

MTIS	Metodologías y Tecnologías de Integración de Sistemas
	Práctica 1
	WSDL: Interoperabilidad WSDL

Preámbulo

Una de las tareas más importantes a la hora de crear una SOA es definir su modelo de servicios: o sea, qué servicios hay y qué tareas en concreto hace cada uno. En una SOA basada en servicios web este modelo se plasmará en documentos WSDL que definan en detalle las interfaces de cada servicio: operaciones, datos recibidos, datos devueltos y errores que pueden ocurrir. Estos WSDLs son casi imprescindibles a la hora de crear los clientes de un servicio, pues facilitan enormemente la tarea de invocarlo y gestionarlo.

El Objetivo de esta práctica es crear un WSDL desde cero (top-down), sin necesidad de crear un servicio web. Esta técnica es la recomendada a la hora de crear servicios web, ya que con ello nos aseguramos la independencia de las herramientas. Lo apropiado es que la herramienta se adapte al WSDL, y no al revés.

Pasos a realizar

Pasos a seguir:

1. Instalar Eclipse
2. Instalar Tomcat 8
3. Configurar Eclipse con Tomcat 8 y Apache Axis 2
4. Creación de un proyecto dinámico
5. **Definir WSDL**
6. **Crear un servicio web que utilice la WSDL definida anteriormente**

Los pasos 1,2,3,4 los vimos en Sistemas Distribuidos, en nuestro caso ya tenemos el Eclipse previamente configurado para trabajar y poder crear servicios web SOAP. Los pasos 5 - 6, están detallados en la presentación anexa.

Enunciado

Se pretende construir un sistema para control de un almacén inteligente, mediante el empleo de terminales de control de presencia y otros dispositivos. Para gestionar dicho edificio inteligente y poder realizar diversas tareas, haremos uso de los servicios web.

En este caso necesitamos varios servicios web encargados de gestionar y validar información de los empleados (Servicio Empleados), gestionar la seguridad (Seguridad) y realizar validaciones (Servicio Utilidades).

Se pide, crear un contrato WSDL para cada servicio de los descritos, incluyendo los métodos indicados a continuación, posteriormente se deberán crear los servicios web asociados a los contratos incluyendo la lógica de negocio necesaria, para el correcto funcionamiento.

1. Crear tres servicios web mediante el enfoque Top-down, mediante el asistente de eclipse para dicha tarea.

El servicio web (**Empleados**), debe de proporcionar los siguientes métodos:

nuevo, el cual creará un nuevo empleado en BD, a partir de la siguiente estructura:

NIF: string
Nombre: string
Apellidos: string
Dirección: string
Población: string
Teléfono: string
Email: string
Fecha Nacimiento: Date
Numero Seguridad Social: string
IBAN: string

Y devolverá un booleano para indicar si ha sido correcta la inserción del registro en BD y un texto el cual mostrará el error en caso de haberlo durante la inserción.

borrar, se encargará de borrar de BD, el registro indicado por el NIF y devolverá un booleano para indicar si ha sido correcta el borrado del registro en BD.

modificar, se encargará de actualizar en BD, se le pasará la estructura completa que define un empleado y devolverá un booleano para indicar si ha sido correcta la actualización del registro en BD.

consultar, devolverá a partir de un NIF la estructura completa de un empleado.

El servicio web (**Seguridad**), debe de proporcionar los siguientes métodos:

validarUsuario, determinará si un Empleado (NIF) tiene permisos para acceder, a una sala determinada.

obtenerNiveles, obtendrá todos los niveles a los que tiene acceso un empleado (NIF).

asignarPermiso, asignará permisos a un empleado a una sala (NIF, Sala). Devolverá un booleano indicando si ha sido exitosa la operación.

eliminarPermiso, eliminará el permiso a un empleado/sala (NIF, Sala). Devolverá un booleano indicando si ha sido exitosa la operación.

Necesitaremos tener en BD una tabla con los permisos (NIF string, Sala string), donde tendremos una serie de NIFs de empleados y una serie de Salas (como en el ejemplo) para poder realizar los métodos citados anteriormente.

52878222R sala1
55652222R sala1
55652222R sala2
55555555R sala2

Utilidades:

validarNIF, el cuál validará un NIF como parámetro de entrada (string), devolviendo como resultado un booleano, comprobando que la letra asociado al número sea correcta, así como el formato presentado, el cual deberá ser:

52777342R (XXXXXXXX-L) donde X{0-9} y L {a-z,A-Z}

Validador para comprobar:

<http://www.migoia.com/website/utilidades/NIF.html>

validarIBAN, el cuál validará el IBAN bancario como parámetro de entrada (string), devolviendo como resultado un booleano, siguiendo la estructura de 24 dígitos, la cual debe de comprobarse que es correcta.

Ejemplo: *ES0690000001210123456789*, ES06 Código del país (dos signos) + código de control (dos dígitos) + Número de cuenta tradicional: (Ejemplo: 9000 0001 21 0123456789), 20 caracteres que identifican la cuenta.

Validador para comprobar (<http://www.ibancalculator.com/>).

validarNAFSS, este método se encargará de validar e número de afiliación de la seguridad social, partiendo como parámetro de entrada (String), devolverá un booleano para indicar si es correcto un número de seguridad social.

Ejemplo: 28 12345678 40. La cadena que se valida está compuesta de doce caracteres, «aabbabbbcc», donde los dos primeros («aa») son el indicativo provincial, los ocho siguientes («bbabbbbb») indican la cuenta de cotización de la persona, y los dos últimos («cc») son los dígitos de control.

Validador para comprobar:

<http://www.migoia.com/website/utilidades/NAF.html>

Consideraciones a tener en cuenta:

Necesitaremos disponer de una Base de Datos creada en MySql, con las tablas necesarias para el enunciado, siendo estas: *empleados* y *seguridad*.

Para la comprobación del correcto funcionamiento de los servicios web, emplearemos la herramienta: SoapUI.

2. **Se deberá crear una aplicación cliente, mediante cualquier lenguaje de programación (.NET, java, php, javascript, etc...), para la comprobación de dichos servicios.**
3. **A la hora de valorar la práctica para la máxima nota, se tendrán en cuenta las siguientes consideraciones:**

Todos los servicios contendrán un parámetro adicional de entrada, a los especificados a continuación, el cual será un *string*, llamado **SoapKey**, este contendrá una clave, la cual deberemos validar que existe en nuestra Base de Datos, para realizar o no, el método solicitado. (Deberemos de crear una tabla en BD llamada soapkey).

Adicionalmente a la salida, contarán todos los métodos de todos los servicios con un parámetro extra (string), para notificar posibles errores.

Por último, se valorará el uso de tipos de datos complejos.

Entrega

Se debe de entregar una pequeña memoria detallando la puesta en marcha y aspectos a destacar de la práctica, así como los proyectos de eclipse y proyecto cliente, por el control creado en el CV.

La fecha límite será el 18/02/2019. En caso de entregar la práctica en una fecha posterior, únicamente se podrá optar a un 5 como máxima nota.

El profesor corregirá la práctica durante la sesión de prácticas del 19/02/2019.