

Desarrollar en lenguaje C o C++ una calculadora de números enteros positivos, basada en dos aplicaciones que se ejecutarán en modo consola, las cuales estarán comunicadas por sockets (una con el rol de servidor y otra con el rol de cliente).

El cliente se conectará al servidor y enviará las operaciones a realizar. El servidor hará el procesamiento y devolverá el resultado al cliente.

## Requerimientos funcionales

### Servidor

- Inicia y espera la conexión del cliente.
- Toda la actividad relacionada con la administración de conexiones realizada por el servidor debe quedar registrada en un archivo de texto llamado **server.log**

- **Notas:**

- a) En caso de haber un cliente conectado, luego de **2 minutos sin actividad** la sesión expira. El servidor lo desconecta.
- b) Cuando un cliente sale o su sesión expira, el servidor se queda esperando a otro.

- **Caracteres aceptados:** los caracteres aceptados dentro de una operación son números enteros positivos y operadores matemáticos:

- ❖ Suma: +
- ❖ Resta: -
- ❖ Multiplicación: \*
- ❖ División: /
- ❖ Potenciación: ^
- ❖ Factorial: !

- **Tipo de operaciones aceptadas:** las operaciones aceptadas por la calculadora pueden ser:

- ✓ *[operando][operación][operando]*: siendo operación **Suma, Resta, Multiplicación, División o Potenciación**, y operando es un número entero positivo. Por ejemplo:

- 15+50
- 20/10
- 4^5

- ✓ *[operando][Factorial]*: operando es un número entero positivo y Factorial el símbolo que lo representa !. Por ejemplo: 8!

- **Validaciones para los cálculos:**

1. **Validación de caracteres de la operación:** el servidor debe validar que la operación recibida sea correcta, verificando que no haya ningún carácter inesperado. En caso de

encontrar un carácter no contemplado, deberá informar al cliente: *“No se pudo realizar la operación, se encontró un carácter no contemplado: [carácter]”*, donde [carácter] es el carácter que provocó el error.

2. **Validación de operación:** en el caso que la operación esté mal formada, el servidor informará *“No se pudo realizar la operación, la operación está mal formada: [abc]”*. Donde b es el carácter donde se detectó el error, a es el carácter anterior (si lo hubiere), y c el carácter siguiente (si lo hubiere). Una operación mal formada se da cuando faltan términos o se mezclan operaciones, por ejemplo:
  - 123/ → Falta el divisor
  - 12\*/3 → Se multiplica y divide al mismo tiempo
  - !15 → El carácter de Factorial debe ir después del número
3. **Longitud de operación:** la longitud máxima de una operación será de 20 caracteres, y la mínima será de 1 caracter. En el caso de recibir una operación fuera del rango mencionado, el servidor responderá *“La operación debe tener entre 1 y 20 caracteres”*.

## Cliente

- Para conectarse al servidor, primero deberá especificar la dirección IP y el puerto, una vez que el servidor acepte la conexión, el cliente dispone de las siguientes opciones:
  - Realizar cálculo
  - Ver registro de actividades
  - Cerrar sesión
- **Realizar operación:** esta opción solicita al usuario el ingreso de una operación y la envía al servidor, según las especificaciones detalladas en la sección de servidor. En caso de querer volver al menú anterior, al escribir *“volver”* en la operación, se mostrará el menú principal nuevamente.
- **Ver registro de actividades:** Mediante esta función se solicita al servidor que **transfiera** al cliente el archivo de registro de actividades. Una vez recibido, se muestra por pantalla el contenido del mismo.

**NOTA:** el cliente no puede acceder directamente al archivo de registro de actividades (*server.log*).

- **Cerrar sesión:** se desconecta del servidor.

## UI

- Utilizar menús para simplificar la selección de las diferentes opciones.

- La presentación de los datos debe ser clara (borrando la pantalla cuando corresponda, presentando páginas si las opciones son demasiadas, etc.).

## Requerimientos no funcionales

Lenguaje de programación C/C++.

Entorno de desarrollo a elección.

Sistema operativo Linux o Windows, a elección.