

Hoja de Trabajo #9

➤ **Repositorio del proyecto en GitHub:**

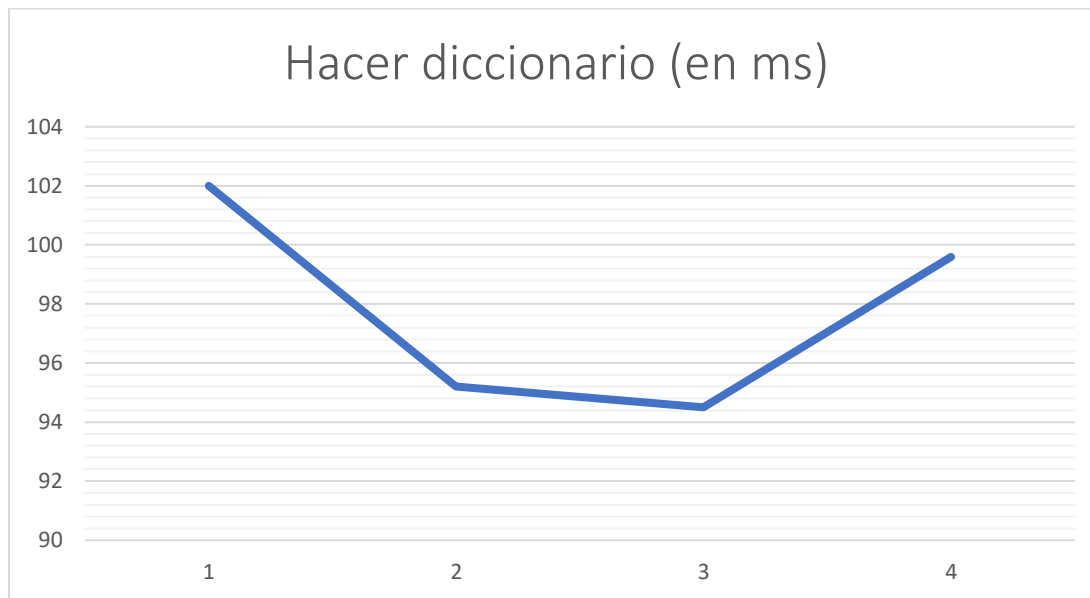
<https://github.com/tej17584/HTB9-Hoja7CopyPaste>

➤ **Profiler para la medición de tiempos:**

Implementación Red-Black Tree:

Corrida	Hacer diccionario (en ms)	Traducir el archivo (en ms)
1	102	3.77
2	95.2	3.85
3	94.5	3.2
4	99.6	3.84

Gráfica de los tiempos:



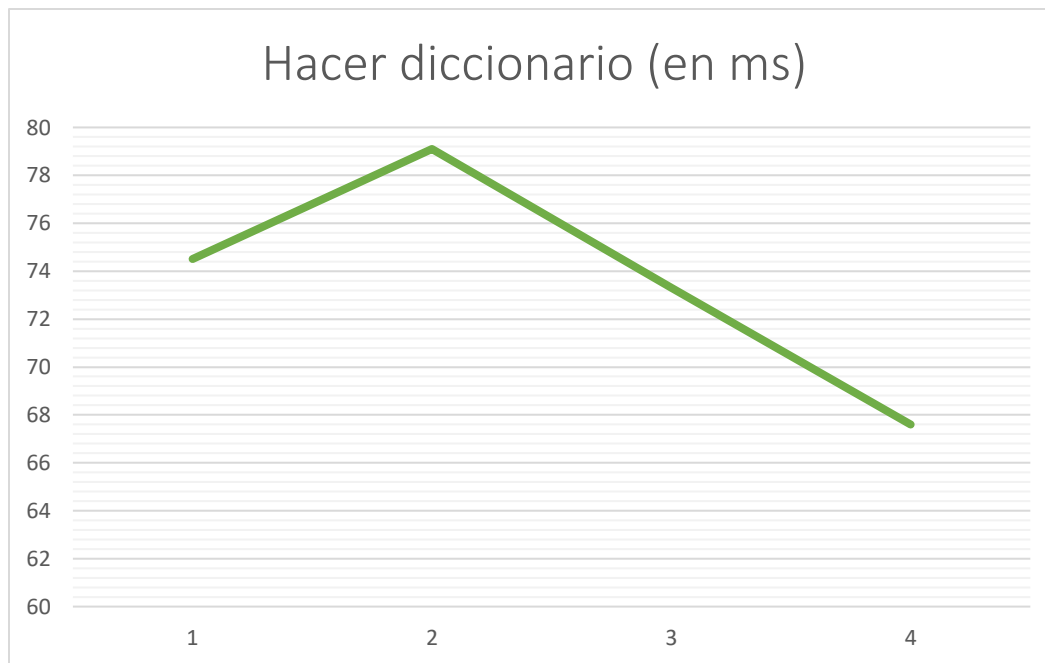
Profiler:

Profile: Selected methods Selected 1 method, edit Limit outgoing calls: 5 Filter outgoing calls: Exclude core Java classes, edit Apply				
Name	Total Time	Total Time (CPU)	Invocations	
main	3,202 ms (100%)	96.6 ms (100%)	1	
hojadetrabajo9.HojaDeTrabajo9.main (String[])	3,202 ms (100%)	96.6 ms (100%)	1	
hojadetrabajo9.ProcesosRBT.diccionarioRBT ()	99.1 ms (3.1%)	81.0 ms (83.8%)	1	
hojadetrabajo9.ProcesosRBT.traducirArchivoRBT ()	3.85 ms (0.1%)	0.0 ms (0%)	1	
hojadetrabajo9.ProcesosRBT.hacerDiccionarioRBT ()	95.2 ms (3%)	81.0 ms (83.9%)	1	
Self time	0.065 ms (0%)	0.0 ms (0%)	1	
hojadetrabajo9.ProcesosRBT.<init> ()	0.009 ms (0%)	0.0 ms (0%)	1	
hojadetrabajo9.Procesos.<init> ()	0.010 ms (0%)	0.0 ms (0%)	1	
hojadetrabajo9.Factory.getImp (String)	1.93 ms (0.1%)	0.0 ms (0%)	1	
hojadetrabajo9.Factory.<init> ()	0.006 ms (0%)	0.0 ms (0%)	1	
Self time	3,101 ms (96.8%)	15.6 ms (16.2%)	1	

Implementación Splay Tree:

Corrida	Hacer diccionario (en ms)	Traducir el archivo (en ms)
1	74.5	14.6
2	79.1	12.5
3	73.3	13.4
4	67.6	16.6

Gráfica de los tiempos:



Profiler:

Profile Results View: Collected data: Process:				
Profile: Selected methods Selected 1 method, edit Limit outgoing calls: 5 Filter outgoing calls: Exclude core Java classes, edit Apply				
Name	Total Time	Total Time (CPU)	Invocations	
main	7,244 ms (100%)	82.3 ms (100%)	1	
hojadetrabajo9.HojaDeTrabajo9.main (String[])	7,244 ms (100%)	82.3 ms (100%)	1	
hojadetrabajo9.ProcesosRBT.<init> ()	0.007 ms (0%)	0.0 ms (0%)	1	
hojadetrabajo9.Procesos.diccionarioST ()	91.6 ms (1.3%)	66.7 ms (81%)	1	
hojadetrabajo9.Procesos.traducirArchivoST ()	12.5 ms (0.2%)	0.0 ms (0%)	1	
hojadetrabajo9.Procesos.hacerDiccionarioST ()	79.1 ms (1.1%)	66.7 ms (81.1%)	1	
Self time	0.061 ms (0%)	0.0 ms (0%)	1	
hojadetrabajo9.Procesos.<init> ()	0.008 ms (0%)	0.0 ms (0%)	1	
hojadetrabajo9.Factory.getImp (String)	0.915 ms (0%)	0.0 ms (0%)	1	
hojadetrabajo9.Factory.<init> ()	0.007 ms (0%)	0.0 ms (0%)	1	
Self time	7,151 ms (98.7%)	15.6 ms (19%)	1	

➤ **Explicación en la diferencia de tiempos:**

La diferencia de los tiempos en las mismas implementaciones se debe a varios factores. Entre ellos, el consumo de PC que se está ejecutando en el mismo momento en el que se llama al método y la frecuente solicitud de operaciones al S.O.

En cuanto a los árboles, se puede ver que el Red-Black Tree es más eficiente al traducir los archivos, es decir, hace en un mejor tiempo la traducción de las palabras en el txt. El Splay Tree, por otra parte, es más eficiente al hacer el diccionario (o árbol) aunque le toma más tiempo traducir el txt. Esto se debe a que, el Splay Tree se forma conforme se ingresan los datos, por lo que no tiene un orden adecuado para hacer un diccionario. De igual forma, cabe mencionar que, al ser un sistema de recomendaciones, puede llegar a ser más útil en cuanto a traducir una palabras con varios significados.