



INSTITUTO POLITÉCNICO
NACIONAL
UNIDAD PROFESIONAL
INTERDISCIPLINARIA EN INGENIERÍA
Y TECNOLOGÍAS AVANZADAS



PRÁCTICA 1: "IMAGEN .BMP"

MATERIA: MULTIMEDIA

ALUMNO: MARTINEZ BARRUETA MARIANA

BOLETA: 2024640160

análisis imagen “example001.bmp”

Utilizando hedex.it

HEADER

1)	Signature	2 bytes	0000h	'BM'
----	-----------	---------	-------	------

Los primeros dos bytes son los siguientes:

00000000 42 4D 36 03 00 00 00 00 | 00 00 36 00 00 00 28 00 BM6.....6....(.

Los cuales corresponden a la **firma del archivo** BM que indica que esto es un archivo **BMP**.

2)	FileSize	4 bytes	0002h	File size in bytes
----	----------	---------	-------	--------------------

Los siguientes cuatro bytes son:

00000000 42 4D 36 03 00 00 00 00 | 00 00 36 00 00 00 28 00 BM6.....6....(.

Los cuales corresponden al tamaño del archivo.

Tamaño del archivo: **36030000** -> **00000336** se convierte a decimal -> **822 bytes**.

3)	reserved	4 bytes	0006h	unused (=0)
----	----------	---------	-------	-------------

Los siguientes cuatro bytes son:

00000000 42 4D 36 03 00 00 00 00 | 00 00 36 00 00 00 28 00 BM6.....6....(.

Los cuales corresponden a los bytes reservados.

4)	DataOffset	4 bytes	000Ah	Offset from beginning of file to the beginning of the bitmap data
----	------------	---------	-------	---

Los siguientes cuatro bytes son:

00000000 42 4D 36 03 00 00 00 00 | 00 00 36 00 00 00 28 00 BM6.....6....(.

Los cuales corresponden donde empieza el data, es decir, el byte de la imagen real.

Byte inicio de imagen: 36000000 -> 00000036 se convierte a decimal -> **byte 54**

INFOHEADER

5)	Size	4 bytes	000Eh	Size of InfoHeader =40
----	------	---------	-------	------------------------

Los siguientes cuatro bytes son:

00000000 42 4D 36 03 00 00 00 00 | 00 00 36 00 00 00 28 00 BM6.....6....(.

00000010 00 00 10 00 00 00 10 00 | 00 00 01 00 18 00 00 00

Los cuales corresponden al tamaño del infoheader.

Tamaño: 28000000 -> 00000028 se convierte a decimal -> **40 bytes**

6)	Width	4 bytes	0012h	Horizontal width of bitmap in pixels
----	-------	---------	-------	--------------------------------------

Los siguientes cuatro bytes corresponden al ancho de la imagen:

00000010 00 00 10 00 00 00 10 00|00 00 01 00 18 00 00 00

Ancho: 10000000 -> 00000010 se convierte en decimal -> **16 pixels**

7)	Height	4 bytes	0016h	Vertical height of bitmap in pixels
----	--------	---------	-------	-------------------------------------

Los siguientes cuatro bytes corresponden al alto de la imagen:

00000010 00 00 10 00 00 00 10 00|00 00 01 00 18 00 00 00

Alto: 10000000 -> 00000010 se convierte en decimal -> **16 pixels**

8)	Planes	2 bytes	001Ah	Number of Planes (=1)
----	--------	---------	-------	-----------------------

Los siguientes dos bytes corresponden a los planos de color:

00000010 00 00 10 00 00 00 10 00|00 00 01 00 18 00 00 00

Planos: 0100 ->0010 convirtiendo a decimal -> **1 plano.**

9)

Bits Per Pixel	2 bytes	001Ch	Bits per Pixel used to store palette entry information. This also identifies in an indirect way the number of possible colors. Possible values are: 1 = monochrome palette. NumColors = 1 4 = 4bit palletized. NumColors = 16 8 = 8bit palletized. NumColors = 256 16 = 16bit RGB. NumColors = 65536 24 = 24bit RGB. NumColors = 16M
----------------	---------	-------	---

Los siguientes dos bytes son:

00000010 00 00 10 00 00 00 10 00|00 00 01 00 18 00 00 00

Los cuales corresponden a los bytes por píxel.

Bytes por píxel: 1800 -> 0018 convirtiendo a decimal -> **24 bits = 3 bytes**, por lo tanto, cada píxel contiene tres bytes y esto se utilizará para identificar los colores de la imagen en el formato **RGB**.

10)	Compression	4 bytes	001Eh	Type of Compression 0 = BI_RGB no compression 1 = BI_RLE8 8bit RLE encoding 2 = BI_RLE4 4bit RLE encoding
-----	-------------	---------	-------	--

Los siguientes cuatro bytes indican si existe compresión en la imagen.

00000010 00 00 10 00 00 00 10 00|00 00 01 00 18 00 00 00

00000020 00 00 00 03 00 00 00 00|00 00 00 00 00 00 00 00

Como el resultado es "0" significa que la imagen no está sin compresión.

11)	ImageSize	4 bytes	0022h	(compressed) Size of Image It is valid to set this =0 if Compression = 0
-----	-----------	---------	-------	---

Los siguientes cuatro bytes corresponden al tamaño de la imagen:

00000020 00 00 00 03 00 00 00 00 | 00 00 00 00 00 00 00 00

Tamaño de la imagen: 00300000 -> 0000300 se convierte a decimal -> **768 bytes**.

Y la imagen es de 16 x 16 y en cada píxel hay 3 bytes, por lo tanto, $(16 \times 16) \times 3 = 768$ bytes, lo cual es igual al resultado anterior.

12)	XpixelsPerM	4 bytes	0026h	horizontal resolution: Pixels/meter
-----	-------------	---------	-------	-------------------------------------

Los siguientes cuatro bytes corresponden a la resolución horizontal en pixeles por metro:

00000020 00 00 00 03 00 00 00 00 | 00 00 00 00 00 00 00 00

Como el resultado es "0" no está definida.

13)	YpixelsPerM	4 bytes	002Ah	vertical resolution: Pixels/meter
-----	-------------	---------	-------	-----------------------------------

Los siguientes cuatro bytes corresponden a la resolución vertical en pixeles por metro:

00000020 00 00 00 03 00 00 00 00 | 00 00 00 00 00 00 00 00

Como el resultado es "0" no está definida.

14)	Colors Used	4 bytes	002Eh	Number of actually used colors. For a 8-bit / pixel bitmap this will be 100h or 256.
-----	-------------	---------	-------	--

Los siguientes cuatro bytes corresponden al número de colores usados en la imagen:

00000020 00 00 00 03 00 00 00 00 | 00 00 00 00 00 00 00 00

00000030 00 00 00 00 00 00 FF FF | FF FF FF FF FF FF FF

Esto solo aplica para BMP de: 1 bit, 4 bits, 8 bits por píxel y la imagen es de 24 bits, entonces usa millones de colores, por lo tanto, como el resultado es "0" no tiene paleta de colores.

15)	Important Colors	4 bytes	0032h	Number of important colors 0 = all
-----	------------------	---------	-------	---------------------------------------

Los siguientes cuatro bytes corresponden al número importante de colores:

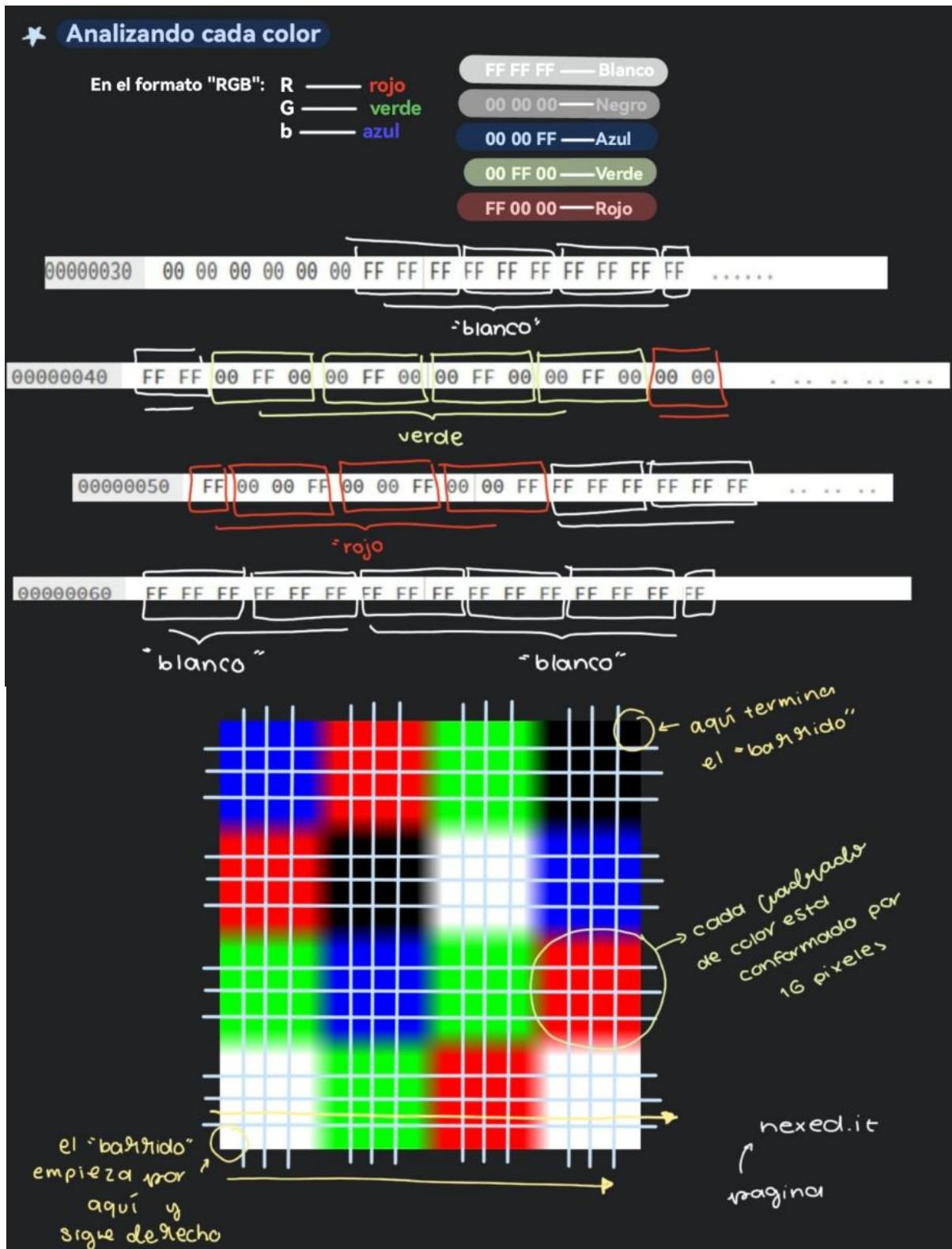
00000030 00 00 00 00 00 00 FF FF | FF FF FF FF FF FF FF

Hasta aquí termina el “metadata” porque el “dataoffset” nos dice que desde el byte 54 se empieza la imagen real.

Ahora sigue el barrido de la imagen, es decir, en qué sentido y orden irán los colores.

BARRIDO

Cada píxel tiene 24 bits y esto es equivalente a 3 bytes y para formar un color se necesita la combinación "RGB" donde cada color está formado por 8 bits equivalente a 1 byte, por lo tanto, cada 3 bytes será un color.



El barrido va renglón por renglón, de izquierda a derecha y de abajo hacia arriba.