

# Trabajo Práctico TCP/IP Socket

## Patricio Arango DNI 29119499

30 de Octubre 2022

El sistema fue desarrollado en Linux. La conexión se hizo tal cual los ejemplos. Cuando el cliente elige algunos de los 3 menús disponibles, el cliente envía un "comando" al servidor. Este comando es de cuatro caracteres y puede ser **"math"**, para realizar una operación; **"file"**, para pedir el archivo de logs; **"clos"**, para pedir el cierre de la conexión y un cuarto que se envía de forma automática por inactividad mayor o igual a 2 minutos que es **"desc"**.

El servidor, todo mensaje que recibe, primero toma los primeros 4 caracteres y se fija qué operación debe realizar.

### Operaciones:

Cuando llega un comando "math" el servidor recorta el comando y envía la operación restante para su validación:

1. Se valida mediante una expresión regular que la operación no contenga letras ni caracteres no válidos (se agregaron algunos pero el universo es mayor)
2. Si no tiene letras o símbolos no admitidos, se valida que sólo tenga números o símbolos válidos
3. Una vez que pasaron estas dos validaciones, entra en el análisis del tipo de operación. Mediante un objeto se analiza: qué tipo de operación es y en qué posición está el signo.
4. Los tipo de operaciones reciben un int según su tipo:
  - a) Factorial = 2
  - b) Potencia = 3
  - c) División = 4
  - d) Multiplicación = 5
  - e) Suma = 1
  - f) Resta = 0
5. Todas las operaciones excepto suma y resta, se determinan con el primer símbolo de esa operación. Es decir, dentro de un loop, se compara la operación con un listado de símbolos de operación y apenas encuentra uno, sale del bucle. Excepto signos más y menos. ¿Para qué? Porque **ESTA CALCULADORA PUEDE REALIZAR** todo tipo de operaciones válidas entre enteros **como  $-1^6$**

6. Al tener tipificada cada operación, también se resuelve cada operación de manera particular: el factorial es un bucle de multiplicación; la potencia se resuelve con la función “pow” de c++, etc.
7. Esta diferenciación también permite una validación específica para cada tipo de operación. Además de las validaciones exigidas, realiza otras de **sentido común** como **EVITAR LAS DIVISIONES POR CERO**.

## Archivo de Log:

El archivo de log se envía línea por línea al cliente y al final, se adosa la palabra “endoffile”. El cliente lee, en este caso, todas las líneas del log y cuando detecta “endoffile” vuelve al menú.

## Desconexión por inactividad:

El cliente, en el menú “Cerrar Sesión”, envía el comando “clos” al servidor para que éste cierre el socket del cliente y vuelva a la posición de escuchando conexiones entrantes. Cuando el cliente está en el menú, se guarda el timestamp (segundos que pasan desde el 1 de enero de 1970) de ese momento. Cuando se ejecuta cualquier acción, se vuelve a guardar un timestamp y se resta al original. Si pasaron 180 segundos o más, se envía el comando “desc” al servidor, este envía el mensaje “cliente cerrado por inactividad”, hace el log correspondiente y cierra el socket.

LINK AL VIDEO DE YOUTUBE

<https://youtu.be/WycJeDv31lw>

LINK A GITHUB

<https://github.com/patricioarango/SocketTCPIP-redes>