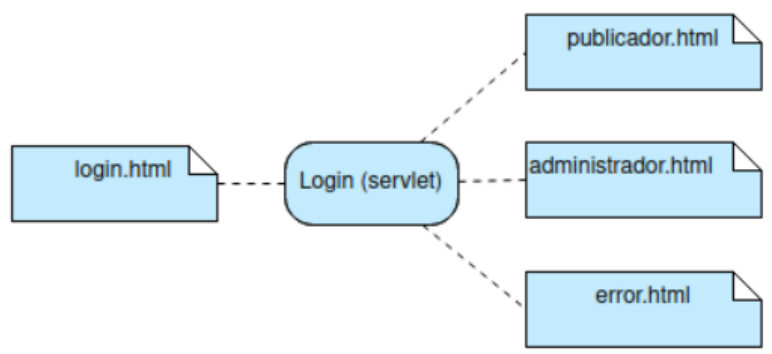


Práctica 2 - Servlets / Alcances / Listeners

1. **Sitio web Clasificados.** Cree un proyecto web dinámico llamado **clasificados**, indicando en el combo **"Dynamic Web module version"** la versión **4.0 o superior**, de esta manera Eclipse utilizará la especificación de servlets que declara los mismos mediante anotaciones. El sitio web clasificados tendrá una "página de login" que proveerá acceso a dos tipos de perfil: "publicador" ó "administrador"



2. Implemente el proyecto según los siguientes incisos:
 - a. Escriba una página HTML llamada **login.html**, que contenga al menos:
 - i. un *título*
 - ii. un *formulario* con dos campos de texto con las etiquetas Usuario y Clave donde el usuario ingresará sus datos de acceso y un botón de tipo *"submit"* para enviar los datos ingresados al servlet del inciso b.
 - b. Escriba un servlet llamado **Login** perteneciente al paquete **tps.clasificados**, que tome del requerimiento http, los datos de un usuario: nombre de usuario y los valide usando datos mantenidos por el propio servlet.

Para la validación, el servlet debe definir en su estado una colección de objetos Usuario (un Usuario tiene un nombre de usuario, clave y perfil) contra los cuales validar. Utilice un objeto List para guardar los usuarios.

Si los datos son válidos: *redirecciona*, usando el método *sendRedirect()*, a una página HTML que contenga un menú de operaciones disponibles de acuerdo al perfil del usuario. Deberá crear una página para el perfil *Publicador* y otra página para el perfil *Administrador*.

perfil	operaciones disponibles
Publicador	Actualizar Datos de Contacto, ABM de Publicaciones, Contestar Consultas
Administrador	Listar Usuarios Publicadores, ABM Administradores, Ver Estadísticas

- c. Si los datos son inválidos: redirecciona a la página `error.html` donde se informa que los datos ingresados no son válidos y se muestra un link a la página `login.html`, para que el usuario vuelva a ingresar sus datos.

3. `sendRedirect()` vs. `forward()`

- a. Modifique el servlet **Login** para que en vez de redireccionar a una página estática, **delegue el requerimiento** (no utilice `sendRedirect()`) **y la respuesta a otro servlet** llamado **Menú**. El servlet **Login** **enviará dentro del request** la información del perfil de usuario al servlet **Menú**. El servlet **Menú** mostrará cual es el perfil y generará dinámicamente el menú correspondiente al perfil de usuario.
- b. ¿Es posible enviar información del perfil desde el servlet Login al servlet Menú usando `sendRedirect()`? De ser posible indique cómo lo implementaría.

4. Listeners

- a. Cree una clase llamada **SitioClasificado** con 3 variables de instancia: nombre, email y teléfono de contacto.
- b. Agregue en el archivo `web.xml` **tres (3) parámetros de contexto** que representan información de un sitio web: nombre del sitio, email y teléfono de contacto.

Nota: si no tiene un archivo `web.xml`, puede crearlo haciendo click con el botón derecho del mouse, usando la opción Java EE Tools -> Generate Deployment Descriptor Stub

- c. Implemente un **Listener de Contexto** que lea del archivo `web.xml` los valores agregados en el punto b. Con esos datos cree una instancia de la clase **SitioClasificado**. Esta instancia será luego guardada **en el contexto de la aplicación Web**.

5. Include

- a. Escriba un servlet llamado **Encabezado** que imprima los datos del sitio Web guardados en el punto anterior.
- b. Modifique el servlet **Menú** para que incluya el encabezado generado en el inciso a).

6. Generación de respuestas binarias a partir de Servlets

Se aproxima el Festival Capital en nuestra ciudad y usted ha sido contratado para implementar una aplicación web que permita participar de una promoción que regala camisetas personalizadas. Las camisetas podrán ser retiradas el día del evento en el mismo Hipódromo de La Plata visitando el stand del sponsor oficial TTPS.

- a. Genere una página `index.html` con un campo para ingresar un texto personalizado que se imprimirá en la remera y en el cupón.

- b. Implemente un servlet llamado **ImprimeCupon** que devuelva una imagen dinámica dada con los siguientes ítems:
- La imagen de la remera del evento (es un archivo `remera.jpg` que reside en el servidor)
 - el texto personalizado ingresado por el usuario
 - Un código de retiro generado aleatoriamente. Por ej: `ThreadLocalRandom.current().nextInt(1000000,99999999)`

NOTA: Utilice la clase **ImageIO** para leer un archivo JPG y generar una salida JPG escribiendo directamente en la respuesta del Servlet la clase **java.awt.image.BufferedImage** para usar su **Graphics** (**java.awt.Graphics2D**) y dibujar sobre la imagen a devolver.

Por ejemplo si se ingresa la frase "**TTPS presente**" el cupón deberá visualizarse así:

