

# FORMATO BMP

Jesús Antonio Jiménez Aviña

Hunab.

ESCOM - IPN.

México D.F. a 6 de Mayo de 1996

En la actualidad existen en el mercado diversos tipos de **formatos para almacenar imágenes**, su diversidad es muy grande. Algunos de los formatos mas populares son GIF, TIFF, JPG, BMP, etc. Todos estos formatos son comerciales por lo cual no es fácil encontrar información interna sobre ellos.

Uno de los formatos mas sencillos es el BMP (Mapa de Bits). Un BMP como todo archivo con formato consta de dos partes primordiales. **Cabecera y cuerpo.**

La cabecera esta compuesta de: la identificaron del formato del archivo, dirección de inicio del cuerpo, tamaño horizontal de la imagen, tamaño vertical de la imagen, el numero de bits por pixel y la paleta de colores.

Con la información de la cabecera obtenemos el tamaño de imagen. el tipo de resolución del archivo (si es de 2, 16 o 256 colores), y la paleta de colores propia de esa imagen.

En el cuerpo de la imagen se encuentra el valor del color de cada pixel, que se ira acomodando en la posición correspondiente de acuerdo al tamaño de horizontal y vertical, podemos pensar en algún momento que la imagen esta guardada de arriba hacia abajo lo cual es falso. El primer pixel del cuerpo corresponde a la parte inferior izquierda de la imagen, los demás pixeles se irán acomodando de izquierda a derecha y de abajo hacia arriba.

La ubicación de la información dentro del archivo es la siguiente:

BYTE	SIGNIFICADO	nombre de la variable
0	Identificación del archivo.	
10	Lugar donde empieza el cuerpo.	CUERPO
18	Tamaño horizontal.	TAM_X
22	Tamaño vertical.	TAM_Y
28	Bits por pixel.	BIT_PIXEL
30	Principio de la paleta de colores.	PALETA
XX	El principio del cuerpo se calcula por medio de los bits por pixel y la paleta de colores.	

La paleta de colores es una tabla donde se tienen los colores de la imagen, cada color esta formado de 4 componentes R, G, B, A donde R, G, B, indican la cantidad de rojo, verde y azul de cada color, A es una mascara que nos indica como se combinara ese color con los demás. Hay Que tener cuidado ya que la paleta de colores esta escrita en el orden de B, G, R, A

A continuación se muestra un pequeño programa que despliega archivo con formato BMP en la pantalla de la PC.

```
/* Programa que despliega una imagen con formato BMP */
/* Realizado por: J. Antonio Jiménez Aviña */
/* 29-04-96 */

#include <conio.h> //Cabeceras
#include <stdio.h>
#include <graphics.h>

FILE * BMP;
//Declaración de variables
long int BIT_PIXEL, PIXEL_BIT, CUERPO,
PALETA, PALETA_TAM, TAM_X, TAM_Y;

typedef struct //Estructura para el color
{
    unsigned char b, g, r, a;
} Color;

int getcol(void);
int ShowBMP(char* fname);

void main(void)
{
    int gdriver=DETECT,gmode;
    initgraph(&gdriver,&gmode,"");
    ShowBMP("ARCHIVO.BMP"); //poner el nombre del archivo donde esta la imagen a desplegar
    getch();
    closegraph();
}

int ShowBMP(char* fname)
//Función que abre el archivo y despliega la imagen en el monitor
{
    int color;
    int k,c;
    Color col;
    int x, y;

    BMP= fopen(fname,"rb");
    if (BMP==NULL)
        return 1;

    fseek(BMP,10,SEEK_SET); //Lectura de la cabecera
    fread(&CUERPO,sizeof(CUERPO),1,BMP);
    fseek(BMP,18,SEEK_SET);
    fread(&TAM_X,sizeof(TAM_X),1,BMP);
    fseek(BMP,22,SEEK_SET);
    fread(&TAM_Y,sizeof(TAM_Y),1,BMP);
    fseek(BMP,28,SEEK_SET);
    fread(&BIT_PIXEL,sizeof(BIT_PIXEL),1,BMP);

    PIXEL_BIT=8/BIT_PIXEL;
    PALETA_TAM=1<<BIT_PIXEL;
    PALETA=CUERPO-4*PALETA_TAM;
```

```

fseek(BMP,PALETA,SEEK_SET);
for (k=0;k<PALETA_TAM;k++) //Cargando la paleta de colores
{
    fread(&col,sizeof(col),1,BMP);
    setrgbpalette(k,(col.r)>>2,(col.g)>>2,(col.b)>>2);
}

for (y=TAM_Y-1; y>=0; y--) //Despliega la imagen en el
for (x=0; x<TAM_X; x++) //monitor
    putpixel(x,y,getcol());
return 0;
}

int getcol(void) //Función que regresa el color
{ //del pixel de acuerdo a sus componentes RGB
static int x=0;
static int n=0;
static long int col[4];
unsigned char b;
if (x>=TAM_X)
{
    x=0;
while (n%4)
{
    fread(&b,sizeof(b),1,BMP);
    n++;
}
n=0;
}
if (x%PIXEL_BIT==0)
{
    int k=PIXEL_BIT;
    int mask=(1<<BIT_PIXEL)-1;
    fread(&b,sizeof(b),1,BMP);
    n++;
    while (k--)
    {
        col[k]= b & mask;
        b>>=BIT_PIXEL;
    }
}
return col[x++%PIXEL_BIT];
}

```