**Ejercicio Práctico 05**

**Programa Concurrente y procesos cooperantes.**

OBJETIVO: usted desarrollará un programa concurrente, para generar reportes del estado de cuenta, por cliente, incluyendo la relación de los movimientos o transacciones de las diferentes cuentas.

Con esta práctica se cubren temas como programa concurrente, procesos concurrentes cooperantes, y paralelismo abstracto entre el thread padre y los hijos.

Deberá subir a Comunidad, este reporte en un .zip, junto con los programas Java y las correspondientes respuestas a las preguntas del final de este texto.

Los archivos del fólder son *movimientos.txt*, *clientes.txt*, *EdoCtaClientes.java* (thread padre *main*) y *MovimientosCliente.java* (threads hijos). El archivo Java *EdoCtaClientes.java* se encuentra parcialmente programado, pero usted los puede rehacer. El otro archivo, *MovimientosCliente.java*, quién constituirá los threads está prácticamente vacio.

En el archivo de *clientes.txt*, por cada cliente, en cada renglón se tiene la clave del cliente y el nombre del cliente. En el archivo *movimientos.txt* se encuentran las diferentes transacciones o movimientos bancarios realizados por los clientes. En cada renglón viene la clave del cliente y a continuación la transacción. Por cada cliente que aparece en *clientes.txt* aparecerá, al menos, una transacción correspondiente al cliente en *movimientos.txt*.

Su programa padre (thread *main*) *EdoCtaClientes.java* deberá crear y ejecutar, por cada cliente, un thread hijo *MovimientosCliente.java* que elaborará un archivo *Reporte<nombredelcliente>.txt* correspondiente al cliente. Así, si uno de los clientes se llama *Laura* entonces el archivo, con su reporte de cuenta, se llamará *ReporteLaura.txt*.

Puede que en el archivo *movimientos.txt* se encuentre información de movimientos de otros clientes que no están registrados en *clientes.txt*; dichas transacciones no las debe tomar en cuenta.

Como siempre, el proceso thread padre debe esperar, haciendo otras cosas (ciclar y volver a preguntar), a que los procesos hijos terminen. Esta espera deberá hacerla de manera iterativa, y de ninguna manera vaya a realizar una espera secuencial.

También, agregue en cada reporte, el tiempo total de ejecución expresado en milisegundos, así como el total de transacciones. También, en el proceso padre (thread *main*), despliegue el tiempo total de ejecución en milisegundos.

Además, recolecte tres “pantallazos” de tres corridas de *EdoCtaClientes*, donde se vea correctamente el despliegue de la ejecución y los tiempos totales de ejecución.

¿Cuántos y cuáles son los programas ordinarios que conforman el programa concurrente de este ejercicio? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

¿Cuántos procesos ligeros (threads) fueron ejecutados? \_\_\_\_\_\_\_\_\_\_\_ y ¿Cuántos procesos tradicionales o pesados? \_\_\_\_\_\_\_\_\_\_\_\_\_