

```
alicia@alicia-Inspiron-1525:~$ lscpu
Arquitectura:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Orden de bytes:         Little Endian
CPU(s):                 1
On-line CPU(s) list:    0
Hilo(s) por núcleo:     1
Núcleo(s) por zócalo: 1
Socket(s):              1
Nodo(s) NUMA:           1
ID del vendedor:        GenuineIntel
Familia de CPU:          6
Modelo:                 22
Stepping:               1
CPU MHz:                1861.849
BogoMIPS:               3723.69
caché L1d:              32K
caché L1i:              32K
caché L2:               1024K
NUMA node0 CPU(s):      0
alicia@alicia-Inspiron-1525:~$
```

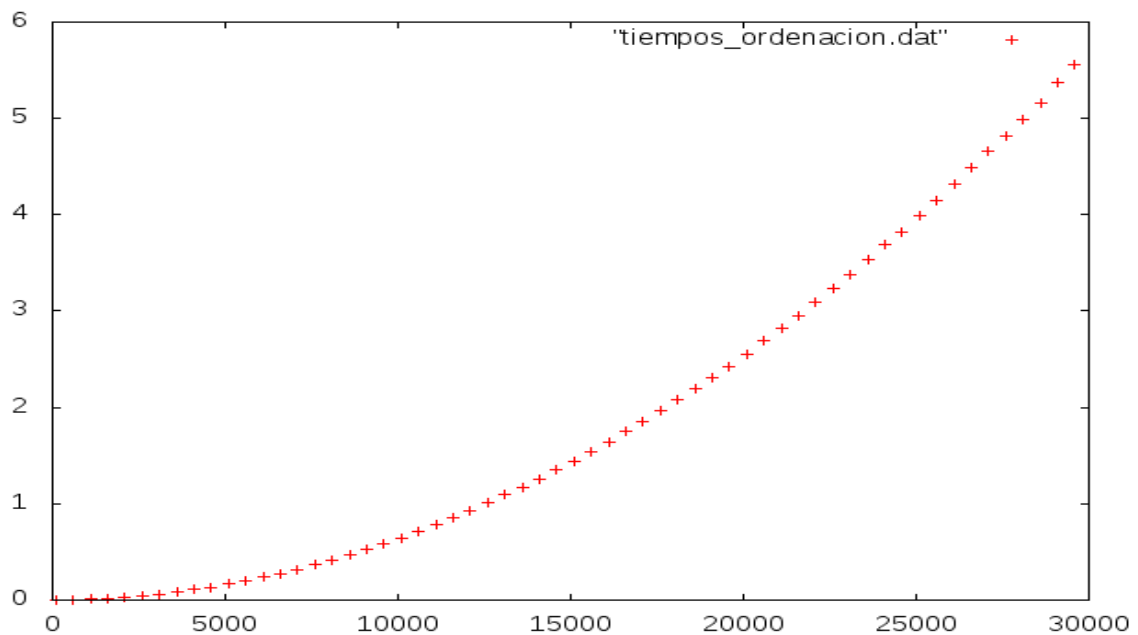
1440 x 900 píxeles 130,0 kB 146 % 1/2

Ejercicio 6: Influencia del proceso de compilación

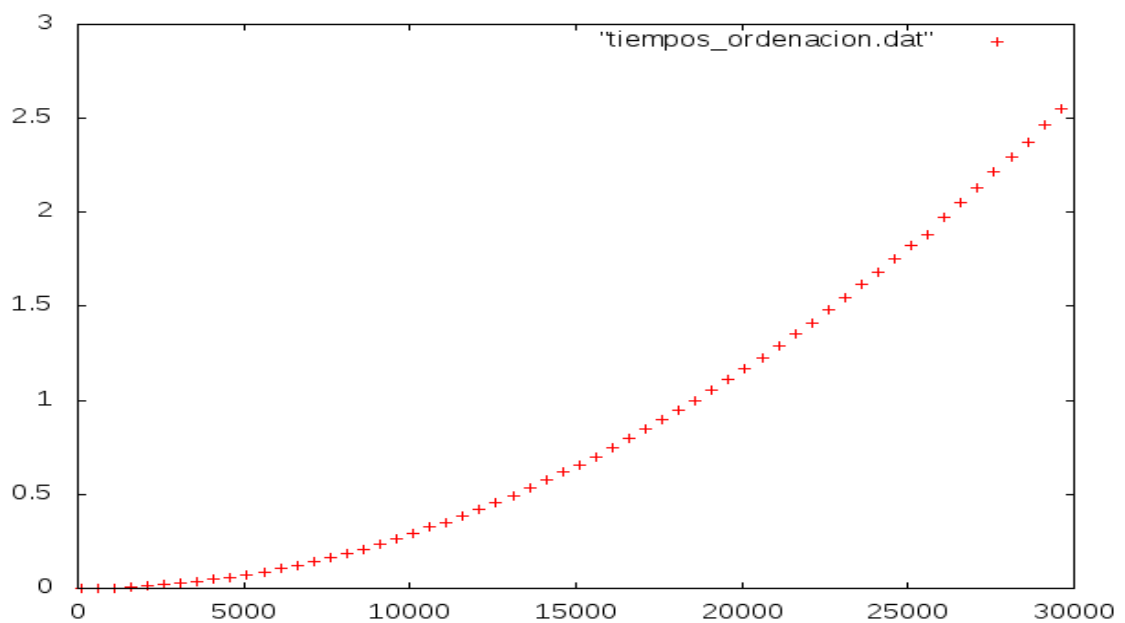
Retome el ejercicio de ordenación mediante el algoritmo de la burbuja. Ahora replique dicho ejercicio pero previamente deberá compilar el programa indicándole al compilador que optimice el código. Esto se consigue así:

```
g++ -O3 ordenacion.cpp -o ordenacion_optimizado
```

Compare las curvas de eficiencia empírica para ver cómo mejora esto la eficiencia del programa.

Eficiencia empírica código ordenación sin optimizar:Eficiencia empírica código ordenación optimizado:

Tras compilar con: **g++ -O3 ordenacion.cpp -o ordenacion_optimizado**
y modificar en **ejecuciones_ordenacion.csh** el ejecutable por **ordenacion_optimizado** la grafica es la siguiente:



Comparación entre las curvas de eficiencia empírica para ver cómo mejora esto la eficiencia del programa:

Se puede observar gran diferencia en los tiempos, ya que en el programa optimizado el tiempo es la mitad comparado con el programa sin optimizar.