Apéndice A

El lenguaje EL y las librerías JSTL

Si bien Struts ofrece sus propias librerías de acciones para la manipulación de datos en las páginas JSP, es muy común y en algunas ocasiones conveniente, utilizar estas librerías junto con las que ofrece la especificación estándar JSTL. La principal causa de la mezcla de estas librerías se debe a la complejidad, eficiencia o funcionalidad que algunas funciones JSTL ofrece sobre algunas pertenecientes a Struts. Esta comparativa es totalmente dependiente para cada desarrollo y necesidades del programador.

A.1 Variables implícitas EL

Como fue mencionado en el capítulo 2. Las acciones de estas librerías se encuentra enfocadas al uso del lenguaje de expresiones EL. Este lenguaje permite que el programador pueda acceder a los datos de petición y propiedades de objetos sin la necesidad de utilizar lenguaje Java. Las llamadas variables implícitas son las encargadas de cargar automáticamente los datos de petición del usuario, de su espacio en sesión o de la configuración de la aplicación. En la siguiente tabla se muestran las principales variables implícitas JSTL:

Nombre	Descripción
param	Es una colección que contiene todos los valores de los parámetros de petición.
header	Es una colección que contiene los valores de las cabeceras de la petición.
cookie	Es una colección que contiene todas las cookies que el navegador haya enviado en la petición.
initParam	Es una colección que contiene todas los parámetros de configuración almacenados en el archivo `web.xml`
pageScope	Es una colección que contiene todas las variables que tengan ámbito de página.
requestScope	Es una colección que contiene todas las variables que tengan ámbito de petición.

Tabla A.1: Variables implícitas de JSTL

Nombre	Descripción
sessionScope	Es una colección que contiene todas las variables que tengan ámbito de sesión.
applicationScope	Es una colección que contiene todas las variables que tengan ámbito de aplicación.

Tabla A.1 (*continuación*)

A.2 Ámbito de datos

En algunas variables implícitas se mencionan los distintos ámbitos que puede tener una variable en la aplicación. El ámbito es el alcance que una variable tiene definido a lo largo de la aplicación. Existen 4 tipos de ámbito:

1. Ámbito de página: Es el ámbito por defecto. La variables con este tipo de alcance sólo se encuentran disponibles en la página actual (en la que está siendo procesada) una vez que hay redirección a otra o se sale de esta la variable deja de estar disponible.

2. Ámbito de petición: La disponibilidad de una variable con este ámbito se da a lo largo de la resolución de una petición HTTP.

3. Ámbito de sesión: Las variables con este ámbito se encuentran disponibles durante el tiempo que el usuario utilice la aplicación. La variable se almacena en una cookie en memoria que estará disponible mientras el navegador se encuentre abierto o el tiempo de disponibilidad de este no sea rebasado.

4. Ámbito de aplicación: Este es el alcance máximo que puede tener una variable y su valor puede ser accedido por todos los usuario (independientemente del navegador).

A.3 Funciones de EL

El lenguaje EL dispone de distintas funciones que permiten un mejor manejo de los datos como se muestra a continuación:

Función	Retorno	Descripción
fn:contains(cadena,subcadena)	boolean	Verifica si la subcadena se encuentra contenida en la cadena.
fn:containdIgnoreCase(cadena,subcadena)	boolean	Verifica si la subcadena se encuentra contenida en la cadena sin importar el uso de mayúsculas y minúsculas.
fn:endsWith(cadena,sufijo)	boolean	Verifica que la cadena termine con el sufijo indicado.
fn:startsWith(cadena,prefijo)	boolean	Verifica que la cadena comience con el prefijo indicado.
fn:indexOf(cadena,subcadena)	int	Devuelve la posición en la que la cadena contiene a la subcadena o -1 si no se encuentra.
fn:length(cadena)	int	Devuelve la longitud de la cadena.
fn:Split(cadena,separador)	String[]	Divide la cadena original por cada ocurrencia que se encuentre del separador.
fn:toLowerCase(cadena)	String	Devuelve la cadena transformando cada uno de sus caracteres en mayúsculas a minúsculas.
fn:toUpperCase(cadena)	String	Devuelve la cadena transformando cada uno de sus caracteres en minúsculas a mayúsculas.
fn:trim(cadena)	String	Devuelve la cadena limpia de espacios iniciales y finales.

Tabla A.2: Funciones EL

A.4 La librería Core

Esta librería se encarga de las condiciones, bucles, asignación de variables y redirecciones. Sus acciones principales son:

<c:if>

Se trata de una etiqueta condicional simple en la que se evalúa el contenido de la variable comparándolo con la condición indicada y así realizar cierta acción. Sus atributos principales son:

Atributo	Tipo	Descripción
test	Boolean	Contiene la expresión a evaluarse. Es el único atributo obligatorio.
var	String	El nombre de la variable donde se guardará, de ser necesario, el resultado de la evaluación.
scope	String	El ámbito que se dará a la variable `var`. Por defecto se le asigna `page`.

Tabla A.3: Atributos de la etiqueta `<c:if>`

<c:choose>, <c:when>, <c:otherwise>

Estas etiquetas conforman entre sí una condición más compleja para ser evaluada. La etiqueta `<c:when>` presenta el mismo comportamiento que `<c:if>`, pero siempre es utilizada dentro de un bloque definido por `<c:choose>`, el cuál controla el procesamiento de los diferentes `<c:when>` anidados, y así comprobar los diferentes valores que puede tomar la condición. Análogamente a la programación estructurada, su uso es similar al `switch`.

<c:choose> → switch

<c:when> → case

<c:otherwise> → default

El único atributo requerido por estas etiquetas es `test` (de funcionamiento idéntico que en `<c:if>`) y es soportado por la etiqueta `<c:when>`

<c:forEach>

Permite la iteración sobre las colecciones para obtener los valores contenidos en estas. Sus atributos son los siguientes:

Atributo	Tipo	Descripción
begin	int	Indica el índice en el que iniciarán las iteraciones.
end	int	Indica el último índice de la colección
items	array	La colección de objetos sobre la que se está iterando.
step	int	Indica el incremento de cada iteración. Por defecto su valor es 1.
var	String	El nombre de la variable que se usará como referencia para el elemento actual de la colección.

Tabla A.4: Atributos de la etiqueta `<c:forEach>`

<c:set>

Útil para la creación de variables requeridas para el procesamiento de la página JSP. Sus atributos son los siguientes:

Atributo	Tipo	Descripción
value	Object	El valor que será asignado a la variable.
var	Object	Nombre de la variable que será creada.
scope	String	Ámbito de dato que será asignado a la variable.

Tabla A.5: Atributos de la etiqueta `<c:set>`

A.5 La librería Format

Esta librería contiene funcionalidades útiles para dar formato a la información usando los patrones dados. Sus etiquetas principales son:

<fmt:formatDate>

Se utiliza para dar un formato específico a las fechas independientemente del formato que tengan en el servidor. Sus atributos principales son:

Atributo	Tipo	Descripción
value	Date	La fecha a la que se dará formato.
pattern	String	El patrón en que debe mostrarse la fecha dada.
type	String	Indica si se desea presentar sólo la fecha, la hora o ambos.
timeZone	String	Especifica la zona horaria
var	String	Indica la variable en que se guarda la fecha una vez que ha sido formateada.
scope	String	Especifica el ámbito de la variable anterior.

Tabla A.6: Atributos de la etiqueta `<ftm:formatDate>`

<fmt:formatNumber>

Su uso es muy parecido a la etiqueta anterior pero aplicada al formato de los números. Sus atributos principales son:

Atributo	Tipo	Descripción
value	String / Número	Especifica el número a formatear.
pattern	String	Indica el patrón que debe seguir el número dado.
type	String	Indica de qué tipo de número de trata: cantidad, moneda, o porcentaje.
currencyCode	String	Indica el código ISO-4217 de la moneda (define todas las monedas del mundo por medio de un código. MXN es para el peso mexicano).
currencySymbol	String	Indica el símbolo de la moneda

Tabla A.7: Atributos de la etiqueta `<ftm:formatNumber>`

Atributo	Tipo	Descripción
maxIntegerDigits	int	Especifica el número máximo de cifras en la parte entera del número.
maxFractionDigits	int	Especifica el número máximo de cifras en la parte decimal del número.
minFractionDigits	int	Especifica el número mínimo de cifras en la parte decimal del número.
var	String	Indica la variable en que se guarda la fecha una vez que ha sido formateada.
scope	String	Especifica el ámbito de la variable anterior.

Tabla A.7 (*continuación*)

A.6 La librería SQL

Esta librería contiene las diferentes tareas requeridas para el acceso a bases de datos realizado directamente en las páginas JSP. Esta acción, como se ha mencionado y demostrado, no es recomendable debido a la mezcla de código de presentación, lógica de negocio y acceso a datos en una JSP. Dando como resultado la nula reutilización de elementos programados, código altamente acoplado, nula escalabilidad y un mantenimiento deficiente. Sus principales etiquetas (para que queden en conocimiento) son:

<sql:query>

Utilizada para definir las consultas encargadas de leer información de la base datos.

<sql:transaction>

Dentro de esta etiqueta se marcan las diferentes sentencias que forman parte de una transacción.

<sql:update>

Utilizada para realizar tareas de actualización tales como `INSERT`, `DELETE`, y `UPDATE`.

Glosario

1. **API (Application Programming Interface):**

   Conjunto de clases que forman parte de una librería o framework.

2. **CSS (Cascade Style Sheet):**

   Lenguaje que define la presentación de un documento HTML.

3. **DI (Dependency Injection):**

   Patrón de diseño propio de la programación orientada a objetos en el que las dependencias de los objetos son suministradas en vez de instanciadas por estos.

4. **EL (Expression Language):**

   Lenguaje de expresiones utilizado por JSTL para acceder a las propiedades de los objetos.

5. **GUI (Graphical User Interface):**

   Componentes gráficos que ayudan a la interacción de un sistema con el usuario.

6. **HTTP (Hypertext Transfer Protocol):**

   Protocolo de petición/respuesta que comunica al cliente con el servidor.

7. **JDBC (Java DataBase Conectivity):**

   API estándar de Java que permite la conectividad con una base de datos.

8. **JSP (Java Server Pages):**

   Tecnología Java que permite la creación de contenido dinámico para la creación de documentos HTML.

9. **JSTL (JSP Standard Tag Library):**

   Librería estándar de JSP para el manejo de contenido dinámico a través de etiquetas de acciones en sustitución de código Java embebido.

10. **MIME (Multipurpose Internet Mail Extensions):**

    Estándar utilizado por el navegador para comprender y manejar los contenidos incluidos en él (image/jpeg, text/html, video/mpeg…)

11. **MVC (Model, View, Controller):**

    Patrón de diseño que divide la lógica de negocio, el acceso a datos (Modelo), la presentación de información (Vista) y el manejo de la petición (Controlador) con la finalidad de desacoplar los componentes.

12. **OLAP (On-Line Analytical Processing):**

    Base de datos aplicada al procesamiento analítico para la toma de decisiones.

13. **OLTP (On-Line Transaction Processing):**

    Base de datos aplicada para las operaciones transaccionales.

14. **ORM (Object Relational Management):**

    Paradigma de conversión de datos relacionales a objetos (y viceversa).

15. **POJO (Plain Old Text Object):**

    Clases que definen atributos con sus métodos getter y setter para el transporte de datos.

16. **RDBMS (Relational DataBase Management System):**

    Sistema gestor de base de datos que proporciona el ambiente para el manejo de datos relacionales.

17. **SCD (Slow Changing Dimension):**

    Dimensiones cuyos datos tienden a modificarse ocasionalmente en el tiempo.

18. **W3C (World Wide Web Consortium):**

    Comunidad internacional encargada de definir los estándares del mundo web.