

## **FORMATO DE ARQUIVO: BMP**

### **1. INTRODUÇÃO: Formato nativo do ambiente Windows**

O formato de arquivos BMP foi desenvolvido pela Microsoft, sendo o formato nativo de mapa de bits do Windows (a partir da versão 3.00). Este sistema operacional utiliza o BMP em sua própria estrutura: no fundo de tela (wallpaper), nos ícones, nos cursores (apontadores de mouse) e nas imagens mapeadas por bits.

O formato BMP que é com frequência o padrão das aplicações Windows que manipulam imagens. Podemos citar por exemplo, a linguagem de programação Delphi que até sua versão 4.0 apenas disponibilizava componentes para manipulação de imagens no formato BMP. Tal fato deve-se, sobretudo, pela simplicidade da estrutura dos arquivos BMP, que tornam mínimas as possibilidades de ocorrência de algum problema ou erro na interpretação dos dados.

### **2. PLATAFORMAS DE UTILIZAÇÃO DO FORMATO BMP**

O formato BMP foi projetado para sistemas operacionais que rodem sobre a plataforma INTEL (windows e OS/2). Deste modo se for necessária a utilização do formato BMP em outros tipos de arquiteturas, como por exemplo Macintosh, deve ser usado outro formato mais adequado (PCX, GIF, TIFF etc). Esta é com certeza uma razão pela qual arquivos BMP não são usados em aplicações para a Internet, que pressupõe padrões abertos e universais.

Vale ressaltar que os arquivos BMP do Windows e OS/2 variam em termos de sua estrutura, entretanto a maioria das aplicações conseguem ler ambos.

### **3. FORMA DE ARMAZENAMENTO DE ARQUIVOS BMP: DIB**

Arquivos BMP são armazenados no formato **DIB** (Device-Independent Bitmap) que permite exibir a imagem em qualquer dispositivo; ou seja o bitmap especifica a cor do pixel numa forma independente do método usado pelo dispositivo para representá-la. A extensão padrão dos arquivos DIB do Windows é ".BMP". Refências a arquivos DIB são em última análise referências a arquivos BMP.

O BMP usa **Formato Posicional**, onde o significado do byte depende de sua posição no arquivo, o que dificulta mudanças em sua estrutura.

### **4. VERSÕES DE BMP QUANTO À QUANTIDADE DE COR**

Os arquivos BMP podem ser classificados conforme a quantidade de bits para representar 1 pixel (bit/Pixel); existindo versões de **1 bit/pixel** ( $2^1=2$  cores), **4 bits/pixel** ( $2^4=16$  cores), **8 bits/pixel** ( $2^8=256$  cores), **24 bits/pixel** (true color com até  $2^{24} \approx 16$  milhões de cores) e mais recentemente 32 bits (true color com até  $2^{32} \approx 4$  bilhões de cores).

## 5. USO DE TÉCNICAS DE COMPRESSÃO ([RLE](#))

É muitíssimo raro, mas arquivos formato BMP podem, nas versões de 4 e 8 bits/pixel, utilizar a compressão [RLE \(Run length encoded\)](#), de forma a reduzir o tamanho do arquivo que armazena o Bitmap. Como a compressão [RLE](#) é raramente implementada, mesmo para imagens de 4 e 8 bits/pixel, os arquivos BMP em geral tendem a ocupar mais espaço em disco do que outros formatos. Logo, se o tamanho do arquivo é importante, o formato BMP não é o mais indicado. Fato que inviabiliza o uso de BMP em aplicações para Internet, que requer, sempre que possível, arquivos pequenos, para maior eficiência.

Vale ressaltar que a técnica de compressão [RLE](#) é usada neste formato somente até 256 cores, por isto arquivos com Bit/Pixel > 8 não podem usá-la. Assim arquivos BMP de 24 (true color) e 32 bits podem ser excessivamente grandes. O formato BMP usado no OS/2 não usa nenhuma técnica de compressão

## 6. ESTRUTURA GERAL DO FORMATO BMP

Todo arquivo BMP está dividido em 3 ou 4 partes, que são:

a) ***Cabeçalho de arquivo***

Contém a assinatura BM e informações sobre o tamanho e lay-out do arquivo BMP (disposição dos dados dentro do arquivo)

b) ***Cabeçalho de mapa de bits***

Contém as informações da imagem contida no arquivo. Define as dimensões, tipo de compressão (se houver) e informações sobre as cores da imagem.

c) ***Paleta ou mapa de cores (opcional)***

Somente estará presente em arquivos de imagens que usam 16 ou 256 cores (4 e 8 bits/pixel). Nas demais, em seu lugar, vem diretamente a parte seguinte: área de dados da imagem.

d) ***Área de dados da imagem contida no arquivo***

Dados que permitem a exibição da imagem propriamente dita, o dados dos pixels a serem exibidos. Podem ser com ou sem compressão.

Obs: Existem dentro da estrutura do BMP alguns campos ditos "[Reservados](#)", destinados a uso futuro, que sempre devem ser setados com [ZERO](#). Outros dados são sempre idênticos ou fornecidos mais de uma vez no arquivo, para possível re-checagem e desvio de possíveis erros. Sente-se que a facilidade de utilização predominava sobre a economia de espaço no projeto do formato!

## 7. ESTRUTURA DETALHADA DO FORMATO BMP

A seguir descreve-se detalhadamente a informação que se encontra em cada byte de cada uma das partes do arquivo BMP.

### A) Cabeçalho de arquivo – informações do arquivo - Tamanho : 14 bytes

Campo	Bytes	Descrição
BfType	2	Assinatura do arquivo: os caracteres ASCII "BM" ou (42 4D) <sub>h</sub> . É a identificação de ser realmente BMP.
BfSize	4	Tamanho do arquivo em Bytes
BfReser1	2	Campo reservado 1; deve ser ZERO
BfReser2	2	Campo reservado 2; deve ser ZERO
BfOffSetBits	4	Especifica o deslocamento, em bytes, para o início da área de dados da imagem, a partir do início deste cabeçalho. - Se a imagem usa paleta, este campo tem tamanho=14+40+(4 x NumeroDeCores) - Se a imagem for true color, este campo tem tamanho=14+40=54

### B) Cabeçalho de mapa de bits – informações da imagem - Tamanho : 40 bytes

Campo	Bytes	Descrição
BiSize	4	Tamanho deste cabeçalho (40 bytes). Sempre (28) <sub>h</sub> .
BiWidth	4	Largura da imagem em pixels
BiHeight	4	Altura da imagem em pixels
BiPlanes	2	Número de planos de imagem. Sempre 1
BiBitCount	2	Quantidade de Bits por pixel (1,4,8,24,32) Este campo indica, indiretamente, ainda o número máximo de cores, que é 2 <sup>Bits por pixel</sup>
BiCompress	4	Compressão usada. Pode ser: 0 = BI_RGB _ sem compressão 1 = BI_RLE8 – compressão RLE 8 bits 2 = BI_RLE4 – compressão RLE 4 bits
BiSizeImag	4	Tamanho da imagem (dados) em byte - Se arquivo sem compressão, este campo pode ser ZERO. - Se imagem em true color, será Tamanho do arquivo (Bfsize) menos deslocamento (BfOffSetBits)
BiXPPMeter	4	Resolução horizontal em pixels por metro
BiYPPMeter	4	Resolução vertical em pixels por metro
BiClrUsed	4	Número de cores usadas na imagem. Quando ZERO indica o uso do máximo de cores possível pela quantidade de Bits por pixel , que é 2 <sup>Bits por pixel</sup>
BiClrImpor	4	Número de cores importantes (realmente usadas) na imagem. Por exemplo das 256 cores, apenas 200 são efetivamente usadas. Se todas são importantes pode ser ZERO. É útil quando for exibir uma imagem em 1 dispositivo que suporta menos cores que a imagem possui.

**C) Paleta cores - definição tabela de cores - Tamanho : 4 bytes x Número de Cores**

Campo	Bytes	Descrição
Blue	1	Intensidade de Azul. De 0 a 255
Green	1	Intensidade de Verde. De 0 a 255
Red	1	Intensidade de Vermelho. De 0 a 255
Reservado	1	Campo reservado deve ser ZERO sempre

A tabela de cores só está presente em imagens com 256 cores ou menos. Para os demais tipos de BMP, vem em seu lugar a área de dados. No BMP a cor é representada de forma diferenciada dos demais formatos de arquivos de imagens.

A paleta é um vetor de bytes da estrutura RGBA, representando a intensidade de cada cor, através de 1 byte. Imagens de 8 bits por pixel, com no máximo 256 cores terão 256 posições na paleta; da mesma forma que imagens de 4 bits por pixel terão 16 posições e imagens de 1 bit por pixel terão 2 posições na paleta. Para um exemplo de imagem com 256 cores, o armazenamento de cada posição da paleta seria :

- ◆ Paleta[0] – contém informação sobre o atributo de cor número ZERO
- ◆ Paleta[1] - contém informação sobre o atributo de cor número UM.
- ◆ .....
- ◆ .....
- ◆ Paleta[255] - contém informação sobre o atributo de cor número 255.

As cores aparecem na tabela em **ordem de importância** o que ajuda o driver de exibição a reproduzir o BitMap em dispositivos que não podem exibir a quantidade de cores com que a imagem foi criada.

**D) Área de dados da imagem - cor que cada pixel deve ser ligado ou esses dados comprimidos - Tamanho: campo BiSizeImg, do cabeçalho de informações da imagem.**

Esta área do arquivo de imagens varia, conforme existência ou não de compressão. Veja adiante a seção de compressão [RLE](#). Para imagens sem compressão, os dados são armazendados em uma ordem sequencial, dentro do arquivo, que corresponde a posições na tela de vídeo. O primeiro pixel refere-se a posição inferior esquerda. O último pixel refere-se a posição superior direita. Como a orientação usual de um sistema de coordenadas cartesiano. Por exemplo, para uma imagem de 150(Largura) x 80(Altura) pixels, teríamos :

- ◆ 1º. Pixel – posição (0 , 80) na imagem.
- ◆ 2º. Pixel – posição (1 , 80) na imagem.
- ◆ 3º. Pixel – posição (2 , 80) na imagem.
- ◆ 4º. Pixel – posição (3 , 80) na imagem.
- ◆ .....
- ◆ Penúltimo Pixel – posição (148 , 0) na imagem.
- ◆ Último Pixel – posição (149 , 0) na imagem.

O valor lido nessa área de dados, se sem compressão, refere-se a cor do pixel de acordo com a tabela de cores (paleta).

Há, entretanto, a restrição de que cada linha deva ter N bytes, sendo N um número divisível por 4. Caso contrário, o BMP deve ser preenchido com bytes não válidos. Por exemplo, se a imagem tem 1 x 100 pixels em 8 bits/pixel, o BMP teria 1 byte válido em cada linha e mais 3 bytes que não tem qualquer significado.

No BMP **monocromático**, cada valor lido corresponde a uma entrada na paleta de cores. Se o bit for ZERO a cor é a da paleta[0]; caso contrário a cor será a da paleta[1].

No BMP de **16 cores**, cada 4 bits (meio byte) correspondem a uma entrada na paleta de cores. Por exemplo se o primeiro byte contiver (1F)<sub>h</sub>, o primeiro pixel tem a cor da paleta[0] e o segundo pixel a cor da paleta[15].

No BMP de **256 cores**, cada byte (8 bits) correspondem a uma entrada na paleta de cores. Se o primeiro byte da área de dados contém (1F)<sub>h</sub> o primeiro pixel tem a cor da paleta[31].

No BMP **true color** (24 bits) cada sequência de 3 bytes correspondem a uma sequência Blue.Green.Red , isso é a composição da cor do pixel diretamente (não tendo neste caso paleta de cores). No **true color** de 32 bits, é a combinação de 4 bytes que estabelecerá a cor do pixel.

## 8. EXEMPLOS DE ARQUIVOS BMP

Um exemplo típico de imagem BMP de 16 cores (4 bits/pixel) é representada abaixo:

### Cabeçalho do arquivo

Type	BM
Size	3118
Reser1	0
Reser2	0
OffSetBits	118 (14+54+(4 x 16)

### Cabeçalho da Imagem

Size	40
Width	80
Height	75
Planos	1
BitCount	4 (bits/pixel)
Compressão	0 (s/compressão)
TamanhoImagem	3000 (3118 – 118)
XporMeter	0
YporMeter	0
Coresusadas	16
CoresImportantes	16

**Paleta de Cores**

<i>Azul</i>	<i>Verde</i>	<i>Vermelho</i>	<i>SemUso</i>	<i>Indice em hexa</i>
84	252	84	0	0000000
252	252	84	0	0000001
84	84	252	0	0000002
252	84	252	0	0000003
84	252	252	0	0000004
252	252	252	0	0000005
0	0	0	0	0000006
168	0	0	0	0000007
0	168	0	0	0000008
168	168	0	0	0000009
0	0	168	0	000000A
168	0	168	0	000000B
0	168	168	0	000000C
168	168	168	0	000000D
84	84	84	0	000000E
252	84	84	0	000000F

**Área de dados da imagem**

cores de cada pixel da imagem....

.....

3000 Bytes com dados da imagem....

.....

Nas telas, das páginas que seguem são mostrados exemplo de arquivos BMP, visualizados em hexadecimal (neste caso foi usado o utilitário ZTreeWin). São identificados abaixo das telas todos os elementos que compõem as partes dos arquivos BMP discutidos

**4 bits/pixel (16 cores)**

```

00000000  42 4D EE 05  00 00 00 00  00 00 76 00  00 00 28 00  BM...v...
00000010  00 00 32 00  00 00 32 00  00 00 01 00  04 00 00 00  .2...2...
00000020  00 00 78 05  00 00 C4 0E  00 00 C4 0E  00 00 00 00  .x.....
00000030  00 00 00 00  00 00 00 00  00 00 00 00  80 00 00 80  80 00 00 80 80
00000040  00 00 00 80  80 00 80 00  00 00 80 00  80 00 80 80  80 00 80 80
00000050  00 00 C0 C0  C0 00 80 80  80 00 00 00  FF 00 00 FF  FF 00 00 FF
00000060  00 00 00 FF  FF 00 FF 00  00 00 FF 00  FF 00 FF  FF  FF 00 FF
00000070  00 00 FF FF  FF 00 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C
00000080  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C
00000090  00 00 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0
000000A0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  00 00 00 0C 0C
000000B0  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C
000000C0  0C 0C 0C 0C  0C 0C 0C 0C  00 00 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0
000000D0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0
000000E0  C0 C0 C0 00  00 00 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C
000000F0  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C
00000100  00 00 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0
00000110  C0 C0 C0 C0  C0 C0 C0 C0  C0 C0 C0 C0  00 00 00 0C 0C
00000120  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C  0C 0C 0C 0C

```

VIEW ASCII Dump Edit Gather Hex Mask OEM Junk Ruler  
 COMMANDS Wordwrap F9 Search SPACE search again ESC cancel  
 ↑ scroll HEX (masked )

**1. CABEÇALHO DO ARQUIVO**

- ♦ ASSINATURA 42 4D
- ♦ TAMANHO ARQ EE 05 00 00 [ formato intel LSB: (5EE)h ]= 1.518 Bytes
- ♦ RESERVADO 00 00 00 00 = 0 (SEM USO)
- ♦ DESLOCAMENTO 76 00 00 00 = 118 Bytes (14+40+64)

**2. CABEÇALHO DA IMAGEM**

- ♦ TAMANHO CABEC 28 00 00 00 = 40
- ♦ LARGURA (WIDTH) 32 00 00 00 = 50
- ♦ ALTURA (HEIGH) 32 00 00 00 = 50
- ♦ PLANOS 01 00 = 1 PLANO
- ♦ BIT/PIXEL 04 00 = 4 Bit/pixel
- ♦ COMPRESSÃO 00 00 00 00 = 0 = S/COMPRESSÃO
- ♦ TAM. BITMAP 78 05 00 00 = 1.400 Bytes (1518-118)
- ♦ Xpixel/Metro C4 0E 00 00 = 3.780
- ♦ Ypixel/Metro C4 0E 00 00 = 3.780
- ♦ CORES USADAS 00 00 00 00 = 0 = max cores
- ♦ CORES IMPORTANT. 00 00 00 00 = 0 = todas

**3. PALETA DE CORES**

- ♦ 4 x 16 (Num Cores) = 64 Bytes (1 RGB + 1 Reservado)

**4. ÁREA DE DADOS (640 x 480) = 1.400 Bytes**

## 8 bits/pixel (256 cores)

```

00000000  42 4D 38 B4  04 00 00 00  00 00 36 04  00 00 28 00  BM8.....6...(-.
00000010  00 00 80 02  00 00 E0 01  00 00 01 00  08 00 00 00  .....
00000020  00 00 00 00  00 00 12 0B  00 00 12 0B  00 00 00 00  .....
00000030  00 00 00 00  00 00 FF FF  FF 00 CC FF  FF 00 99 FF  .....
00000040  FF 00 66 FF  FF 00 33 FF  FF 00 00 FF  FF 00 F0 FB  ...f...3.....
00000050  FF 00 FF CC  FF 00 CC CC  FF 00 99 CC  FF 00 66 CC  .....
00000060  FF 00 33 CC  FF 00 00 CC  FF 00 FF 99  FF 00 CC 99  ...3.....
00000070  FF 00 99 99  FF 00 66 99  FF 00 33 99  FF 00 00 99  .....
00000080  FF 00 80 7C  FF 00 FF 66  FF 00 CC 66  FF 00 99 66  .....
00000090  FF 00 66 66  FF 00 33 66  FF 00 00 66  FF 00 50 50  .....
000000A0  FF 00 FF 33  FF 00 CC 33  FF 00 99 33  FF 00 66 33  .....
000000B0  FF 00 33 33  FF 00 00 33  FF 00 FF 00  FF 00 CC 00  .....
000000C0  FF 00 99 00  FF 00 66 00  FF 00 33 00  FF 00 00 00  .....
000000D0  FF 00 F8 F8  F8 00 F1 F1  F1 00 C6 D6  EF 00 EA EA  .....
000000E0  EA 00 D6 E7  E7 00 E3 E3  E3 00 DD DD  DD 00 D7 D7  .....
000000F0  D7 00 93 00  D6 00 FF FF  CC 00 CC FF  CC 00 99 FF  .....
00000100  CC 00 66 FF  CC 00 33 FF  CC 00 00 FF  CC 00 FF EC  ...f...3.....
00000110  CC 00 FF CC  CC 00 CC CC  CC 00 99 CC  CC 00 66 CC  .....
00000120  CC 00 33 CC  CC 00 00 CC  CC 00 FF 99  CC 00 CC 99  ...3.....

```

VIEW ASCII Dump Edit Gather Hex Mask OEM Junk Ruler  
COMMANDS Wordwrap F9 Search SPACE search again ESC cancel  
↑ scroll HEX (masked)

### 1. CABEÇALHO DO ARQUIVO

- ♦ ASSINATURA 42 4D
- ♦ TAMANHO ARQ 38 B4 04 00 [ formato intel LSB: (4B438)h ] = 308.280 Bytes
- ♦ RESERVADO 00 00 00 00 = 0 (SEM USO)
- ♦ DESLOCAMENTO 36 04 00 00 [ formato intel LSB: (436)h ] = 1.078 Bytes (14+40+1024)

### 2. CABEÇALHO DA IMAGEM

- ♦ TAMANHO CABEC 28 00 00 00 = 28(H) = 40
- ♦ LARGURA (WIDTH) 80 02 00 00 = 640
- ♦ ALTURA (HEIGH) E0 01 00 00 = 480
- ♦ PLANOS 01 00 = 1 PLANO
- ♦ BIT/PIXEL 08 00 = 8 Bit/pixel
- ♦ COMPRESSÃO 00 00 00 00 = 0 = S/COMPRESSÃO
- ♦ TAM. BITMAP 00 00 00 00 = 0 = não compressão
- ♦ Xpixel/Metro 12 0B 00 00 = 2.834
- ♦ Ypixel/Metro 12 0B 00 00 = 2.834
- ♦ CORES USADAS 00 00 00 00 = 0 = max cores
- ♦ CORES IMPORTANT. 00 00 00 00 = 0 = todas

### 3. PALETA DE CORES

- ♦ 4 x 256 (Num Cores) = 1024 Bytes (1 RGB + 1 Reservado)

### 4. ÁREA DE DADOS (640 x 480) = 307.200 bytes

## 24 bits/pixel (true color)

```

00000000  42 4D 1E DE 16 00 00 00 00 00 36 00 00 00 28 00 BM.....6...(. .
00000010  00 00 4E 03 00 00 4E 02 00 00 01 00 18 00 00 00 ..N...N.....
00000020  00 00 E8 DD 16 00 12 17 00 00 12 17 00 00 00 00 .....
00000030  00 00 00 00 00 00 FF FF FF FF EA FF FF EC FF FF ...
00000040  FF F0 FF FF F8 FF FF F2 FF FF EA FF FF EC FF FF ...
00000050  F2 FF FF EE FD FF F0 FF FF F3 FF FF F1 FF FF F6 ...
00000060  FF FF EE FF FF ED FF FF ED FF EF FC FF E9 FF ...
00000070  FF EA FC FF EB FF FF F2 FF FF E9 FE FF E8 FF FF ...
00000080  F0 FD FF E9 FF FF E8 FA FF EC FF FF E9 FF FF E3 ...
00000090  F8 FF E7 FF FF E1 FF FF EE FF FF EB FF FF F2 FC ...
000000A0  FF EC FF FF EC FF FF EA FD FF ED FE FF F3 FE FF ...
000000B0  ED FF FF ED FD FF EE FF FF F2 FF FF EB FD FF FF ...
000000C0  FD FF F6 FF FF F1 FF FF EB F8 FF FD FF FF F0 FE ...
000000D0  FF EA FF FF EB FA FF EE FC FF ED FF FF EC FB FF ...
000000E0  EA F3 FF EA F5 FF E4 F8 FF E6 FD FF EF FF FF EA ...
000000F0  FE FF EC FA FF F2 FF FF F0 F8 FF EE FF FF E4 FD ...
00000100  FF EF F8 FF E9 FF FF EE FF FF EF FA FF F0 FF FF ...
00000110  F2 F8 FF F2 FD FF F2 FA FF EB FC FF F3 FF FF ED ...
00000120  FF FF F0 FB FF ED FF FF F2 FB FF EC FF FF F7 FF ...

```

VIEW    ASCII    Dump    Edit    Gather    Hex    Mask    OEM    Junk    Ruler  
 COMMANDS    Wordwrap    F9 Search    SPACE search again    ESC cancel  
 scroll    HEX (masked )

### 1. CABEÇALHO DO ARQUIVO

- ♦ ASSINATURA 42 4D
- ♦ TAMANHO ARQ 1E DE 16 00 [ formato intel LSB: (16DE1E)h ] = 1.498.654 Bytes
- ♦ RESERVADO 00 00 00 00 = 0 (SEM USO)
- ♦ DESLOCAMENTO 36 00 00 00 = 54 Bytes (14+40)

### 2. CABEÇALHO DA IMAGEM

- ♦ TAMANHO CABEC 28 00 00 00 = 28(H) = 40
- ♦ LARGURA (WIDTH) 4E 03 00 00 = 846
- ♦ ALTURA (HEIGH) 4E 02 00 00 = 590
- ♦ PLANOS 01 00 = 1 PLANO
- ♦ BIT/PIXEL 18 00 = 24 Bit/pixel
- ♦ COMPRESSÃO 00 00 00 00 = 0 = S/COMPRESSÃO
- ♦ TAM. BITMAP E8 DD 16 00 = 1.498.600
- ♦ Xpixel/Metro 12 17 00 00 = 5.906
- ♦ Ypixel/Metro 12 17 00 00 = 5.906
- ♦ CORES USADAS 00 00 00 00 = 0 = max cores
- ♦ CORES IMPORTANT. 00 00 00 00 = 0 = todas

### 3. ÁREA DE DADOS (640 x 480) = 1.498.600 Bytes (cada 3 bytes 1 pixel)

**CURIOSIDADES:** Algumas incoerências inexplicáveis podem ser achadas em arquivos BMP, corretamente vistos nas telas, repare nestas:

```

FILE: C:\VS\mestrado\compvisual\procimg\trab2\PARADISE.BMP
00000000  42 4D 0E 2D 01 00 00 00 00 00 36 04 00 00 28 00 BM - . . . 6 . .
00000010  00 00 80 02 00 00 E0 01 00 00 01 00 08 00 00 00 . . . . .
00000020  00 00 00 B0 04 00 00 00 00 00 00 00 00 00 00 01 . . . . .
00000030  00 00 00 00 00 00 FF FF FF 00 CC FF FF 00 99 FF . . . . .
00000040  FF 00 66 FF FF 00 33 FF FF 00 FF CC FF 00 CC CC . f . . 3 . .
00000050  FF 00 99 CC FF 00 66 CC FF 00 33 CC FF 00 00 CC . . f . . 3 . .
00000060  FF 00 CC 99 FF 00 99 99 FF 00 66 99 FF 00 33 99 . . . f . . 3 . .
00000070  FF 00 00 99 FF 00 99 66 FF 00 66 66 FF 00 33 66 . . . f . ff . 3f
00000080  FF 00 00 66 FF 00 FF FF CC 00 CC FF CC 00 99 FF . f . . .
00000090  CC 00 66 FF CC 00 FF CC CC 00 CC CC CC 00 99 CC . f . . .
000000A0  CC 00 66 CC CC 00 33 CC CC 00 00 CC CC 00 FF 99 . f . . 3 . .
000000B0  CC 00 CC 99 CC 00 99 99 CC 00 66 99 CC 00 33 99 . . . f . . 3 . .
000000C0  CC 00 00 99 CC 00 99 66 CC 00 66 66 CC 00 33 66 . . . f . ff . 3f
000000D0  CC 00 00 66 CC 00 66 33 CC 00 33 33 CC 00 00 33 . f . f3 . 33 . 3
000000E0  CC 00 FF FF 99 00 CC FF 99 00 99 FF 99 00 FF CC . . . .
000000F0  99 00 CC CC 99 00 99 CC 99 00 66 CC 99 00 33 CC . f . . 3 . .
00000100  99 00 00 CC 99 00 FF 99 99 00 CC 99 99 00 99 99 . . . .
00000110  99 00 66 99 99 00 33 99 99 00 00 99 99 00 CC 66 . f . . 3 . f
00000120  99 00 99 66 99 00 66 66 99 00 33 66 99 00 00 66 . f . ff . 3f . f

```

VIEW    ASCII    Dump    Edit    Gather    Hex    Mask    OEM    Junk    Ruler  
COMMANDS    Wordwrap    F9 Search    SPACE search again    ESC cancel  
↑ scroll    HEX (masked)

## 1. CABEÇALHO DO ARQUIVO

- ◆ ASSINATURA            42 4D
- ◆ TAMANHO ARQ          0E 2D 01 00 [ formato intel LSB: (12D0E)h ] = 77.070 (mas não deveria ser : 308 278?????)
- ◆ RESERVADO             00 00 00 00 = 0 (SEM USO)
- ◆ DESLOCAMENTO        36 04 00 00 = 1.078 Bytes (14+40+1024)

## 2. CABEÇALHO DA IMAGEM

- ◆ TAMANHO CABEC      28 00 00 00 = 28(H) = 40
- ◆ LARGURA (WIDTH)     80 02 00 00 = 640
- ◆ ALTURA (HEIGH)      E0 01 00 00 = 480
- ◆ PLANOS               01 00 = 1 PLANO
- ◆ BIT/PIXEL            08 00 = 8 Bit/pixel
- ◆ COMPRESSÃO          00 00 00 00 = 0 = S/COMPRESSÃO
- ◆ TAM. BITMAP          00 B0 04 00 = 0 = ???
- ◆ Xpixel/Metro        00 00 00 00 = 0
- ◆ Ypixel/Metro        00 00 00 00 = 0
- ◆ CORES USADAS       00 01 00 00 = 256 ?
- ◆ CORES IMPORTANT.    00 00 00 00 = 0 = todas

## 3. PALETA DE CORES

- ◆ 4 x 256 (Num Cores) = 1024 Bytes (1 RGB + 1 Reservado)

## 4. ÁREA DE DADOS (640 x 480) = 307 200 Bytes?????

(explicação? talvez as redundâncias de informações do arquivo!!!)

## 9. COMPRESSÃO

A seguir apresentaremos a forma como os dados da imagem são representados se comprimidos. Dividiremos esta seção de acordo com as 2 formas possíveis: imagens com 8 ou 4 bits por pixel. A existência de compressão é indicada pelo campo **BiCompress** do cabeçalho de mapa de bits ser diferente de ZERO.

### Compressão de imagens com 8 bits/pixel

Neste caso o campo *BiCompress* do cabeçalho de mapa de bits está setado com *BI\_RLE8* (01). O BMP usa o formato de compressão RLE para imagens de 256 cores. Este formato o RLE tem 2 modos, denominados **Encoded mode** e **Absolute mode**. Vejamos cada um deles.

- ◆ **Encoded mode**

Este modo usa compressão **RLE em 2 bytes**. O 1º byte especifica o **número de pixels** consecutivos que serão desenhados usando o **índice de cor** contido no 2º byte.

Mas caso o 1º byte esteja "setado" com ZERO, isso é um indicativo de que (4 coisas): pode ser **fim de linha**, **fim do bitmap**, haverá um "**Delta**" ou ainda a **codificação RLE se interromperá por alguns bytes**. Qual será a interpretação correta do ZERO, depende do 2º byte do par (vide abaixo). Esse byte pode assumir valores de 0x00 a 0xFF e os significados podem ser os definidos na tabela abaixo:

2º Byte	Significado
0	Fim de linha
1	Fim de BitMap
2	Delta.
3 a 255	Modo absoluto

**Delta** é entendido como um deslocamento do próximo pixel a ser representado na tela, da posição de sucessor do pixel anterior usual para a posição indicada pelos próximos e bytes do arquivo que seguem. Neste caso os 2 bytes seguintes (ao flag 2) contém valores que, indicam o deslocamento horizontal e vertical do próximo pixel a partir da posição corrente (veja melhor no exemplo que segue).

- ◆ **Absolute mode**

Este modo é sinalizado pelo 1º byte do par ser setado com ZERO e o 2º byte ser setado por um valor entre 0x03 e 0xff. Neste caso o valor do 2º byte representa o número de bytes seguintes que serão descritos na forma não comprimida isso é: cada qual com seu valor representando um índice de cor da paleta. Siga o exemplo abaixo para entender completamente o que ocorre!

- ◆ **Exemplo de uso dos 2 modos de compressão RLE. Suponha que no arquivo na área de dados da imagem estejam os valores abaixo.**

03 04 05 06 00 03 45 56 67 00 02 78 00 02 05 01 02 78 00 00 09 1E 00  
01

Acompanhe na tabela abaixo o que seria representado na matriz de dados da imagem, quando os dados fossem expandidos:

Dados comprimidos	dados expandidos
03 04	04 04 04
05 06	06 06 06 06
00 03 45 56 67	45 56 67
00 02 78	78 78
00 02 05 01	delta de 5 pixels p/esquerda e 1 p/baixo
02 78	78 78
00 00	Fim de linha
09 1E	1E 1E 1E 1E 1E 1E 1E 1E
00 01	Fim da Imagem

### Compressão de imagens com 4 bits/pixel

Neste caso o campo *BiCompress* do cabeçalho de mapa de bits está setado com *BI\_RLE4* (02), o BMP usa o formato de compressão RLE para imagens de 16 cores. Este formato usa os mesmos 2 modos anteriores, mas agora cada 4 bits representam um dado. Um diferencial é o conceito de cor primária e secundária. Vejamos como ficam os 2 modos.

- ♦ **Encoded mode**

Como no caso anterior usa 2 bytes. O 1º byte do par contém o número de pixels que serão desenhados usando os índices de cores do 2º byte. O 2º byte contém 2 índices de cores (já que o BMP está codificado com 4 bits/pixel).

O primeiro pixel usa a cor de ordem primária, o 2º pixel a cor de ordem secundária e assim por diante até que todos os pixels especificados no 1º. Byte tenham sido desenhados.

Como no caso anterior o 1º byte do par pode ser setado com ZERO, o que estará indicando que pode ser: fim de linha, fim de BMP, delta ou interrupção do modo codificado (encode) e passagem para o modo absoluto. A interpretação dependerá do valor do 2º byte , que pode variar de 0x00 a 0xFF. O significado é idêntico ao explicado anteriormente para compressão de arquivos BMP com 8 bits/pixel.

- ♦ **Absolute mode**

Neste modo o 1º byte contém ZERO e o 2º byte contém o número de índices de cores que se seguem e os bytes seguintes indicam os índices de cores primárias e secundárias, sendo 1 índice de cores para cada pixel.

- ♦ **Exemplo de uso dos 2 modos de compressão RLE. Suponha que no arquivo na área de dados da imagem estejam os valores abaixo.**

03 04 05 06 00 03 45 56 67 00 02 78 00 02 05 01 02 78 00 00 09 1E 00 01

Acompanhe na tabela abaixo o que seria representado na matriz de dados da imagem, quando os dados fossem expandidos:

Dados comprimidos	dados expandidos
03 04	0 4 0
05 06	0 6 0 6 0
00 03 45 65 67	4 5 6 5 6 7
00 02 78	7 8 7 8
00 02 05 01	Delta: move 5 p/esquerda e 1 p/baixo
02 78	7 8 7 8
00 00	Fim de Linha
09 1E	1 E 1 E 1 E 1 E 1
00 01	Fim dos dados da imagem

#### 10. RESUMO: Na tabela que segue apresenta-se um resumo do comentado acima:

<b>Nome</b>	Windows BitMaP
<b>Extensão</b>	BMP
<b>Características</b>	Formato simples → poucos erros Arquivos grandes → não usáveis na Internet. Raramente usam compressão
<b>Tipo de Formato</b>	Mapa de Bits, armazenado em formato DIB (device independent BitMap). Formato Posicional → dificulta mudanças → byte menos significativo é descrito primeiro nos valores maiores que 1 byte ( ou seja formato LSB )
<b>Versões</b>	Presente no Microsoft Windows (3.x e acima) e no Presentation Maganer do OS/2
<b>Variações</b>	Os arquivos BMP do Windows e OS/2 variam em termos de estrutura de arquivos, entretanto a maioria das aplicações conseguem tratar ambos.
<b>Arquiteturas</b>	Baseadas nos processadores Intel = LSB
<b>Capacidade de cor</b>	2 (1 bit/pixel), 16 (4 bits/pixel), 256 (8 bits/pixel) 16 milhões de cores (24 bits/pixel = true color) e 4 bilhões de cores (32 bits/pixel = true color)
<b>Compressão</b>	Usada apenas na plataforma Windows. O formato OS/2 não é comprimido. Muito raro de ser usada. Quando utilizada é para imagens de 4 e 8 bits/pixel. Técnica de compressão usada RLE, diferenciada para 4 e 8 bits/pixel. Este técnica é aplicável para no máximo 256 cores e tem características particulares, que permitem que trechos não comprimidos sejam também incluídos.
<b>Softwares</b>	Praticamente todas as aplicações Windows que utilizam mapa de bits podem abrir ou importar BMP.

#### 11. BIBLIOGRAFIA

- [www.upv.es/protel/usr/jotrofer/graficos/fmtgraf.htm](http://www.upv.es/protel/usr/jotrofer/graficos/fmtgraf.htm)
- [www.prudente.unesp.br/dcartog/arlete/hparlete/portfolio/cartnet.htm](http://www.prudente.unesp.br/dcartog/arlete/hparlete/portfolio/cartnet.htm)
- [www.microsoft.com](http://www.microsoft.com)
- [www.imagem-digital.com/chapter07.htm](http://www.imagem-digital.com/chapter07.htm)