

WSDL :

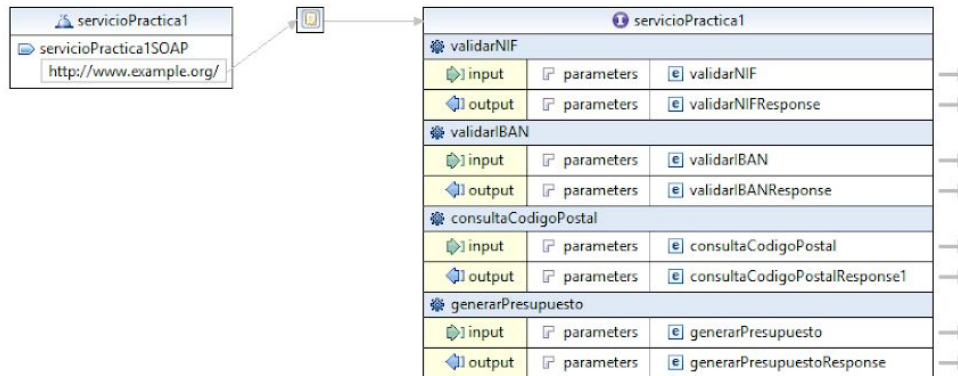
Interoperabilidad

WSDL

DESARROLLO DE LA PRÁCTICA

Para el desarrollo de la práctica se han seguido los pasos especificados en la guía de interoperabilidad que se nos ha proporcionado, creando primero el WSDL y utilizando la vista de diseño que nos proporciona eclipse para agregar los nuevos servicios y datos de entrada y salida que necesitan.

Los servicios son los siguientes:



Cada uno dispone de los siguientes parámetros de entrada y salida.

	Entrada	Salida
validarNIF	nif : String SoapKey:String	out:Boolean
validarIBAN	iban : String SoapKey:String	existe: Boolean error: String
consultaCodigoPostal	cp : String SoapKey:String	codigoPostal: String poblacion: String provincia: String
generarPresupuesto	fechaPresupuesto : Date SoapKey:String referenciaProducto:String cantidadProducto: Int idCliente: Int	idPresupuesto: int presupuestoGeneradoCorrectamente: Boolean

Tras agregar el Web Service, se crean todas las clases necesarias para la gestión de estos servicios. Se agrega la lógica de negocio al skeleton. Todos tienen algo en común, que es la comprobación de la SoapKey en la base de datos.

La base de datos tiene dispone de dos claves: “clave1”, “clave2”. En caso de que la clave introducida no coincide con ninguna de estas, se lanzará una excepción que será mostrada en el cliente. El resto de servicios también tienen ciertas características.

Validar NIF: Este servicio devolverá un verdadero o falso en función del resultado de la comprobación. Además puede lanzar excepción si ha habido algún problema al acceder a la base de datos para comprobar la clave.

Validar IBAN: Para validar el IBAN se ha utilizado una librería de Apache Commons Validator, que es capaz de validar el IBAN de muchos países.

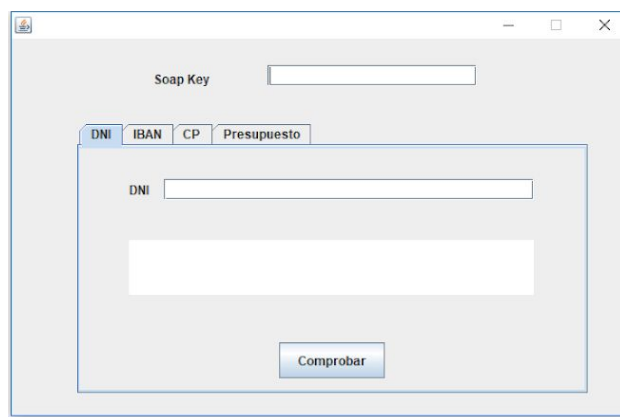
Consultar CP: La base de datos está poblada con los códigos postales y sus datos correspondientes de todo el territorio. Como el servicio devuelve únicamente una estructura compleja, se devuelve el primer resultado que coincida con el CP introducido. En caso de que no encuentre ningún dato, se devuelve una estructura vacía, por lo que puede ser interpretado por el cliente como que no hay ningún dato al respecto.

Generar Presupuesto: Ya hemos visto los datos que recibe este servicio. Si se introduce dos veces el mismo presupuesto con los mismos datos, el servicio devuelve la id del primero que genere, aunque genere una entrada nueva en la base de datos.

Cliente.

El cliente está desarrollado en java utilizando SWING para conseguir la interfaz. Para comunicarse con los servicios en necesario añadirle un Web Service Client con la dirección del WSDL que contiene la información de nuestro servicios.

Arriba se incluye la SoapKey y en cada uno de los servicios aparecen los datos que se solicitan, así como un apartado en el que se mostrara la respuesta.



DESPLIEGUE DE LA PRÁCTICA

La aplicación consta de tres partes que necesitan sincronizarse para funcionar:

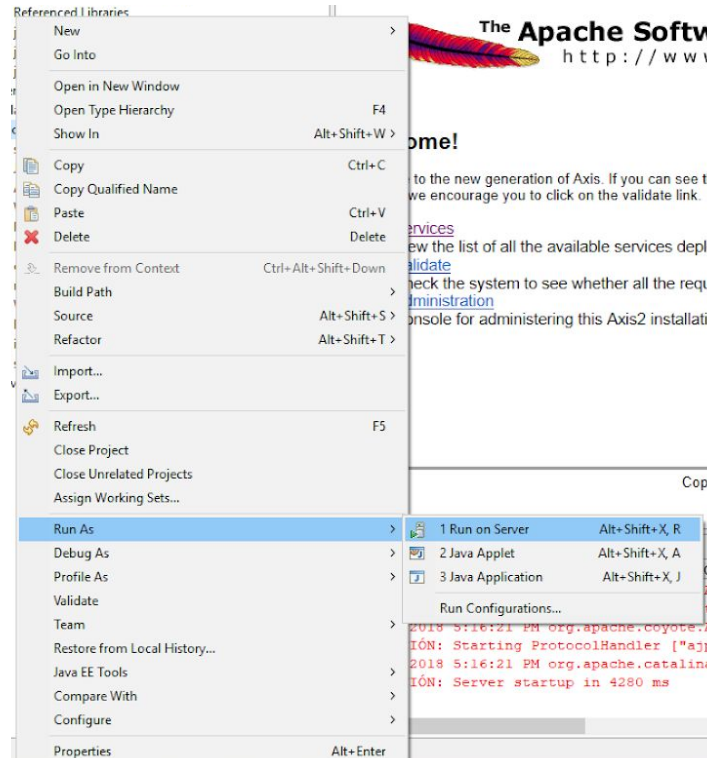
- Web Services

- BBDD
- Cliente

Lanzamiento de Web Services:

En la entrega se incluye el proyecto eclipse de lo servicios. Ha de ser desplegado en Apache Tomcat, por lo que lo primero que haremos será importar el proyecto en eclipse.

Una vez tengamos el proyecto, se puede desplegar en el servidor haciendo botón derecho sobre el mismo y seleccionando las siguientes opciones.



Al pulsar estas opciones, se nos abrirá una pantalla con el servidor localhost de Tomcat, vale con darle a finalizar para lanzar el servicio web en el servidor.

Base de datos.

Se necesita un servicio MYSQL corriendo en el que crear la base de datos que se va a poblar para que los servicios puedan hacer uso de ella.

En mi caso he utilizado XAMPP que me permite lanzar un servicio de mysql, y gestionarlo con phpmyadmin. Junto con la práctica se adjunta un archivo sql (mtis.sql), que es el que tendremos que importar para regenerar la base de datos. Para ello se necesita crear dicha base de datos con el nombre de "mtis", además, los servicios hacen uso de un usuario llamado "user", con la contraseña "user". Este usuario es necesario tanto para consultar como para crear los presupuesto del servicio correspondiente, por lo que necesita dichos privilegios en la base de datos que se acaba de crear. Una vez esté creada, basta con importar el archivo .sql.



Cliente

Aunque se adjunta el proyecto del cliente, por si se quieren consultar los fuentes. También hay un archivo ejecutable, que permite utilizar el cliente sin necesidad de abrirlo en eclipse.