

Redes y Comunicaciones

Trabajo Práctico

2do Cuatrimestre de 2022

Documento	Nombre	Evaluación

Objetivo

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**

A modo de ejemplo:

Archivo: server.log

Contenido:

```
2022-11-01_11:50: =====
2022-11-01_11:50: =====Inicia Servidor=====
2022-11-01_11:50: =====
2022-11-01_11:50: Socket creado. Puerto de escucha:5000.
2022-11-01_11:53: Conexión Aceptada.
2022-11-01_11:58: Conexión Cerrada por Inactividad.
```

- **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 carácter. 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 “*vo/ver*” 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.

Presentación

17/10/2022: **Consultas.**

30/10/2022 20:00hs: **Entrega del TP vía Campus Virtual.**

31/10/2022: **Defensa del TP (presencial).**

Normas de entrega.

El trabajo entregado deberá contener un documento incluyendo:

En la entrega vía campus se debe enviar:

- El enunciado del trabajo práctico (TODO este documento, incluyendo el anexo).
- La estrategia de resolución del trabajo práctico. Es un texto descriptivo de cómo se estructuró la aplicación, cómo se separaron las capas, relaciones entre las entidades, es decir, todo aquello que consideren significativo para explicar la resolución del trabajo.
- El código fuente del proyecto anexado en una carpeta aparte o subido a un repositorio público (Github, GDrive, etc).
- Video (o enlace a video) en formato mp4 explicando (narrando) las pruebas del sistema. La especificación de las pruebas a realizar será indicada oportunamente.
- El archivo de log resultante de las pruebas.

Para la defensa del TP:

- Deberán entregar el TP impreso en su totalidad (enunciado, estrategia de resolución, código fuente, el archivo resultante de las pruebas) en hojas A4 numeradas con un tamaño de fuerza mínimo de 11.

El incumplimiento de cualquiera de las normas de entrega implicará la desaprobación del trabajo práctico.

Evaluación

La evaluación será oral y escrita. Los temas a evaluar serán, por ejemplo: preguntas sobre la resolución del trabajo práctico, explicación de las estructuras de datos utilizadas, codificación o modificación de alguna función, etc.

No se harán correcciones parciales. Por este motivo, SOLO deberán presentarse quienes hayan concluido TODO el trabajo práctico.

Cuestiones de autoría

- Todo el código debe ser desarrollado íntegramente por el alumno.
- No se permite la reutilización de código de cuatrimestres anteriores o de otras materias.
- Ante cualquier duda se deberá consultar con los docentes.
- La reutilización de código sin consulta previa será condición suficiente para la desaprobación de la materia.

