



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE
INGENIERÍA Y TECNOLOGÍAS AVANZADAS



CARRERA: INGENIERÍA TELEMÁTICA

UNIDAD DE APRENDIZAJE: MULTIMEDIA

PRACTIUCA: 5 - FILTRO PALETA DE 16 COLORES



ALUMNO:

- GARNICA GONZALEZ CHRISTOPHER ALDAIR
- 2023640373

DOCENTE: NOE SIERRA ROMERO

FECHA: 16/02/25

GRUPO: 3TM2

-Cuantización y Mapeo de Color

El código implementado realiza un proceso de Cuantización de Color utilizando una paleta predefinida de 16 tonos. A continuación, se detallan los conceptos clave aplicados:

- Manipulación Binaria de Archivos BMP: Se accede directamente al archivo en modo binario ('rb'), respetando los primeros 54 bytes que corresponden a la cabecera (header) del formato BMP, donde se almacena la información de dimensiones y profundidad de color.
- Segmentación del Espacio de Color: Dado que una imagen de 24 bits puede representar hasta 2^{24} colores (16,777,216 combinaciones), el programa divide este rango total entre 16 para crear "escalones" o rangos de intensidad uniformes.
- Mapeo de Índices: Cada píxel original es leído y convertido a un valor entero. Dependiendo de su valor, se le asigna un índice del 0 al 15, sustituyendo su color original por el tono correspondiente de la paleta dorada definida en formato BGR (Blue, Green, Red).

Imagen original.



-Código

```
# 1. Definimos la paleta de 16 tonos DORADOS (Formato BGR para BMP)
# Nota: En BMP el orden es [Azul, Verde, Rojo]
paleta_dorada = [
    [0x00, 0x22, 0x44], # 0: Marrón oscuro (Sombra)
    [0x05, 0x2B, 0x5C], # 1: Bronce profundo
    [0x0A, 0x3D, 0x7A], # 2: Dorado ocre
    [0x10, 0x4E, 0x8B], # 3: Dorado mate
    [0x15, 0x5F, 0x9C], # 4: Dorado medio bajo
    [0x1A, 0x70, 0xAD], # 5: Dorado estándar
    [0x20, 0x81, 0xBE], # 6: Dorado vibrante
    [0x25, 0x92, 0xCF], # 7: Oro puro
    [0x30, 0xA3, 0xDF], # 8: Oro claro
    [0x40, 0xB4, 0xEF], # 9: Brillo metálico
    [0x55, 0xC5, 0xFF], # 10: Oro pálido
    [0x70, 0xD6, 0xFF], # 11: Reflejo brillante
    [0x8A, 0xE7, 0xFF], # 12: Oro suave
    [0xA5, 0xF2, 0xFF], # 13: Destello
    [0xC0, 0xF8, 0xFF], # 14: Casi blanco dorado
    [0xE0, 0xFF, 0xFF] # 15: Punto de luz (Brillo máximo)
]

# 2. Configuración de archivos (Asegúrate de subir 'volcan.bmp' a Colab)
input_path = './images/volcan (1).bmp'
output_path = 'volcan_dorado.bmp'

try:
    with open(input_path, 'rb') as file:
        with open(output_path, 'wb') as fileo:

            # Copiar metadata (54 bytes iniciales del encabezado BMP)
            metadata = file.read(54)
            fileo.write(metadata)

            # 3. Configuración de rangos
            # Un pixel de 24 bits tiene un valor máximo de 2^24 - 1
            valor_maximo = pow(2, 24) - 1
            tamaño_rango = valor_maximo / 16

            no_pix = 0
            print("Transformando imagen a tonos dorados...")

            while True:
                pixel_data = file.read(3) # B, G, R

                if len(pixel_data) == 3:
                    # Convertimos bytes a entero (Little Endian)
                    valor_int = int.from_bytes(pixel_data, byteorder='little')

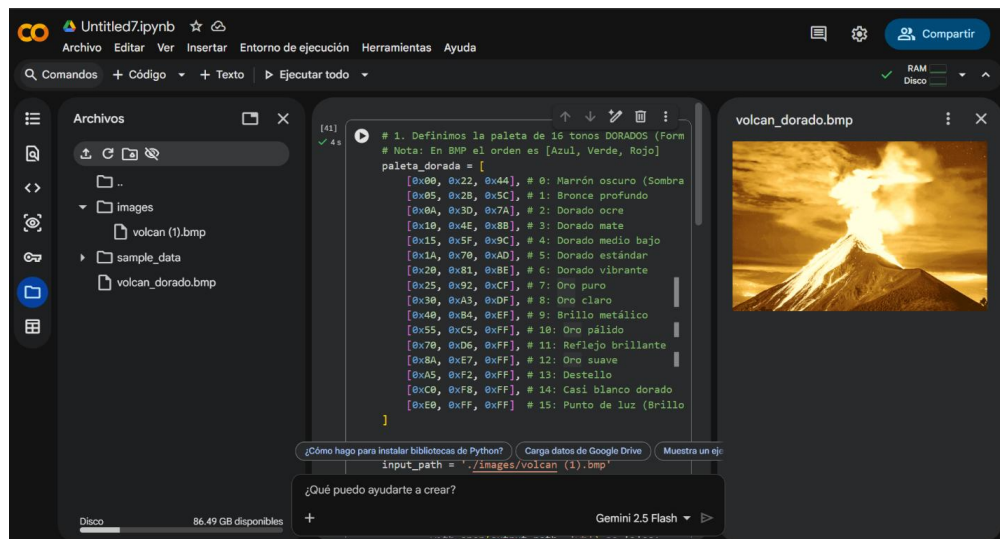
                    # Clasificamos el pixel en uno de los 16 niveles
                    indice = min(int(valor_int / tamaño_rango), 15)

                    # Escribimos el color de la paleta dorada
                    color_seleccionado = paleta_dorada[indice]
                    fileo.write(bytes(color_seleccionado))

                    no_pix += 1
                else:
                    break

            print(f'¡Listo! Imagen guardada como {output_path}')
            print(f'Píxeles procesados: {no_pix}')

except FileNotFoundError:
    print(f"Error: No encontré el archivo '{input_path}'.")
    print("Sugerencia: Sube el archivo 'volcan.bmp' a la carpeta principal de Colab (icono de carpeta a la izquierda).")
```





-Resumen de la practica

Esta practica demuestra la capacidad de manipular datos multimedia a bajo nivel mediante Python. El objetivo fue transformar una imagen original (el volcán) en una representación monocromática estilizada basada en una **paleta de 16 tonos dorados**.

A diferencia de un filtro convencional, este método reconstruye la imagen píxel por píxel basándose en la intensidad de los datos binarios. Además, se menciona la relación con la **esteganografía**, ya que el dominio de la estructura de archivos de imagen es el primer paso para ocultar información dentro de los bits menos significativos de los píxeles, una técnica fundamental en la seguridad informática y el espionaje.