarda en el contexto de la aplicación el mensaje ingresado con el nombre del usuario antepuesto. Una vez guardado el mensaje muestra la página visualizarMensajes.jsp
salir.jsp	Termina la sesión del usuario y muestra la página visualizarMensajes.jsp

NOTAS:

- Cuando el usuario está autenticado, visualizarMensaje.jsp muestra el siguiente menú:
 - Escribir Mensaje** (agregarMensaje.jsp)
 - Salir**

- b. Para iterar sobre los mensajes a mostrar use simplemente scriptlets.
- c. Para resolver la autenticación de usuarios utilice un listener donde se cargue manualmente una lista de objetos Usuario (usuario, clave) en el alcance de aplicación.

Desde el navegador trate de acceder a la página agregarMensaje.jsp sin haberse autenticado. ¿Es correcto que un usuario ingrese a cualquier página de la aplicación sin previa autenticación?. ¿Cómo implementaría el rechazo a usuarios no autenticados considerando lo visto hasta el momento en las prácticas?

JSTL

JavaServer Pages Standard Tag Library es una colección de Tags JSP que encapsulan funcionalidad básica en aplicaciones con páginas JSP.

- a. Agregue en su proyecto MAVEN la dependencia que permita descargar las librerías necesarias para trabajar con JSTL. Para realizar esto, busque en la URL: mvnrepository.com la dependencia. Use la que tiene groupId javax.servlet, version 1.2
- b. Modifique la visualización de los mensajes en la página visualizarMensajes.jsp para que la iteración sobre los mensajes se realice usando los tags **forEach** y **out** de JSTL.