

Informe de Laboratorio 05

Tema: Método del trapecio

Nota

Estudiante	Escuela	Asignatura
Jose Carlos Chino Nina jchino@ulasalle.edu.pe	Carrera Profesional de Ingeniería de Software	Lenguaje de Programación II Semestre: III Código: 20231001

Laboratorio	Tema	Duración
05	Método del trapecio	06 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2025 - B	Del 09 Octubre 2025	Al 15 Octubre 2025

1. Tarea

- Implementar una clase en Java, llamada **Registro**, que utilice un **constructor con un parámetro** para inicializar un atributo.
- El constructor debe hacer uso de la palabra clave **this** para diferenciar entre el atributo de la instancia y el parámetro.
- Utilizar Git para evidenciar su trabajo.
- Enviar trabajo al profesor en un repositorio GitHub Privado, dándole permisos como colaborador.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.
- VIM 9.0.
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos (**POO**).
- **Constructores** y palabra clave **this** en Java.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/Jchino24/lp2.git`
- URL para el laboratorio 04 en el Repositorio GitHub.
- `https://github.com/Jchino24/lp2/tree/main/Lab04`

4. Actividades con el repositorio GitHub

4.1. Creando codigo

- Como es el primer laboratorio se creo el repositorio GitHub.
- Se realizaron los siguientes comandos en la computadora:

Listing 1: Creando carpeta laboratorio 05

```
$ mkdir -p $HOME/Jchino24/
```

Listing 2: Dirigiéndonos al directorio de trabajo

```
$ cd $HOME/Jchino24/
```

Listing 3: Creando directorio para repositorio GitHub

```
$ mkdir -p $HOME/Jchino24/lp
```

Listing 4: Inicializando directorio para repositorio GitHub

```
$ cd $HOME/Jchino24/lp
$ echo "# programacion" >> README.md
$ git init
$ git config --global user.name "Jose Carlos Chino Nina"
$ git config --global user.email jjchino@ulasalle.edu.pe
$ git add README.md
$ git commit -m "first commit"
$ git branch -M main
$ git remote add origin https://github.com/Jchino24/lp2.git
$ git push -u origin main
```

4.2. Commits

Listing 5: Primer Commit Creando carpeta/archivo para laboratorio 04

```
$ mkdir lab04
$ touch lab04/Trapezio.java % ARCHIVO ACTUALIZADO
$ git add .
$ git commit -m "Creando carpeta/archivo para lab 04: Trapecio.java"
$ git push -u origin main
```

- Se creo el archivo **.gitignore** para no considerar los archivos ***.class**.

Listing 6: Creando .gitignore

```
$ vim lab04/.gitignore
```

Listing 7: lab04/.gitignore

```
*.class
```

Listing 8: Commit: Creando .gitignore para archivos *.class

```
$ git add .  
$ git commit -m "Creando .gitignore para archivos *.class"  
$ git push -u origin main
```

- Se implementó el **Método del Trapecio** utilizando **Threads** para la integración numérica, basándose en la fórmula de la suma de Riemann. La sumatoria interna fue distribuida entre el número de núcleos disponibles en el sistema.

Listing 9: Creando Trapecio.java

```
$ vim lab04/Trapecio.java
```

Listing 10: Insertion.java

```
1 import java.util.*;
2
3 public class Trapecio {
4
5     static class TrapecioWorker extends Thread {
6         private double a, h;
7         private int inicio, fin;
8         private double constante;
9         private double sumaParcial = 0;
10
11         public TrapecioWorker(double a, double h, int inicio, int fin, double constante) {
12             this.a = a;
13             this.h = h;
14             this.inicio = inicio;
15             this.fin = fin;
16             this.constante = constante;
17         }
18
19         public void run() {
20             for (int i = inicio; i < fin; i++) {
21                 double x = a + i * h;
22                 sumaParcial += 2 * f(x);
23             }
24         }
25
26         private double f(double x) {
27             return constante;
28         }
29
30         public double getSumaParcial() {
31             return sumaParcial;
32         }
33     }
34
35     public static void main(String[] args) throws InterruptedException {
36         Scanner sc = new Scanner(System.in);
37
38         System.out.println("=== MTODO DEL TRAPECIO BSICO CON HILOS ===");
39         System.out.print("Ingresa el valor constante f(x) = ");
40         double constante = sc.nextDouble();
41
42         System.out.print("Ingresa el lmite inferior (a): ");
43         double a = sc.nextDouble();
44
45         System.out.print("Ingresa el lmite superior (b): ");
46         double b = sc.nextDouble();
47
48         System.out.print("Ingresa el nmero de subintervalos (n): ");
49         int n = sc.nextInt();
50     }
```

```
51     int numHilos = Runtime.getRuntime().availableProcessors();
52     System.out.println("Usando " + numHilos + " hilos...");
53
54     double h = (b - a) / n;
55
56     double sumaTotal = fGlobal(constante, a) + fGlobal(constante, b);
57
58     TrapecioWorker[] workers = new TrapecioWorker[numHilos];
59     int paso = n / numHilos;
60
61     for (int i = 0; i < numHilos; i++) {
62         int inicio = i * paso + 1;
63         int fin = (i == numHilos - 1) ? n : (i + 1) * paso + 1;
64         workers[i] = new TrapecioWorker(a, h, inicio, fin, constante);
65         workers[i].start();
66     }
67
68     for (TrapecioWorker worker : workers) {
69         worker.join();
70         sumaTotal += worker.getSumaParcial();
71     }
72
73     double resultado = (h / 2) * sumaTotal;
74     System.out.printf("\nResultado aproximado de la integral: %.6f\n", resultado);
75 }
76
77 private static double fGlobal(double constante, double x) {
78     return constante;
79 }
80 }
```

Listing 11: Compilando y probando código para la función $f(x) = 10$

```
$ cd lab04
$ javac Trapecio.java
$ java Trapecio
=== MTODO DEL TRAPECIO BSICO CON HILOS ===
Ingrese el valor constante f(x) = 10
Ingrese el lmite inferior (a): 0
Ingrese el lmite superior (b): 5
Ingrese el nmero de subintervalos (n): 1000
Usando 8 hilos...

Resultado aproximado de la integral: 50.000000
```

- **Verificación:** Para $f(x) = 10$, la integral $\int_0^5 10 dx$ analíticamente es $10 \cdot (5 - 0) = 50,0$. El resultado de la simulación concurrente coincide con el valor exacto, demostrando la correcta implementación de la concurrencia.

Listing 12: Commit: Implementación final del Método del Trapecio Concurrente

```
$ git add .
$ git commit -m "Implementacion del metodo del Trapecio usando Java Threads"
$ git push -u origin main
```

subsectionEstructura de laboratorio 04

- El contenido que se entrega en este laboratorio es el siguiente:

```
.
|--- lab04 [DIRECTORY]
|   |--- README.md
|   |--- .gitignore
|   |--- exercises [DIRECTORY]
|   |--- Trapecio.java
|   |--- latex [DIRECTORY]
|   |--- Informe_lab04.pdf
|
3 directories, 4 files
```

5. Calificación

Tabla 1: Rúbrica para contenido del Informe y evidencias

	Contenido y demostración	Puntos	Checklist	Estudiante	Profesor
1. GitHub	Repositorio se pudo clonar y se evidencia la estructura adecuada para revisar los entregables. (Se descontará puntos por error o observación)	4	X	4	
2. Commits	Hay porciones de código fuente asociado a los commits planificados con explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
3. Ejecución	Se incluyen comandos para ejecuciones y pruebas del código fuente explicadas gradualmente que permitirían replicar el proyecto. (Se descontará puntos por cada omisión)	4	X	4	
4. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos. (Se descontará puntos por error encontrado)	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente con explicaciones puntuales pero precisas, agregando diagramas generados a partir del código fuente y refleja un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4			
Total		20		16	

6. Referencias

- <https://www.w3schools.com/java/default.asp>
- <https://www.geeksforgeeks.org/insertion-sort/>