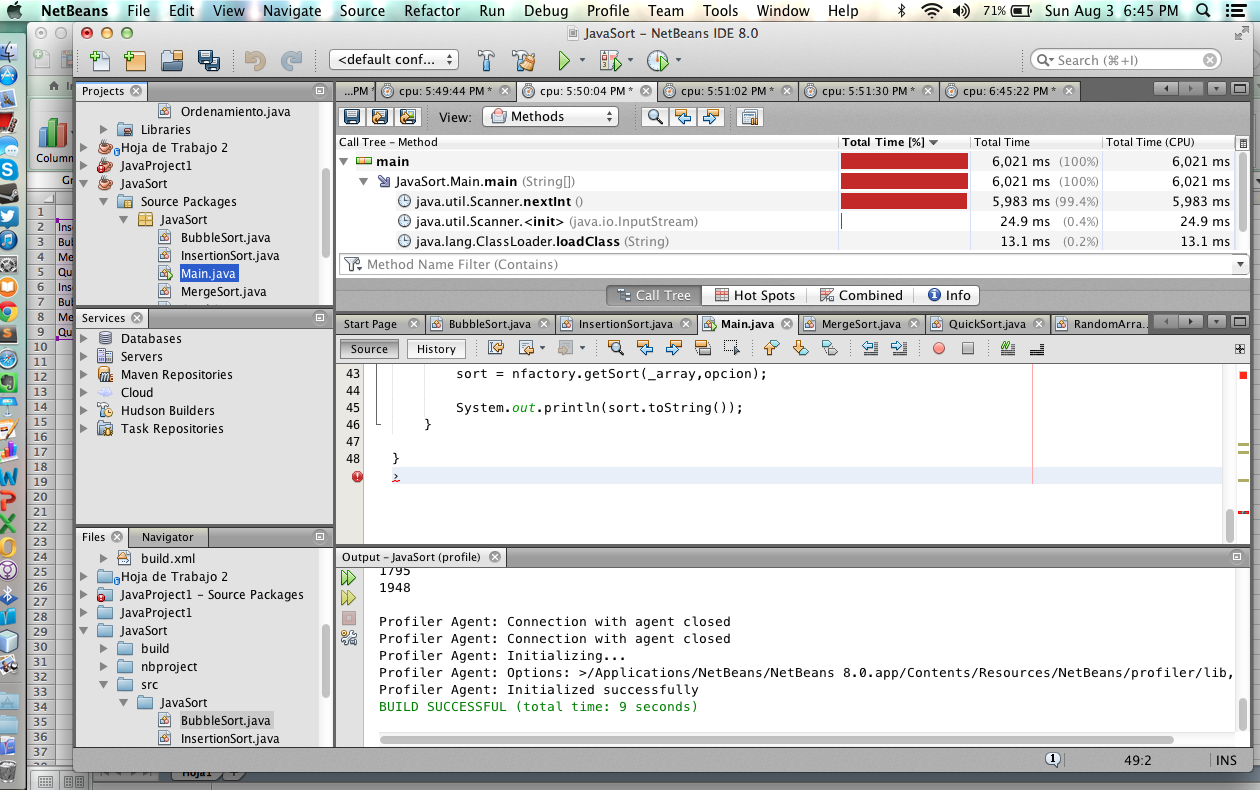
HOJA DE TRABAJO 3

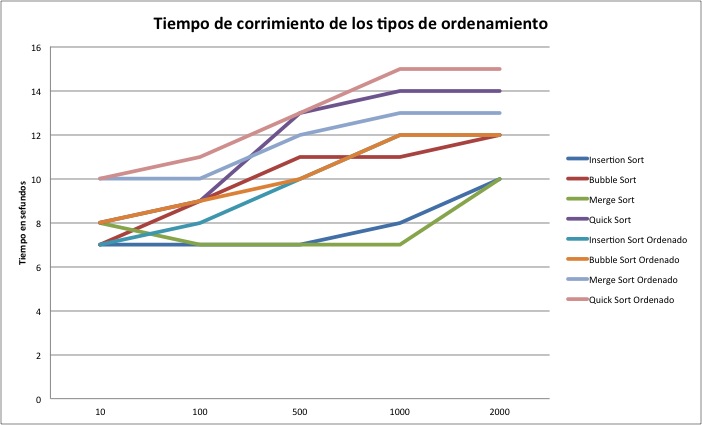
Se utilizó el profiler de Neatbeans IDE 8.0 se empleo corriendo el programa y seleccionando la opción de sort que se deseaba utilizar, al finalizar el ordenamiento apareció el tiempo de corrimiento del programa; se hizo éste procedimiento con cada tipo de sort. Como evidencia de ésto se puede observar la imagen 1.



*Imagen 1. Utilización del profiler*

Los resultados obtenidos por el profiler fueron los siguientes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tiempo en segundos** | | | | |
| **Cantidad de números** | **10** | **100** | **500** | **1000** | **2000** |
| Insertion Sort | 7 | 7 | 7 | 8 | 10 |
| Bubble Sort | 7 | 9 | 11 | 11 | 12 |
| Merge Sort | 8 | 7 | 7 | 7 | 10 |
| Quick Sort | 8 | 9 | 13 | 14 | 14 |
| Insertion Sort Ordenado | 7 | 8 | 10 | 12 | 12 |
| Bubble Sort Ordenado | 8 | 9 | 10 | 12 | 12 |
| Merge Sort Ordenado | 10 | 10 | 12 | 13 | 13 |
| Quick Sort Ordenado | 10 | 11 | 13 | 15 | 15 |



*Gráfica1. Tipos de ordenamientos y sus tiempos obtenidos con el profiler*

La gráfica muestra el tiempo de corrimiento de los tipos de ordenamiento utilizando arrays ordenados y desordenados de tamaño 10, 100, 500, 1000 y 2000. Como se puede observar en la gráfica 1. El tipo de ordenamiento más rápido para arrays presentados de forma desordenada es el merge sort, teniendo un tiempo de 7 segundos con arrays de tamaño 100, 500 y 1000.

Cómo se ve en la gráfica es más rápido ordenar los números cuándo éstos están de forma desordenada, que ordenada, debido a que cuando están de forma ordenada la computadora realiza doble trabajo, ya que vuelve a revisar casilla por casilla del arreglo y así hasta llegar al final, por lo lleva más tiempo recorrer todo el array que irlo ordenando de una vez. Sin embargo el tipo de ordenamiento que menos tiempo se hace en verificar que el array esté arreglado ya es el bubble sort.