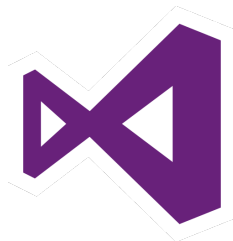


Universidad de Oviedo
Escuela Politécnica de Ingeniería de Gijón

Grado en Tecnologías y Servicios de la Telecomunicación
Aplicaciones Telemáticas

Sesión 1.2 - Especificación, diseño e implementación de un protocolo

Marco Martínez Ávila



1. Descripción de la práctica

En esta práctica se ampliará la funcionalidad del servicio web implementado, añadiendo la posibilidad de resolver sistemas de ecuaciones de 2 y 3 incógnitas. Además, se consumirá ese servicio desde un cliente Java, utilizando el IDE Eclipse.

2. Descripción del trabajo realizado

La inclusión de la nueva funcionalidad, la resolución de sistemas de ecuaciones, tiene su origen en la modificación de la interfaz “IMath.cs”. Es necesario definir un nuevo *OperationContract* (método), de forma que los clientes puedan utilizarlo para consumir esta nueva opción.

```
[OperationContract]
double[] SolveSystem(double[] system);
```

La implementación de la funcionalidad necesaria se realiza por medio de la biblioteca “EcSystem”. No se encontró ninguna biblioteca externa que permitiera añadir esta opción de forma directa, por lo que se optó por modificar un fragmento de código que permitía resolver ecuaciones utilizando el método de Cramer para adaptarlo a los requerimientos de esta práctica.

Se decidió que el argumento que recibe la función “SolveSystem” sea un array de *double*, conteniendo los valores del sistema de ecuaciones sin resolver. A su vez, el método devolverá otro array del mismo tipo, conteniendo los valores para las variables que correspondan.

Posteriormente, y una vez corroborado el buen funcionamiento del servicio, se amplió la interfaz gráfica creada con anterioridad, de forma que incluyera las opciones necesarias para resolver sistemas de ecuaciones.

Por su parte, el consumo del servicio utilizando otro lenguaje de programación se consiguió utilizando Java y el IDE Eclipse. Tras instalar unos paquetes necesarios utilizando el instalador de este último, fue necesario añadir un “Web Service Client” al proyecto existente. Para poder finalizar con éxito este procedimiento, fue necesario añadir una referencia al archivo WSDL, obtenido a partir de la pantalla de descripción del servicio.

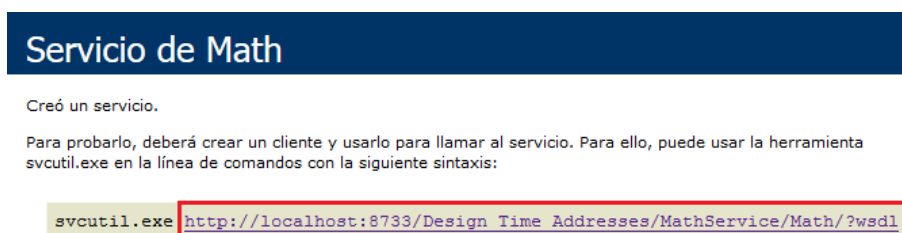


Figura 1: URL a través de la que obtener el archivo WSDL.

De esta forma, se generaron en Eclipse las clases necesarias para poder hacer uso del servicio web. Instanciando una de ellas, en cuyo constructor se creaba el *endpoint* que permitía acceder al mismo. Gracias a los métodos existentes en esa clase, fue posible hacer uso de las funciones que determinaban si un número era primo, o resolvían el sistema de ecuaciones.

La parte más complicada al consumir el servicio desde Java, además de la cantidad de información ambigua y/o errónea encontrada en Internet, fue la cantidad de clases generadas a partir del archivo WSDL. Al contrario que con el Visual Studio, que sólo generaba 2, y con un código mucho más sencillo de entender, el generado en Eclipse resultaba de comprensión mucho más difícil. Inicialmente, no teníamos claro como acceder a las funciones necesarias, ya que parecía que se repetían en distintos archivos, y las pruebas iniciales resultaron en errores de todo tipo. Con un poco de investigación, se llegó a la solución aparentemente correcta, instanciando la clase “IMathProxy” desde otra clase llamada “Consumer”. De esta forma, y siguiendo el flujo de ejecución del programa, fue posible acceder a los métodos implementados en el servicio web.

3. Conclusiones

La resolución de esta práctica conllevó más tiempo del que, visto una vez hecha, parece necesario. Ambas ampliaciones requirieron bastante investigación y búsquedas en distintas páginas. Además, si bien la primera supuso un problema mucho más directo, el consumo del servicio desde Java era algo que no habíamos

estudiado con anterioridad. Por lo tanto, aunque no había más que replicar el funcionamiento conseguido con el Visual Studio, este inconveniente y la aparición de constantes errores dificultó bastante la consecución de un resultado satisfactorio.