

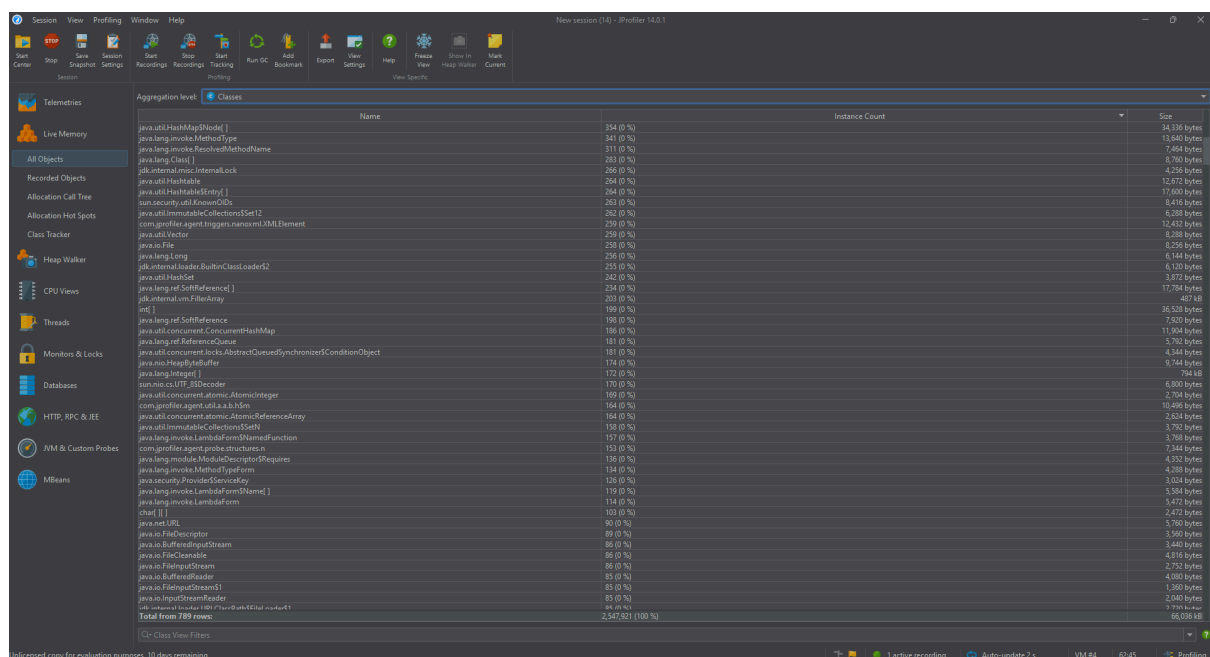
Hoja de Trabajo 3

Github: <https://github.com/HUG0BA/sorting.git>

Profiler Utilizado:

El profiler utilizado es JProfiler que básicamente es una herramienta de java que permite tener un control del tiempo de ejecución de cualquier programa. Este es capaz de dar el rendimiento y su tiempo de ejecución.

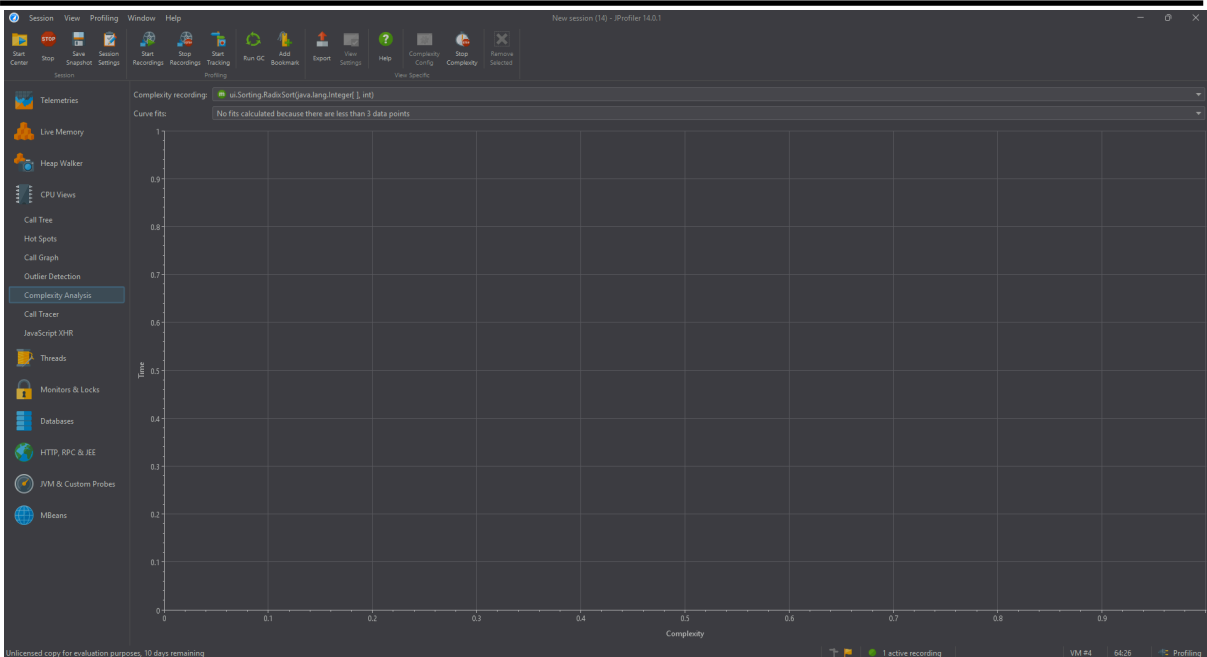
Gráficas:



The screenshot shows the JProfiler interface with the 'Classes' tab selected. The table lists various Java classes and their memory usage statistics.

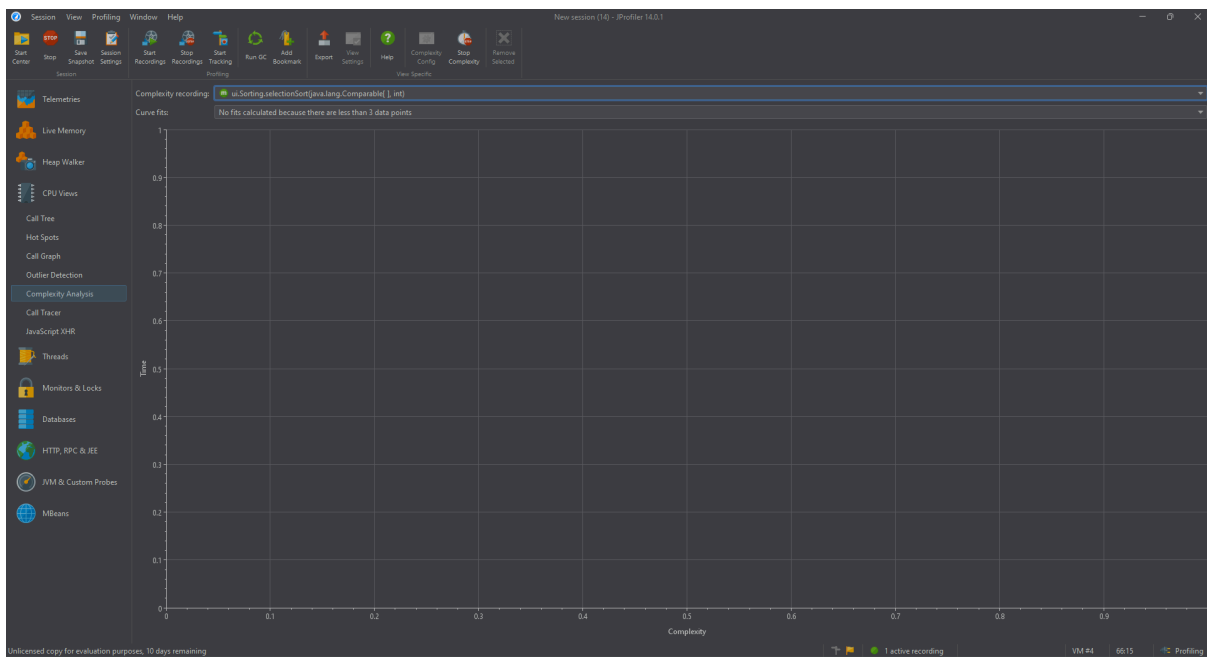
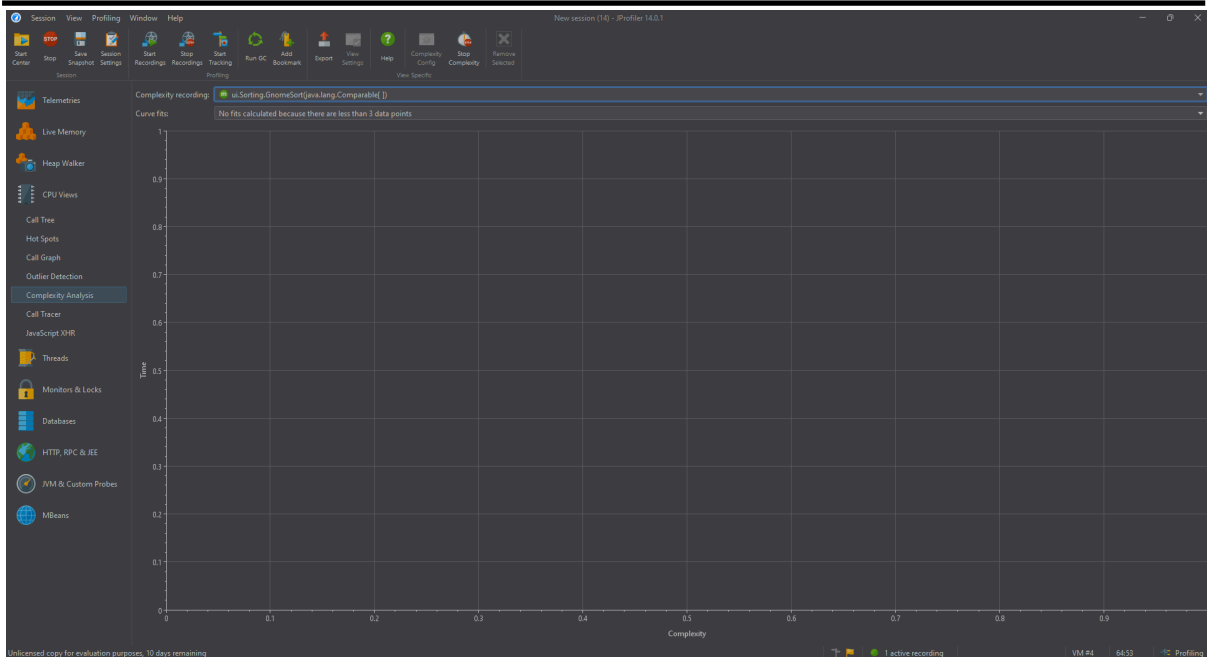
Name	Instance Count	Size
java.util.HashMap\$Node[]	354 (0 %)	34,330 bytes
java.lang.invoke.MethodType	341 (0 %)	13,640 bytes
java.lang.invoke.ResolvedMethodName	311 (0 %)	2,464 bytes
java.lang.Class[]	283 (0 %)	6,760 bytes
jdk.internal.misc.InternedLock	266 (0 %)	4,256 bytes
java.util.HashMap	264 (0 %)	12,672 bytes
java.util.HashMap\$Entry[]	264 (0 %)	17,600 bytes
sun.security.util.KnownOIDs	263 (0 %)	8,416 bytes
java.util.ImmutableCollections\$Set12	262 (0 %)	8,208 bytes
com.jprofiler.agent.triggers.nanoxml.XMLElement	259 (0 %)	12,432 bytes
java.util.Vector	259 (0 %)	8,288 bytes
java.io.File	258 (0 %)	2,224 bytes
java.lang.Long	256 (0 %)	6,144 bytes
jdk.internal.loader.BuiltinClassLoader\$2	255 (0 %)	6,120 bytes
java.util.HashSet	242 (0 %)	3,872 bytes
java.lang.ref.SoftReference[]	234 (0 %)	17,764 bytes
jdk.internal.vm.FillerArray	203 (0 %)	487 kB
int[]	199 (0 %)	36,528 bytes
java.lang.ref.SoftReference	198 (0 %)	7,620 bytes
java.util.concurrent.ConcurrentHashMap	186 (0 %)	11,904 bytes
java.lang.ref.ReferenceQueue	181 (0 %)	5,728 bytes
java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject	181 (0 %)	4,344 bytes
java.nio.HeapByteBuffer	174 (0 %)	9,744 bytes
java.lang.Integer[]	172 (0 %)	744 kB
sun.misc.UTF_32Decoder	170 (0 %)	6,800 bytes
java.util.concurrent.atomic.AtomicInteger	169 (0 %)	2,704 bytes
com.jprofiler.agent.util.abi.bdm	164 (0 %)	16,496 bytes
java.util.concurrent.atomic.AtomicReferenceArray	164 (0 %)	2,624 bytes
java.util.ImmutableCollections\$SetN	158 (0 %)	3,792 bytes
java.lang.invoke.LambdaForm\$NameRefFunction	157 (0 %)	3,768 bytes
com.jprofiler.agent.probe.threadcurator	153 (0 %)	7,344 bytes
java.lang.module.ModuleDescriptor\$Requires	138 (0 %)	4,352 bytes
java.lang.invoke.MethodType\$Form	134 (0 %)	4,208 bytes
java.security.Provider\$ServiceKey	126 (0 %)	3,024 bytes
java.lang.invoke.LambdaForm\$Name[]	119 (0 %)	5,584 bytes
java.lang.invoke.LambdaForm	114 (0 %)	5,472 bytes
char[]	103 (0 %)	2,472 bytes
java.net.URL	90 (0 %)	5,760 bytes
java.io.PushbackInputStream	89 (0 %)	3,568 bytes
java.io.BufferedInputStream	86 (0 %)	3,440 bytes
java.io.FileChannel	86 (0 %)	4,816 bytes
java.io.FileInputStream	86 (0 %)	2,728 bytes
java.io.BufferedReader	85 (0 %)	4,080 bytes
java.io.FileInputStream\$1	85 (0 %)	1,360 bytes
java.io.InputStreamReader	85 (0 %)	2,040 bytes
jdk.internal.loader.Loader\$FileClassLoader\$Module\$1	84 (0 %)	9,728 bytes
Total from 789 rows	2,547,921 (100 %)	66,038 kB

Hugo Barillas #23556
Ernesto Ascencio #23009



Hugo Barillas #23556

Ernesto Ascencio #23009



Hugo Barillas #23556

Ernesto Ascencio #23009

