

# Programación de Hilos en Java

## Ejercicios

### Ejercicio 1: Cálculo de Media

Implementa un programa en Java que calcule la media de un array de números enteros, utilizando hilos. El número de hilos que se deben utilizar será especificado por el usuario. El trabajo se debe dividir equitativamente entre los hilos, y cada hilo calculará la suma de una porción del array. Al final, la media total del array se deberá calcular a partir de las sumas parciales obtenidas por cada hilo.

1. Debes implementar un método `public static double calculateMean(int[] array, int numThreads)` que reciba como parámetros el array de números y el número de hilos que se utilizarán.
2. El array debe dividirse entre los hilos de forma equitativa. Si el número de elementos no es divisible por el número de hilos, el último hilo se encargará de la porción restante.
3. El método principal debe retornar la media total calculada a partir de los resultados obtenidos por los hilos.

### Ejercicio 2: Cálculo del máximo y mínimo

Implementar un programa en Java que calcule el valor máximo y mínimo de un array de números enteros, utilizando hilos. El número de hilos que se utilizarán será especificado por el usuario.

1. Debes implementar un método `public static int calculateMax(int[] array, int numThreads)` que reciba como parámetros el array de números y el número de hilos que se utilizarán.
2. Debes implementar un método `public static int calculateMin(int[] array, int numThreads)` que reciba como parámetros el array de números y el número de hilos que se utilizarán.
3. El array debe dividirse entre los hilos de forma equitativa. Si el número de elementos no es divisible por el número de hilos, el último hilo se encargará de la porción restante.

### Ejercicio 3: Generación de números aleatorios

Implementa un programa en Java que genere una cantidad específica de números aleatorios utilizando varios hilos. Cada hilo debe generar una parte de los números aleatorios y almacenarlos en un array compartido. El número de números aleatorios y el número de hilos que se deben utilizar será especificado por el usuario.

1. Debes implementar un método `public static int[] generateRandomNumbers(int numNumbers, int numThreads)` que reciba como parámetros el número de números aleatorios y el número de hilos.

2. El conjunto de números aleatorios debe dividirse equitativamente entre los hilos.
3. Cada hilo debe generar su porción de números aleatorios y almacenarlos en un array compartido.
4. El método debe retornar el array completo de números aleatorios generados.

## Ejercicio 4: Cálculo de números primos

Implementa un programa en Java que calcule cuántos números primos hay en un rango especificado, utilizando varios hilos. El rango y el número de hilos que se deben utilizar será especificado por el usuario.

### Requisitos:

1. Debes implementar un método `public static int countPrimes(int rangeStart, int rangeEnd, int numThreads)` que reciba como parámetros el inicio y el final del rango, y el número de hilos que se deben utilizar.
2. El rango de números debe dividirse equitativamente entre los hilos.
3. Cada hilo debe contar los números primos en su porción del rango.
4. El método principal debe retornar el número total de números primos en el rango.

## Ejercicio 5: Carrera de Galgos

Implementar un programa en Java que simule una carrera de galgos utilizando hilos. Cada hilo representará un galgo, y el progreso de cada caballo se debe simular utilizando `Thread.sleep()` con tiempos aleatorios. El número de caballos y la distancia de la carrera será especificado por el usuario.

### Requisitos:

1. Debes implementar un método `public static void startRace(int numDogs, int distance)` que reciba como parámetros el número de galgos y la distancia de la carrera.
2. Cada hilo debe representar un galgo y debe avanzar en la carrera en intervalos de tiempo aleatorios.
3. El programa debe imprimir el progreso de cada galgo y notificar cuándo llegue a la meta, indicando el tiempo total que ha tardado.

