

<b>MTIS</b>	<b>Metodologías y Tecnologías de Integración de Sistemas</b>
	<b>Práctica 1</b>
	<b>WSDL: Interoperabilidad WSDL</b>

## 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.

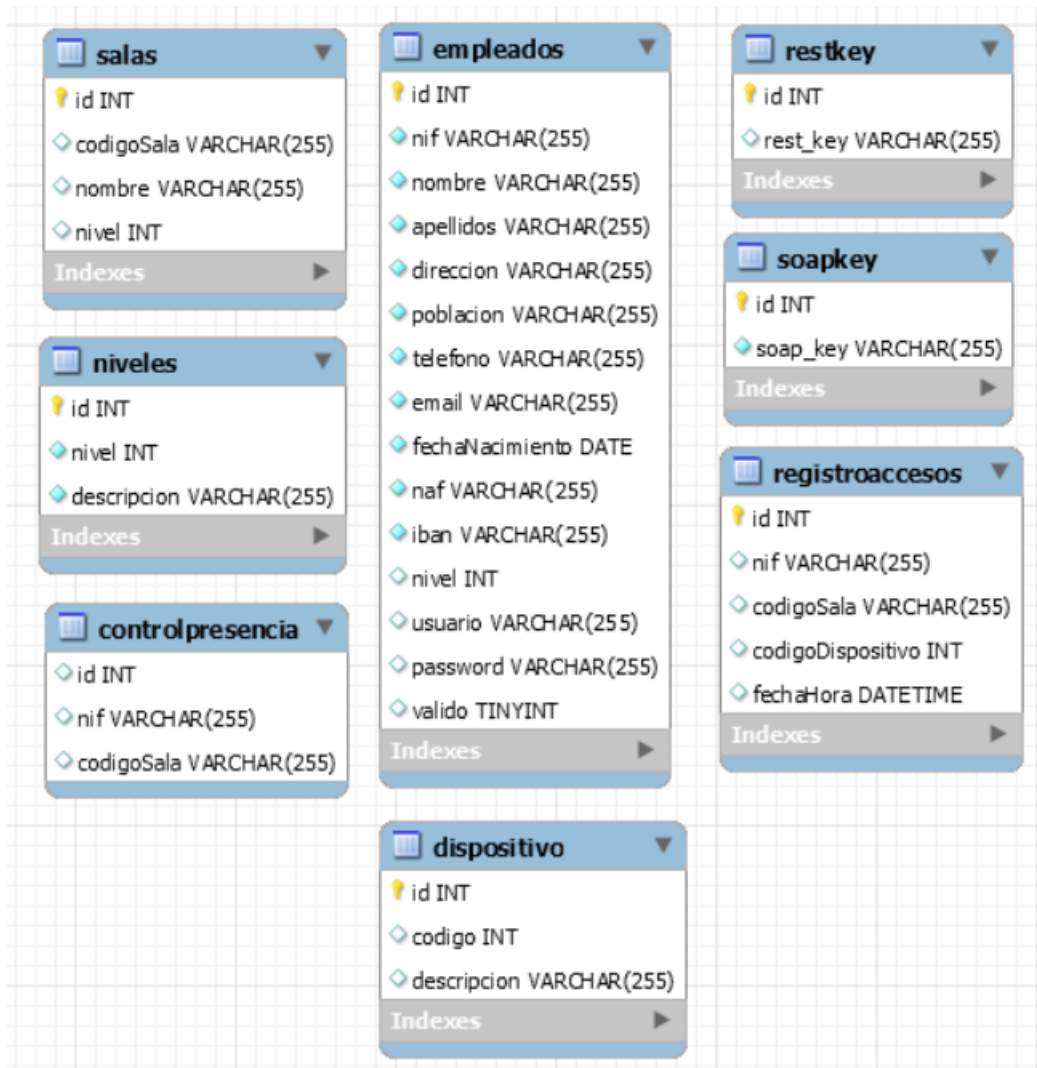
El Objetivo de esta práctica es crear un WSDL desde cero (top-down), sin necesidad de crear un servicio web. Esta técnica es la recomendada a la hora de crear servicios web, ya que con ello nos aseguramos la independencia de las herramientas. Lo apropiado es que la herramienta se adapte al WSDL, y no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

## Enunciado

Se pretende construir un sistema para control de un edificio inteligente, mediante el empleo de terminales de control de presencia y otros dispositivos de control. 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, gestionar los niveles de seguridad, las diferentes salas con los niveles, controlar la presencia de personas en salas, mantener un registro de accesos, así como realizar determinadas validaciones.

Nuestros Servicios Web se basarán en el siguiente esquema de BD (el script de BD se adjunta).



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

# 1. Crear cuatro contratos WSDL y sus correspondientes servicios web mediante el enfoque Top-down, con la ayuda del asistente de eclipse para dicha tarea.

El primer servicio web se denominará **Empleados** y debe de proporcionar los siguientes métodos:

**nuevo**, el cual creará un nuevo empleado en BD, a partir de la estructura indicada en BD.

**borrar**, se encargará de borrar de BD, el registro indicado por el *NIF*.

**modificar**, se encargará de actualizar en BD el registro, 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 segundo servicio se denominará **ControlAccesos** y debe proporcionar los siguientes métodos:

**registrar**, a partir de un *NIF*, un *codigosala* y un *codigodispositivo* debe de crear en BD un registro con la fecha-hora actual.

**consultar**, a partir de un *NIF*, *codigosala*, *codigodispositivo* y rango fechas, debe obtener todos los registros que cumplan dichos requisitos en BD.

El tercer servicio se denominará **ControlPresencia** y debe proporcionar los siguientes métodos:

**registrar**, a partir de un *NIF* y un *codigosala*, creará un nuevo registro en la tabla *controlpresencia*.

**eliminar**, a partir de un *NIF* y un *codigosala*, eliminará de BD un registro de la tabla *controlpresencia*.

**controlEmpleadosSala**, a partir de un *codigosala*, obtendrá los usuarios con presencia en una sala.

Por último, contaremos con un servicio de chequeos **Validaciones**, el cual contará con los siguientes métodos:

**validarNIF**, el cuál validará un NIF como parámetro de entrada (*string*), devolviendo como resultado un booleano, comprobando que la letra asociada 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 un 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, para ello se adjunta un script de BD.

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.

Adicionalmente a la salida (en caso de haberla), todos los métodos de todos los servicios contarán con un parámetro de salida booleano (indicará si se ha ejecutado correctamente dicho método) y un string (para notificar posibles errores durante la ejecución del método del servicio web).

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

## Entrega

Se deberá de entregar por la entrega habilitada para tal fin en Moodle:

- Una pequeña memoria detallando la puesta en marcha y aspectos a destacar de la práctica como puedan ser las implementaciones para la máxima nota.
- Proyectos de eclipse y proyecto cliente.

La fecha límite de entrega será el 21.02.2023. En caso de entregar la práctica en una fecha posterior, únicamente se podrá optar a un 5 como nota máxima, siendo la fecha límite para cualquier entrega el 24.05.2023.