

# **INTRODUCCIÓN AL PARALELISMO – HILOS**

## **AUTORES:**

YORKS GOMEZ – CESAR VÁSQUEZ

## **PROFESOR:**

JAVIER IVAN TOQUICA

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

ARQUITECTURAS DE SOTFWARE – GRUPO # 1

INGENIERÍA DE SISTEMAS

BOGOTÁ D.C.

2023

## INTRODUCCIÓN

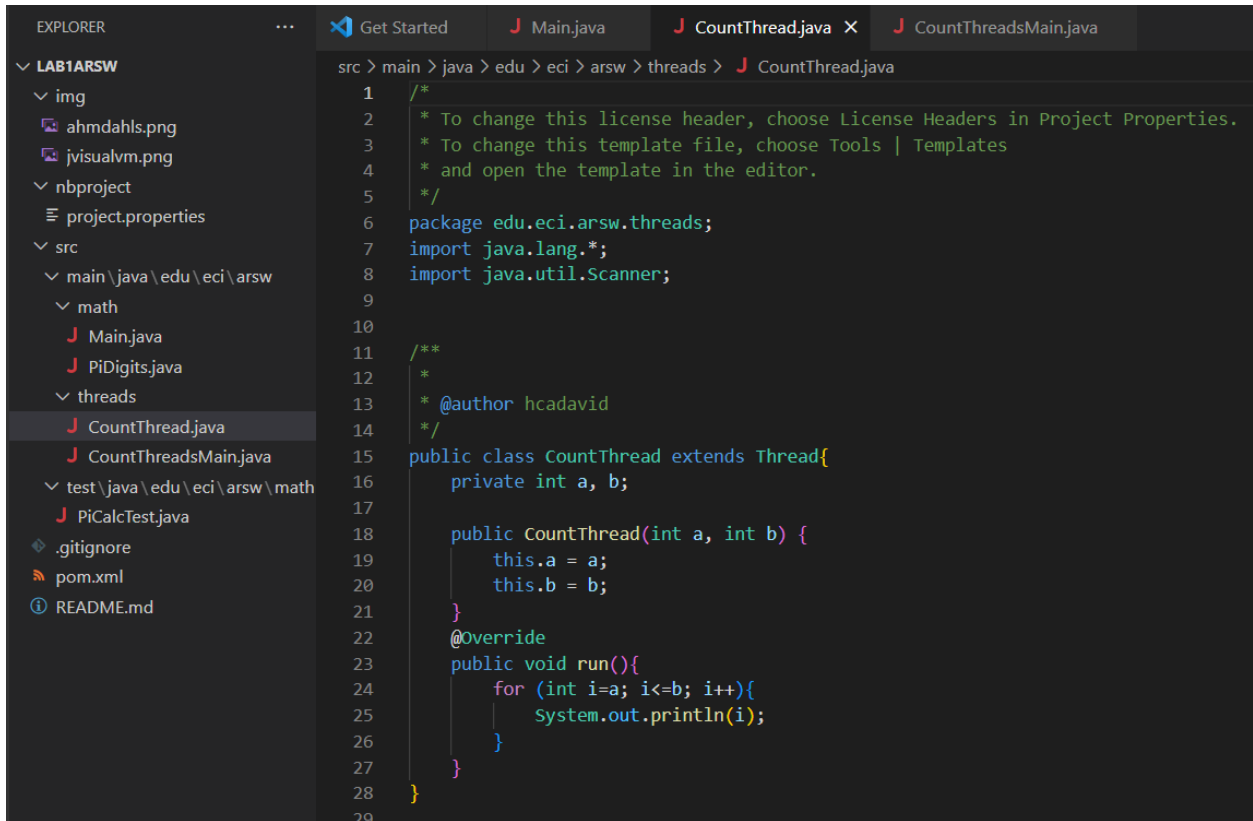
En este informe de laboratorio podremos profundizar y entender de lleno el concepto de paralelismo, para qué nos sirve, cómo se maneja, de que consta, etc. Lo podremos realizar a través de las diferentes formas de uso orientadas a **hilos** en java, aprenderemos a usarlas adecuadamente entendiendo el funcionamiento que tiene cada una de ellas y aplicándose cuando sea necesario.

Para ello utilizaremos correspondientemente todas las herramientas dadas, como lo son lecturas, videos, conocimientos, equipos, entre otras.

## DESARROLLO

### Parte I

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  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package edu.eci.arsw.threads;
7  import java.lang.*;
8  import java.util.Scanner;
9
10
11  /**
12   *
13   * @author hcadavid
14   */
15  public class CountThread extends Thread{
16      private int a, b;
17
18      public CountThread(int a, int b) {
19          this.a = a;
20          this.b = b;
21      }
22
23      @Override
24      public void run(){
25          for (int i=a; i<=b; i++){
26              System.out.println(i);
27          }
28      }
29  }
  
```

2. 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].
  - Inicie los tres hilos con 'start()'.

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package edu.eci.arsw.threads;
7
8  /**
9   *
10  * @author hcadavid
11  */
12  public class CountThreadsMain {
13
14      public static void main(String a[]){
15          CountThread countThread1 = new CountThread(0, 99);
16          CountThread countThread2 = new CountThread(99, 199);
17          CountThread countThread3 = new CountThread(200, 299);
18          Thread hilo1 = new Thread(countThread1);
19          Thread hilo2 = new Thread(countThread2);
20          Thread hilo3 = new Thread(countThread3);
21          hilo1.start();
22          hilo2.start();
23          hilo3.start();
24      }
25  }

```

- Ejecute y revise la salida por pantalla.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

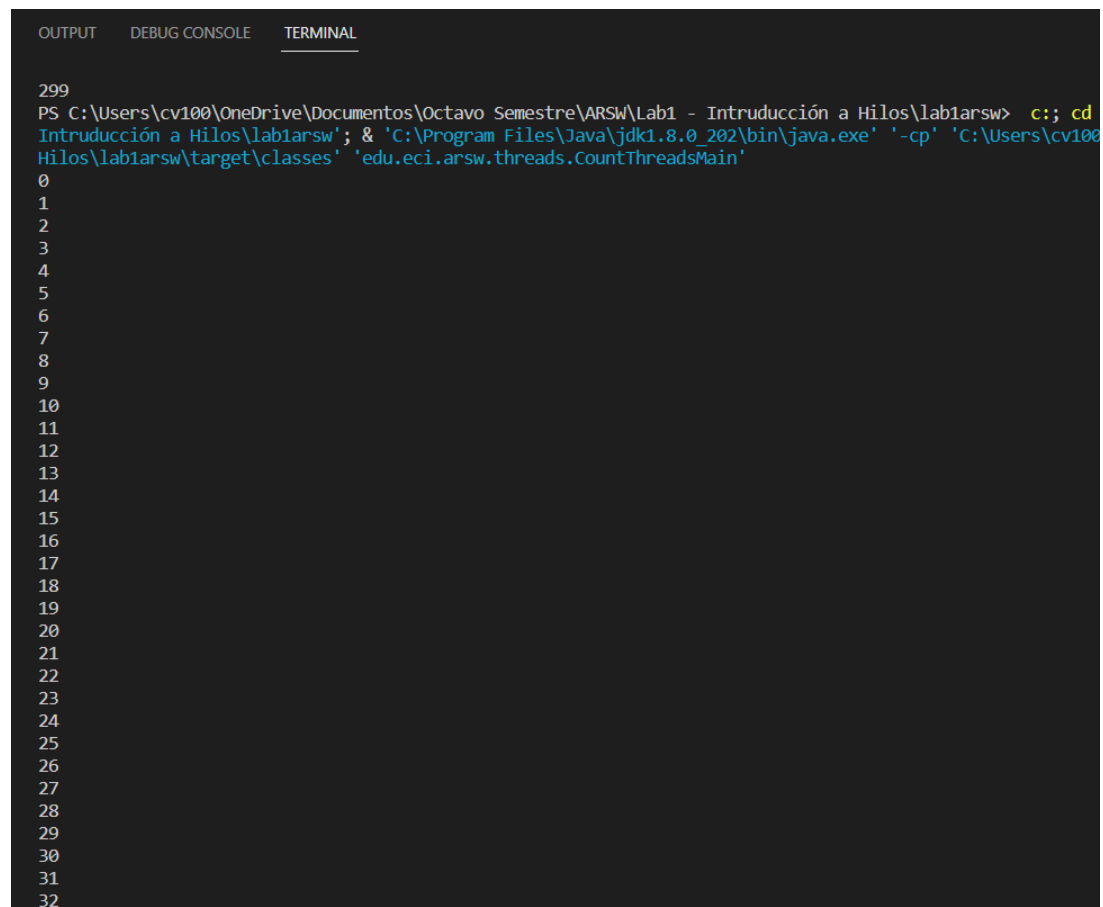
PS C:\Users\cv100\OneDrive\Documentos\Octavo Semestre\ARSW\Lab1 - Intrucción a Hilos\lab1arsw> & 'C:\Program Files\Java\jdk-10.0.2\bin\java.exe' -cp 'C:\Users\cv100\OneDrive\Documentos\Octavo Semestre\ARSW\Lab1 - Intrucción a Hilos\lab1arsw\target\classes' 'edu.eci.arsw.threads.CountThreadsMain'
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

```

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

Cuando utilizamos el método start() para el objeto CountThread lo que está haciendo es crear el hilo y cambiar el estado a Runnable, cuando le decimos al objeto de CountThread start() cierto número de ocasiones, está cambiando el estado de Runnable a Runnable, esto hace que imprima en desorden, pues cada hilo está interrumpiendo a los demás utilizando el medio cierto momento. El método run() lo que hace es ejecutar el hilo sin tener que cambiarle el estado más de una vez, por lo que hace el llamado a un método de una clase esto hace que no haga otra instrucción hasta que acabe el método, luego continuaran los hilos restantes hasta finalizar, por eso se puede ver en orden la ejecución.

Así se ve con el método run():



```
OUTPUT  DEBUG CONSOLE  TERMINAL
299
PS C:\Users\cv100\OneDrive\Documentos\Octavo Semestre\ARSw\Lab1 - Intruducción a Hilos\lab1arsw> c::; cd
Intruducción a Hilos\lab1arsw'; & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\cv100
Hilos\lab1arsw\target\classes' 'edu.eci.arsw.threads.CountThreadsMain'
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
```