

# Práctica 2 Apache Thrift

Nombre: Ana López Mohedano

Curso: 3º / Grupo: 2

En esta práctica se realiza una calculadora que recibe un tipo de operación, envía los datos al servidor, que procesará los datos y hará los cálculos necesarios, y devolverá el resultado al cliente.

En este caso, usaremos Apache Thrift que genera automáticamente los ficheros para realizar todo el proceso. Se implementa solamente el cliente y el servidor, en este caso en python. Las operaciones que puede hacer pueden ser con reales o con vectores, y en ambos puede hacer suma, resta, multiplicación y división. Con los vectores, hace operaciones con dos vectores del mismo tamaño y el resultado también es un vector del mismo tamaño.

## Fichero .thrift

En calculadora.thrift hemos definido las funciones que realizará la calculadora, que serán las descritas anteriormente, suma, resta, multiplicación y división para reales y para vectores.

```
1  calculadora.thrift
2  service Calculadora{
3      void ping(),
4      i32 suma(1:i32 num1, 2:i32 num2),
5      i32 resta(1:i32 num1, 2:i32 num2),
6      i32 multiplicacion(1:i32 num1, 2:i32 num2),
7      i32 division(1:i32 num1, 2:i32 num2),
8      list<double> sumaVectores(1:list<double> n1, 2:list<double> n2),
9      list<double> restaVectores(1:list<double> n1, 2:list<double> n2),
10     list<double> multiplicaVectores(1:list<double> n1, 2:list<double> n2),
11     list<double> divideVectores(1:list<double> n1, 2:list<double> n2)
12 }
13
```

## Cliente

Primero, tendremos un bucle while en que estaremos mientras el programa se esté ejecutando, a no ser que se elija la opción 3 que es salir del programa. Si se elige la 1 o la 2, operaciones con números reales o con vectores, entramos en otro bucle while en el que mostraremos otro menú.

```

while(salir == False):
    print("\nMenu:")
    print("1. Operaciones con numeros reales")
    print("2. Operaciones con vectores")
    print("3. Salir del programa\n")
    seleccion = input("Seleccion: ")

    v1 = []
    v2 = []
    v_resultado = []
    resultado = 0

    # Si no se ha elegido salir, se entra en el bucle
    if(seleccion != 3):
        menu = False
    else: # Si se elige salir, se sale del programa entero
        menu = True
        salir = True

```

Con la primera opción, haremos operaciones con números reales. Elegimos una opción y esa función será realizada por el servidor, y cuyo resultado será devuelto al cliente. Si se elige la opción 5 se sale del bucle while del menú y vuelve al menú principal, al while de la ejecución del programa.

```

41     while(menu == False):
42         if(seleccion == 1):
43             # Menu para elegir operacion
44             print("\nOperaciones con reales:")
45             print("1. Suma.")
46             print("2. Resta.")
47             print("3. Multiplicacion")
48             print("4. Division")
49             print("5. Volver al menu principal\n")
50             opcion = input("Que quiere realizar?: ")
51
52             # Introducir valores
53             if(opcion != 5):
54                 n1 = input("Ingrese el primer operando: ")
55                 n2 = input("Ingrese el segundo operando: ")
56
57                 if(opcion == 1):
58                     resultado = client.suma(n1, n2)
59                 if(opcion == 2):
60                     resultado = client.resta(n1, n2)
61                 if(opcion == 3):
62                     resultado = client.multiplicacion(n1, n2)
63                 if(opcion == 4):
64                     resultado = client.division(n1, n2)
65                 if(opcion == 5):
66                     menu = True
67

```

Las operaciones con vectores funcionan igual, con su menú correspondiente y sus funciones según la elección. Se hace un manejo de errores por si se divide entre 0.

```
68 elif(seleccion == 2):
69     # Menu para elegir operacion
70     print("\nOperaciones con vectores:")
71     print("1. Suma.")
72     print("2. Resta.")
73     print("3. Multiplicacion")
74     print("4. Division")
75     print("5. Volver al menu principal\n")
76     opcion = input("Que desea realizar?: ")
77
78     # Introducir valores
79     if(opcion != 5):
80         print("Introduzca el primer vector.")
81         for i in range(5):
82             valor = input("Introduce el valor {} del vector: ".format(i+1))
83             valor_float = float(valor)
84             v1.append(valor_float)
85
86         print("Introduzca el segundo vector.")
87         for i in range(5):
88             valor = input("Introduce el valor {} del vector: ".format(i+1))
89             valor_float = float(valor)
90             v2.append(valor_float)
91
92
93     # Realizar operacion con los operandos
94     if(opcion == 1):
95         v_resultado = client.sumaVectores(v1, v2)
96     if(opcion == 2):
97         v_resultado = client.restaVectores(v1, v2)
98     if(opcion == 3):
99         v_resultado = client.multiplicaVectores(v1, v2)
100     if(opcion == 4):
101         try:
102             v_resultado = client.divideVectores(v1, v2)
103         except ValueError as error:
104             print("Se produjo un error: ", error)
105             resultado = None
106     if(opcion == 5):
107         menu = True
```

Dentro del while del menú, procesaremos los resultados y los mostraremos por pantalla, según eran números reales o vectores.

```
113 if(menu == False and salir == False):
114     if(seleccion == 1): # si la seleccion eran numeros reales
115         # Imprimir operandos y resultado
116         print("\nLos dos operandos son: " + str(n1) + ", " + str(n2))
117         print("El resultado es: " + str(resultado))
118     elif(seleccion == 2): # si la seleccion eran vectores
119         print("\nPrimer vector: " + str(v1))
120         print("Segundo vector: " + str(v2))
121         if (resultado == None):
122             print("No se pudo calcular, no se puede dividir entre cero")
123         else:
124             print("Vector resultado: " + str(v_resultado))
125
```

# Servidor

En el servidor definimos las funciones de las operaciones que queremos realizar. Para los números reales:

```
27     def suma(self, n1, n2):
28         print("sumando " + str(n1) + " con " + str(n2))
29         return n1 + n2
30
31     def resta(self, n1, n2):
32         print("restando " + str(n1) + " con " + str(n2))
33         return n1 - n2
34
35     def multiplicacion(self, n1, n2):
36         print("multiplicando " + str(n1) + " con " + str(n2))
37         return n1 * n2
38
39     def division(self, n1, n2):
40         print("dividiendo " + str(n1) + " con " + str(n2))
41         if(n2 == 0):
42             resultado = -1
43         else:
44             resultado = n1 / n2
45         return resultado
46
```

Y las operaciones con vectores:

```
47     def sumaVectores(self, n1, n2):
48         print("sumando vectores...")
49         v = []
50         for i in range(len(n1)):
51             v.append(n1[i] + n2[i])
52         return v
53
54     def restaVectores(self, n1, n2):
55         print("restando vectores...")
56         v = []
57         for i in range(len(n1)):
58             v.append(n1[i] - n2[i])
59         return v
60
61     def multiplicaVectores(self, n1, n2):
62         print("multiplicando vectores...")
63         v = []
64         for i in range(len(n1)):
65             v.append(n1[i] * n2[i])
66         return v
67
68     def divideVectores(self, n1, n2):
69         print("sumando vectores...")
70         v = []
71         try:
72             for i in range(len(n1)):
73                 if(n2[i] == 0):
74                     raise ValueError("NO SE PUEDE DIVIDIR ENTRE 0")
75                 else:
76                     v.append(n1[i] / n2[i])
77         except ValueError as e:
78             raise TApplicationException(TApplicationException.INTERNAL_ERROR, str(e))
79         return v
80
```

# Ejecución

El principio del programa es así:

```
ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python cliente.py
hacemos ping al server

Menu:
1. Operaciones con numeros reales
2. Operaciones con vectores
3. Salir del programa

Seleccion: █

ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python servidor.py
iniciando servidor...
me han hecho ping()
█
```

Elegimos operaciones con números reales, y dentro de ahí elegimos por ejemplo la multiplicación.

```
Menu:
1. Operaciones con numeros reales
2. Operaciones con vectores
3. Salir del programa

Seleccion: 1

Operaciones con reales:
1. Suma.
2. Resta.
3. Multiplicacion
4. Division
5. Volver al menu principal

Que quiere realizar?: █

ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python servidor.py
iniciando servidor...
me han hecho ping()
█
```

Introducimos 25 y 4, y su correspondiente resultado, 100. Volvemos al menú de operaciones reales, pero volvemos al menú principal para hacer operaciones con vectores.

```
Que quiere realizar?: 3
Ingrese el primer operando: 25
Ingrese el segundo operando: 4

Los dos operandos son: 25, 4
El resultado es: 100

Operaciones con reales:
1. Suma.
2. Resta.
3. Multiplicacion
4. Division
5. Volver al menu principal

Que quiere realizar?: █

ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python servidor.py
iniciando servidor...
me han hecho ping()
multiplicando 25 con 4
█
```

Ahora elegimos vectores, y la operación resta.

```
Operaciones con reales:
1. Suma.
2. Resta.
3. Multiplicacion
4. Division
5. Volver al menu principal

Que quiere realizar?: 5

Menu:
1. Operaciones con numeros reales
2. Operaciones con vectores
3. Salir del programa

Seleccion: █

ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python servidor.py
iniciando servidor...
me han hecho ping()
multiplicando 25 con 4
█
```

```
PROBLEMAS 9 SALIDA TERMINAL CONSOLA DE DEPURACION
1. Suma.
2. Resta.
3. Multiplicacion
4. Division
5. Volver al menu principal

Que desea realizar?: 2
Introduzca el primer vector.
Introduce el valor 1 del vector: 5
Introduce el valor 2 del vector: 5
Introduce el valor 3 del vector: 5
Introduce el valor 4 del vector: 5
Introduce el valor 5 del vector: 5
Introduzca el segundo vector.
Introduce el valor 1 del vector: 1
```

```
ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python servidor.py
iniciando servidor...
me han hecho ping()
multiplicando 25 con 4
[]
```

Introduciremos el vector [5, 5, 5, 5, 5] y lo restaremos con el vector [1, 2, 3, 4, 5].  
Aquí vemos el resultado:

```
Introduzca el segundo vector.
Introduce el valor 1 del vector: 1
Introduce el valor 2 del vector: 2
Introduce el valor 3 del vector: 3
Introduce el valor 4 del vector: 4
Introduce el valor 5 del vector: 5

Primer vector: [5.0, 5.0, 5.0, 5.0, 5.0]
Segundo vector: [1.0, 2.0, 3.0, 4.0, 5.0]
Vector resultado: [4.0, 3.0, 2.0, 1.0, 0.0]

Operaciones con vectores:
1. Suma.
2. Resta.
3. Multiplicacion
4. Division
5. Volver al menu principal

Que desea realizar?:
```

```
ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python servidor.py
iniciando servidor...
me han hecho ping()
multiplicando 25 con 4
restando vectores...
[]
```

Ahora para salir del programa, pulsaremos 5 para volver al menú principal, y una vez allí pulsaremos 3 para salir del programa.

```
Segundo vector: [1.0, 2.0, 3.0, 4.0, 5.0]
Vector resultado: [4.0, 3.0, 2.0, 1.0, 0.0]

Operaciones con vectores:
1. Suma.
2. Resta.
3. Multiplicacion
4. Division
5. Volver al menu principal

Que desea realizar?: 5

Menu:
1. Operaciones con numeros reales
2. Operaciones con vectores
3. Salir del programa

Seleccion: 3
ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$
```

```
ana@ana:~/UNI/2022_2023/DSD/Practicas_DSD/Practica2-2_DSD/gen-py$ python servidor.py
iniciando servidor...
me han hecho ping()
multiplicando 25 con 4
restando vectores...
[]
```