# Unidad 2: Persistencia en ficheros

## Actividad propuesta Act 1.3

**Acceso a datos**

**Actividades 4.4 (pág. 57)**

**1. Reescribir el programa anterior para que, en vez de leer y escribir byte a byte, lo haga en bloques de 128 bytes.**

**2. Escribir un programa para detectar el formato de un archivo de imagen. Para detectar el tipo de archivo, se deben leer los primeros bytes. Aquí tienes una tabla con los bytes de cabecera para cada formato en notación hexadecimal:**

|  |  |
| --- | --- |
| Tipo de archivo | Bytes de encabezado (offset = 0) |
| BMP | 42 4D |
| .GIF | 47 49 46 38 39 61 / 37 61 |
| ICO | 00 00 01 00 |
| JPEG | FF D8 FF |
| .PNG | 89 50 4E 47 |

Para obtener más información sobre los encabezados de los ficheros se puede consultar

el siguiente enlace: <https://en.wikipedia.org/wiki/List_of_file_signatures>

Se debe tener en cuenta también que, si se usa directamente el valor hexadecimal (en Java se pueden escribir anteponiendo 0x, por ejemplo, 0x42), estos valores, al contrario que los bytes, son sin signo, por lo que se deberá hacer el casting adecuado o usar el método Byte.ToUnsignedInt.

**3. Escribir un programa para leer el encabezado de un archivo BMP e informar su tamaño, ancho, alto y si la imagen está comprimida. La cabecera de un fichero BMP es un bloque de 54 bytes con la siguiente estructura (se indica sólo la información necesaria para completar el ejercicio):**

|  |  |  |
| --- | --- | --- |
| Offset | Tamaño | Descripción |
| 0 | 2 | Firma, debe ser 42 4D (HEXA) |
| 2 | 4 | Tamaño del fichero en bytes |
| … |  |  |
| 18 | 4 | Ancho de la imagen en pixels |
| 22 | 4 | Alto de la imagen en pixels |
| … |  |  |
| 30 | 4 | Compresión (0=no, 1=RLE-8, 2=RLE-4) |

Se puede consultar el formato completo en el siguiente enlace: <https://es.wikipedia.org/wiki/Windows_bitmap>

Como posiblemente habréis notado, a la hora de almacenar un valor, por ejemplo, entero, en un archivo binario, hay que hacerlo byte a byte. Esto implica que, para leerlo, tendremos que reconstruir el valor, en función de si el formato elegido para almacenar ha sido liteEndian (como es el caso) o BioEndian. Se puede dite más información sobre esto en el siguiente enlace: <https://es.wikipedia.org/wiki/Endianness>