

Sistemas Web II

Práctica 2

SOAP

Carlos Ambrosio Habela

David Recio Arnés

1.Introducción

La elección del proyecto del documento para trabajar con SOAP en el entorno de desarrollo es sobre un recetario y recetas como en el proyecto anterior. La elección se llevó a cabo por dos integrantes del grupo.

El proyecto se desarrolla en Java, en el entorno de desarrollo *Neatbeans*. Se ha usado también para el correcto desarrollo por parte de los dos integrantes del grupo el control de versiones *Github* para llevar un control y facilitar el reparto de tareas y el seguimiento del proyecto.

2.Resumen

En el caso del proyecto, como no se pide ninguna interfaz gráfica se prefirió decantarse por la terminal de salida sirviendo de un menú simple, para poder enfocarse en el tema principal de la práctica.

Para darle un cierto control de la programación se decidió implementar un repositorio como se dijo anteriormente, además de definir una buena estructura, todo ello permitió trabajar paralelamente ya que se tocaban zonas de código diferente, e incluso poder hacer uso del *pair programming*.

El proyecto consta de tres partes diferenciadas o tres proyectos distintos, un proyecto servidor en web y dos proyectos que actúan de clientes, uno en Java y otro en web.

Los clientes le harán peticiones al servidor y este consumirá los recursos y los ofrecerá.

Los ficheros XML necesarios para el correcto funcionamiento del proyecto se encuentran en la carpeta *files* de los respectivos proyectos, es decir, en el caso por ejemplo del servidor web la carpeta sería (Practica2SOAPClass/files/xml), el funcionamiento es similar con los otros proyectos.

Respecto a la ejecución, solo hay un dato importante, se debe crear o importar un recetario antes de crear las recetas, ya que este es el que actuara de almacén.

3.Desarrollo y despliegue

El proyecto cuyo nombre es *Practica2SOAP*, está desarrollado en Java y consta como se dijo anteriormente de tres proyectos separados, dos clientes (web y java) y el servidor. Cada uno de ellos tiene una función y unos paquetes con unas clases diferenciadas de las cuales se hablará más adelante. Primero se hacen las recetas y posteriormente el recetario.

El proyecto *Practica2SOAPClass* es el servidor y consta de tres paquetes: *Funcionalidad*, *Recursos* y *serviciosWeb* con sus respectivas clases.

- El paquete *Funcionalidad* tiene dos clases *Marshalling* y *ValidarXSD* que son idénticos a los de la primera práctica, salvo que el *Marshalling* recibe ahora ficheros. La clase *Marshalling* realiza el *marshalling* y *Unmarshalling*, que sirve para hacer volcados de

objetos a XML o viceversa (*unmarshalling*). La clase *ValidarXSD* se usa, como su nombre indica para validar los XML correspondientes en la carpeta files y dentro de esta en la subcarpeta con nombre *xsd* y el fichero *recetario*.

- El paquete *Recursos* tiene dos clases que son *Receta* y *Recetario*, que tiene la misma funcionalidad que en la practica1 siendo estas las clases con las que se va a trabajar a lo largo de los proyectos. Estas clases son idénticas a las de la primera práctica.
- El paquete *serviciosWeb* contiene una clase del mismo nombre en el que están todas las funciones necesarias para el desarrollo de la practica como pueden ser añadir *Receta* y *recetario*, borrarlas, listar recetas y *recetarios*, etc.

El proyecto *Practica2SOAPClienteJava* es el servidor Java y consta dos paquetes a parte de los que se generan automáticamente (esenciales para la correcta comunicación con el servidor) que son: *practica2soapclientejava*, *Funcionalidad* con sus respectivas clases:

- El paquete *practica2soapclientejava*, es como el *launcher* que se encarga de llamar al menú.
- El paquete *Funcionalidad* tiene dos clases, que son el menú y modelo que se encarga de hacer el trabajo pesado de la aplicación, donde participan las diferentes clases.

El proyecto *Practica2SOAPClienteWeb* es el servidor web y consta de dos paquetes *beans*, en los que hay dos clases que son *Receta* y *Recetario* necesarias para las *servlets* de las que se hablará posteriormente. Y el segundo paquete contiene las *servlets* implementadas.

A continuación, se explicarán las clases *servlets* y los respectivos archivos *html* para su correcto funcionamiento.

- *CrearRecetarioServlet*: Esta *servlet* crea *recetarios*.
- *CrearRecetaServlet*: Esta *servlet* crea recetas.
- *ListarRecetarioServlet*: Se obtiene un listado de *recetarios* disponible.
- *ListarRecetaServlet*: Se obtiene un listado de recetas disponible.
- *ExportReceta*: Exporta la receta seleccionada
- *ExportRecetario*: Exporta el *recetario* seleccionado
- *ImpotReceta*: Importa la receta seleccionada
- *ImportRecetario*: Importa el *recetario* seleccionado
- *RmvReceta*: Borra las recetas seleccionadas.
- *ValidarXSD*: Comprueba que el fichero seleccionado esta correctamente validado.
- *Index*: Es el menú de las *servlets* ofrecidas.

En la carpeta *web pages* dentro del proyecto de *Practica2SOAPClienteWeb*, se encuentran unos archivos *html* que son necesarios que actúan como formularios para la creación o bien para obtener la lista de recetas y *recetarios*.

Cabe destacar, que, para la compilación y ejecución de este proyecto, se han de tener en cuenta que se debe tener desplegado el servidor y al menos un cliente ejecutándose y se tiene que seguir unos pasos para su ejecución. En primer lugar, se debe “*deployar*” o desplegar el proyecto del servidor, es decir, el *Practica2SOAPClass*. Esto se hace de la siguiente manera (hay que pulsar botón derecho y pinchar en *Deploy*). Una vez hecho esto, ya se pueden compilar el resto del proyecto. Para el cliente web se debe abrir con el navegador *Internet Explorer*. Los archivos de

prueba correctos están ubicados dentro de *files/xml* en el cliente Java. Y los ficheros erróneos (que están adrede), se encuentran en *files/xml* pero en esta ocasión en el cliente web.

Para crear un cliente Java debes irte al cliente Java y ejecutar el *launcher*, una vez ahí simplemente debes importar un recetario o crear uno nuevo.

Para crear un cliente web debes ejecutar el *html index* y una vez ahí, simplemente se debe importar un recetario o crear uno nuevo.

Cabe destacar que, para exportar recetas, se debe haber creado alguna, no hay problema de solapamiento a la hora de almacenar datos en el servidor temporalmente, ya que sus funciones son atómicas y se trabaja siempre sobre las mismas variables (almacenadas en el servidor), de tal forma que no se tiene de ellas ninguna copia local en el cliente, impidiendo así problemas de inconsistencia por falta de sincronía.

Si se requiere de algún recetario/receta que se quiera importar al servidor o que este lo mande, se trabajará bajo la carpeta *xml*, ubicada en el cliente. En el traspaso de ficheros de cliente a servidor, se realizará mediante una transmisión de *arrays* de bytes.

Respecto al WSDL se genera automáticamente a la hora de realizar la creación del cliente web, lo único que se hizo, fue desplegar el servidor, y crear un cliente web en el cliente web de cada proyecto, el cual se veía enlazado con el proyecto seleccionado que previamente había sido desplegado.