

PRÁCTICA 1. Complejidad.

Sesión 1. Manipulación de Imágenes. Cálculo de la Complejidad Teórica.

Es evidente que la manipulación de imágenes está en auge. Por ello, es importante que los programas que las manipulan sean eficientes. En esta práctica vais a analizar la eficiencia de algunos métodos existentes de tratamiento básico de imágenes. Por tanto, debéis hacer lo siguiente:

1. Implementar un programa que invoque a los siguientes métodos, cuyo código se os proporciona en el fichero *Auxiliar.java*:
 - a. Método **GenerarImagenGris**(String ImagenEntrada, String ImagenSalida): toma como entrada el nombre del archivo *ImagenEntrada*, que contiene una imagen, la transforma a escala de grises, y la guarda en otro archivo, de nombre *ImagenSalida*. El nombre de la imagen de salida será el mismo que la de entrada añadiendo la terminación *_g*, por ejemplo, si la entrada es "320x214.png" la salida será "320x214_g.png". Se debe comprobar si la imagen de salida está representada en escala de grises.
 - b. Método **int[] Histograma**(String Ruta): recibe como entrada la ruta de un archivo que contiene una imagen, transforma esta a escala de grises, calcula su histograma y lo proporciona como resultado.
 - c. Método **ImprimeHistograma**(int[] Histograma): recibe como entrada un vector de enteros que contiene los valores del histograma y los muestra por pantalla.
 - d. Método **GenerarImagenOrdenandoColumnas**(String ImagenEntrada, String ImagenSalida, int Metodo): transforma la imagen del archivo *ImagenEntrada* a escala de grises y ordena sus columnas de manera ascendente, según el algoritmo de la burbuja si *Metodo* es 0, o según el algoritmo Quicksort si *Metodo* es 1. Una vez ordenadas todas las columnas, vuelca la información en el archivo de imagen *ImagenSalida*. El nombre de la imagen de salida será el mismo que la de entrada añadiendo la terminación *"_b"*, si el método de ordenación es el de la burbuja, y la terminación *"_q"* si el método de ordenación es el Quicksort. Por ejemplo, si la entrada es "320x214.png" las salidas serán "320x214_b.png" y "320x214_q.png", respectivamente.

Para invocar a los métodos, se os proporcionan 5 archivos de imágenes de distintas resoluciones, con nombres "320x214.png", "640x360.png", "640x427.png", "1024x1024.png" y "1536x1536.png". Están disponibles en Moodle en el fichero **Imágenes.zip**. Además, podéis hacer uso de los métodos de ordenación y de lectura de datos que se os proporcionan en los archivos *Ordenar.java* y *leerAlumnos.java*, respectivamente, junto con el archivo *Auxiliar.java* en **Auxiliares.zip**.

2. Calcular, **de forma teórica**, la complejidad de cada uno de los métodos anteriores.

Sesión 2. Manipulación de Imágenes. Cálculo de la Complejidad Empírica y Representación Gráfica.

Debéis realizar lo siguiente:

1. Implementar un programa que sirva para calcular empíricamente la complejidad de los métodos utilizados en la Sesión 1. El tiempo de ejecución se calculará en milisegundos o en nanosegundos, en función de lo que desee el usuario. Esta opción se escogerá al inicio del programa. Para los cálculos se pueden usar los métodos estáticos **nanoTime()** y **currentTimeMillis()** de la clase **System**, que proporcionan un valor de tipo *long* que representa la hora actual del sistema en nanosegundos y milisegundos, respectivamente.
2. Determinar, cada uno de los miembros del equipo en vuestro computador, los tiempos de ejecución de los métodos especificados anteriormente, tomando como entrada de cada uno de ellos las imágenes "320x214.png", "640x360.png", "640x427.png", "1024x1024.png" y "1536x1536.png".
3. Generar un fichero con formato .csv (texto separado con comas) con los resultados.
4. Abrir el archivo con un programa de hoja de cálculo, tipo MS Excel, y generar un gráfico con los datos del archivo .csv.
5. Comparar los resultados obtenidos por cada miembro del equipo o, incluso, en distintas ejecuciones.
6. Justificar, razonadamente, si los resultados empíricos se adaptan a los teóricos.

A continuación, se muestra parte de una ejecución del programa en un ordenador **(1)**, el contenido de la carpeta de imágenes después de la ejecución **(2)**, y la representación del archivo de resultados junto con el gráfico asociado con MS Excel **(3)**.

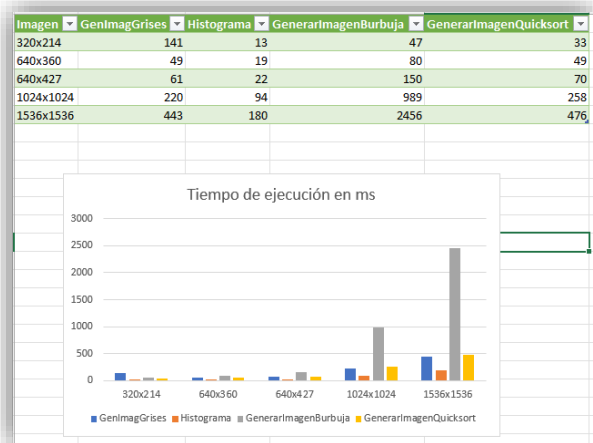
```
----- Procesando imagen 320x214 -----
Imagen de grises generada
Histograma generado
29 39 37 30 27 10 17 6 10 28 21 18 21 14 26 17
Columnas ordenadas por burbuja
Columnas ordenadas por quicksort

----- Procesando imagen 640x360 -----
Imagen de grises generada
Histograma generado
135054 14619 5647 6909 3415 2882 2079 1701
Columnas ordenadas por burbuja
Columnas ordenadas por quicksort
```

(1) Parte de los resultados de un ejemplo de ejecución (observa que el histograma está cortado en la figura)

```
imágenes
1024x1024_b.png
1024x1024_g.png
1024x1024_q.png
1024x1024.png
320x214_b.png
320x214_g.png
320x214_q.png
320x214.png
```

(2) Contenido de la carpeta de imágenes con las generadas por el programa



(3) Archivo de resultados abierto en MS Excel y gráfico asociado