**Práctica 07 win**

**Programa Concurrente y procesos cooperantes.**

OBJETIVO: usted desarrollará un programa concurrente, para generar estados 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 proceso padre y los hijos.

Los cuatro archivos, dos *.txt* y dos *.java*, se encuentran en el fólder ***LabBG07*** de Comunidad.

Deberá subir a Comunidad, este reporte en un .zip, junto con los programas Java y las correspondientes respuestas. Antes o después que haya subido a Comunidad, en el laboratorio el profesor revisará el funcionamiento de los programas.

Los archivos del fólder son *movimientos.txt*, *clientes.txt*, *EdoCtaClientes.java* (padre) y *MovimientosCliente.java* (hijo). Los archivos Java se encuentran parcialmente programados, pero usted los puede rehacer.

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.

Su programa padre *EdoCtaClientes.java* deberá crear y ejecutar, por cada cliente, un subproceso *MovimientosCliente.java* que elaborará un archivo *EdoCta<nombredelcliente>.txt* correspondiente al cliente. Así, si uno de los clientes se llama *Laura* entonces el archivo, con su estado de cuenta, se llamará *EdoCtaLaura.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 padre debe esperar, haciendo otras cosas (ciclar y volver a preguntar), a que los procesos hijos terminen.

También, agregue en cada estado de cuenta, el tiempo total de ejecución expresado en milisegundos, así como el total de transacciones. También, en el proceso padre, despliegue el tiempo total de ejecución en milisegundos.

Después de que la anterior parte funcione, modifique la aplicación considerando que puede darse que algún *cliente* no aparezca en *movimientos.txt*; en este caso no debe generar el archivo del estado de cuenta. El hijo correspondiente deberá “avisarle” al padre que dicho estado de cuenta no se generó y el padre al final, y sólo cuando todos los procesos hayan terminado, deberá desplegar el reporte de aquellos estados de cuenta que no se produjeron, por nombre del cliente.

Además conteste:

¿Cuáles son los programas secuenciales ordinarios que conforman este programa concurrente?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

¿Cuántos procesos concurrentes cooperantes interactúan en memoria en este programa concurrente? Expréselo en términos de procesos padres e hijos. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Agregar imágenes de *ProcessExplorer*, que muestre el árbol jerárquico mostrando varios hijos.\_\_\_\_\_\_\_\_

**OBSERVACIONES:**

No use PIPEs.

No deberá hacer procesamiento secuencial.

El programa padre *EdoCtaClientes.java* está parcialmente programado; muy poco.

Se recomienda que maneje un arreglo de procesos, tanto para lanzar la ejecución de los procesos hijos así como para esperar su terminación.

El programa hijo *MovimientosCliente.java* es un esqueleto con algún código.

Recuerde que las formas de “comunicarse” son: al lanzar la ejecución del proceso hijo y cuando el hijo termina a través de *exit(…)*;

En el fólder ***LabBG07/archiv*** vienen una serie de ejercicios en JAVA para manipulación de archivos y fechas. Dentro del fólder, el archivo *inf.txt* le indica que hace cada programa.