



**Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas**

Alumna: García Ortiz Martha Lesly

Profesor: Sierra Romero Noe

Carrera: Telemática

Materia: Multimedia

Grupo: 3TM2

Practica 1: Esteganografía LSB en Imágenes BMP

Fecha: 18/02/2026

```
import struct import math

import os

def leer_bmp(filepath):

    with open(filepath, 'rb') as f:

        data = f.read()

        offset = struct.unpack_from('<I', data, 10)[0]

        width = struct.unpack_from('<i', data, 18)[0]

        height = struct.unpack_from('<i', data, 22)[0]

        row_size = (width * 3 + 3) & ~3

        header = bytearray(data[:offset])

        pixels = bytearray(data[offset:])

        return header, pixels, width, height, row_size

def guardar_bmp(filepath, header, pixels):

    """Guarda los datos en un nuevo archivo BMP."""

    with open(filepath, 'wb') as f:

        f.write(header)

        f.write(pixels)

def embed_lsb(src_path, dst_path, mensaje):

    """Oculta un mensaje de texto dentro de una imagen BMP usando LSB."""

    header, pixels, width, height, row_size = leer_bmp(src_path)

    msg_bytes = mensaje.encode('utf-8')

    msg_len = len(msg_bytes)

    datos = struct.pack('<I', msg_len) + msg_bytes

    bits = []

    for byte in datos:

        for i in range(7, -1, -1):

            bits.append((byte >> i) & 1)
```

```
if len(bits) > len(pixels):
    raise ValueError(f"Mensaje muy largo. Necesitas {len(bits)} px, tienes {len(pixels)}.")

pixels_mod = bytearray(pixels)

for idx, bit in enumerate(bits):
    pixels_mod[idx] = (pixels_mod[idx] & 0xFE) | bit

guardar_bmp(dst_path, header, pixels_mod)
print(f"[OK] Se ocultaron {msg_len} bytes en '{dst_path}'")

def extract_lsb(stego_path):
    """Recupera el mensaje oculto de una imagen BMP."""

    header, pixels, width, height, row_size = leer_bmp(stego_path)

    len_bits = [pixels[i] & 1 for i in range(32)]

    len_bytes = bytearray()
    for i in range(0, 32, 8):
        byte_val = 0
        for bit in len_bits[i:i+8]:
            byte_val = (byte_val << 1) | bit
        len_bytes.append(byte_val)

    msg_len = struct.unpack('<I', len_bytes)[0]
    start = 32
    end = 32 + (msg_len * 8)
    if end > len(pixels):
        return "Error: Longitud inválida o archivo corrupto."
    msg_bits = [pixels[i] & 1 for i in range(start, end)]
```

```

msg_bytes = bytearray()

for i in range(0, len(msg_bits), 8):

    byte_val = 0

    for bit in msg_bits[i:i+8]:

        byte_val = (byte_val << 1) | bit

    msg_bytes.append(byte_val)

return msg_bytes.decode('utf-8')

def calcular_psnr(original_path, stego_path):

    """Calcula el Peak Signal-to-Noise Ratio entre dos imágenes."""

    _, p_orig, w, h, _ = leer_bmp(original_path)

    _, p_stego, _, _, _ = leer_bmp(stego_path)

    mse = 0

    limit = min(len(p_orig), len(p_stego))

    for i in range(limit):

        diff = p_orig[i] - p_stego[i]

        mse += diff ** 2

    mse = mse / limit

    if mse == 0:

        return float('inf')

    max_pixel = 255.0

    psnr = 10 * math.log10((max_pixel ** 2) / mse)

    return psnr

if __name__ == "__main__":

    print("--- INICIO PRÁCTICA 1: ESTEGANOGRÁFÍA LSB ---")

    img_original = "volcan.bmp"

    img_stego = "stego.bmp"

    mensaje_secreto = "TELEMÁTICA SECRETA 2025"

```

```
if not os.path.exists(img_original):

    print(f"ERROR: No se encuentra '{img_original}'. Por favor coloca un BMP en la
carpeta.")

else:

    try:

        print(f"\n1. Ocultando: '{mensaje_secreto}'")

        embed_lsb(img_original, img_stego, mensaje_secreto)

        print(f"\n2. Extrayendo mensaje de '{img_stego}'...")

        recuperado = extract_lsb(img_stego)

        print(f" Mensaje recuperado: '{recuperado}'")

        if recuperado == mensaje_secreto:

            print(" ¡ÉXITO! El mensaje coincide perfectamente.")

        else:

            print(" FALLO: El mensaje no coincide.")

        print("\n3. Calculando calidad de imagen (PSNR)...")

        val_psnr = calcular_psnr(img_original, img_stego)

        print(f" PSNR: {val_psnr:.2f} dB")

        if val_psnr > 40:

            print(" Resultado excelente (Imperceptible al ojo humano).")

        else:

            print(" Calidad baja (Podría notarse ruido visual).")

    except Exception as e:

        print(f"\n Ocurrió un error inesperado: {e}")
```

... --- INICIO PRÁCTICA 1: ESTEGANOGRÁFÍA LSB ---

1. Ocultando: 'TELEMÁTICA SECRETA 2025'
[OK] Se ocultaron 24 bytes en 'stego.bmp'
2. Extrayendo mensaje de 'stego.bmp'...
Mensaje recuperado: 'TELEMÁTICA SECRETA 2025'
 ¡ÉXITO! El mensaje coincide perfectamente.
3. Calculando calidad de imagen (PSNR)...
PSNR: 96.35 dB
 Resultado excelente (Imperceptible al ojo humano).

Fig1. Resultado del mensaje oculto



Fig2. Imagen original



Fig3. Imagen stego.bmp

Imagen (px)	Tamaño mensaje	PSNR obtenido	¿Imperceptible?
200 × 200	50 bytes	75.32 dB	Si
200 × 200	500 bytes	65.84 dB	Sí
512 × 512	5,000 bytes	64.07 db	‘Sí
512 × 512	Capacidad máx.	51.14 dB	Sí

Fig4. Tabla con resultados obtenidos

Preguntas:

1. ¿Qué sucede si se intenta ocultar un mensaje más largo que la capacidad de la imagen?

Respuesta Técnica:

Si se intenta escribir más bits de los que la imagen soporta, el índice del arreglo de píxeles excederá el tamaño máximo de la lista de datos , eso provoca un error y el programa se detendría sin guardar nada.

```
if len(bits) > len(pixels):  
    raise ValueError('Mensaje demasiado largo para esta imagen')
```

2. Comparación visual e histograma: ¿Hay diferencias observables?

No hay diferencias. Al modificar solo el bit menos significativo, el valor de un color cambia como máximo 1 unidad (ej. de 254 a 255).

3. Proponga una estrategia para aumentar la capacidad a 2 LSBs por canal. ¿Cómo afecta al PSNR?

El PSNR **disminuiría** (es decir, la calidad de la imagen bajaría).

Con 1 LSB, el error máximo por píxel es 1. Con 2 LSBs, puedes cambiar un valor hasta en 3 unidades (ej. cambiar 00 por 11 en binario añade 3). Al ser el error cuadrático medio (MSE) mayor, la relación señal-ruido (PSNR) cae. Aunque probablemente seguiría siendo imperceptible a simple vista, la degradación matemática es mayor.

4. ¿Por qué BMP es preferible a JPEG para esteganografía LSB? ¿Qué ocurre si se guarda como JPEG?

El formato BMP (en su forma estándar) es un formato **sin compresión**. Guarda el valor exacto de cada píxel tal cual es.

El formato JPEG utiliza **compresión con pérdida**, para ahorrar espacio, JPEG usa algoritmos matemáticos que eliminan información que el ojo humano "no nota mucho", como los cambios sutiles de alta frecuencia.