

PRÁCTICA 2: Apache Thrift

Pablo Jiménez Jiménez

Granada, 17 de abril de 2020

Índice

1. Tarea realizada	1
2. Invocación del programa	2
3. Capturas de pantalla	3
3.1. Python	3
3.2. Java	4

1. Tarea realizada

He implementado una calculadora que hace operaciones básicas con números: sumar, restar, multiplicar, dividir, factorial y la potencia de un número. Por otro lado puede llevar a cabo operaciones con vectores: suma, producto vectorial, producto escalar, calcula la media de un vector y multiplicar una matriz por un número real (producto escalar de matrices).

- Los vectores se escriben en un fichero. Véase `v1.txt`
- Las matrices se escriben en un fichero separando las filas con comas. Véase `matriz.txt`

El programa ha sido escrito tanto en `Python3` como en `Java`. Para `Java` se ha utilizado el IDE `Netbeans` para gestionar cómodamente las bibliotecas de `Apache Thrift` a continuación se detalla la forma de ejecución y se adjuntan capturas de la ejecución. Para ejecutar los programas siempre hay que ejecutar el servidor escrito en `Python`.

2. Invocación del programa

Para ejecutar el programa escrito en Python:

```
$ python3 servidor.py
```

y en otra ventana arrancamos el cliente

```
$ python3 cliente.py <arg1> <operador> <arg2>
```

Operaciones con números	
Operación	Orden
Suma	<invocación> <n> + <n>
Resta	<invocación> <n> - <n>
Multiplicación	<invocación> <n> x <n>
División	<invocación> <n> / <n>
Factorial	<invocación> <n> !
Potencia	<invocación> <n> ^ <n>

Operaciones con vectores	
Operación	Orden
Suma	<invocación> <file> +v <file>
Producto escalar	<invocación> <file> . <file>
Producto vectorial	<invocación> <file> xv <file>
Media aritmética	<invocación> <file> m
Producto escalar	<invocación> <file> .m <n>

Siendo <n> un número real, y <file> un fichero `txt` con el contenido del vector/matriz en el formato indicado anteriormente. El producto escalar es válido tanto para vectores como matrices.

3. Capturas de pantalla

3.1. Python

```
→ Calculadora git:(master) X python3 cliente.py 12 + 10
12.0 + 10.0 = 22.0
→ Calculadora git:(master) X python3 cliente.py 12 - 10
12.0 - 10.0 = 2.0
→ Calculadora git:(master) X python3 cliente.py 12 x 10
12.0 x 10.0 = 120.0
→ Calculadora git:(master) X python3 cliente.py 12 / 10
12.0 / 10.0 = 1.2
→ Calculadora git:(master) X python3 cliente.py 12 !
12.0 ! = 479001600
→ Calculadora git:(master) X python3 cliente.py 12 ^ 10
12.0 ^ 10.0 = 61917364224.0
→ Calculadora git:(master) X
```

Figura 1: Calculadora operaciones básicas Python

```
→ Calculadora git:(master) X python3 cliente.py v1.txt +v v2.txt
[2.0, 3.0, 4.0, 5.0] + [2.0, 3.0, 4.0, 5.0] = [4.0, 6.0, 8.0, 10.0]
→ Calculadora git:(master) X python3 cliente.py v1.txt . v2.txt
[2.0, 3.0, 4.0, 5.0] . [2.0, 3.0, 4.0, 5.0] = 54.0
→ Calculadora git:(master) X python3 cliente.py v1.txt xv v2.txt
[2.0, 3.0, 4.0, 5.0] x [2.0, 3.0, 4.0, 5.0] = [4.0, 9.0, 16.0, 25.0]
→ Calculadora git:(master) X python3 cliente.py v1.txt m
[2.0, 3.0, 4.0, 5.0] m = 3.5
→ Calculadora git:(master) X python3 cliente.py v1.txt .m 3
[[2.0, 3.0, 4.0, 5.0]] x 3.0 = [[6.0]]
→ Calculadora git:(master) X python3 cliente.py matriz.txt .m 3
[[1.0, 2.0, 3.0], [4.0, 5.0, 6.0], [8.0, 9.0, 0.0]] x 3.0 = [[3.0, 6.0, 9.0], [12.0, 15.0, 18.0], [24.0, 27.0, 0.0]]
→ Calculadora git:(master) X
```

Figura 2: Calculadora operaciones con vectores Python

```

→ Calculadora git:(master) X python3 cliente.py matriz.txt tr 07
instrucciones incorrectas...
Python3 cliente.py <n> + <n>
Python3 cliente.py <n> - <n>
Python3 cliente.py <n> x <n>
Python3 cliente.py <n> / <n>
Python3 cliente.py <n> !
Python3 cliente.py <file> +v <file>
Python3 cliente.py <file> xv <file>
Python3 cliente.py <file> . <file>
Python3 cliente.py <file> m:
Python3 cliente.py <file> .m <n> float(sys.argv[1])

```

Figura 3: Operación no contemplada Python

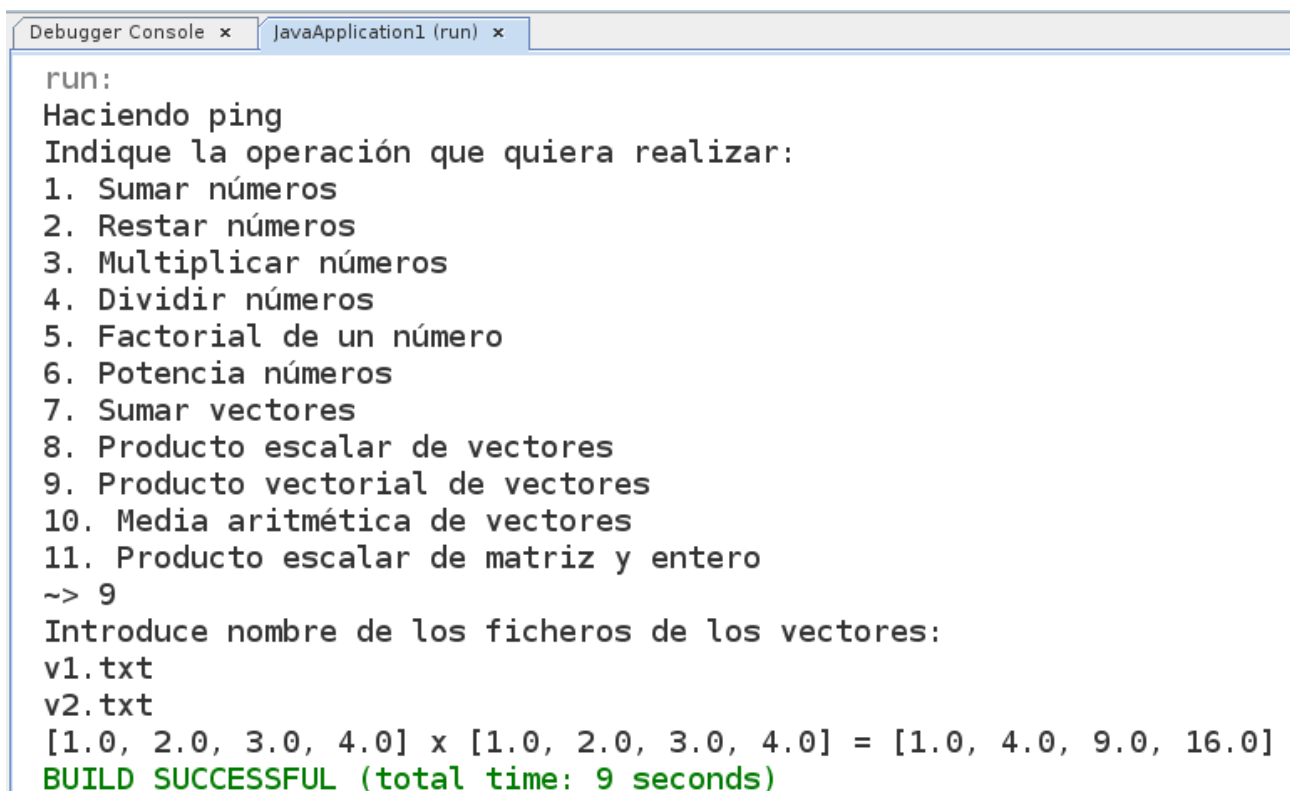
3.2. Java

```

Debugger Console x  javaApplication1 (run) x
run:
Haciendo ping
Indique la operación que quiera realizar:
1. Sumar números
2. Restar números
3. Multiplicar números
4. Dividir números
5. Factorial de un número
6. Potencia números
7. Sumar vectores
8. Producto escalar de vectores
9. Producto vectorial de vectores
10. Media aritmética de vectores
11. Producto escalar de matriz y entero
~> 7
Introduce nombre de los ficheros de los vectores:
v1.txt
v2.txt
[1.0, 2.0, 3.0, 4.0] + [1.0, 2.0, 3.0, 4.0] = [2.0, 4.0, 6.0, 8.0]
BUILD SUCCESSFUL (total time: 9 seconds)

```

Figura 4: Menu y operación suma de vectores con Java



```
run:
Haciendo ping
Indique la operación que quiera realizar:
1. Sumar números
2. Restar números
3. Multiplicar números
4. Dividir números
5. Factorial de un número
6. Potencia números
7. Sumar vectores
8. Producto escalar de vectores
9. Producto vectorial de vectores
10. Media aritmética de vectores
11. Producto escalar de matriz y entero
~> 9
Introduce nombre de los ficheros de los vectores:
v1.txt
v2.txt
[1.0, 2.0, 3.0, 4.0] x [1.0, 2.0, 3.0, 4.0] = [1.0, 4.0, 9.0, 16.0]
BUILD SUCCESSFUL (total time: 9 seconds)
```

Figura 5: Producto vectorial de vectores con Java

```
Debugger Console x  javaApplication1 (run) x
run:
Haciendo ping
Indique la operación que quiera realizar:
1. Sumar números
2. Restar números
3. Multiplicar números
4. Dividir números
5. Factorial de un número
6. Potencia números
7. Sumar vectores
8. Producto escalar de vectores
9. Producto vectorial de vectores
10. Media aritmética de vectores
11. Producto escalar de matriz y entero
~> 10
Introduce nombre del fichero con el vector:
v1.txt
[1.0, 2.0, 3.0, 4.0] media = 2.5
BUILD SUCCESSFUL (total time: 9 seconds)
```

Figura 6: Media aritmética de un vector con Java

```
Debugger Console x  JavaApplication1 (run) x
run:
Haciendo ping
Indique la operación que quiera realizar:
1. Sumar números
2. Restar números
3. Multiplicar números
4. Dividir números
5. Factorial de un número
6. Potencia números
7. Sumar vectores
8. Producto escalar de vectores
9. Producto vectorial de vectores
10. Media aritmética de vectores
11. Producto escalar de matriz y entero
~> 11
Introduce nombre con la matriz y el escalar:
matriz.txt
3
[[1.0, 2.0, 3.0], [1.0, 2.0, 3.0], [4.0, 5.0, 6.0]] * 3,00
= [[3.0, 6.0, 9.0], [3.0, 6.0, 9.0], [12.0, 15.0, 18.0]]
BUILD SUCCESSFUL (total time: 7 seconds)
```

Figura 7: Producto escalar de una matriz con Java

Índice de figuras

1.	Calculadora operaciones básicas Python	3
2.	Calculadora operaciones con vectores Python	3
3.	Operación no contemplada Python	4
4.	Menu y operación suma de vectores con Java	4
5.	Producto vectorial de vectores con Java	5
6.	Media aritmética de un vector con Java	6
7.	Producto escalar de una matriz con Java	7