

## Laboratorio 3: Hilos y Procesos



Juan Bello Durango  
Arley Bernal Muñetón  
Kevin Garay Gaitán

Introducción a los Sistemas Distribuidos  
Profesor: John Corredor Franco

# 1. Introducción

En el presente trabajo se desarrolla una serie de pruebas orientadas a analizar el comportamiento de la ejecución de procesos en Java, tanto de forma secuencial como concurrente. Para ello, se emplean diferentes enfoques de programación, desde la ejecución simple sin hilos, hasta la utilización de clases que implementan Thread y Runnable, con el fin de comprender las ventajas del paralelismo frente a la ejecución tradicional.

El objetivo principal es observar cómo varía el desempeño y la estructura del programa según el mecanismo de ejecución utilizado. De esta manera, se busca ilustrar de manera práctica las diferencias entre concurrencia y paralelismo, resaltando cómo el uso de hilos en Java permite optimizar la gestión de tareas simultáneas.

## 2. Batería de pruebas

### Prueba 3: Ejecución Concurrente sin hilos.

Ejecutado de manera secuencial, se ejecuta el código principal utilizando las cajeras normales y los clientes. no utiliza hilos, por lo que lo hace de manera concurrente.

Se ejecutan los archivos **Main.java**, **Cajera.java** y **Cliente.java**

```
estudiante@NGEN184:~/Documents/Sistemas_Distribuidos_KG/ThreadsJarroba-master/src/threadsJarroba$ make
javac -d bin Cajera.java CajeraThread.java Cliente.java Main.java MainRunnable.java MainThread.java
estudiante@NGEN184:~/Documents/Sistemas_Distribuidos_KG/ThreadsJarroba-master/src/threadsJarroba$ make secuencial
javac -d bin Cajera.java CajeraThread.java Cliente.java Main.java MainRunnable.java MainThread.java
java -cp bin threadsJarroba.Main
La cajera Cajera 1 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 1 EN EL TIEMPO: 0seg
Procesado el producto 1 ->Tiempo: 2seg
Procesado el producto 2 ->Tiempo: 4seg
Procesado el producto 3 ->Tiempo: 5seg
Procesado el producto 4 ->Tiempo: 10seg
Procesado el producto 5 ->Tiempo: 12seg
Procesado el producto 6 ->Tiempo: 15seg
La cajera Cajera 1 HA TERMINADO DE PROCESAR Cliente 1 EN EL TIEMPO: 15seg
La cajera Cajera 2 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 2 EN EL TIEMPO: 15seg
Procesado el producto 1 ->Tiempo: 16seg
Procesado el producto 2 ->Tiempo: 19seg
Procesado el producto 3 ->Tiempo: 24seg
Procesado el producto 4 ->Tiempo: 25seg
Procesado el producto 5 ->Tiempo: 26seg
La cajera Cajera 2 HA TERMINADO DE PROCESAR Cliente 2 EN EL TIEMPO: 26seg
```

## Prueba 2: Ejecución con hilos.

Ejecuta los procesos con hilos. Se está utilizando el `MainThread`, el cual invoca el método `start ()` de la clase `CajeraThread`, lo que a su vez ejecuta el método `run ()`. De esta manera, cada cajera se ejecuta en un hilo independiente creado durante la instanciación de cada objeto Cajera. Así, la ejecución se realiza de forma paralela.

Se ejecutan los archivos **MainThread.java**, **CajeraThread.java** y **Cliente.java**

```
estudiante@NGEN184:~/Documents/Sistemas_Distribuidos_KG/ThreadsJarroba-master/src/threadsJarroba$ make threads
javac -d bin Cajera.java CajeraThread.java Cliente.java Main.java MainRunnable.java MainThread.java
java -cp bin threadsJarroba.MainThread
La cajera Cajera 2 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 2 EN EL TIEMPO: 0seg
La cajera Cajera 1 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 1 EN EL TIEMPO: 0seg
Procesado el producto 1 del cliente Cliente 2->Tiempo: 1seg
Procesado el producto 1 del cliente Cliente 1->Tiempo: 2seg
Procesado el producto 2 del cliente Cliente 1->Tiempo: 4seg
Procesado el producto 2 del cliente Cliente 2->Tiempo: 4seg
Procesado el producto 3 del cliente Cliente 1->Tiempo: 5seg
Procesado el producto 3 del cliente Cliente 2->Tiempo: 9seg
Procesado el producto 4 del cliente Cliente 1->Tiempo: 10seg
Procesado el producto 4 del cliente Cliente 2->Tiempo: 10seg
Procesado el producto 5 del cliente Cliente 2->Tiempo: 11seg
La cajera Cajera 2 HA TERMINADO DE PROCESAR Cliente 2 EN EL TIEMPO: 11seg
Procesado el producto 5 del cliente Cliente 1->Tiempo: 12seg
Procesado el producto 6 del cliente Cliente 1->Tiempo: 15seg
La cajera Cajera 1 HA TERMINADO DE PROCESAR Cliente 1 EN EL TIEMPO: 15seg
```

## Prueba 3: Ejecución con hilos usando MainRunnable

Ejecuta los procesos con hilos, pero a diferencia de la prueba anterior, en este caso es el `MainRunnable` el encargado de crear los hilos. Para ello utiliza la clase común Cajera. Así, por cada cajera que se instancia, se crea también un hilo desde el `main`, en lugar de enviar señales.

De esta manera, al usar hilos, la ejecución se realiza de forma paralela.

Se ejecutan los archivos **MainRunnable.java**, **Cajera.java** y **Cliente.java**.

```
estudiante@NGEN184:~/Documents/Sistemas_Distribuidos_KG/ThreadsJarroba-master/src/threadsJarroba$ make runnable
javac -d bin Cajera.java CajeraThread.java Cliente.java Main.java MainRunnable.java MainThread.java
java -cp bin threadsJarroba.MainRunnable
La cajera Cajera 2 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 2 EN EL TIEMPO: 0seg
La cajera Cajera 1 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 1 EN EL TIEMPO: 0seg
Procesado el producto 1 ->Tiempo: 1seg
Procesado el producto 1 ->Tiempo: 2seg
Procesado el producto 2 ->Tiempo: 4seg
Procesado el producto 2 ->Tiempo: 4seg
Procesado el producto 3 ->Tiempo: 5seg
Procesado el producto 3 ->Tiempo: 9seg
Procesado el producto 4 ->Tiempo: 10seg
Procesado el producto 4 ->Tiempo: 10seg
Procesado el producto 5 ->Tiempo: 11seg
La cajera Cajera 2 HA TERMINADO DE PROCESAR Cliente 2 EN EL TIEMPO: 11seg
Procesado el producto 5 ->Tiempo: 12seg
Procesado el producto 6 ->Tiempo: 15seg
La cajera Cajera 1 HA TERMINADO DE PROCESAR Cliente 1 EN EL TIEMPO: 15seg
```

#### Prueba 4: Ejecución con Make demo

En este caso, el comando `make demo` ejecuta las tres pruebas, lo que permite comparar de forma directa los resultados obtenidos por cada uno de los tres programas.

```
estudiante@NGEN184:~/Documents/Sistemas_Distribuidos_KG/ThreadsJarroba-master/src/threadsJarroba$ make demo
javac -d bin Cajera.java CajeraThread.java Cliente.java Main.java MainRunnable.java MainThread.java
>>> Ejecutando versión secuencial
java -cp bin threadsJarroba.Main
La cajera Cajera 1 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 1 EN EL TIEMPO: 0seg
Procesado el producto 1 ->Tiempo: 2seg
Procesado el producto 2 ->Tiempo: 4seg
Procesado el producto 3 ->Tiempo: 5seg
Procesado el producto 4 ->Tiempo: 10seg
Procesado el producto 5 ->Tiempo: 12seg
Procesado el producto 6 ->Tiempo: 15seg
La cajera Cajera 1 HA TERMINADO DE PROCESAR Cliente 1 EN EL TIEMPO: 15seg
La cajera Cajera 2 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 2 EN EL TIEMPO: 15seg
Procesado el producto 1 ->Tiempo: 16seg
Procesado el producto 2 ->Tiempo: 19seg
Procesado el producto 3 ->Tiempo: 24seg
Procesado el producto 4 ->Tiempo: 25seg
Procesado el producto 5 ->Tiempo: 26seg
La cajera Cajera 2 HA TERMINADO DE PROCESAR Cliente 2 EN EL TIEMPO: 26seg

>>> Ejecutando versión con Thread
java -cp bin threadsJarroba.MainThread
La cajera Cajera 2 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 2 EN EL TIEMPO: 0seg
La cajera Cajera 1 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 1 EN EL TIEMPO: 0seg
Procesado el producto 1 del cliente Cliente 2->Tiempo: 1seg
Procesado el producto 1 del cliente Cliente 1->Tiempo: 2seg
Procesado el producto 2 del cliente Cliente 1->Tiempo: 4seg
Procesado el producto 2 del cliente Cliente 2->Tiempo: 4seg
Procesado el producto 3 del cliente Cliente 1->Tiempo: 5seg
Procesado el producto 3 del cliente Cliente 2->Tiempo: 9seg
Procesado el producto 4 del cliente Cliente 1->Tiempo: 10seg
Procesado el producto 4 del cliente Cliente 2->Tiempo: 10seg
Procesado el producto 5 del cliente Cliente 2->Tiempo: 11seg
La cajera Cajera 2 HA TERMINADO DE PROCESAR Cliente 2 EN EL TIEMPO: 11seg
Procesado el producto 5 del cliente Cliente 1->Tiempo: 12seg
Procesado el producto 6 del cliente Cliente 1->Tiempo: 15seg
La cajera Cajera 1 HA TERMINADO DE PROCESAR Cliente 1 EN EL TIEMPO: 15seg

>>> Ejecutando versión con Runnable
java -cp bin threadsJarroba.MainRunnable
La cajera Cajera 1 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 1 EN EL TIEMPO: 0seg
La cajera Cajera 2 COMIENZA A PROCESAR LA COMPRA DEL CLIENTE Cliente 2 EN EL TIEMPO: 0seg
Procesado el producto 1 ->Tiempo: 1seg
Procesado el producto 1 ->Tiempo: 2seg
Procesado el producto 2 ->Tiempo: 4seg
Procesado el producto 2 ->Tiempo: 4seg
Procesado el producto 3 ->Tiempo: 5seg
Procesado el producto 3 ->Tiempo: 9seg
Procesado el producto 4 ->Tiempo: 10seg
Procesado el producto 4 ->Tiempo: 10seg
Procesado el producto 5 ->Tiempo: 11seg
La cajera Cajera 2 HA TERMINADO DE PROCESAR Cliente 2 EN EL TIEMPO: 11seg
Procesado el producto 5 ->Tiempo: 12seg
Procesado el producto 6 ->Tiempo: 15seg
La cajera Cajera 1 HA TERMINADO DE PROCESAR Cliente 1 EN EL TIEMPO: 15seg
```

### 3. Conclusiones

Las pruebas realizadas demuestran que la ejecución con hilos ofrece un rendimiento superior frente a la ejecución secuencial, al permitir la paralelización de tareas que, de otro modo, se realizarían de manera lineal. Asimismo, se evidenció que tanto la implementación mediante la clase Thread como la interfaz Runnable cumplen con el objetivo de generar concurrencia, aunque difieren en su estructura y forma de instanciación. Finalmente, el uso del comando make demo resultó útil para comparar los resultados de cada enfoque de manera integral.