**Informe:**

**Taller de Programación Java EE para la SUNARP**



Proyecto de Migración de Módulo Web

Calculadora Registral - Versión 1.0

2018

**INDICE**

[1. Antecedentes 1](#_Toc519601973)

[2. Objetivos 1](#_Toc519601974)

[3. Selección del Framework 2](#_Toc519601975)

[4. Desarrollo de la Capa de Servicio o Modelo 2](#_Toc519601976)

[4.1. DTO (Data Transfer Object) 3](#_Toc519601977)

[4.2. Capa de Servicio 3](#_Toc519601978)

[4.2.1. JDBC – Acceso a Datos 3](#_Toc519601979)

[5. Desarrollo del Aplicativo Web 4](#_Toc519601980)

[5.1. Servlets 4](#_Toc519601981)

[5.2. JSP – (Java Server Pages) 5](#_Toc519601982)

[5.3. Bootstrap 5](#_Toc519601983)

[5.4. JSTL 5](#_Toc519601984)

[5.5. JQuery 5](#_Toc519601985)

[6. Conclusiones 6](#_Toc519601986)

1. Antecedentes

La SUNARP tiene como misión el Inscribir y publicitar actos, contratos, derechos y titularidades de las personas de manera oportuna, inclusiva, transparente, predictible y eficiente.

Cuando la SUNARP inicia operaciones (en 1995), se logra mejorar la calidad de la atención mediante la implementación de nuevos sistemas informáticos. Sin embargo, algunos productos informáticos se continúan utilizando como es el caso del módulo de pre-liquidación desarrollado en FoxPro y que permitía que el área de orientación al usuario pueda calcular las Tasas Registrales que los usuarios debían pagar por sus tramites

En 2004 se rediseña el módulo y se desarrolla una solución de escritorio en Visual Basic 5 - Access para que sea utilizado tanto por los usuarios internos como para las Notarías mediante un convenio.

A inicios de 2007 se migra la solución de escritorio a una solución web y se implementa el módulo de pre-liquidaciones con tecnología ASP 1.0, el cual a la fecha aún se encuentra en producción.

1. Objetivos

Como parte del Taller de Programación Java EE, se tiene por objetivo migrar el Módulo Calculadora Registral a una solución de tipo Java.

El presente documento describe las etapas que se tomaron para desarrollar la migración:

1. Selección del Framework
2. Desarrollo de la Capa de Servicio
3. Desarrollo del Aplicativo Web
4. Selección del Framework

El presente proyecto planea utilizar el Framework MVC (Modelo, Vista Controlador) más el uso de Servlets para la capa de la vista en conjunción con páginas JSP.

1. Desarrollo de la Capa de Servicio o Modelo

El desarrollo de aplicativo ha considerado crear dos proyectos java, uno para los objetos DTO y otro específicamente para el servicio; la conectividad se realiza a través de JDBC.

|  |  |
| --- | --- |
| **DTO** | **SERVICE** |
|  |  |

El desarrollo de estas capas se ha efectuado utilizando:

* 1. DTO (Data Transfer Object)

Es un objeto que transporta datos entre procesos. La motivación de su uso tiene relación con el hecho que la comunicación entre procesos se realiza generalmente mediante interfaces remotas (por ejemplo, servicios web), donde cada llamada es una operación costosa.

Como la mayor parte del costo de cada llamada está relacionado con la comunicación de ida y vuelta entre el cliente y servidor, una forma de reducir el número de llamadas es usando un objeto (el DTO) que agrega los datos que habrían sido transferidos por cada llamada, pero que son entregados en una sola llamada.

La diferencia entre un objeto de transferencia de datos y un objeto de negocio (BUSINESS OBJECT) o un objeto de acceso a datos (DATA ACCESS OBJECT, DAO) es que un DTO no tiene más comportamiento que almacenar y entregar sus propios datos.

* 1. Capa de Servicio

La capa de servicio siguiendo la estrategia del uso de interfaces más su respectiva implementación nos permite ordenar el código y separar la especificación de los métodos a utilizarse.

**Interfaz (Java)** Una interfaz en Java es una colección de métodos abstractos y propiedades constantes. En las interfaces se especifica qué se debe hacer, pero no su implementación. Serán las clases que implementen estas interfaces las que describan la lógica del comportamiento de los métodos.

El uso de las interfaces Java proporciona las siguientes ventajas:

* Organizar la programación, permiten declarar constantes que van a estar disponibles para todas las clases que queramos (implementando esa interfaz).
* Obligar a que ciertas clases utilicen los mismos métodos (nombres y parámetros).
* Establecer relaciones entre clases que no estén relacionadas.
  + 1. JDBC – Acceso a Datos

JAVA DATABASE CONNECTIVITY, más conocida por sus siglas JDBC, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

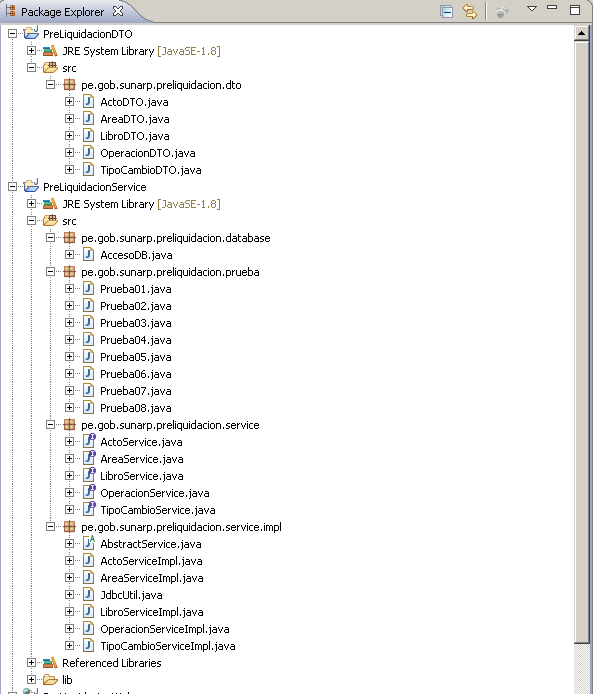


Figura 1.- Proyecto DTO y Proyecto Service

1. Desarrollo del Aplicativo Web

La aplicación Web se ha desarrollado con el empleo de las siguientes tecnologías:

* 1. Servlets

Son objetos que corren dentro del contexto de un contenedor de servlets (ej: Tomcat) y extienden su funcionalidad. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

* 1. JSP – (Java Server Pages)

Es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML y XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

* 1. Bootstrap

Es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo.

* 1. JSTL

JSTL - La tecnología JavaServer Pages Standard Tag Library (JSTL) es un componente de Java EE. Extiende las ya conocidas JavaServer Pages (JSP) proporcionando cuatro bibliotecas de etiquetas (Tag Libraries) con utilidades ampliamente utilizadas en el desarrollo de páginas web dinámicas.

Estas bibliotecas de etiquetas extienden de la especificación de JSP (la cual a su vez extiende de la especificación de Servlet). Su API nos permite además desarrollar nuestras propias bibliotecas de etiquetas.

Las bibliotecas englobadas en JSTL son:

* core, iteraciones, condicionales, manipulación de URL y otras funciones generales.
* xml, para la manipulación de XML y para XML-Transformation.
* sql, para gestionar conexiones a bases de datos.
* fmt, para la internacionalización y formateo de las cadenas de caracteres como cifras.

Estas bibliotecas de etiquetas extienden de la especificación de JSP (la cual a su vez extiende de la especificación de Servlet). Su API permite además desarrollar bibliotecas propias de etiquetas.

* 1. JQuery

Es una biblioteca multiplataforma de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizada.1?

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados.2? jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Es requisito para algunas de las funcionalidades utilizadas por Bootstrap.

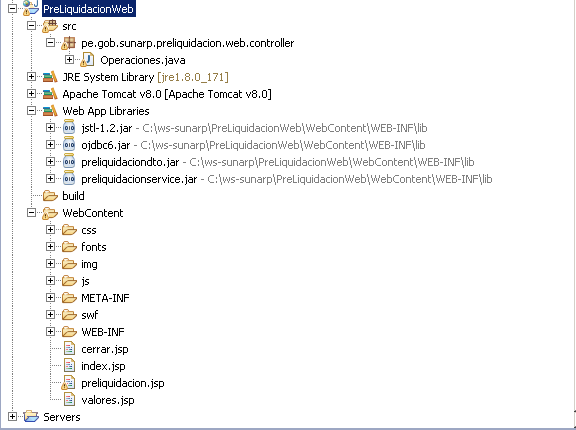


Figura 2.- Proyecto Solución Web

1. Conclusiones

El curso ha cumplido con las expectativas de proporcionar el conocimiento teórico-práctico para desarrollar aplicaciones web con el uso de herramientas Java. Nos hubiese gustado que el curso incluya en acápite de buenas prácticas para una mejor aplicación de los conocimientos recopilados.