
PRÁCTICA 2 - PRIMERA PARTE
CALCULADORA

Tercer Curso - Grado en Ingeniería Informática
Curso 2019-2020

PEDRO SERRANO PÉREZ

Contents

| | | |
|----------|--|----------|
| 1 | Introducción | 1 |
| 2 | Solución | 1 |
| 2.1 | Funcionalidad del Servidor | 1 |
| 2.2 | Funcionalidad del Cliente | 2 |
| 3 | Ejemplo de ejecución | 3 |
| 4 | Capturas de pantalla de ejecución | 8 |

1 Introducción

Esta práctica consiste en el desarrollo de una calculadora, que opere con números enteros, utilizando RPC Sun. La calculadora tendrá una arquitectura *Cliente-Servidor* y, por tanto, constará de dos partes:

- Servidor. Aporta las funcionalidades que permiten *sumar*, *restar*, *multiplicar* y *dividir*. Estas operaciones las podrá utilizar el cliente para que, el servidor, realice la operación con los dos números enteros que el cliente desee.
- Cliente. El lado del cliente se encarga de, por un lado, recibir la operación que desea calcular el usuario y, por otro, utilizar las operaciones del servidor para obtener el cálculo.

2 Solución

En esta sección se explicarán los distintos detalles de la solución al ejercicio.

La calculadora solo acepta números enteros por entrada pero, al utilizar divisiones, internamente se trabaja con enteros decimales (double).

La calculadora acepta por entrada el siguiente formato:

< entero > < operación > < entero >

Donde entero es cualquier número entero y operación incluye los símbolos +, -, "*" (debe ir con "" debido a la terminal) y /.

Es posible realizar operaciones concatenadas, en las cuales se respeta el orden de prioridad de las operaciones, por ejemplo:

$$1 + 2 * 3 - 5 = 2$$

2.1 Funcionalidad del Servidor

Como hemos visto en la introducción, el servidor aporta las distintas funcionalidades para realizar cálculos básicos: sumar, restar, multiplicar y dividir. Esto tiene como finalidad evitar que el servidor pase demasiado tiempo ocupado con un único cliente.

Lo único en lo que se diferencian estas funciones es en la operación matemática que realizan. Cabe destacar de las funciones lo siguiente:

- Devuelve un puntero a double (double*).
- Reciben por parámetro de numeros enteros sobre los cuales se realiza la operación. En el caso de la división el primer argumento sería el numerador y el segundo el denominador.

2.2 Funcionalidad del Cliente

El Cliente recibe las operaciones que el usuario desea calcular, por tanto debe comprobar que el formato en el que han sido introducidas es correcto. Como nos encontramos realizando el Cliente en C, el usuario envía las operaciones cuando ejecuta el cliente (parámetro char *argv[] del main), por tanto se realizan las siguientes comprobaciones:

- Comprueba que como mínimo se han introducido 5 argumentos (./cliente host entero operador entero)
- Debido al formato de las operaciones, el número de argumentos perteneciente a estas siempre es impar.
- Como los argumentos se tratan de cadenas de caracteres es necesario comprobar que se ha introducido el elemento correcto, por lo que se comprueba que inmediatamente después del elemento se encuentre un '\0'.
- Una operación tiene un entero a la izquierda y otro a la derecha
- Un entero va precedido de una operación (excepto el primero).
- A un entero le sigue una operación (excepto al último).
- Como se aceptan números enteros estos pueden ser negativos, por lo que se aceptaría, por ejemplo el número -10, introducirlo como:

$$1 + 2 * 3 - -10 = 17$$

Una vez explicadas las comprobaciones que realiza el cliente, vamos a ver cómo gestiona las operaciones que desea realizar el usuario.

En primer lugar almacena, por orden de izquierda a derecha, los números y las operaciones que encuentra en dos vectores.

Después, realiza 2 recorridos, uno en el que realiza las operaciones con prioridad (multiplicación y división), y otro en el que realiza las sumas y restas. Para realizar las distintas operaciones matemáticas llama a las funcionalidades del Servidor, descritas anteriormente, utilizando como parámetros el elemento actual del vector de números, el siguiente y un puntero de cliente creado para dicha operación.

Cuando obtiene un resultado actualiza la posición actual del vector de números con este resultado y elimina el segundo operando usado, desplazando el resto de elementos una posición hacia la izquierda.

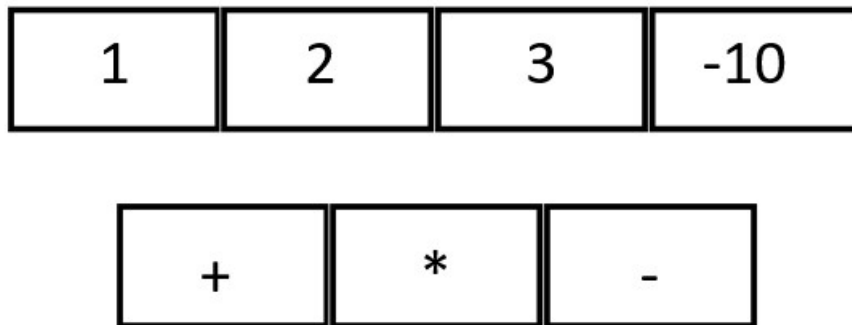
Finalmente, cuando termina ambos recorridos, el primer elemento del vector de los números es el que contiene el resultado.

3 Ejemplo de ejecución

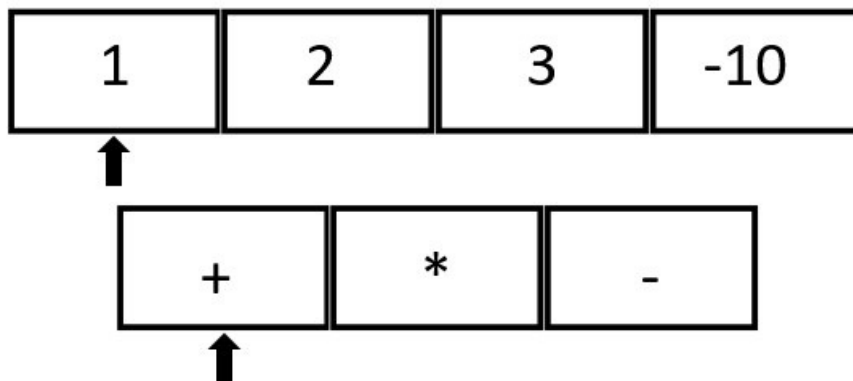
A continuación vamos a ver un ejemplo de ejecución para la siguiente operación:

$$1 + 2 * 3 - -10 = 17$$

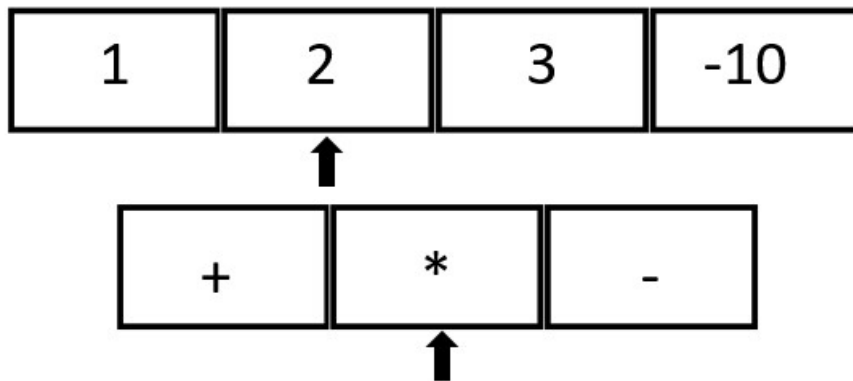
Una vez se realicen las comprobaciones el Cliente tendrá dos vectores, uno con los número y otro con las operaciones. El estado de los vectores en este punto sería el siguiente:



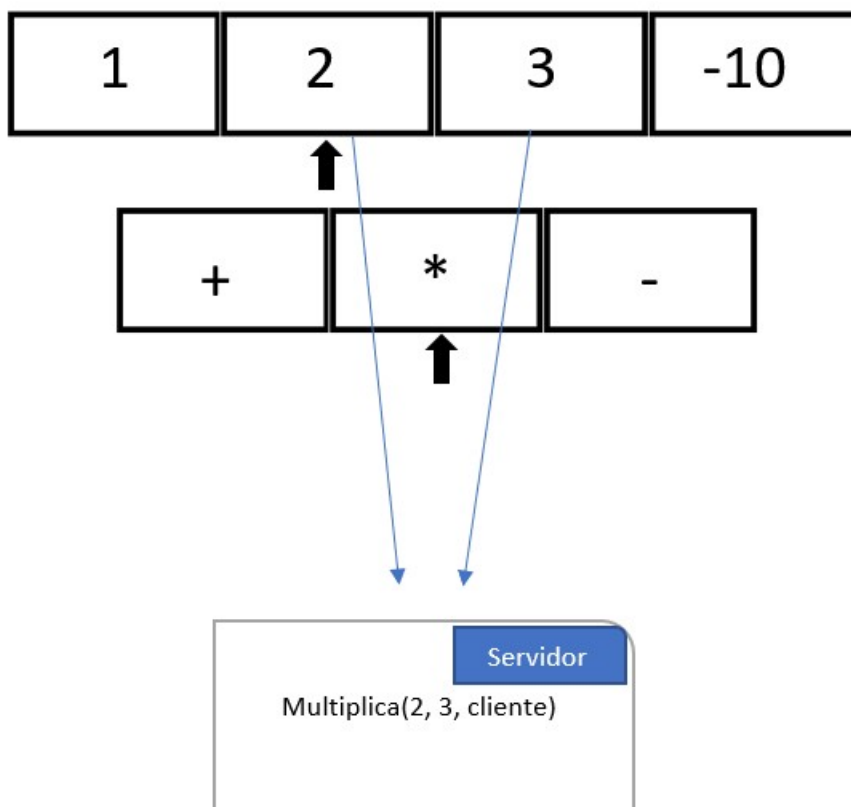
Como hemos explicado antes, se realizan dos recorridos sobre estos vectores. En el primero de ellos se realizarán las operaciones con prioridad, por tanto, en la primera iteración el estado sería:



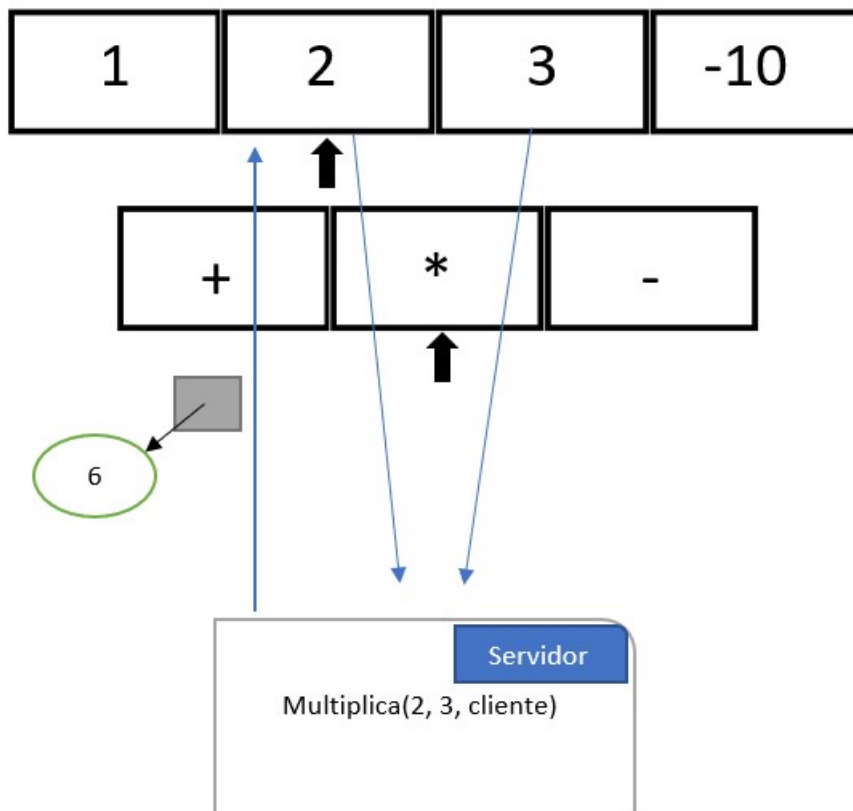
Como la operacion es $+$ se ignora y se avanza en ambos vectores. Este proceso se repite hasta encontrar $*$ o $/$



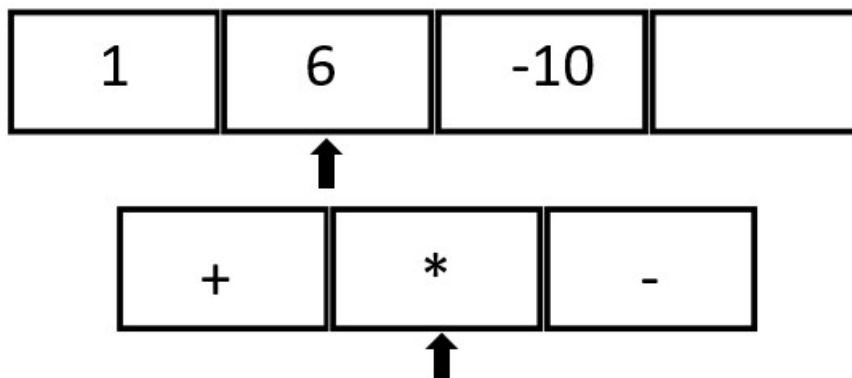
En este punto se encuentra con una multiplicación, por lo que llama a la función multiplicar del servidor con el número actual, el siguiente y el puntero a cliente:



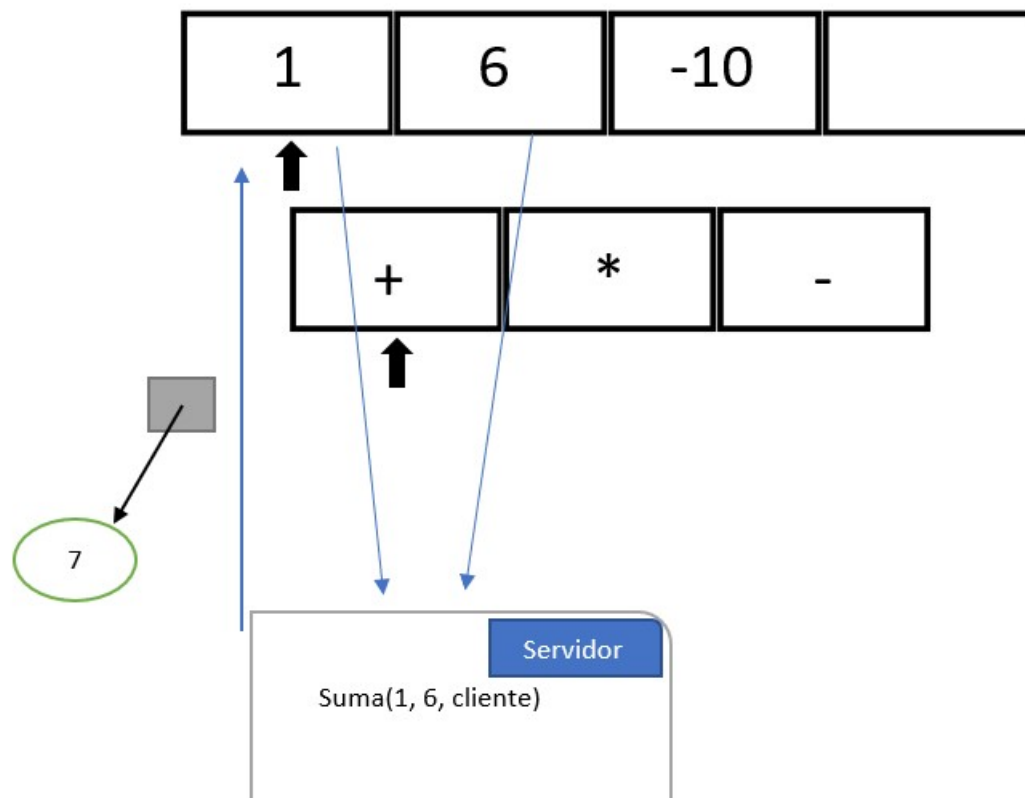
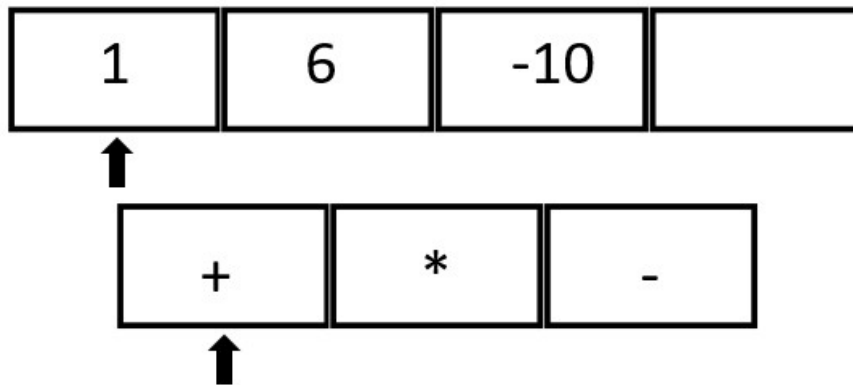
El servidor devuelve un puntero que apunta al resultado:



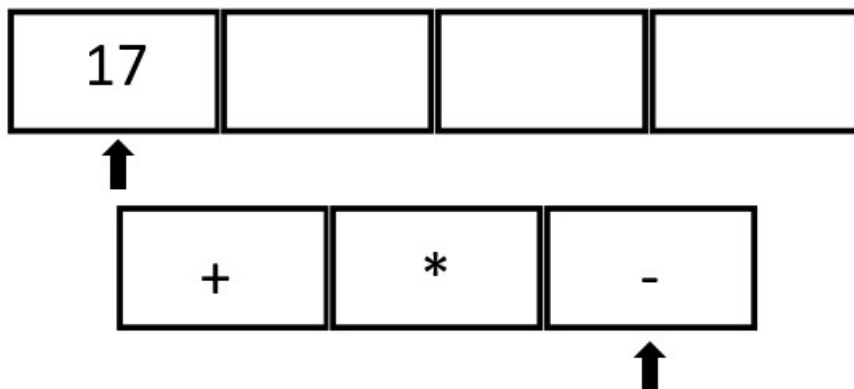
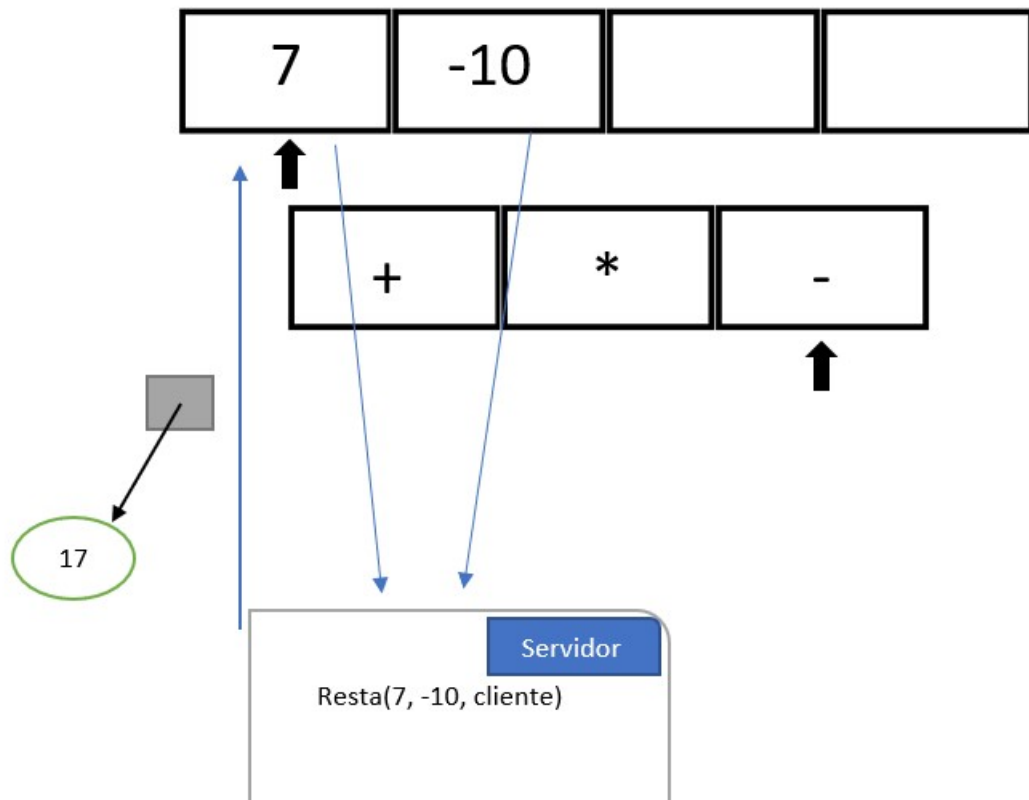
El cliente se encarga de actualizar la posición actual del vector de números con el resultado y de eliminar el segundo operando. Eliminando un elemento del vector y, desplazando el resto hacia la izquierda, por tanto el estado sería:



En la siguiente iteración, avanza en ambos vectores y se encuentra el último elemento del vector de operaciones, un -, por lo que terminaría el recorrido.

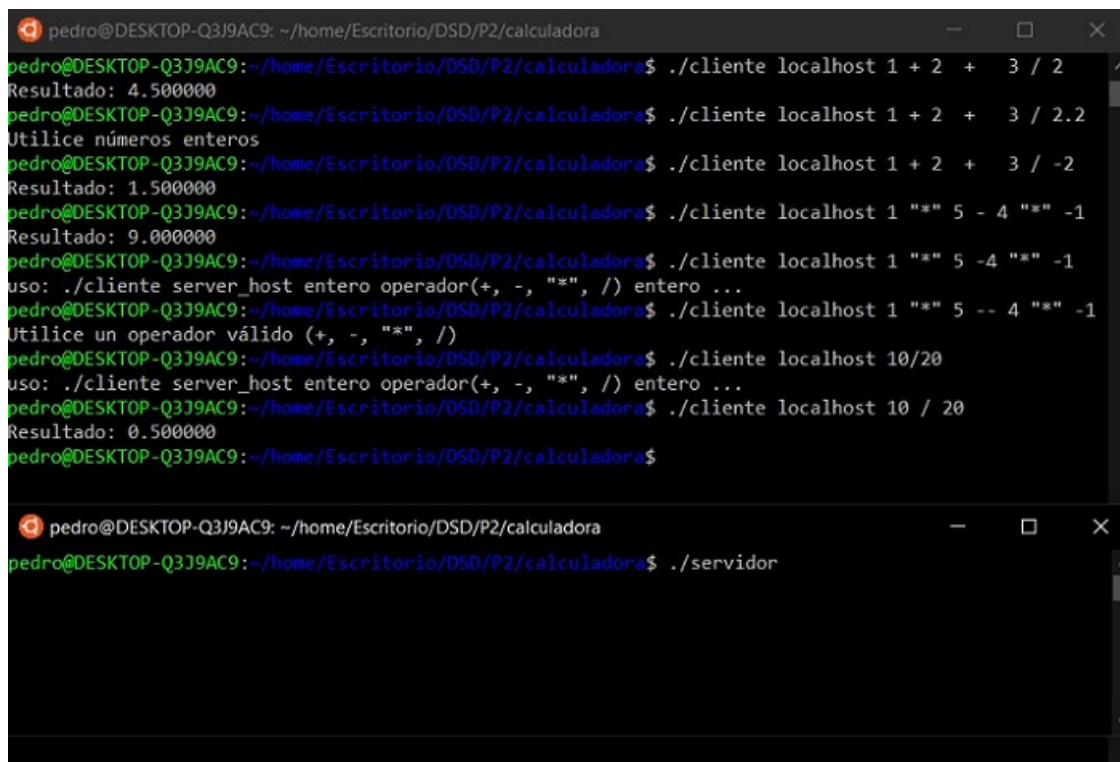


El cliente actualiza el vector de la misma forma que hemos visto antes, reemplaza el valor actual del vector de números por el resultado al que apunta el puntero y elimina el siguiente elemento, desplazando el resto una posición hacia la izquierda.



4 Capturas de pantalla de ejecución

A continuación se muestra una captura de pantalla con varias ejecuciones de la calculadora en el cliente. En la captura se muestran ejemplos del formato que se debe utilizar para ejecutar el cliente.

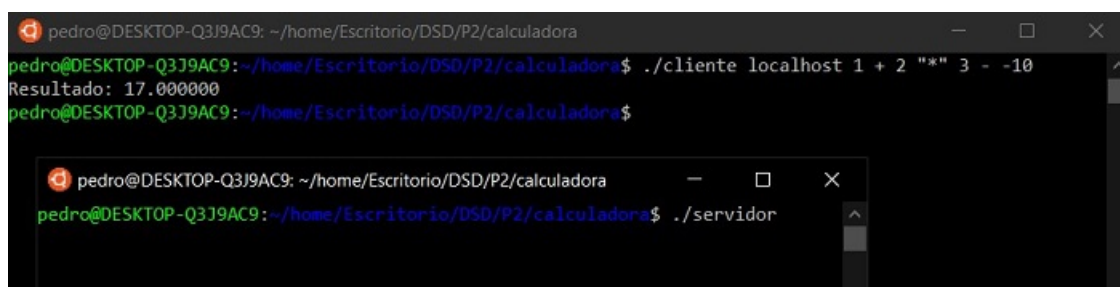


The image contains two terminal window screenshots. The top window shows the client application being run with various commands and their outputs. The bottom window shows the server application being run.

```
pedro@DESKTOP-Q3J9AC9: ~/home/Escritorio/DSD/P2/calculadora
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 1 + 2 + 3 / 2
Resultado: 4.500000
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 1 + 2 + 3 / 2.2
Utilice números enteros
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 1 + 2 + 3 / -2
Resultado: 1.500000
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 1 "*" 5 - 4 "*" -1
Resultado: 9.000000
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 1 "*" 5 -4 "*" -1
uso: ./cliente server_host entero operador(+, -, "*", /) entero ...
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 1 "*" 5 -- 4 "*" -1
Utilice un operador válido (+, -, "*", /)
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 10/20
uso: ./cliente server_host entero operador(+, -, "*", /) entero ...
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 10 / 20
Resultado: 0.500000
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$

pedro@DESKTOP-Q3J9AC9: ~/home/Escritorio/DSD/P2/calculadora
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./servidor
```

Captura de pantalla de ejecución del ejemplo descrito previamente:



The image shows two terminal windows. The top window shows the client application being run with a specific command and its output. The bottom window shows the server application being run.

```
pedro@DESKTOP-Q3J9AC9: ~/home/Escritorio/DSD/P2/calculadora
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./cliente localhost 1 + 2 "*" 3 - -10
Resultado: 17.000000
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$

pedro@DESKTOP-Q3J9AC9: ~/home/Escritorio/DSD/P2/calculadora
pedro@DESKTOP-Q3J9AC9:~/home/Escritorio/DSD/P2/calculadora$ ./servidor
```