



INSTITUTO POLITÉCNICO  
NACIONAL  
UNIDAD PROFESIONAL  
INTERDISCIPLINARIA EN INGENIERÍA  
Y TECNOLOGÍAS AVANZADAS



# *Práctica 2:* **IMAGEN VOLCAN.BMP**



Alumno: Martínez Barrueta Mariana

Profesor: Sierra Romero Noe

Boleta: 2024640160

*Multimedia*

# análisis imagen "volcan.bmp"

Utilizando hedex.it

## HEADER

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

Los primeros dos bytes son los siguientes:

42 4D 36 80 70 00 00 00 00 00 36 00 00 00 28 00 BM6Çp.....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:

42 4D 36 80 70 00 00 00 00 00 36 00 00 00 28 00 BM6Çp.....6... (

Los cuales corresponden al tamaño del archivo.

Tamaño del archivo: **36807000** -> **00708036** se convierte a decimal -> **7,372,854**

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

Los siguientes cuatro bytes son:

42 4D 36 80 70 00 00 00 00 00 36 00 00 00 28 00 BM6Çp.....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:

42 4D 36 80 70 00 00 00 00 00 36 00 00 00 28 00 BM6Çp.....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:

42 4D 36 80 70 00 00 00 00 00 36 00 00 00 28 00 BM6Çp.....6... (

00 00 80 07 00 00 00 05 00 00 01 00 18 00 00 00 ..Ç.....

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:

00 00 80 07 00 00 00 05 00 00 01 00 18 00 00 00 ..Ç.....

Ancho: 80070000 -> 00000780 se convierte en decimal -> **1920 pixeles**

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

Los siguientes cuatro bytes corresponden al alto de la imagen:

00 00 80 07 00 00 00 05 00 00 01 00 18 00 00 00 ..Ç.....

Alto: 00050000 -> 00000500 se convierte en decimal -> **1280 pixeles**

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

Los siguientes dos bytes corresponden a los planos de color:

00 00 80 07 00 00 00 05 00 00 01 00 18 00 00 00 ..Ç.....

Planos: 0100 -> 0001 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:

00 00 80 07 00 00 00 05 00 00 01 00 18 00 00 00 ..Ç.....

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.

00 00 80 07 00 00 00 05 00 00 01 00 18 00 00 00 ..Ç.....

00 00 00 80 70 00 00 00 00 00 00 00 00 00 00 00 ...Çp.....

Como el resultado es "0" significa que la imagen no esta 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:

00 00 00 80 70 00 00 00 00 00 00 00 00 00 00 00 ...Çp.....

Tamaño de la imagen: 00807000 ->00708000 se convierte a decimal -> **7,372,800 bytes.**

Y la imagen es de 1920 x 1280 y en cada píxel hay 3 bytes, por lo tanto, (1920x1280) x3 = 7,372,800 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:

00 00 00 80 70 00 00 00 00 00 00 00 00 00 00 00 ...Çp.....

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:

00 00 00 80 70 00 00 00 00 00 00 00 00 00 00 00 ...Çp.....

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:

00 00 00 80 70 00 00 00 00 00 00 00 00 00 00 00 ...Çp.....

00 00 00 00 00 00 05 01 0D 02 00 0A 03 00 0B 00 .....

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:

00 00 00 00 00 00 05 01 0D 02 00 0A 03 00 0B 00 .....

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