

**Adrian Hernández Mesa**  
**NIA 10041439**

### Practica I

Para la resolución de la practica se utilizó el IDE ECLIPSE.

#### I. Practica-Medición Simple de Latencias.

En este ejercicio se usó la librería HdrHistogram, luego de la importación de la librería se siguieron los pasos mostrados en la parte no presencial para la creación del histograma asignando los valores mínimos(lowestDiscernibleValue), máximos(highestTrackableValue) y el valor de significancia(significantValueDigits) que determina la precisión de los cálculos:

```
Histogram hg = new Histogram(lowestDiscernibleValue, highestTrackableValue, significantValueDigits);
```

Luego se utilizó la función autorezisable para asegurarse que el histograma funcione, aunque salga del rango esperado. `hg.setAutoResize(true);`

Luego se realiza un bucle donde se definen el startTime y el endTime mediante la función `System.nanoTime()`, antes y después de `syncOpSimulator.executeOp()` que permite ver el tiempo transcurrido de la aplicación en nanosegundos.

Finalmente se imprime el histograma, escalando a microsegundos,

```
hg.outputPercentileDistribution(System.out, 1000.00);
```

Resultando,

Value	Percentile	TotalCount	1/(1-Percentile)	
862.207	0.999780273438	99979	4551.11	
883.711	0.999804687500	99981	5120.00	
937.983	0.999829101563	99983	5851.43	
1173.503	0.999853515625	99986	6826.67	
1491.967	0.999877929688	99988	8192.00	
1599.487	0.999890136719	99990	9102.22	
1673.215	0.999902343750	99991	10240.00	
1806.335	0.999914550781	99992	11702.86	
2353.151	0.999926757813	99993	13653.33	
2387.967	0.999938964844	99994	16384.00	
2697.215	0.999945068359	99995	18204.44	
2971.647	0.999951171875	99996	20480.00	
2971.647	0.999957275391	99996	23405.71	
3102.719	0.999963378906	99997	27306.67	
3102.719	0.999969482422	99997	32768.00	
5763.071	0.999972534180	99998	36408.89	
5763.071	0.999975585938	99998	40960.00	
5763.071	0.999978637695	99998	46811.43	
6381.567	0.999981689453	99999	54613.33	
6381.567	0.999984741211	99999	65536.00	
6381.567	0.999986267090	99999	72817.78	
6381.567	0.999987792969	99999	81920.00	
6381.567	0.999989318848	99999	93622.86	
12664.831	0.999990844727	100000	109226.67	
12664.831	1.000000000000	100000		
#[Mean	=	51.835, StdDeviation	=	62.541]
#[Max	=	12664.831, Total count	=	100000]
#[Buckets	=	14, SubBuckets	=	2048]

## II. Practica-Medición con y sin calentamiento.

En este ejercicio se usó la librería HdrHistogram, al igual que para la creación del histograma se usó la misma función Histogram definiendo el mínimo, el máximo y el valor de significancia,

```
Histogram hg = new Histogram(1,100000L,2);  
hg.setAutoResize(true);
```

Luego se utilizó un bucle en el cual dentro de la misma llamada se ejecuta carga y se resetea (hg.rset()) el histograma, para ir calentando el sistema, dentro de este se pintan los valores obtenido para ver los resultados.

```
for (int j=0; j<100; j++) {  
    for (int i = 0; i < 200000; i++) {  
        long startTime = System.nanoTime();  
        syncOpSimulator.executeOp();  
        long endTime = System.nanoTime() - startTime;  
        hg.recordValue(endTime);  
    }  
    hg.reset();  
}
```

Se utilizaron los métodos de hg.getMean(), hg.getMinValue(), hg.getMaxValue(), hg.getValueAtPercentile(99) para la obtención de los resultados.

Sistema Frio	Sistema Caliente, repetición número:100
Mean: 9593.9424	Mean: 5477.3824
Min: 6784	Min: 4672
Max: 128974847	Max: 905215
percentile 99: 14911	percentile 99: 7423
percentile 99,9: 72703	percentile 99,9: 56319

Los resultados muestran que el sistema en frio muestra peores resultados de latencia que los resultados en caliente. Cuando el sistema esta caliente los resultados se optimizan.

## III. Practica-Medición de latencias acumuladas y multi-hilo.

Nuevamente se usó la librería HdrHistogram, se crearon dos histogramas con SynchronizedHistogram, ya que permite acceder desde varios hilos a todos sus métodos sin que produzca problemas de concurrencia, uno que mida la latencia normal y el otro que mida la latencia acumulada.

```
SynchronizedHistogram hg = new SynchronizedHistogram(1,100000L,2);  
SynchronizedHistogram hgAcumulado = new SynchronizedHistogram(1,100000L,2);
```

Para la realización de la latencia acumulada se calcula el tiempo antes de la ejecución y luego se va almacenando en una variable para ver su tiempo de ejecución, obteniendo:

Salida para 1 hilo	
Latencia Mean: 10.63 Min: 10 Max: 35	Latencia Acumulada Mean: 11.04 Min: 10 Max: 38
Salida para 8 hilos	
Latencia Mean: 71.375 Min: 10 Max: 743	Latencia Acumulada Mean: 3285.54375 Min: 10 Max: 6911

Se puede notar que a mayor cantidad de hilos ejecutados mayor es la latencia promedio.