

Hoja de Trabajo No. 6

Conjuntos de Desarrolladores

*Respuestas a los incisos c y d

c) Después de correr el programa con las tres implementaciones y realizar el profiler con ayuda de NetBeans; se obtuvieron los siguientes tiempos:

	TreeSet (ns)	HashSet (ns)	LinkedHashSet (ns)
1 nombre	67920	61230	62590
2 nombres	69890	65420	66783
3 nombres	39130	45930	58425
Promedio	58980.00	57526.67	62599.33

Se realizó el mismo procedimiento, 3 veces. En cada una de las pruebas se agregaba un nombre más; esto con el fin de tener mayor cantidad de datos para hacer una mejor conclusión con respecto a cuál de las implementaciones es más rápida. Y con los datos anteriores, se puede afirmar que la implementación de HashSet fue la más rápida de los tres; quedando casi empatada TreeSet (comparándolas con respecto al promedio).

d) Debido a que solo se utiliza el método containsAll; como se muestra a continuación:

```
/**
 * @return conjunto de desarrolladores Java es un subconjunto de Desarrolladores Android
 */
public String javaSubConjAndroid(){
    if(DesAndroid.containsAll(DesJava)== true)
        return "Si";
    else
        return "No";
}
```

Se llega a la conclusión de que la complejidad es $O(c)$.

Información Extra:

Set implementations:

	add	contains	next	notes
HashSet	$O(1)$	$O(1)$	$O(h/n)$	h is the table capacity
LinkedHashSet	$O(1)$	$O(1)$	$O(1)$	
CopyOnWriteArraySet	$O(n)$	$O(n)$	$O(1)$	
EnumSet	$O(1)$	$O(1)$	$O(1)$	
TreeSet	$O(\log n)$	$O(\log n)$	$O(\log n)$	
ConcurrentSkipListSet	$O(\log n)$	$O(\log n)$	$O(1)$	