**Tabla comparación de tiempos:**

|  |  |  |
| --- | --- | --- |
| Numero de Datos | Tiempo Bubble Sort (segundos) | Tiempo Bucket Sort (segundos) |
| 10000 | 0.118 | 0.003 |
| 20000 | 0.886 | 0.005 |
| 30000 | 1.688 | 0.007 |
| 40000 | 3.948 | 0.008 |
| 50000 | 5.543 | 0.01 |
| 60000 | 7.947 | 0.012 |
| 70000 | 11.804 | 0.013 |
| 80000 | 14.769 | 0.015 |
| 90000 | 17.586 | 0.016 |
| 100000 | 22.422 | 0.017 |
| 110000 | 29.119 | 0.018 |
| 120000 | 33.445 | 0.019 |
| 130000 | 37.844 | 0.019 |
| 140000 | 44.723 | 0.021 |
| 150000 | 54.128 | 0.021 |
| 160000 | 60.29 | 0.022 |
| 170000 | 66.605 | 0.024 |
| 180000 | 76.47 | 0.023 |
| 190000 | 87.183 | 0.025 |
| 200000 | 94.534 | 0.026 |
| 210000 | 106.748 | 0.026 |
| 220000 | 117.601 | 0.027 |
| 230000 | 131.629 | 0.028 |
| 240000 | 144.964 | 0.029 |
| 247047 | 153.738 | 0.033 |

**Conclusiones:**

A partir de los datos obtenidos podemos ver que entre más cantidad de datos más eficiente es el Bucket Sort en comparación del Bubble Sort. Mientras que el Bubble Sort aumenta rápidamente a medida que incrementamos los datos, el Bucket Sort mantiene un tiempo casi constante, lo cual refleja su mejor rendimiento cuando trabaja con grandes volúmenes de datos. Por lo cual, si vamos a trabajar con muchos datos Bucket Sort sería la mejor opción.