**LABORATORIO 1 ARSW**

Presentado por:

Daniela García

Edwar Lozano

Presentado a:

Diego Triviño

Escuela Colombiana de Ingeniería Julio Garavito

Materia: ARSW

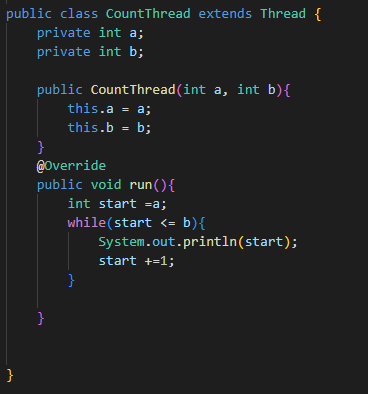
Grupo: 2

1 de febrero de 2023

**LABORATORIO 1 ARSW**

**Parte I Hilos Java**

1. De acuerdo con lo revisado en las lecturas, complete las clases CountThread, para que las mismas definan el ciclo de vida de un hilo que imprima por pantalla los números entre A y B.



1. Complete el método main de la clase CountMainThreads para que:

* Cree 3 hilos de tipo CountThread, asignándole al primero el intervalo [0..99], al segundo [99..199], y al tercero [200..299].

Texto

Descripción generada automáticamente

* Inicie los tres hilos con 'start()'.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

* Ejecute y revise la salida por pantalla.

Imagen que contiene Tabla

Descripción generada automáticamente

* Cambie el inicio con 'start()' por 'run()'. ¿Cómo cambia la salida?, por qué?

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Imagen que contiene Tabla

Descripción generada automáticamente

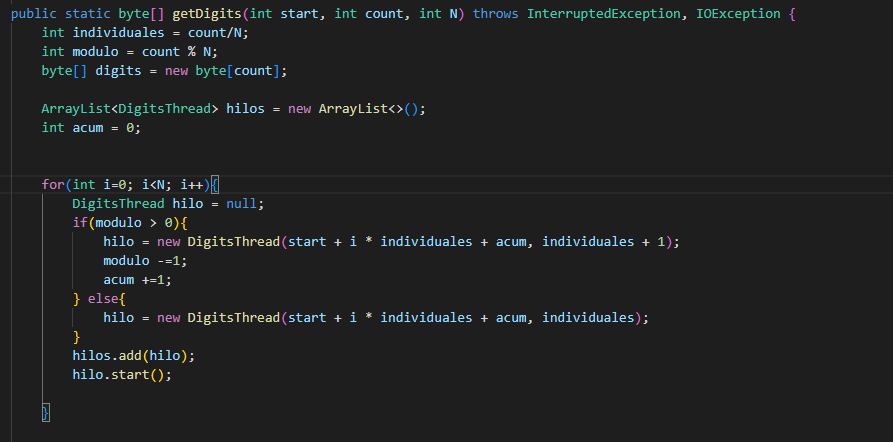
**Parte II Hilos Java**

1. Cree una clase de tipo Thread que represente el ciclo de vida de un hilo que calcule una parte de los dígitos requeridos.

Texto

Descripción generada automáticamente

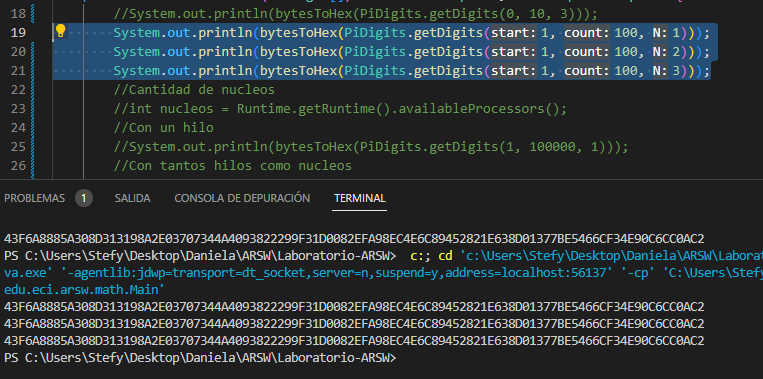
1. Haga que la función PiDigits.getDigits() reciba como parámetro adicional un valor N, correspondiente al número de hilos entre los que se va a paralelizar la solución. Haga que dicha función espere hasta que los N hilos terminen de resolver el problema para combinar las respuestas y entonces retornar el resultado. Para esto, revise el método join del API de concurrencia de Java.



Texto

Descripción generada automáticamente

1. Ajuste las pruebas de JUnit, considerando los casos de usar 1, 2 o 3 hilos (¡este último para considerar un número impar de hilos!)



**Parte III Evaluación de Desempeño**

A partir de lo anterior, implemente la siguiente secuencia de experimentos para calcular el millon de dígitos (hex) de PI, tomando los tiempos de ejecución de los mismos (asegúrese de hacerlos en la misma máquina):

1. Un solo hilo.

Texto

Descripción generada automáticamente

1. Tantos hilos como núcleos de procesamiento (haga que el programa determine esto haciendo uso del [API Runtime](https://docs.oracle.com/javase/7/docs/api/java/lang/Runtime.html)).

Texto

Descripción generada automáticamente

1. Tantos hilos como el doble de núcleos de procesamiento.

Texto

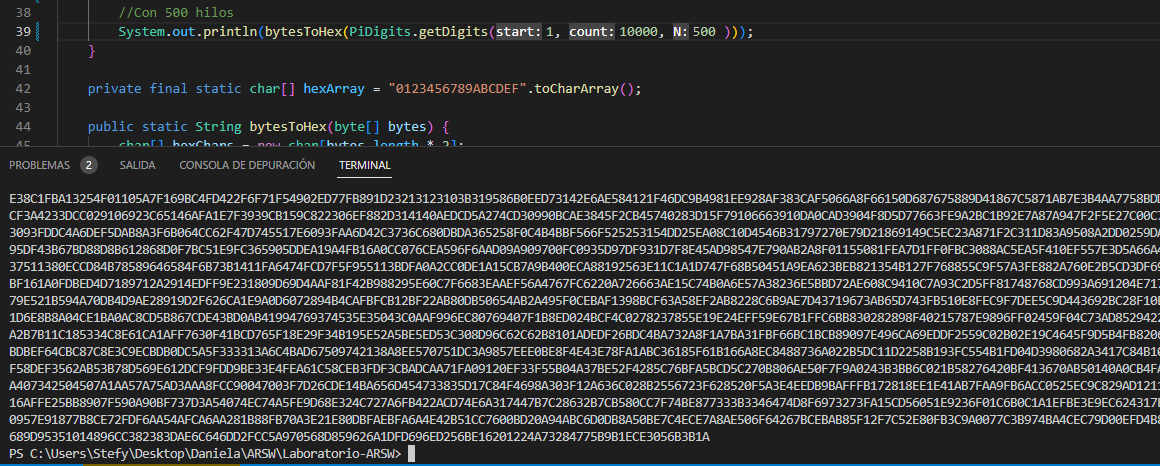
Descripción generada automáticamente

1. 200 hilos.

Texto

Descripción generada automáticamente

1. 500 hilos.



Al iniciar el programa ejecute el monitor jVisualVM, y a medida que corran las pruebas, revise y anote el consumo de CPU y de memoria en cada caso.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Con lo anterior, y con los tiempos de ejecución dados, haga una gráfica de tiempo de solución vs. número de hilos. Analice y plantee hipótesis con su compañero para las siguientes preguntas (puede tener en cuenta lo reportado por jVisualVM):

1. Según la [ley de Amdahls](https://www.pugetsystems.com/labs/articles/Estimating-CPU-Performance-using-Amdahls-Law-619/#WhatisAmdahlsLaw?):

[Diagram

Description automatically generated](https://github.com/ARSW-ECI-archive/Parallelism-JavaThreads-Introduction/blob/master/img/ahmdahls.png), donde S(n) es el mejoramiento teórico del desempeño, P la fracción paralelizable del algoritmo, y n el número de hilos, a mayor n, mayor debería ser dicha mejora. ¿Por qué el mejor desempeño no se logra con los 500 hilos?, cómo se compara este desempeño cuando se usan 200?

La comparar el desempeño al utilizar 500 hilos se logra ver que hay una disminución con respecto al que se presenta con 200 hilos, esto se debe sobre todo a que la cantidad de hilos que pueden trabajar en paralelo depende de la cantidad de núcleos con las que cuenta equipo, hacia los 200 hilos se alcanza la máxima capacidad por lo que no se podrá aumentar el rendimiento.

1. ¿Cómo se comporta la solución usando tantos hilos de procesamiento como núcleos comparados con el resultado de usar el doble de éste?

El rendimiento se ve afectado por la cantidad de hilos que estén trabajando en paralelo dentro de cada núcleo, por lo que tener el doble de estos disminuirá el tiempo de solución.

1. De acuerdo con lo anterior, si para este problema en lugar de 500 hilos en una sola CPU se pudiera usar 1 hilo en cada una de 500 máquinas hipotéticas, ¿la ley de Amdahls se aplicaría mejor? Si en lugar de esto se usaran c hilos en 500/c máquinas distribuidas (siendo c es el número de núcleos de dichas máquinas), se mejoraría? Explique su respuesta.

Para el primer caso no mejoraría la aplicación de la ley, ya que básicamente esta nos indica que el rendimiento mejorar cuando se aumentan los hilos que se utilizan por máquina. Para el segundo caso sí se vería una mejoraría ya que se estaría utilizando la programación en paralelo, además de esto se estaría utilizando la máxima capacidad de cada núcleo en cada equipo.