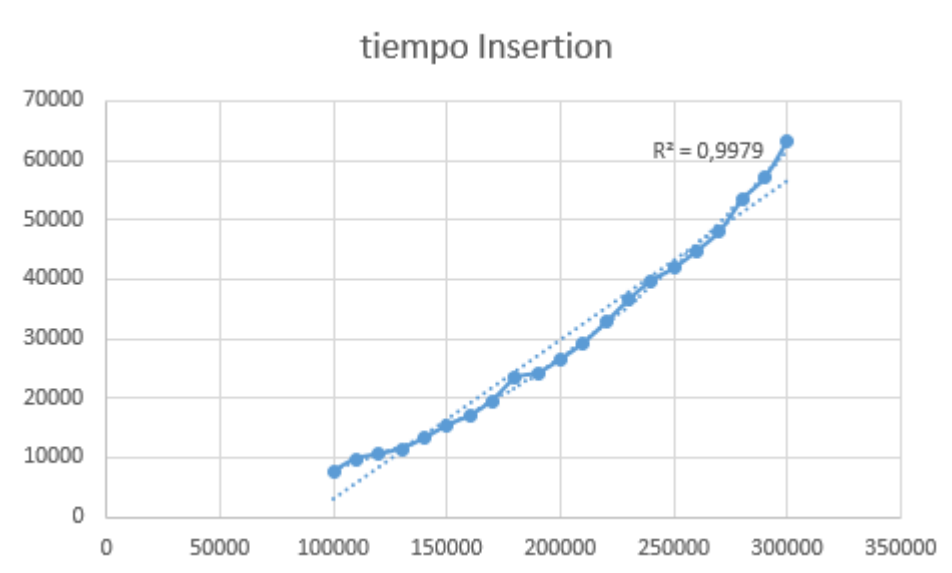


## Laboratorio Nro. X: Notación O grande

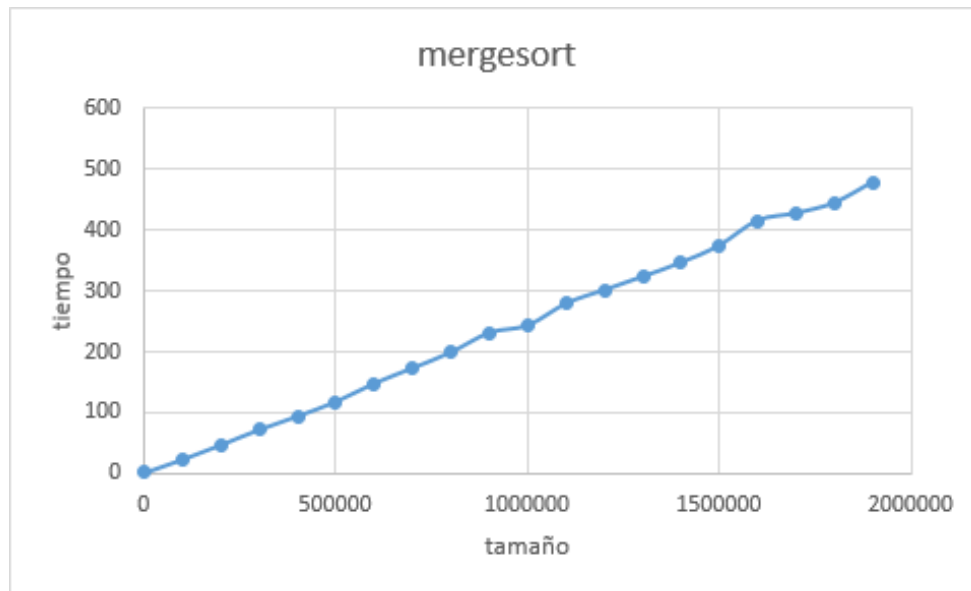
**Jose Miguel Gil Valencia**  
Universidad Eafit  
Medellín, Colombia  
jmgilv@eafit.edu.co

**Mauricio Castaño Uribe**  
Universidad Eafit  
Medellín, Colombia  
mcastanou@eafit.edu.co

3)



**DOCENTE MAURICIO TORO BERMÚDEZ**  
Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627  
Correo: mtorobe@eafit.edu.co



tamaño	tiempo Insertion
100000	7655
110000	9758
120000	10515
130000	11474
140000	13177
150000	15383
160000	17003
170000	19568
180000	23297
190000	24212
200000	26464
210000	29221
220000	32732
230000	36508
240000	39752
250000	41844
260000	44741
270000	48029
280000	53385
290000	57120
300000	63348

Tamaño	mergesort tiempo
1000	1
101000	23
201000	47
301000	72
401000	94
501000	118
601000	148
701000	173
801000	199
901000	231
1001000	243
1101000	279
1201000	301
1301000	323
1401000	346
1501000	374
1601000	415
1701000	427
1801000	444
1901000	478

**3.3** Notablemente Merge Sort es mucho mejor que Insertion Sort para resolver problemas de altos volúmenes de datos, debido a su complejidad.

Insertion Sort requiere menos memoria que el Merge Sort, al momento de crear variables.

Complejidad:

Merge Sort:  $n \log(n)$  Vs Insertion Sort:  $n^2$

### 3.5 y 3.6

count events

La complejidad es de:

$O(n)$

Donde n es el tamaño del arreglo

maxSpam

La complejidad es de:

$O(n^2)$

Donde n es el tamaño del arreglo

lucky13

La complejidad es de:

$O(n)$

Donde n es el tamaño del arreglo

more14

La complejidad es de:

$O(n)$

Donde n es el tamaño del arreglo

tripleUp

La complejidad es de:

$O(n)$

Donde n es el tamaño del arreglo

zeroFront

La complejidad es de:

$O(n)$

Donde n es el tamaño del arreglo

Fix45

La complejidad es de:

$O(n^2)$

Donde n es el tamaño del arreglo