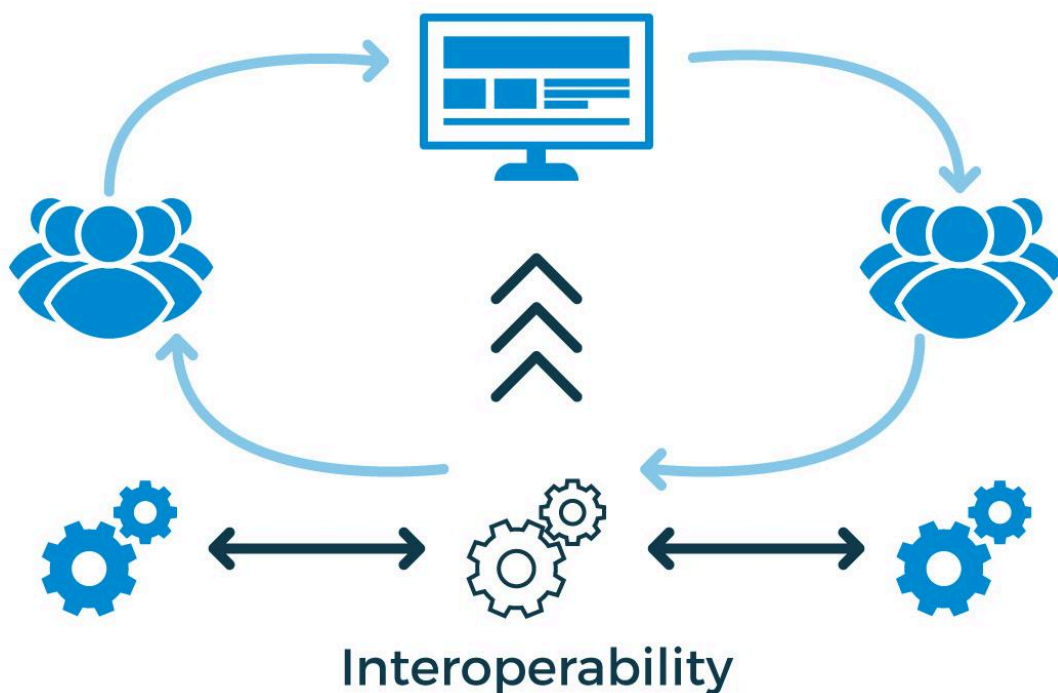


Metodologías y Tecnologías de Integración de Sistemas

Práctica 1

Interoperabilidad Servicios Web

Alumno: Georg Usin
DNI/NIE: X8174555X



ÍNDICE

Introducción	3
Base de datos	4
Servicios SOAP	5
Servicios REST	6
Cliente	7

Introducción

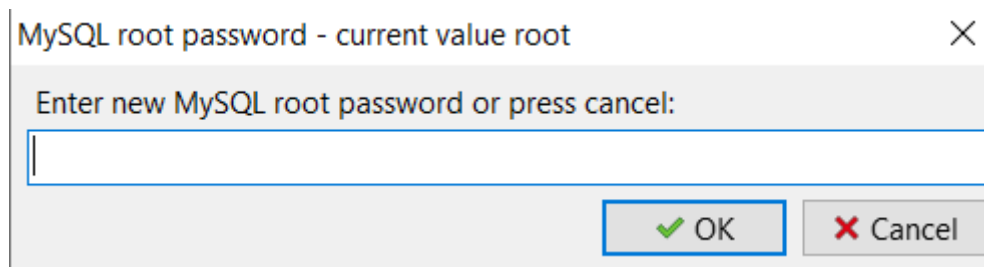
En este proyecto, se ha desarrollado una API REST con .NET y una API SOAP, ambas operando sobre la misma base de datos llamada mtis, la cual se ejecuta en el puerto 3307 bajo un servidor MySQL en Uniform Server Zero.

Para interactuar con estas APIs, se ha implementado un cliente en C# .NET, que permite consumir ambos servicios y proporciona una aplicación de escritorio basada en terminal. Esta aplicación ofrece una interfaz sencilla y eficiente para realizar peticiones tanto a la API REST como a la API SOAP, facilitando la interacción con los servicios de una manera estructurada y organizada.

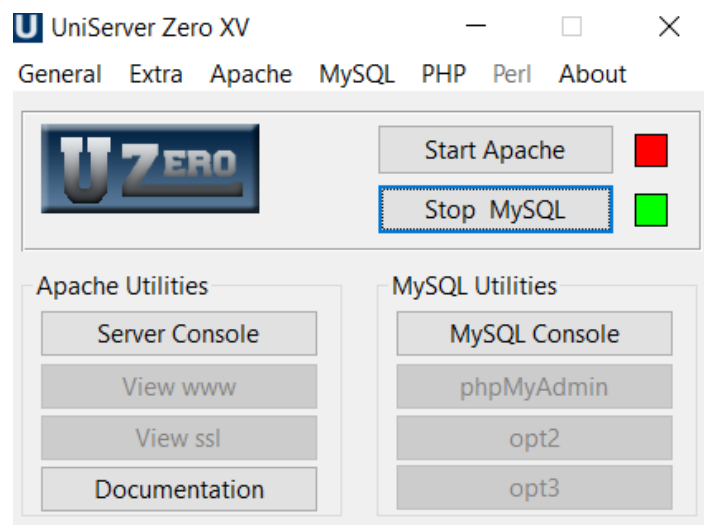
El correcto funcionamiento de la API REST se ha verificado mediante Swagger y OpenAPI, permitiendo una documentación interactiva y pruebas directas de los endpoints. Por otro lado, la API SOAP ha sido diseñada utilizando contratos WSDL, y su correcto funcionamiento se ha validado con SOAPUI, asegurando la compatibilidad con los estándares de comunicación basados en XML.

Base de datos

Primero de todo debemos iniciar la base de datos.



Iniciamos sesión en UniServer como root con la contraseña root y activamos la opción StartMySQL.



Luego ingresamos en la consola de MySQL y para poder iterar sobre nuestra base de datos ejecutamos el comando USE mtis;

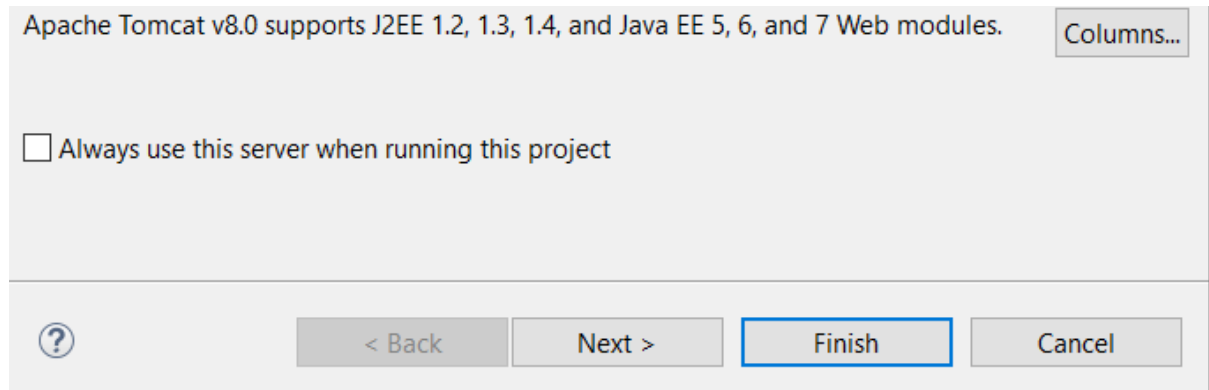
```
mysql> USE mtis;  
Database changed  
mysql>
```

Una vez hecho esto ya tenemos la base de datos mySQL lista para trabajar.

Servicios SOAP

Para poner en marcha nuestro servidor TomCat para poder utilizar los servicios de SOAP simplemente tenemos que entrar en Eclipse y ejecutar el servidor.

Clic Derecho en EdificioInteligente -> Run As -> Run on server y le damos al botón Finish si nos sale el menú para reiniciar el servidor.



Hecho esto ya está corriendo el servidor Tomcat y podemos observar que el servicio está Activo en la siguiente captura:

Empleados

Service Description : Empleados

Service EPR : <http://localhost:8080/EdificioInteligente/services/Empleados>

Service Status : Active

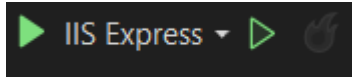
Available Operations

- modificar
- consultar
- nuevo
- borrar

Con el servidor activo ya podemos utilizar los servicios SOAP, podríamos comprobar el funcionamiento con SoapUI pero no lo voy a hacer en esta mini guía de despliegue, pues ya se comprobará el funcionamiento en el cliente.

Servicios REST

Para ejecutar el servidor REST para poder utilizar los servicios definidos en la API, simplemente le tenemos que dar al botón de ejecutar en el visual studio y se pone en marcha.



Si todo va bien se nos abrirá esta pestaña en el navegador mostrando los distintos endpoints disponibles.

API Edificio Inteligente 1.0.0 OAS3
/swagger/1.0.0/swagger.json
API Edificio Inteligente (ASP.NET Core 7.0)
[Swagger Codegen Contributors - Website](#)

Servers
/api

DefaultApi ^

DELETE

/dispositivos/{codigo} ✓

GET

/dispositivos/{codigo} ✓

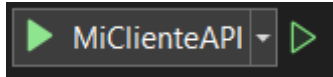
PUT

/dispositivos/{codigo} ✓

Al igual que antes, se comprobará el correcto funcionamiento de los servicios a través del cliente.

Cliente

Para ejecutar el cliente para poder utilizar los servicios REST y SOAP, simplemente le tenemos que dar al botón de ejecutar en el visual studio y se pone en marcha.



Si todo va bien, se abrirá una terminal con un menú un tanto rudimentario pero no por ello peor.

```
--- Menú Principal ---
1. Dispositivos
2. Salas
3. Niveles
4. Notificar
5. Empleados
6. ControlAccesos
7. ControlPresencia
8. Salir
Seleccione una opción:
```

Probaremos uno de los servicios definidos en REST, vamos a crear un nuevo Dispositivo, para ello seleccionamos la primera opción.

```
--- Menú Dispositivos ---
1. Crear Dispositivo
2. Consultar Dispositivo
3. Modificar Dispositivo
4. Borrar Dispositivo
5. Volver al Menú Principal
Seleccione una opción:
```

Se nos mostrará un menú secundario orientado a Dispositivos y volveremos a seleccionar la primera opción.

Ingresamos los datos uno a uno y como se puede observar se muestra un mensaje de confirmación.

```
Seleccione una opción: 1
Ingrese código del dispositivo: 152
Ingrese descripción del dispositivo: Practica1
Dispositivo creado correctamente.
```

Para confirmar que se ha creado correctamente podemos hacer una pequeña select en nuestra base de datos y vemos que efectivamente se ha creado el nuevo dispositivo.

```
mysql> select * from dispositivo where codigo = 152;
+----+-----+-----+
| id | codigo | descripcion |
+----+-----+-----+
| 11 | 152 | Practica1 |
+----+-----+-----+
1 row in set (0.01 sec)
```

Ahora comprobaremos el funcionamiento de un servicio SOAP. Vamos a probar a crear un nuevo Control Acceso. Rellenamos los datos uno a uno y enviamos la petición.

```
--- Registrar Acceso ---
Ingrese el NIF del empleado: 13345678A
Ingrese el código de la sala: 101
Ingrese el código del dispositivo: 152
<?xml version="1.0" encoding="utf-16"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns1:registrarResponse xmlns:ns1="http://www.example.org/ControlAccesos/">
      <out xmlns="">Acceso registrado correctamente</out>
    </ns1:registrarResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Recibiremos un XML con la respuesta. Observamos que ha sido creado correctamente, para confirmarlo haremos la misma prueba que antes y lo consultaremos en la base de datos.

```
mysql> select * from registroaccesos where idDispositivo = 11;
+-----+-----+-----+-----+-----+
| id | idEmpleado | idSala | idDispositivo | fechaHora          |
+-----+-----+-----+-----+-----+
| 5 | 6 | 1 | 11 | 2025-03-05 14:05:35 |
+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```