

# Programación Avanzada

## Taller Individual Nro. 3

Si bien el taller es individual, una vez que resuelto y completada la información que se solicita, usted deberá trabajar en grupo de 2 o 3 personas para elaborar una tabla que se detalla más adelante.

El taller consiste en completar el proyecto que descargó desde el EVA, el mismo tiene como propósito obtener información del tiempo que tarda en ejecutar un programa, que usa programación concurrente, para determinar cuáles de los números de una lista son primos.

Entre el código encontrará una clase denominada *DataGenerator* que tiene dos métodos: *loadFromFile* y *generateTestData*, que le permitirán obtener una lista de números que usará.

Usted debe realizar lo siguiente:

1. Completar la clase *TaskCountPrime*, para que implemente el método *call* que devuelve *true* si el número que recibe como atributo es primo y *false* si no lo es.
2. Para determinar si el número es primo, debe usar los siguientes algoritmos
  1. `return !IntStream.range(2, num).anyMatch(div -> num % div == 0);`
  2. `return !IntStream.rangeClosed(2, num/2).anyMatch(div -> num % div == 0);`
  3. Un método que usted implemente, buscando que sea lo más rápido posible.
3. Completar el método *execWorkSeveralThreadsListNums* en la clase *AppExec*, para que compile y pueda ejecutar el programa.
4. Trabajar con los datos que devuelva el método *generateTestData* y ejecutar el programa, con cada algoritmo del punto 2, con 2, 4, 8, 16 y 32 hilos, para lo cual debe asignar cada uno de esos valores a la variable de la línea 20.
5. Debe rellenar la información que se muestra a continuación. Con el programa completo se generara una salida con tres tiempos expresados en segundos. Usted debe registrar ese tiempo en una hoja electrónica que tendrá la forma que se muestra más abajo y debe calcular el tiempo promedio por cada fila.  
Información a completar:

**Algoritmo:** 1, 2 o 3 (ver punto 2)

**Nombre sistema operativo:**

**Versión del sistema operativo:**

**Versión de Java:**

**Velocidad del procesador:**

### Número de cores:

Nro.Hilos	Ejecución 1	Ejecución 2	Ejecución 3	Promedio
2				
4				
8				
16				
32				

- Determinar cuál es el número ideal de hilos con los que trabajar, es decir, cuando el tiempo de ejecución es menor (ver la columna promedio).
- Repetir los puntos desde 4 hasta 6, pero desde 32 hilos en adelante (con potencias de 2, es decir: 32, 64, 128, etc.) con los datos que devuelve el método *loadFromFile* y encontrar el número ideal de hilos con los que trabajar. Recuerde crear una tabla similar para estos datos.

Finalmente, cuando termine su trabajo, comparta sus datos con 1 o 2 compañeros (a quienes tiene sentado a lado) y elaboren tablas resumen, para los tiempos promedio que obtuvieron de forma individual, identifique por columnas los datos de cada compañero. Comparen sus resultados, vean similitudes en hardware y software y traten de encontrar posibles razones que expliquen las diferencias de tiempo si las hubieran.

**Nota:** De forma individual generará 6 tablas, una por cada algoritmo, son 3 algoritmos, y 2 grupos de datos, generados por los métodos: *generateTestData* y *loadFromFile*. Cuando trabaje en grupo generará el mismo número de tablas, pero sólo con las columnas: números de hilos y promedio de cada uno de los integrantes.

Opcionalmente puede generar una gráfica por cada tabla (denominada de dispersión o scatter plot) como se muestra a continuación.

