

# **Multiprocesamiento y arquitecturas alternativas**

## **Proyecto 1er bimestre**

**Título:** Pruebas y comparaciones de programas multihilo.

**Nombre:** Luis Enrique Pérez Señalín.

### **1. INTRODUCCIÓN:**

Los computadores actuales cuentan con múltiples hilos, también conocidos como “procesadores lógicos” los cuales son una división “lógica” de un núcleo de procesador, lo que significa que de un núcleo físico pueden sacar 1 o más núcleos lógicos.

Los procesadores actuales cuentan con una gran cantidad de hilos, pero no todos los programas utilizan todos los hilos para funcionar, lo que en ciertos casos es un ahorro de consumo de recursos y en otros un desperdicio de recursos por falta de uso.

En este proyecto vamos a comparar las velocidades de procesamiento entre varios procesadores, que son de distintos fabricantes, generación y arquitectura, así como sus gamas y la cantidad de núcleos e hilos que poseen.

Vamos a utilizar el lenguaje Java implementando un algoritmo para el calculo del número de Fibonacci, el cual contará con la capacidad de elegir el número de hilos y el número de cálculos que se harán.

### **2. DESARROLLO:**

#### **2.1 Descripción del programa.**

El código del proyecto cuenta con dos partes, la principal es la llamada al algoritmo para el cálculo del número de Fibonacci que se hará utilizando multihilos en java y el algoritmo recursivo para el código de Fibonacci.

#### **2.2 Descripción del programa principal.**

El algoritmo principal funciona utilizando un grupo de números que se van a utilizar para calcular su número de Fibonacci, con el fin de distribuir estos cálculos a los demás hilos del procesador, significa que podemos tener 5 cálculos del número 45 de Fibonacci. Utilizamos de 1 a n cantidad de hilos dentro del programa, la máxima especificación solo permitirá la cantidad de hilos que tenga el equipo, si este se excede solo usará la cantidad que tiene.

#### **2.3 Descripción del programa secundario.**

Para realizar el cálculo del número de Fibonacci implementamos una función recursiva de la forma  $F_n = F_{n-1} + F_{n-2}$ , con la condición de cuando  $n \leq 1$  retorne el valor de n.

## 2.4 Parámetros de entrada del programa.

Tendremos de entrada 3 parámetros por medio de los argumentos del sistema, significa que se ingresarán al momento de ejecutar el comando de inicialización del programa, en concreto tenemos este orden:

1. El número de Fibonacci a calcular, como recomendación números mayores a 45 suelen tardar más 3 segundos en computadores actuales entonces se evitaría usar números muy elevados.
2. La cantidad de veces que se va a calcular el número de Fibonacci ingresado anteriormente.
3. El número de hilos que el programa va a utilizar, el cuál debe ser mínimo 1, si se ingresa más hilos de los que el equipo tiene, entonces solo se usarán el número de hilos máximos que tenga.

## 2.5 Utilización de múltiples hilos.

Para hacer uso de múltiples hilos en nuestro programa, usamos la librería de “ExecutorService” para crear un objeto “executor” que importamos de esta manera:

```
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;
```

La librería “ExecutorService” y “executor” nos permitirá crear un objeto de la siguiente manera:

```
ExecutorService executor = Executors.newFixedThreadPool(threads);
```

Para `fibonacci` el objeto “executor” podemos hacerlo en un ciclo repetitivo cualquiera o asignando manualmente un proceso de la siguiente manera:

```
executor.submit(() -> { fibonacci(number) });
```

De esta forma le asignamos un proceso al objeto executor.

## 2.6 Descripción de los equipos de cómputo.

Para estas pruebas se tomaron 3 diferentes equipos, de los cuales 2 tienen procesadores de tipo laptop y 1 de tipo escritorio.

- Equipo: Laptop marca Dell  
Procesador: Intel I3-7020U – 2 núcleos 4 hilos  
Memoria: 8gb ram ddr4 2400Mhz  
SO: Ubuntu 22.04 LTS.
- Equipo: Minipc Beelink SER5  
Procesador: Ryzen 5 5560U – 6 núcleos 12 hilos  
Memoria: 21gb ram ddr4 3200Mhz  
SO: Windows 11.
- Equipo: Pc escritorio  
Procesador: Intel I5-1400F – 6 núcleos 12 hilos

Memoria: 16gb ram ddr4 3200Mhz  
SO: Windows 10.

**CPU-X**

< CPU Antememorias Placa madre >

Procesador

Proveedor	Intel
Nombre clave	Kaby Lake-U (Core i3)
Paquete	
Tecnología	14 nm
Voltaje	

Especificación Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz

Familia	0x6	Modelo	0xE	Temp.	32,00°C
Ext. Familiar	0x6	Ext. Modelo	0x8E	Stepping	9

Instrucciones HT, MMX, SSE(1, 2, 3, 3S, 4.1, 4.2), AVX(1, 2), FMA(3), AES, CLMUL, RdRand, SGX, VT-x, x86-64

Relojes

Core Speed	815 MHz
Multiplificador	
Velocidad Bus	
Uso	6,05 %

Cache

L1 Datos	2 x 32 kB, 8-way
L1 Inst.	2 x 32 kB, 8-way
Nivel 2	2 x 256 kB, 4-way
Nivel 3	3 MB, 12-way

Núcleo #0 ▾ Socket(s) 1 Núcleo(s) 2 Hilos(s) 4

**CPU-Z**

CPU | Mainboard | Memory | SPD | Graphics | Bench | About |

Processor

Name	AMD Ryzen 5
Code Name	Cezanne
Package	Socket FP6
Technology	7 nm
Core Voltage	1.140 V

Specification AMD Ryzen 5 5560U with Radeon Graphics

Family	F	Model	0	Stepping	0
Ext. Family	19	Ext. Model	50	Revision	CZN-A0

Instrucciones MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AES, AVX, AVX2, FMA3, SHA

Clocks (Core #0)

Core Speed	1992.73 MHz
Multiplier	x 20.0
Bus Speed	99.65 MHz
Rated FSB	

Cache

L1 Data	6 x 32 KBytes	8-way
L1 Inst.	6 x 32 KBytes	8-way
Level 2	6 x 512 KBytes	8-way
Level 3	8 MBytes	16-way

Selection Socket #1 ▾ Cores 6 Threads 12

**CPU-Z** Ver. 2.09.0.x64 Tools ▾ Validate Close

**CPU-Z**

CPU | Mainboard | Memory | SPD | Graphics | Bench | About |

Processor

Name	Intel Core i5 11400F
Code Name	Rocket Lake
Max TDP	65.0 W
Package	Socket 1200 LGA
Technology	14 nm
Core Voltage	1.160 V

Specification 11th Gen Intel® Core™ i5-11400F @ 2.60GHz

Family	6	Model	7	Stepping	1
Ext. Family	6	Ext. Model	A7	Revision	B0

Instrucciones MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, EM64T, AES, AVX, AVX2, AVX512, FMA3, SHA

Clocks (Core #0)

Core Speed	4189.75 MHz
Multiplier	x 42.0 (8.0 - 44.0)
Bus Speed	99.76 MHz
Rated FSB	

Cache

L1 Data	6 x 48 KBytes	12-way
L1 Inst.	6 x 32 KBytes	8-way
Level 2	6 x 512 KBytes	8-way
Level 3	12 MBytes	16-way

Selection Socket #1 ▾ Cores 6 Threads 12

**CPU-Z** Ver. 2.09.0.x64 Tools ▾ Validate Close

Capturas de pantalla de las especificaciones del CPU de los equipos.

## 2.7 Comparaciones.

Vamos a realizar las comparaciones utilizando una cantidad de 36 cálculos con todos los núcleos disponibles y vamos a ir variando el número de Fibonacci que vamos a calcular.

1. Fibonacci número 44.

- 13-7020U:



Imagen i3-7020U Número 44 = 79.246 ms

- Ryzen 5 5560U:

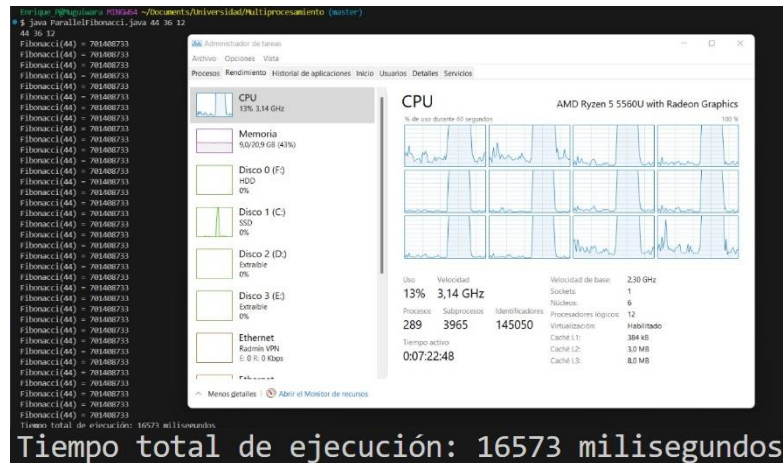


Imagen Ryzen 5 5560U Número 44 = 16.573 ms

- I5-11400F:

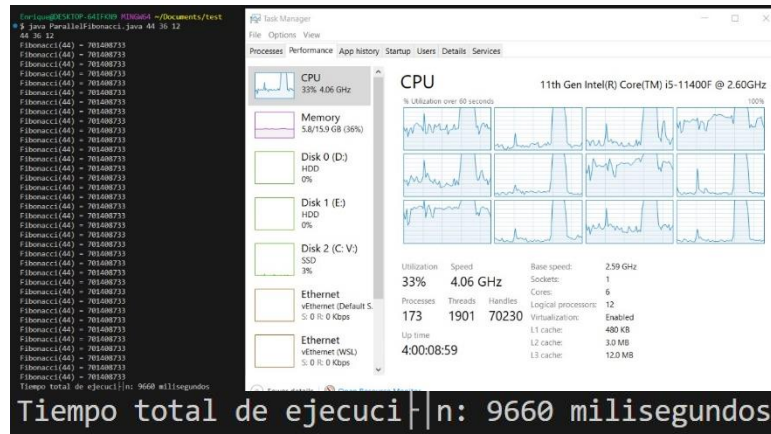


Imagen i5-11400F Número 44 = 9.660 ms

## 2. Fibonacci número 45.

- I3-7020U:



Imagen i3-7020U Número 45 = 108.668 ms

- Ryzen 5 5560U:

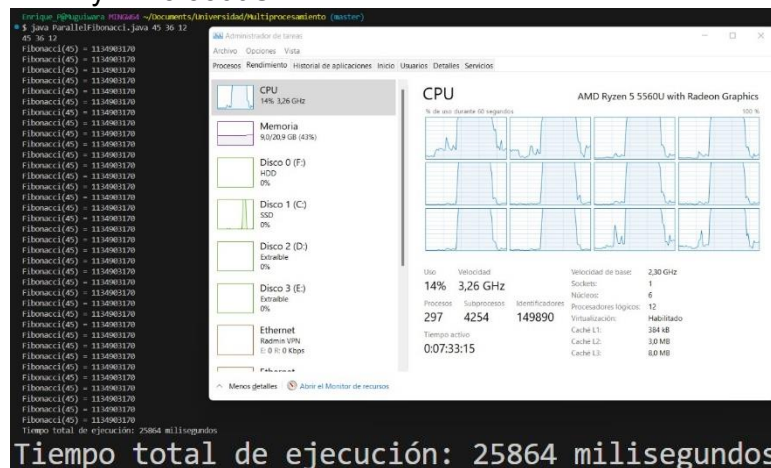


Imagen Ryzen 5 5560U Número 45 = 25.864 ms

- I5-11400F:

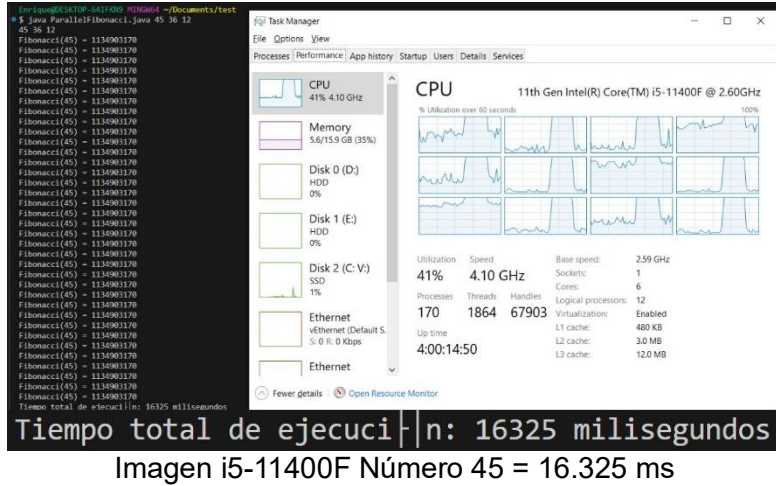


Imagen i5-11400F Número 45 = 16.325 ms

### 3. Fibonacci número 46.

- 13-7020U:

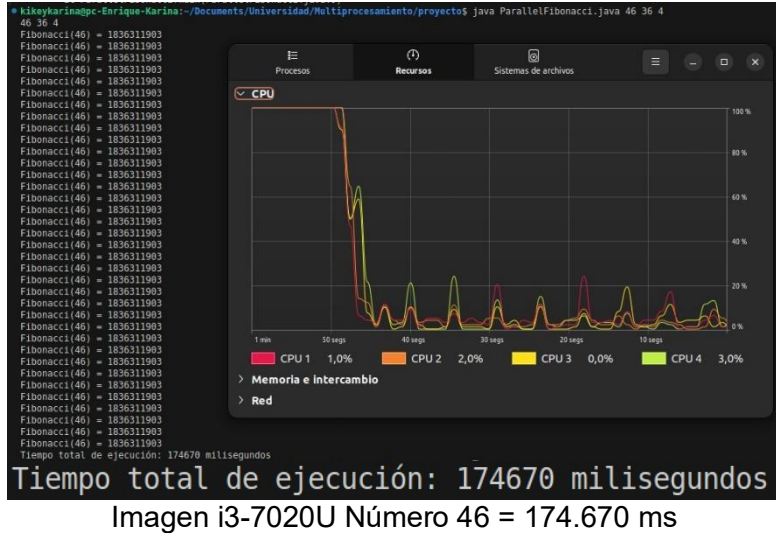


Imagen i3-7020U Número 46 = 174.670 ms

- Ryzen 5 5560U:

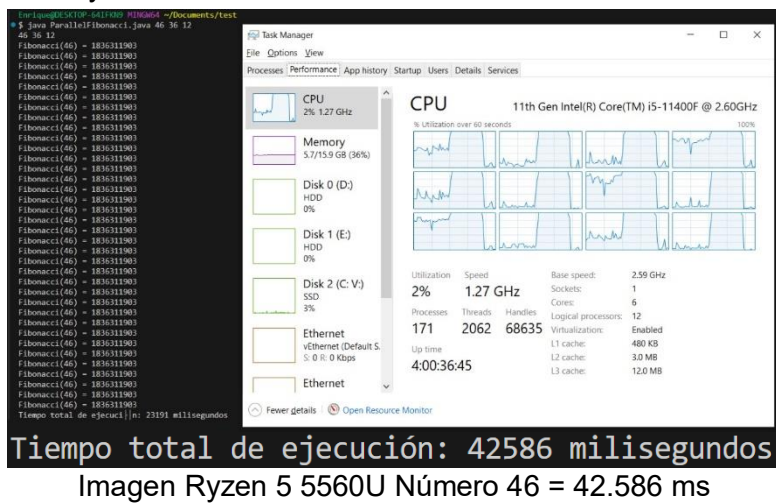


Imagen Ryzen 5 5560U Número 46 = 42.586 ms



- I5-11400F:

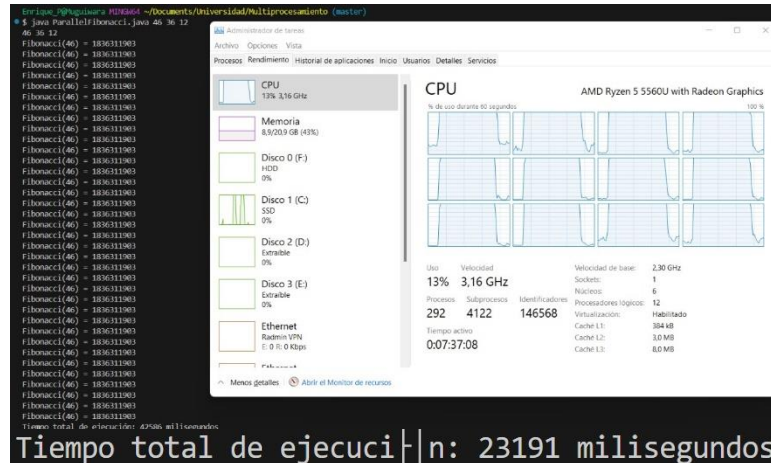


Imagen i5-11400F Número 46 = 23.191 ms

Podemos observar en las imágenes las salidas de los resultados, los cuales equivaldrían a las 36 veces del número de Fibonacci calculado, utilizando el administrador de tareas vemos el porcentaje de uso del cpu y sus hilos, y en todas las pruebas podemos observar la utilización completa de todos los hilos disponibles en cada procesador.

#### Resumen de pruebas:

- Número 44 de Fibonacci  
i3-7020U = 79.246 ms  
Ryzen 5 5560U = 16.573 ms  
i5-11400F = 9.660 ms
- Número 45 de Fibonacci  
i3-7020U = 108.668 ms  
Ryzen 5 5560U = 25.864 ms  
i5-11400F = 16.325 ms
- Número 46 de Fibonacci  
i3-7020U = 174.670 ms  
Ryzen 5 5560U = 42.586 ms  
i5-11400F = 23.191 ms

### 3. CONCLUSIONES

El uso correcto del multihilo agiliza bastante el tiempo y eficiencia de ejecución de los programas, la cantidad de hilos influye mucho en el tiempo de duración ejecutando una tarea. Existen grandes diferencias de potencia incluso entre procesadores con características similares donde puede influir el consumo del procesador como su año de salida (generación por consiguiente).