UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMAS DISTRIBUIDOS

PROFESORA ING. GUADALUPE LIZETH PARRALES ROMAY

TAREA 3: IMPLEMENTACIÓN DE UN COORDINADOR DE TRANSACCIONES

ROJAS MÉNDEZ GABRIEL

FECHA DE ENTREGA: 1 DE MAYO DE 2022

SEMESTRE 2022-2

**Planteamiento de la solución.**

Para poder realizar con más facilidad la implementación de los diversos métodos solicitados, se hizo uso de hilos los cuales ya cuentan con métodos que permiten modificar el nombre del hilo lo cual sirvió para determinar el origen de las transacciones, ya fuera de cliente o de servidor. Además, también existe la función que retorna el ID del hilo lo que permitió combinarse con el nombre y tener un ID único para cada hilo instanciado, por lo tanto, para esto se crearon dos clases, una denominada Cliente y la otra, Servidor.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

En cada una de estas clases se implementaron los métodos de deposito y retiro de manera implícita, ya que mediante el constructor de cada una de la clase se modifico el atributo de nombre del hilo y de esta manera se emplearon estructuras de control que determinaban que transacción realizar y quién debía hacerla.

Captura de pantalla de computadora

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Con esta estructura las instancias de hilos a nivel de procesador se ejecutarían conforme ganaran tiempo de Quantum y así se podría ver las anomalías de lecturas sucias y escrituras prematuras, para que se logrará esto se declaró una clase denominada Cuenta con una variable estática para que así tanto cliente como servidor tuvieran acceso a un solo tipo de variable y no hicieran copia de esta para observar el comportamiento del valor de esta.

Texto

Descripción generada automáticamente

Finalmente, para la implementación de ir guardando las transacciones en un archivo de texto se empleó otra clase que hacia uso de la clase FileWriter a la cual se le paso un buffer para que así se capturaran los datos de cada transacción

Texto

Descripción generada automáticamente

Y los resultados de la implementación de esta clase quedaron de la siguiente manera, pudiendo observar la fecha y hora de la transacción, el origen y la acción realizada y el estado de la variable estática al termino de cada transacción.

Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Esos son los datos de la memoria, pero también se emplearon unos printf para poder observar esos datos en tiempo de ejecución y ahí era más evidente lo de las anomalías ya mencionadas.

Texto

Descripción generada automáticamente

Y aunque los print y la ejecución de los hilos tenían un orden en el código en la consola se veía otro al igual que los valores esto debido a lo ya antes mencionado de la carrera por recursos otorgados por el sistema operativo.

Captura de pantalla de computadora

Descripción generada automáticamente

Y así es como se logró generar un coordinador de transacciones sin algún tipo de optimización para evitar las lecturas sucias y las escrituras prematuras, pero el mismo uso de hilos permite implementar esta optimización si es requerida mediante el uso de métodos synchronized, los cuales bloquean sus respectivos atributos hasta que se termina de operar con ellos, pero esto ya será implementación para la segunda parte de este coordinador.