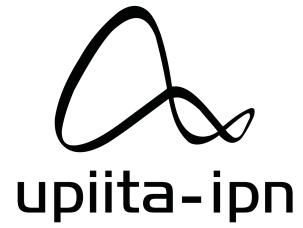




INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL
INTERDISCIPLINARIA EN
INGENIERÍA Y TECNOLOGÍAS
AVANZADAS



MULTIMEDIA

PRÁCTICA 3 **“Filtro BMP”**

Alumno: De la Rosa Valdes Itztli Fernanda

Boleta: 2024640062

Abriendo el archivo “example001.bmp” dentro de Google Colab.

The screenshot shows a Jupyter Notebook interface. On the left, the 'Archivos' sidebar lists a 'images' folder containing 'example001.bmp' and a 'sample_data' folder. The main area displays a code cell with the following Python script:

```
#Read and display data from BMP
file = open("./images/example001.bmp", "rb")
firm = file.read(2)
print(firm)
file.seek(54,0)
pixel_first = file.read(3)
print(pixel_first)

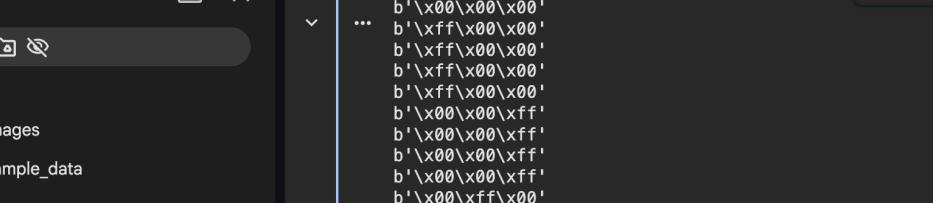
file.seek(54,0)
no_pix = 0
while(True):
    pixel_data = file.read(3)
    if(len(pixel_data) > 0):
        print(pixel_data)
        no_pix += 1
    else:
        break
print('No Pixels: '+str(no_pix))
file.close()

... b'BM'
b'\xff\xff\xff'
b'\xff\xff\xff'
b'\xff\xff\xff'
b'\xff\xff\xff'
b'\x00\x00\x00'
b'\x00\x00\x00'
b'\x00\x00\x00'
b'\x00\x00\x00'
b'\x00\x00\x00'
```

On the right, the 'Notas de la versión' (Version Notes) section lists the following package upgrades:

- accelerate 1.11.0 -> 1.12.0
- astropy 7.1.1 -> 7.2.0
- bigframes 2.28.0 -> 2.31.0
- cachetools 5.5.2 -> 6.2.4
- dataprocspark-connect 0.8.3 -> 1.0.1
- diffusers 0.35.2 -> 0.36.0
- flax 0.10.7 -> 0.11.2
- google 2.0.3 -> 3.0.0
- google-adk 1.17.0 -> 1.21.0
- google-auth 2.38.0 -> 2.43.0
- google-genai 1.49.0 -> 1.55.0
- gradio 5.49.1 -> 5.50.0
- holidays 0.84 -> 0.88
- humanize 4.14.0 -> 4.15.0

Número de píxeles.



```
from PIL import Image
img = Image.open('image.bmp')
print(img)
print(img.getdata())
print(img.size)
print('No Pixels: ', img.size[0] * img.size[1])
```

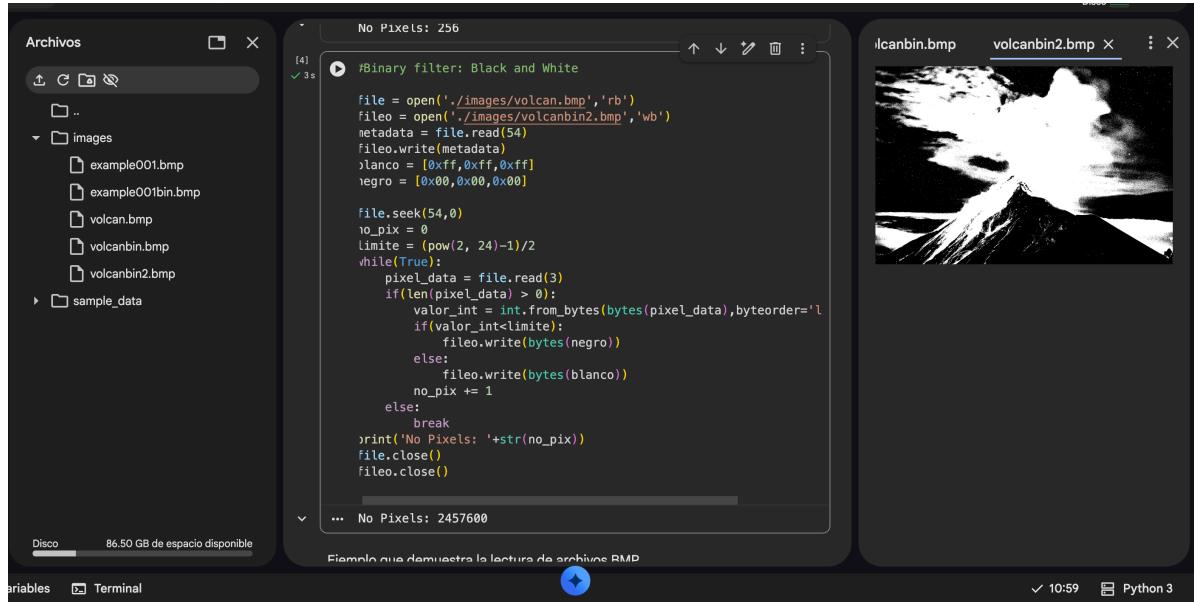
Código para el filtro blanco y negro.

```
[2]  ✓ 0s  #Binary filter: Black and White
      file = open('./images/example001.bmp', 'rb')
      fileo = open('./images/example001bin.bmp', 'wb')
      metadata = file.read(54)
      fileo.write(metadata)
      blanco = [0xff, 0xff, 0xff]
      negro = [0x00, 0x00, 0x00]

      file.seek(54, 0)
      no_pix = 0
      limite = (pow(2, 24)-1)/2
      while(True):
          pixel_data = file.read(3)
          if(len(pixel_data) > 0):
              valor_int = int.from_bytes(bytes(pixel_data), byteorder='l
              if(valor_int<limite):
                  fileo.write(bytes(blanco))
              else:
                  fileo.write(bytes(negro))
              no_pix += 1
          else:
              break
      print('No Pixels: '+str(no_pix))
      file.close()
      fileo.close()

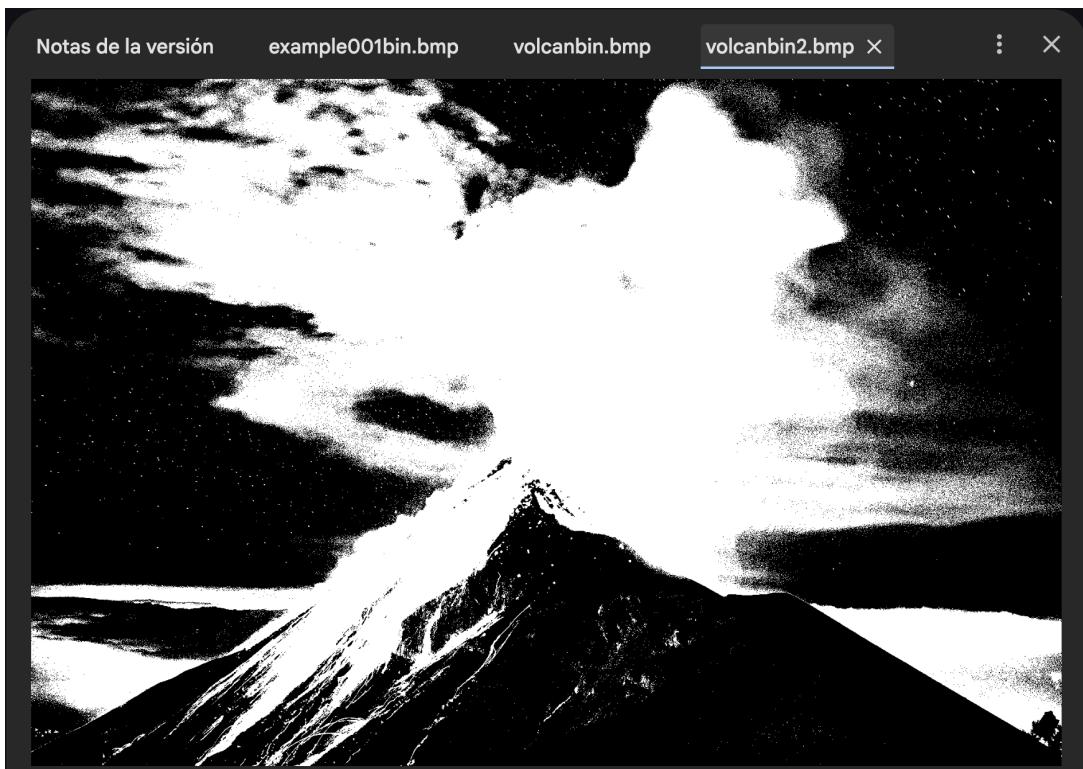
...  No Pixels: 256
```

Prueba de código con imagen “volcan”.



The screenshot shows a Jupyter Notebook environment with the following components:

- File Browser:** On the left, a sidebar shows a file structure with a folder named "images" containing files: "example001.bmp", "example001bin.bmp", "volcan.bmp", "volcanbin.bmp", and "volcanbin2.bmp".
- Code Editor:** The central cell contains the same Python code as the previous screenshot, with the file paths changed to "volcan.bmp" and "volcanbin2.bmp". The output of the code is "No Pixels: 256".
- Image Viewer:** On the right, there are two image preview windows. The top window is titled "volcanbin.bmp" and shows a high-contrast, black-and-white version of the volcano image. The bottom window is titled "volcanbin2.bmp" and shows a similar high-contrast image, likely a different version of the processed image.



Inversión de filtro.

Archivos

- ..
- images
- example001.bmp
- example001bin.bmp
- volcan.bmp
- volcanbin.bmp

Variables Terminal

```
#Binary filter: Black and White
file = open('./images/volcan.bmp','rb')
fileo = open('./images/volcanbin.bmp','wb')
metadata = file.read(54)
fileo.write(metadata)
blanco = [0xff,0xff,0xff]
negro = [0x00,0x00,0x00]

file.seek(54,0)
no_pix = 0
limite = (pow(2, 24)-1)/2
while(True):
    pixel_data = file.read(3)
    if(len(pixel_data) > 0):
        valor_int = int.from_bytes(bytes(pixel_data),byteorder='l')
        if(valor_int<limite):
            fileo.write(bytes(blanco))
        else:
            fileo.write(bytes(negro))
        no_pix += 1
    else:
        break
print('No Pixels: '+str(no_pix))
file.close()
fileo.close()
```

Ejemplo que demuestra la lectura de archivos BMP

✓ 10:51 Python 3

The screenshot shows a Jupyter Notebook interface. On the left, a file browser displays local files including 'example001.bmp', 'example001bin.bmp', 'volcan.bmp', and 'volcanbin.bmp'. The central area contains a code cell with Python code for a binary filter. The code reads a BMP file ('volcan.bmp'), processes it using a black and white filter, and writes the result to a new file ('volcanbin.bmp'). The code also prints the total number of pixels. On the right, a preview image shows the processed binary representation of the volcano. The interface includes a top navigation bar with tabs for 'Notas de la versión', 'example001bin.bmp', 'volcanbin.bmp', and 'volcanbin2.bmp' (which is currently active). The bottom status bar shows disk space (86.53 GB) and the current time (10:51).

Notas de la versión

example001bin.bmp

volcanbin.bmp X

⋮ X

