



INSTITUTO POLITECNICO  
NACIONAL  
UNIDAD PROFESIONAL  
INTERDISCIPLINARIA EN INGENIERÍA  
Y TECNOLOGÍAS AVANZADAS



# *Practica 6:* **ESTENOGRAFÍA LSB EN IMÁGENES BMP**

Alumno: Martínez Barrueta Mariana

Profesor: Sierra Romero Noe

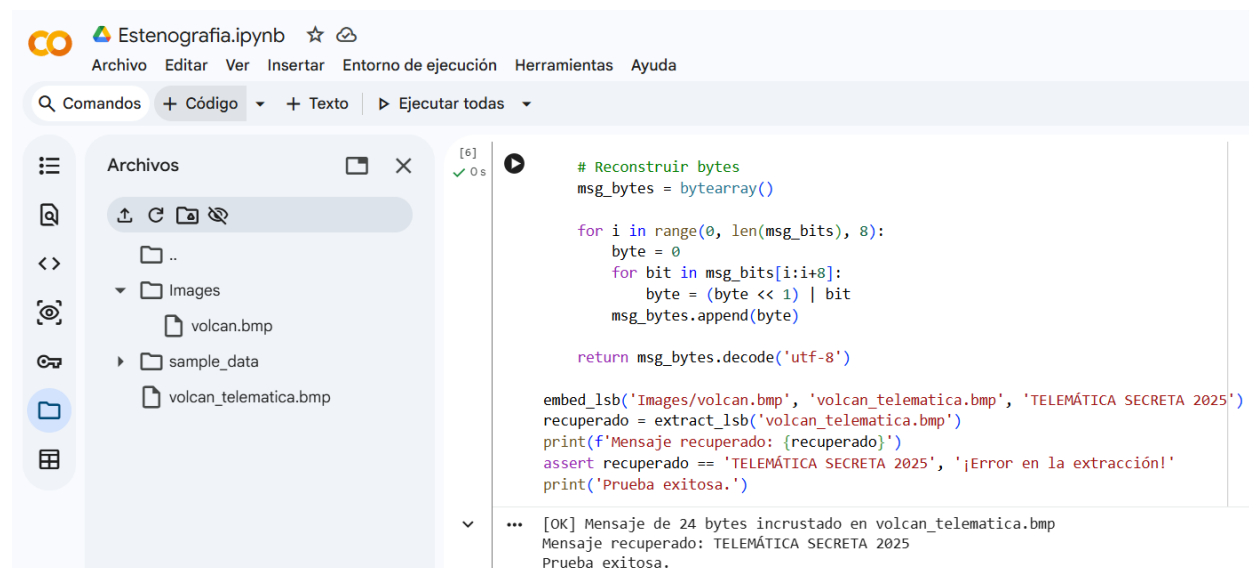
Boleta: 2024640160

*Multimedia*

# Estenografía LSB

## 2.2 Procedimiento

Compilando el código de la función auxiliar de lectura/escritura de píxeles, de incrustación (embedding) y extracción (extraction) en “Colab” y al mismo tiempo ejecutando la prueba de ida y vuelta:



```
# Reconstruir bytes
msg_bytes = bytearray()

for i in range(0, len(msg_bits), 8):
    byte = 0
    for bit in msg_bits[i:i+8]:
        byte = (byte << 1) | bit
    msg_bytes.append(byte)

return msg_bytes.decode('utf-8')

embed_lsb('Images/volcan.bmp', 'volcan_telematica.bmp', 'TELEMÁTICA SECRETA 2025')
recuperado = extract_lsb('volcan_telematica.bmp')
print(f'Mensaje recuperado: {recuperado}')
assert recuperado == 'TELEMÁTICA SECRETA 2025', '¡Error en la extracción!'
print('Prueba exitosa.')
```

[OK] Mensaje de 24 bytes incrustado en volcan\_telematica.bmp  
Mensaje recuperado: TELEMÁTICA SECRETA 2025  
Prueba exitosa.

Y poniendo la imagen incrustada “volcán\_telematica.bmp” en “hexed.it” y comparándola con la imagen original “volcán.bmp”:

-Sin título- x	volcan_telematica.bmp x	volcan.bmp x
00000000	42 4D 36 80 70 00 00 00	00 00 36 00 00 00 28 00
00000010	00 00 80 07 00 00 00 05	00 00 01 00 18 00 00 00
00000020	00 00 00 80 70 00 00 00	00 00 00 00 00 00 00 00
00000030	00 00 00 00 00 00 04 00	0C 02 00 0A 02 00 0A 00
00000040	00 08 08 02 14 0E 08 1A	06 00 14 14 0C 26 18 0E
00000050	32 15 09 30 18 0C 36 07	00 27 16 0D 36 2A 22 4B
00000060	08 00 28 03 00 1D 02 01	16 1E 11 31 0C 00 1E 27
00000070	18 38 2C 19 3A 15 02 23	16 02 23 1B 06 27 11 01
00000080	16 20 0A 26 1D 07 23 20	0C 24 18 06 1C 17 04 1B
00000090	12 03 14 23 0E 2A 30 13	44 3E 21 54 4A 31 5C 27
000000A0	0E 36 26 0E 3B 33 1C 49	1C 08 34 14 04 27 0A 00
000000B0	15 04 00 0A 10 04 1A 11	00 21 5E 4A 77 63 4E 7D
000000C0	40 2E 58 13 02 25 02 01	10 26 1E 2E 0D 09 10 0F
000000D0	0A 0F 04 00 07 1E 14 25	2E 20 3E 37 24 4D 3A 27
000000E0	50 17 02 2B 10 00 1E 1F	10 28 08 00 10 09 00 10
000000F0	0B 00 14 0A 00 12 02 00	09 03 02 04 05 06 00 06
00000100	09 05 08 04 0E 08 00 16	27 19 3A 14 05 2A 0E 00
00000110	23 21 10 35 26 13 40 36	21 52 25 0C 46 31 19 4F
00000120	19 05 2F 2F 1F 41 0D 00	20 21 16 37 22 15 3B 2F
00000130	21 4A 23 14 40 06 00 22	00 00 1B 19 0E 2F 08 00
00000140	1C 00 00 0F 05 00 12 00	00 0D 0F 08 1D 20 17 31
00000150	01 00 13 0E 04 22 13 08	29 12 07 27 00 00 15 04
00000160	00 18 0E 05 20 04 00 16	11 07 25 11 06 26 09 00
00000170	1F 06 00 1E 0D 01 25 0D	02 23 0F 04 25 0F 05 23
00000180	05 00 19 05 00 17 0A 00	1E 00 00 14 0D 03 21 21
00000190	18 33 11 08 23 0B 03 1A	00 00 0D 00 00 0A 02 00
000001A0	09 00 00 06 03 00 09 08	04 0F 0C 0A 16 23 20 30
000001B0	0E 09 1E 00 00 11 39 26	47 31 20 41 18 10 2F 07
000001C0	04 2B 08 04 3F 10 10 4C	06 06 36 0C 0B 33 0B 00



-Sin título- x	volcan_telematica.bmp x	volcan.bmp x	
00000000	42 4D 36 80 70 00 00 00	00 00 36 00 00 00 28 00	BM6Çp.....6... (.
00000010	00 00 80 07 00 00 00 05	00 00 01 00 18 00 00 00	..Ç.....
00000020	00 00 00 80 70 00 00 00	00 00 00 00 00 00 00 00	...Çp.....
00000030	00 00 00 00 00 00 05 01	0D 02 00 0A 03 00 0B 00	.....
00000040	00 08 09 03 14 0E 08 1B	06 00 15 15 0C 27 19 0F	.....'
00000050	33 14 09 31 19 0D 37 07	00 26 17 0D 37 2B 22 4A	3..1..7..&..7+"J
00000060	09 00 28 02 00 1C 02 00	16 1F 10 30 0D 00 1E 27	..(.....0...'
00000070	18 38 2C 19 3A 15 02 23	17 02 22 1B 06 26 10 00	.8,..:..#...".&..
00000080	17 21 0B 27 1D 07 23 21	0D 24 18 06 1D 16 05 1A	..!..'..#!.\$.....
00000090	12 02 14 22 0E 2B 30 13	44 3E 20 55 4B 30 5C 27	..."+0.D> UK0\'
000000A0	0F 37 27 0E 3A 33 1C 49	1D 09 34 15 04 26 0B 00	.7'..:3.I..4..&..
000000B0	15 05 00 0B 10 04 1A 10	01 20 5E 4B 76 62 4E 7D	..... ^KvBN}
000000C0	41 2F 58 13 03 25 02 00	11 26 1E 2F 0D 08 11 0E	A/X..%.&./....
000000D0	0A 0F 05 00 07 1E 14 24	2F 20 3F 37 25 4C 3B 27	.....\$/ ?%L;'
000000E0	51 16 03 2A 10 00 1F 1E	10 28 09 00 11 08 00 10	Q..*.....(.....
000000F0	0A 00 14 0A 00 13 02 00	08 03 02 04 04 06 00 07	.....
00000100	08 04 08 05 0E 08 01 16	27 19 3B 14 05 2B 0E 00	.....'..;..+..
00000110	22 21 10 35 27 13 40 36	21 52 25 0C 46 31 19 4F	"!.5'..@6!R%.F1.0
00000120	19 05 2F 2F 1F 41 0D 00	20 21 16 37 22 15 3B 2F	..//.A..!.7"..;/
00000130	21 4A 23 14 40 06 00 22	00 00 1B 19 0E 2F 08 00	!J#..@.."....../..
00000140	1C 00 00 0F 05 00 12 00	00 0D 0F 08 1D 20 17 31	..... .1
00000150	01 00 13 0E 04 22 13 08	29 12 07 27 00 00 15 04	....."..).'. ....
00000160	00 18 0E 05 20 04 00 16	11 07 25 11 06 26 09 00	.....%.&..
00000170	1F 06 00 1E 0D 01 25 0D	02 23 0F 04 25 0F 05 23	.....%.#..%.#
00000180	05 00 19 05 00 17 0A 00	1E 00 00 14 0D 03 21 21	.....!!
00000190	18 33 11 08 23 0B 03 1A	00 00 0D 00 00 0A 02 00	.3..#.....
000001A0	09 00 00 06 03 00 09 08	04 0F 0C 0A 16 23 20 30	.....# 0
000001B0	0E 09 1E 00 00 11 39 26	47 31 20 41 18 10 2F 07	.....9&G1 A.. /.
000001C0	04 2B 08 04 3F 10 10 4C	06 06 36 0C 0B 33 0B 00	..+..?..L..6..3..

Y comparando los resultados, en el byte 54 en ambas imágenes termina el header y empiezan los pixeles de colores en “RGB” y allí es donde se hace la modificación y se esconde el texto “TELEMATICA SECRETA 2025”. Se cambie el bit menos significativo de cada uno de los pixeles sin alterar su apariencia visual.

Después se realiza el calculo de PSNR que es la relación señal-ruido de pico entre la imagen original y la de estenografiada para evaluar la degradación visual:

```

Estenografia.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
Comandos + Código + Texto Ejecutar todas
Archivos
  ..
  Images
    volcan.bmp
  sample_data
    volcan_telematica.bmp
Disco 86.46 GB de espacio disponible

import math

def calcular_psnr(original_path, stego_path):
    _, pix_orig, w, h, rs = leer_bmp(original_path)
    _, pix_steg, _, _ = leer_bmp(stego_path)

    mse = sum((a - b)**2 for a, b in zip(pix_orig, pix_steg)) / (w * h * 3)

    if mse == 0:
        return float('inf')

    psnr = 10 * math.log10(255**2 / mse)

    print(f'MSE: {mse:.6f}')
    print(f'PSNR: {psnr:.2f} dB (>40 dB: cambio imperceptible)')

    return psnr

calcular_psnr('Images/volcan.bmp', 'volcan_telematica.bmp')

MSE: 0.000015
PSNR: 96.43 dB (>40 dB: cambio imperceptible)
96.43286315998367

```

## 2.3 Experimento de capacidad

Imagen(px)	Tamaño mensaje	PSNR obtenido	¿Imperceptible?
200x200	50 bytes	76.54 dB	Cambio imperceptible
200x200	500 bytes	66.35 dB	Cambio imperceptible
512x512	5000 bytes	56.30 dB	Cambio imperceptible
512x512	Capacidad máx.	76.54 dB	Cambio imperceptible

Con imagen de 200x200 pixeles con 50 bytes

The screenshot shows a Jupyter Notebook titled "Estenografia.ipynb". The left sidebar displays a file explorer with a directory structure including "Images" (containing hongo.bmp, hongo.png, and volcan.bmp) and "sample\_data" (containing hongo\_50bytes.bmp and volcan\_telematica.bmp). The main code cell, labeled [21], contains the following Python code:

```
msg_bytes = [b'PACAP[4] W A L U I A A I I m p e r c e p t i b l e , v o l c a n _ m a t a /']

# Reconstruir bytes
msg_bytes = bytearray()

for i in range(0, len(msg_bits), 8):
    byte = 0
    for bit in msg_bits[i:i+8]:
        byte = (byte << 1) | bit
    msg_bytes.append(byte)

return msg_bytes.decode('utf-8')

embed_lsb('Images/hongo.bmp', 'hongo_50bytes.bmp', 'Imagen 200x200 con tamaño del mensaje de 50 bytes')
recuperado = extract_lsb('hongo_50bytes.bmp')
print(f'Mensaje recuperado: {recuperado}')
assert recuperado == 'Imagen 200x200 con tamaño del mensaje de 50 bytes', '¡Error en la extracción!'
print('Prueba exitosa.')
```

The output of the cell shows a successful execution: "[OK] Mensaje de 50 bytes incrustado en hongo\_50bytes.bmp" and "Mensaje recuperado: Imagen 200x200 con tamaño del mensaje de 50 bytes".

The screenshot shows the same Jupyter Notebook interface, but with a new code cell, labeled [22], added. This cell defines a function to calculate the PSNR (Peak Signal-to-Noise Ratio) between two images.

```
import math

def calcular_psnr(original_path, stego_path):
    _, pix_orig, w, h, rs = leer_bmp(original_path)
    _, pix_steg, _, _, _ = leer_bmp(stego_path)

    mse = sum((a - b)**2 for a, b in zip(pix_orig, pix_steg)) / (w * h * 3)

    if mse == 0:
        return float('inf')

    psnr = 10 * math.log10(255**2 / mse)

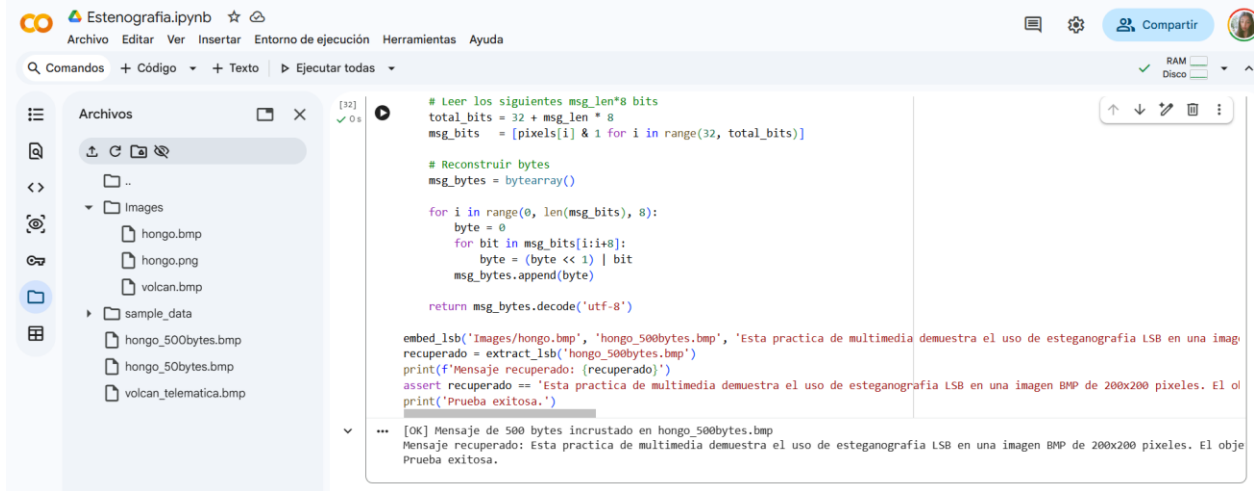
    print(f'MSE: {mse:.6f}')
    print(f'PSNR: {psnr:.2f} dB (>40 dB: cambio imperceptible)')

    return psnr

calcular_psnr('Images/hongo.bmp', 'hongo_50bytes.bmp')
```

The output of the cell shows the calculated values: "MSE: 0.001442" and "PSNR: 76.54 dB (>40 dB: cambio imperceptible)".

Con imagen de 200x200 pixeles con 500 bytes



```
[32] ✓ 0.8
# Leer los siguientes msg_len*8 bits
total_bits = 32 + msg_len * 8
msg_bits = [pixels[i] & 1 for i in range(32, total_bits)]

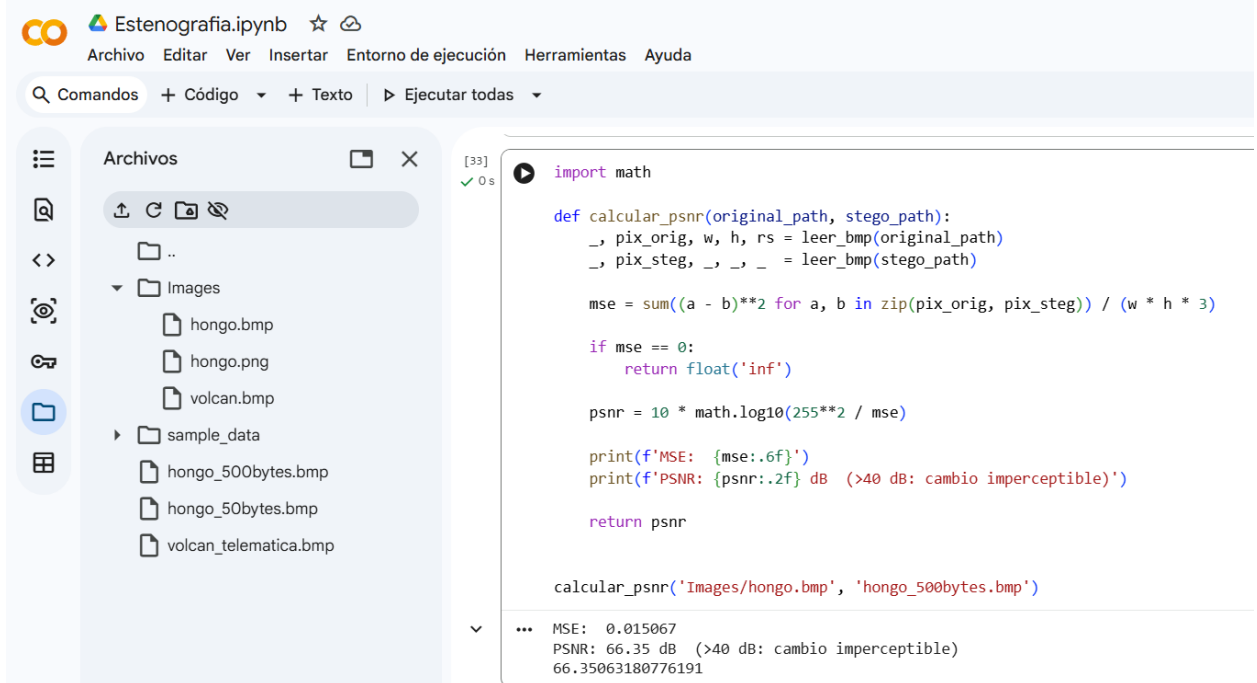
# Reconstruir bytes
msg_bytes = bytearray()

for i in range(0, len(msg_bits), 8):
    byte = 0
    for bit in msg_bits[i:i+8]:
        byte = (byte << 1) | bit
    msg_bytes.append(byte)

return msg_bytes.decode('utf-8')

embed_lsb('Images/hongo.bmp', 'hongo_500bytes.bmp', 'Esta practica de multimedia demuestra el uso de esteganografia LSB en una imagen recuperado = extract_lsb('hongo_500bytes.bmp')
print(f'Mensaje recuperado: {recuperado}')
assert recuperado == 'Esta practica de multimedia demuestra el uso de esteganografia LSB en una imagen BMP de 200x200 pixeles. El ol
print('Prueba exitosa.')
```

... [OK] Mensaje de 500 bytes incrustado en hongo\_500bytes.bmp  
Mensaje recuperado: Esta practica de multimedia demuestra el uso de esteganografia LSB en una imagen BMP de 200x200 pixeles. El obje  
Prueba exitosa.



```
[33] ✓ 0.5
import math

def calcular_psnr(original_path, stego_path):
    _, pix_orig, w, h, rs = leer_bmp(original_path)
    _, pix_steg, _, _ = leer_bmp(stego_path)

    mse = sum((a - b)**2 for a, b in zip(pix_orig, pix_steg)) / (w * h * 3)

    if mse == 0:
        return float('inf')

    psnr = 10 * math.log10(255**2 / mse)

    print(f'MSE: {mse:.6f}')
    print(f'PSNR: {psnr:.2f} dB (>40 dB: cambio imperceptible)')

    return psnr

calcular_psnr('Images/hongo.bmp', 'hongo_500bytes.bmp')
```

... MSE: 0.015067  
PSNR: 66.35 dB (>40 dB: cambio imperceptible)  
66.35063180776191

## Con imagen de 512x512 pixeles con 500 bytes



The screenshot shows a Jupyter Notebook titled "Estenografia.ipynb". The left sidebar displays a file explorer with a directory structure: `..`, `Images` (containing `pajaro.bmp` and `pajaro.png`), and `sample_data` (containing `pajaro.bmp` and `pajaro_5000bytes.bmp`). The main area shows a code cell with the following Python code:

```
[6] 0s
return msg_bytes.decode('utf-8')

# ===== (BLOQUE 2) Mensaje tal como lo tienes =====
mensaje = ("Esta practica de multimedia demuestra el uso de esteganografia LSB en una imagen BMP de 200x200 pixeles. "
"El objetivo es ocultar informacion dentro de los bits menos significativos de los colores sin modificar la "
"apariencia visual de la imagen. El proceso incluye lectura de la estructura del archivo, modificacion de "
"pixeles, almacenamiento de la imagen resultante y recuperacion del mensaje. Tambien se analiza la capacidad "
"de almacenamiento, la integridad del texto y la calidad visual usando PSNR.") * 10

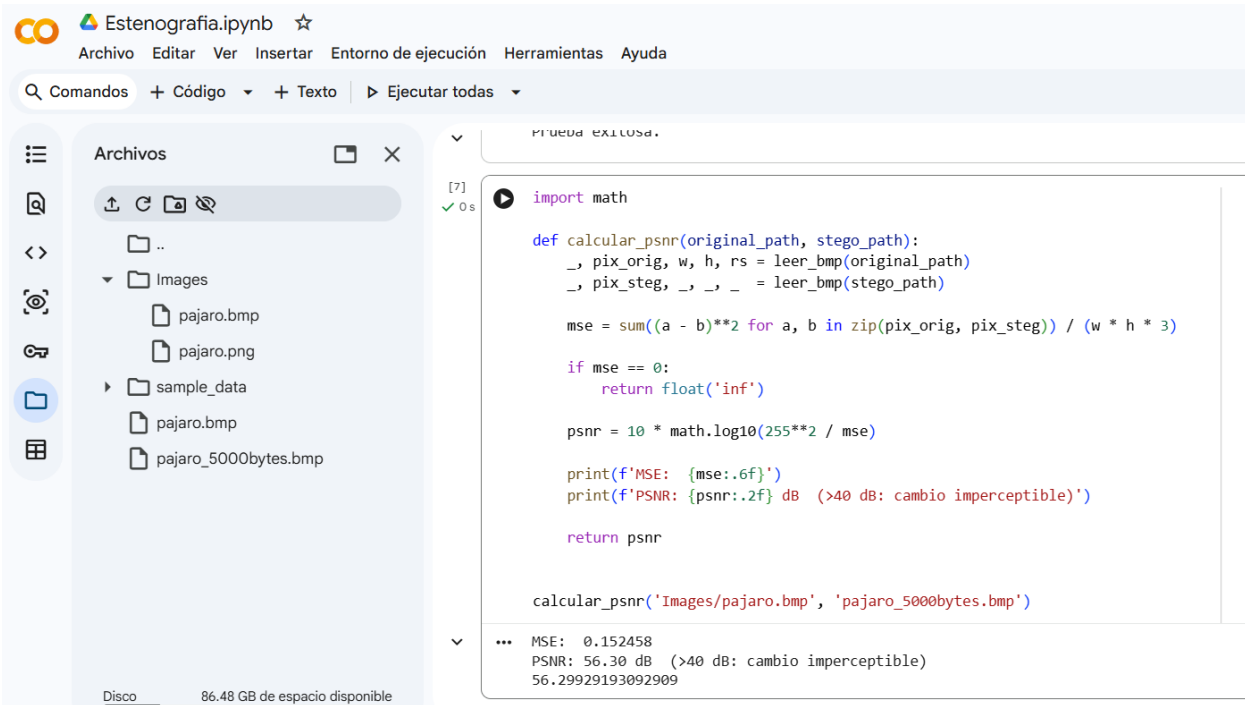
print(len(mensaje.encode('utf-8')))

# ===== Prueba (misma logica, solo nombres consistentes) =====
embed_lsb('Images/pajaro.bmp', 'pajaro_5000bytes.bmp', mensaje)
recuperado = extract_lsb('pajaro_5000bytes.bmp') # <- era 50000, por eso fallaba

print(f'Mensaje recuperado: {recuperado}')
assert recuperado == mensaje, '¡Error en la extracción!'
print('Prueba exitosa.')
```

The output of the cell shows:

```
... 5000
[OK] Mensaje de 5000 bytes incrustado en pajaro_5000bytes.bmp
Mensaje recuperado: Esta practica de multimedia demuestra el uso de esteganografia LSB en una imagen BMP de 200x200 pixeles. El obje
Prueba exitosa.
```



The screenshot shows a Jupyter Notebook titled "Estenografia.ipynb". The left sidebar displays a file explorer with a directory structure: `..`, `Images` (containing `pajaro.bmp` and `pajaro.png`), and `sample_data` (containing `pajaro.bmp` and `pajaro_5000bytes.bmp`). The main area shows a code cell with the following Python code:

```
[7] 0s
import math

def calcular_psnr(original_path, stego_path):
    _, pix_orig, w, h, rs = leer_bmp(original_path)
    _, pix_steg, _, _, _ = leer_bmp(stego_path)

    mse = sum((a - b)**2 for a, b in zip(pix_orig, pix_steg)) / (w * h * 3)

    if mse == 0:
        return float('inf')

    psnr = 10 * math.log10(255**2 / mse)

    print(f'MSE: {mse:.6f}')
    print(f'PSNR: {psnr:.2f} dB (>40 dB: cambio imperceptible)')

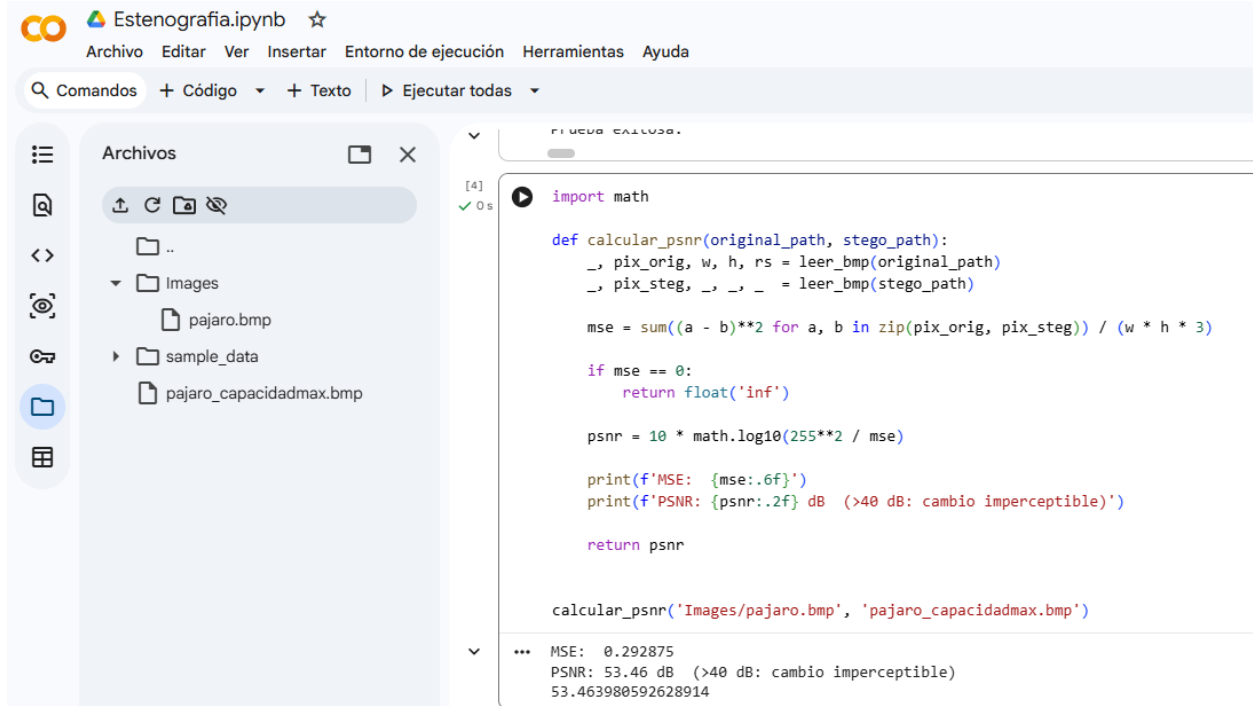
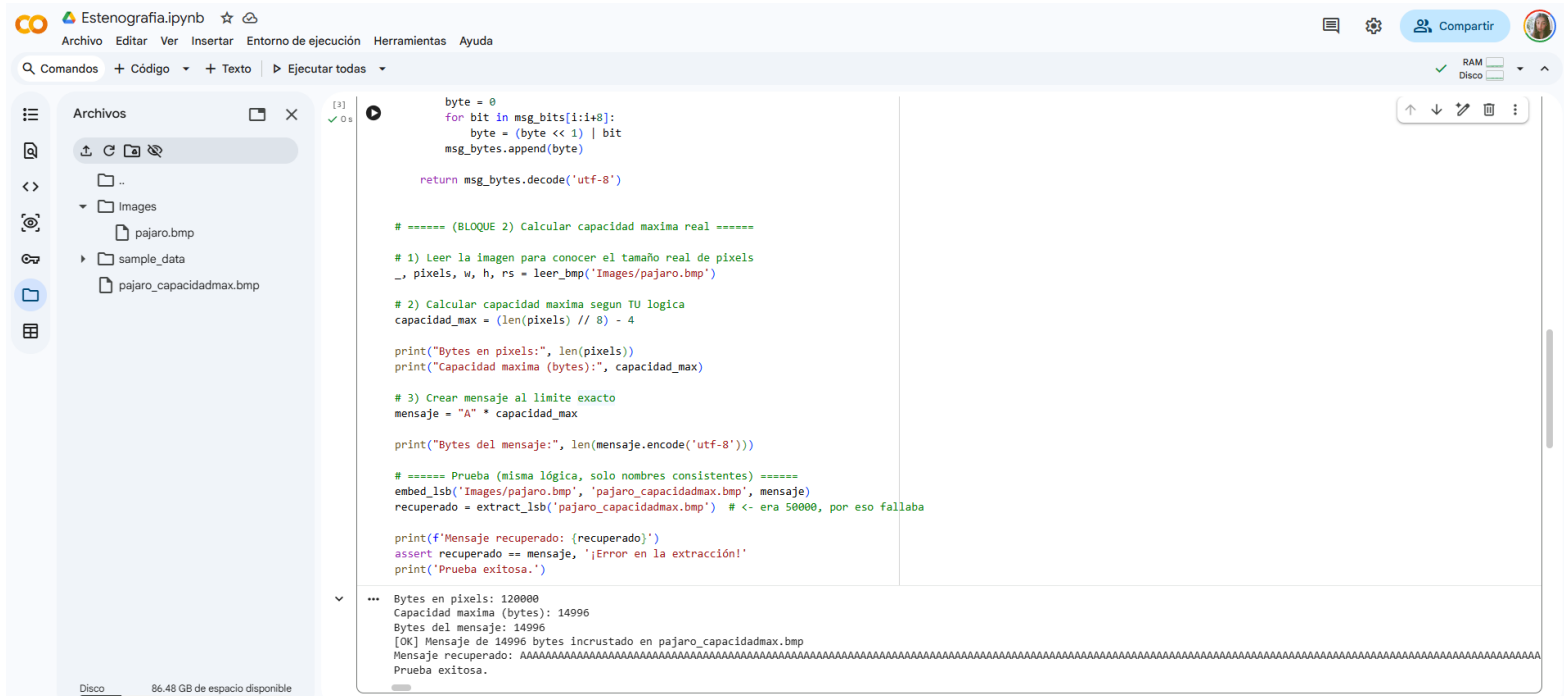
    return psnr

calcular_psnr('Images/pajaro.bmp', 'pajaro_5000bytes.bmp')
```

The output of the cell shows:

```
... MSE: 0.152458
PSNR: 56.30 dB (>40 dB: cambio imperceptible)
56.29929193092909
```

Con imagen de 512x512 pixeles con capacidad máxima



## 2.4 Preguntas de análisis

1. **¿Qué sucede si intenta ocultar un mensaje más largo que la capacidad de la imagen? Implemente un control de error apropiado.**

Si intentas esconder un mensaje más largo que lo que la imagen puede guardar:

- No caben todos los bits.
- El programa puede marcar error.
- O el mensaje se guardaría incompleto y saldría mal al extraerlo.

Por eso se pone este control:

***if len(bits) > len(pixels):***

***raise ValueError("Mensaje demasiado largo para esta imagen")***

Esto evita que el programa falle y se muestre un mensaje claro.

2. **Compare visualmente (o mediante histograma) la imagen original y la estenografiada. ¿Hay diferencias observables en le histograma de intensidades?**

Visualmente casi no se nota diferencia cuando usamos 1 LSB porque solo cambiamos el último bit del color. Eso significa que el color cambia máximo en 1 unidad (por ejemplo 120 puede pasar a 121).

Y en el histograma la forma general casi no cambia, pero puede haber pequeñas diferencias entre números pares e impares.

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

Una forma de aumentar la capacidad de ocultamiento es usar **2 bits menos significativos (2 LSB)** en lugar de solo 1 por cada canal de color (R, G, B).

Normalmente se modifica solo el último bit del color y cada canal guarda 1 bit de información.

Si usamos 2 LSB, se modifican los dos últimos bits de cada canal por lo tanto puede guardar 2 bits y la capacidad de la imagen se duplica.

Por ejemplo:

Antes: un píxel guardaba 3 bits (1 por R, G, B).

Ahora: un píxel guarda 6 bits.

El PSNR mide la calidad de la imagen comparada con la original.

Al usar 2 LSB: se modifican más los valores de color, el cambio ya no es de  $\pm 1$ , puede ser hasta  $\pm 3$  y la imagen se altera más.

Por lo tanto, aumenta la cantidad de información oculta pero baja la calidad de la imagen y el PSNR disminuye.



**4. ¿Por qué el formato BMP es preferible al JPEG para esteganografía LSB? ¿Qué ocurriría si se guarda la imagen stego como JPEG?**

El formato BMP es preferible porque **no comprime la imagen**. Guarda los valores de los píxeles exactamente como están, por lo que los bits que se modifican con la técnica LSB se conservan sin cambios.

En cambio, el formato JPEG: usa compresión con pérdida, modifica los valores de los píxeles al guardar la imagen, cambia los bits menos significativos.

Si una imagen con mensaje oculto se guarda como JPEG: los bits donde estaba el mensaje se alteran, el mensaje se daña o se pierde, al intentar extraerlo ya no se recupera correctamente.