

Práctica 7. Filtro multicolor

```

file = open('./images/volcan.bmp','rb')
fileo = open('./images/volcan16.bmp','wb')

metadata = file.read(54)
fileo.write(metadata)
paleta_hex = [
    "#ffffff", "#fefafb", "#fdf0f7", "#fceff4",
    "#fbcb6", "#f3bfd2", "#f2b9ce", "#f1b4ca",
    "#de4a83", "#dc497f", "#db447b", "#da3e78",
    "#8b1b44", "#861941", "#81183f", "#7b173e"
]

paleta = []
for color in paleta_hex:
    r = int(color[1:3], 16)
    g = int(color[3:5], 16)
    b = int(color[5:7], 16)
    paleta.append([b, g, r]) # BMP usa BGR

file.seek(54,0)
no_pix = 0

while True:
    pixel_data = file.read(3)

    if len(pixel_data) > 0:
        b = pixel_data[0]
        g = pixel_data[1]
        r = pixel_data[2]
        intensidad = int((r + g + b) / 3)
        indice = int((intensidad / 256) * 16)
        if indice == 16:
            indice = 15

        nuevo_color = paleta[indice]

        fileo.write(bytes(nuevo_color))
        no_pix += 1
    else:
        break

print('No Pixels: ' + str(no_pix))
file.close()
fileo.close()

```

En el código anterior se realiza la aplicación del filtro en base a lo que se venia trabajando en la ultimas prácticas. Se tomó como base una paleta de colores de tonos *magenta* por ello se utilizó la siguiente paleta de colores:

```
#ffffff  
#fefafb  
#fdf0f7  
#fceff4  
#fbc4d6  
#f3bfd2  
#f2b9ce  
#f1b4ca  
#de4a83  
#dc497f  
#db447b  
#da3e78  
#8b1b44  
#861941  
#81183f  
#7b173e
```

La función del código es analizar los pixeles que hay dentro de la imagen donde se realizará un promedio RGB para analizar la intensidad de cada pixel, una vez hecho esto se procede a mapear cada píxel donde al más acercarse a negro su índice será cero, caso contrario, al ser lo más cercano a blanco su índice será 15, estos índices se encuentran indicados en un vector al inicio del código donde dependiendo del mismo se reescribirán los nuevos bytes, es decir, se obtendrá una nueva salida.



Figura 1. Imagen original



Figura 1.1. Imagen con filtro de colores.