

Apache Thrift

PRÁCTICA 2.2

CARMEN CHUNYIN FERNÁNDEZ NÚÑEZ

1 INTRODUCCIÓN

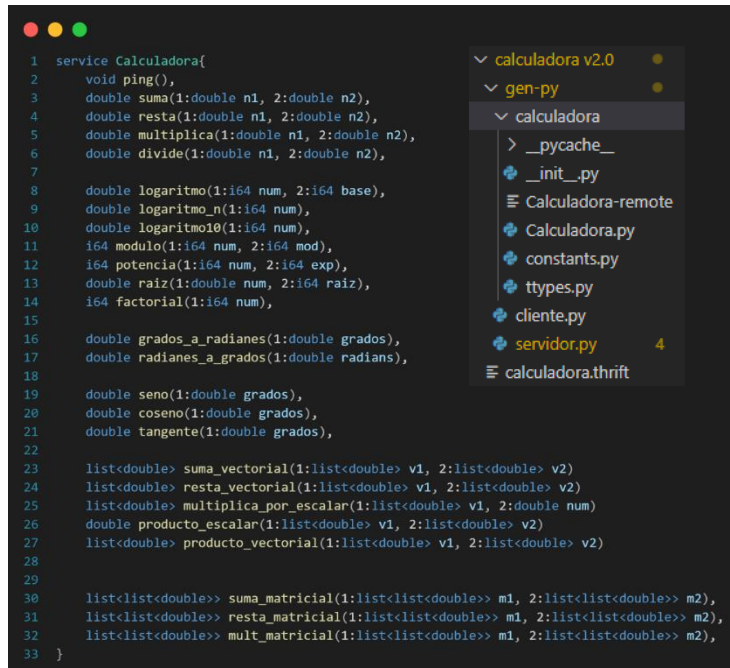
Esta práctica consiste en el desarrollo de un programa distribuido, para ser más exactos una calculadora, utilizando Apache Thrift. Este programa realizará desde varias operaciones básicas como sumas, restas, multiplicaciones y divisiones hasta operaciones con estructuras más complejas como son las matrices, además de la implementación del mismo en diferentes lenguajes de programación.

2 EXPLICACIÓN DE LA SOLUCIÓN

2.1 CALCULADORA.THRIFT

En este archivo se definen las rutinas a las que se puede acceder remotamente, escrito en el lenguaje Apache Thrift.

Como se puede observar, hemos definido una variedad de operaciones desde algunas más básicas hasta operaciones con vectores y matrices. A partir de este archivo, con la orden “thrift-gen py calculadora.thrift” se generan el resto de archivos necesarios. Además, añadiremos el cliente y el servidor



2.2 SERVIDOR.PY

El archivo del servidor, deberá contener todas las funciones definidas en el archivo base. Por un lado, tenemos la parte donde definimos el handler y lanzamos el servidor. Por otro lado, en el handler definiremos las operaciones y daremos el resultado.

```
if __name__ == "__main__":
    handler = CalculadoraHandler()
    processor = Calculadora.Processor(handler)
    transport = TSocket.TServerSocket(host="127.0.0.1", port=9090)
    tfactory = TTransport.TBufferedTransportFactory()
    pfactory = TBinaryProtocol.TBinaryProtocolFactory()

    server = TServer.TSimpleServer(processor, transport, tfactory, pfactory)

    print("iniciando servidor...")
    server.serve()
    print("fin")
```

2.3 CLIENTE.PY

En este archivo, recogeremos el cálculo que nos piden desde teclado y se llama al servidor. Para ello debemos definir primero el socket de transporte y abrirlo. Una vez terminadas las operaciones, lo cerraremos.

```
transport = TSocket.TSocket("localhost", 9090)
transport = TTransport.TBufferedTransport(transport)
protocol = TBinaryProtocol.TBinaryProtocol(transport)

client = Calculadora.Client(protocol)

transport.open()
```

Como en la práctica anterior, el cuerpo de nuestro cliente serán una serie de elecciones que nos llevarán a la operación deseada mediante switches. Aunque en este caso he definido cada "submenú" en una función por separado (realmente no tiene mucha importancia).

```
while True:
    menu_principal()
    opcion = input()
    print('')

    #En python3 introdujeron el match-case a modo de switch-case
    match opcion:
        case '1':
            operaciones_basicas()
        case '2':
            operaciones_avanzadas()
        case '3':
            operaciones_trigonometricas()
        case '4':
            operaciones_vectoriales()
        case '5':
            operaciones_matriciales()
        case '6':
            print('Saliendo del programa...\n')
            break
        case _:
            print("Opcion no valida.\n")

transport.close()
```

```
def operaciones_matriciales():
    while True:
        print("ELIJA UNA OPERACION:")
        print("\t1. Suma Matricial")
        print("\t2. Resta Matricial")
        print("\t3. Multiplicacion Matricial")
        print("\t4. Volver al menu principal")
        operacion = input()
        print('')

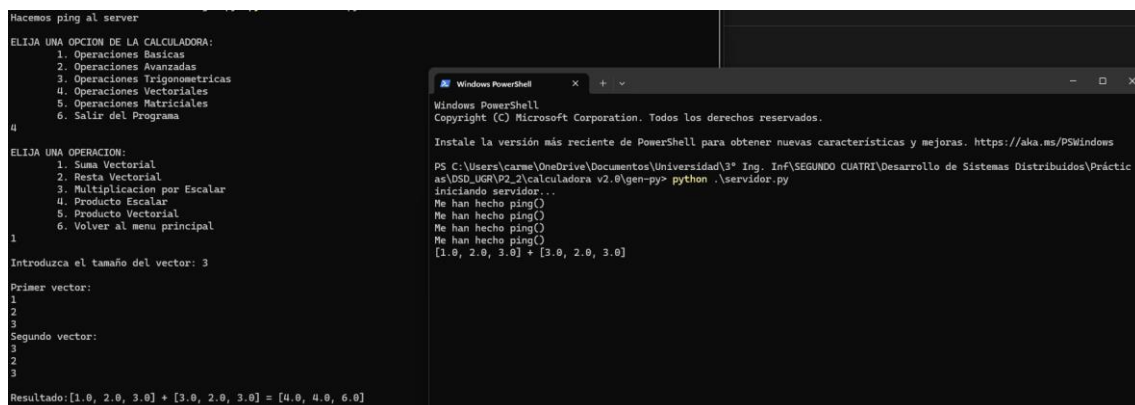
        match operacion:
            case '1': # SUMA
                fil = int(input('Matriz num filas: '))
                col = int(input('Matriz num columnas: '))
                print('')

                m1 = [[0 for j in range(col)] for i in range(fil)]
                m2 = [[0 for j in range(col)] for i in range(fil)]

                print('Primera matriz: ')
                rellenar_matriz(m1)
                print('Segunda matriz: ')
                rellenar_matriz(m2)
                print('')
```

3 FUNCIONAMIENTO

El funcionamiento es exactamente el mismo que en la práctica anterior, llamamos a cada programa en una terminal diferente y el cliente nos pedirá que escojamos que operación queremos realizar y, a continuación, nos pedirá los datos necesarios para realizarla.



```
Hacemos ping al server
ELIJA UNA Opcion DE LA CALCULADORA:
1. Operaciones Basicas
2. Operaciones Avanzadas
3. Operaciones Trigonometricas
4. Operaciones Vectoriales
5. Operaciones Matriciales
6. Salir del Programa
4

ELIJA UNA OPERACION:
1. Suma Vectorial
2. Resta Vectorial
3. Multiplicacion por Escalar
4. Producto Escalar
5. Producto Vectorial
6. Volver al menu principal
1

Introduzca el tamaño del vector: 3

Primer vector:
1
2
3
Segundo vector:
1
2
3

Resultado: [1.0, 2.0, 3.0] + [3.0, 2.0, 3.0] = [4.0, 4.0, 6.0]
```