

Memoria de práctica

Calculadora

(DSD P2S2)



Román Larrosa Lewandowska
marzo 2020

Introducción

En el siguiente documento se detalla la forma en la que he resuelto la práctica 2 sesión 2 de la asignatura DSD, en la que hay que desarrollar un servicio RCP con la tecnología thrift.

Archivo .thrift

Para generar los archivos necesarios, primero he dado formato a las operaciones que tendrá disponibles el servidor de la calculadora, quedando un archivo tal que así:

```
1  service Calculadora {  
2      void ping(),  
3      double suma(1:double num1, 2:double num2),  
4      double sub(1:double num1, 2:double num2),  
5      double mult(1:double num1, 2:double num2),  
6      double div(1:double num1, 2:double num2),  
7  
8      list<double> sumVec(1:list<double> vec1, 2:list<double> vec2),  
9      list<double> subVec(1:list<double> vec1, 2:list<double> vec2),  
10     double escVec(1:list<double> vec1, 2:list<double> vec2),  
11  
12 }
```

El servicio implementa las operaciones básicas, y además la suma, resta y producto escalar de dos vectores de cualquier dimensión.

Una vez acabado, procedo a generar los archivos necesarios mediante la orden `thrift -r --gen py calculadora.thrift`

Esto genera los archivos que necesitaremos para implementar un cliente y un servidor del servicio en python

Archivo cliente.py

En este archivo se maneja el funcionamiento del programa y es desde el cual se realizarán las llamadas a procedimiento remoto. Aquí es donde se determina que operación es la que hay que pedirle al servidor. Si no se establece ninguna operación básica en la llamada del cliente éste ejecutará una operación de cada una de las disponibles para probar el funcionamiento.

Archivo servidor.py

En este archivo lo único que hay que rellenar es el comportamiento de cada una de las operaciones disponibles que se determinaron en calculadora.thrift

■ ■ ■

Para la prueba del servicio, abrimos el servicio del servidor, y en un terminal distinto ejecutamos el cliente de la siguiente forma:

Si queremos realizar una operación sencilla: `python cliente.py <arg1> <operator> <arg2>`

Si queremos probar todas las operaciones: `python cliente.py`