



# Trabajo Práctico de Implementación: Estenografía

72.04 - Criptografía y Seguridad

Segundo Cuatrimestre 2025

Grupo 8

Ana Negre - 63074

Camila Lee - 63382

Matías Leporini - 62872

# Índice general

1. Análisis del documento de Majeed y Sulaiman.....	2
1.1. Organización del documento.....	2
1.2. Descripción del algoritmo.....	2
1.3. Notación utilizada.....	2
2. Comparación de algoritmos de esteganografía.....	5
2.1. Métricas evaluadas.....	5
2.2. Resultados experimentales.....	6
2.3. Análisis de resultados.....	6
2.4. Análisis visual de diferencias.....	10
3. Estegoanálisis.....	13
4. Análisis de LSBI.....	16
4.1. Comparación con LSB1.....	16
4.2. Almacenamiento de patrones.....	16
4.3. Dificultades de implementación.....	17
5. Mejoras o futuras extensiones.....	18
6. Bibliografía.....	19
7. Apéndice.....	20

# 1. Análisis del documento de Majeed y Sulaiman

Para la implementación del algoritmo LSBI (LSB Improved), se utilizó como referencia el *paper* de Majeed y Sulaiman (2015), que propone una mejora sobre el método estándar LSB mediante inversión selectiva de bits en imágenes de 24 bits. Dado que este documento constituye la única especificación técnica del método, resulta relevante evaluar su claridad, precisión y reproducibilidad. En esta sección se analiza la estructura del *paper*, la descripción del algoritmo y las ambigüedades o inconsistencias encontradas durante el proceso de implementación.

## 1.1. Organización del documento

El *paper* presenta una estructura académica convencional dividida en 8 secciones que sigue un orden lógico: contextualización histórica (sección 1), revisión de fundamentos técnicos (secciones 2-4), presentación de la contribución (secciones 5-6), resultados experimentales (sección 7) y conclusiones (sección 8).

La principal debilidad del *paper* es que no incluye una sección dedicada a trabajos relacionados. Las comparaciones con otros métodos aparecen solo de forma breve en la introducción y en una tabla dentro de los resultados. Esto hace que sea difícil entender con claridad cómo se posiciona el método propuesto frente a otros existentes o cuál es su verdadero aporte dentro del campo.

## 1.2. Descripción del algoritmo

Los autores presentan el método con dos enfoques: una explicación coloquial paso a paso (sección 5) y un pseudocódigo formal (sección 6). Sin embargo, ambos presentan limitaciones. La descripción coloquial explica la lógica intuitivamente -aplicar LSB1, analizar patrones, contar cambios, invertir selectivamente- pero carece de detalles de implementación sobre estructuras de datos e iteración. El pseudocódigo es más formal pero introduce ambigüedades: no especifica cuándo calcular los patrones, cómo gestionar eficientemente las agrupaciones por patrón, ni el orden exacto de acceso a memoria.

Para nuestra implementación, nos basamos principalmente en la estructura conceptual de la sección 5, pero tuvimos que tomar decisiones no especificadas.

## 1.3. Notación utilizada

El *paper* presenta diversas deficiencias en claridad y precisión técnica que dificultan su reproducibilidad. A lo largo del documento se observan errores gramaticales, tipográficos y de puntuación que, si bien individualmente podrían considerarse menores, en conjunto afectan la capacidad del lector para comprender e implementar el método propuesto. Entre los ejemplos identificados se encuentran:

#### **Peak Signal-to-Noise Ratio (PSNR):**

In In steganography technique, PSNR is the standard measurement uses to test the quality of the

must not affect both capacity and the quality of stego image□

#### **Extractos 1 y 2 - Errores variados en el paper.**

Los errores más problemáticos ocurren en secciones técnicas críticas. En la Sección 5 (página 3), al describir los pasos del algoritmo, el texto indica: "Calculate the pattern occurrences of these two bits on the cover-image, which are either 00, 10, 10, or 11" [1]. El patrón 10 aparece dos veces cuando debería listar los cuatro patrones únicos: 00, 01, 10 y 11.

1. Calculate the pattern occurrences of these two bits on the cover-image, which are either 00, 10, 10, or 11. Classify the cover image according to these four patterns.

#### **Extracto 3 - Error de enumeración en el paper.**

Aún más grave es el error en el único ejemplo numérico detallado (Sección 5, página 4): al byte D le falta su último bit, y su inferencia se hace en base a descripciones posteriores. Como es el único ejemplo trabajado en detalle, su corrección es crítica para la comprensión del algoritmo.

Secret message	:	1011	
Cover image	:	10001100	10101101
		A	B
		10101011	1010110□
		C	D
LSB stego-image	:	10001 <b>101</b>	10101 <b>100</b>
		A	B
		10101 <b>011</b>	10101 <b>101</b>
		C	D

#### **Extracto 4 - Error en ejemplo del paper.**

El documento también presenta ambigüedades en definiciones fundamentales. La fórmula del PSNR (página 6) incluye la variable MAX pero nunca define explícitamente qué representa. Solo consultando fuentes externas se confirma que MAX = 255 para imágenes de 8 bits. Existen también inconsistencias numéricas significativas. En la Sección 7, los autores afirman que su mensaje de prueba contiene "226 characters or 2881 bits" [1], pero  $226 \times 8 = 1808$  bits. Incluso considerando el overhead del protocolo (aproximadamente 1,880 bits totales), la discrepancia de ~1000 bits no se explica.

En la conclusión aparece una contradicción importante. Los autores afirman que “*the goal of the technique is not just to increase the capacity of the message...*” [1]. Sin embargo, el método LSBI no aumenta la capacidad en comparación con el LSB1 estándar: ambos esconden solo 1 bit por byte. Incluso, LSBI agrega un pequeño *overhead* de 4 bytes, lo que en realidad reduce ligeramente la capacidad total. La frase citada da a entender que el objetivo es aumentar la capacidad, cuando en realidad la mejora propuesta apunta a mantener la misma capacidad pero con una mejor calidad visual.

En síntesis, el documento de Majeed y Sulaiman presenta una idea teóricamente valiosa pero una descripción insuficientemente detallada y ocasionalmente errada, lo que dificulta su implementación fiel y su validación experimental independiente.

## 2. Comparación de algoritmos de esteganografía

Para evaluar las características de los tres métodos implementados (LSB1, LSB4 y LSBI), se realizaron pruebas sobre un conjunto de imágenes ampliamente utilizadas en la literatura de esteganografía: *Lenna*, *Pepper* y *Airplane* ( $512 \times 512$  píxeles, 24 bits RGB). Se utilizaron dos tamaños de mensaje: 1 KB y 50 KB, con el objetivo de evaluar el comportamiento de los algoritmos tanto con cargas mínimas como con ocupación significativa de la capacidad disponible.

### 2.1. Métricas evaluadas

Siguiendo la clasificación de Majeed y Sulaiman [1], una técnica de esteganografía ideal debe optimizar tres aspectos principales: (1) la capacidad máxima de almacenamiento, (2) la imperceptibilidad de la modificación, y (3) la robustez frente a detección. Para evaluar estos aspectos, se midieron:

#### Capacidad teórica

Método	Bits por byte	Capacidad ( $512 \times 512$ RGB)	Overhead	Capacidad neta
LSB1	1	786,432 bytes	0	786,432 bytes
LSB4	4	1,572,864 bytes	0	1,572,864 bytes
LSBI	1	786,432 bytes	4 bytes	786,428 bytes

**Tabla 1** - Capacidad teórica de los métodos de esteganografía.

#### Imperceptibilidad

El PSNR (Peak Signal-to-Noise Ratio) mide qué tan similar es la imagen stego respecto a la original. Valores superiores a 40 dB indican que el cambio es imperceptible al ojo humano. El MSE (Mean Squared Error) cuantifica el error promedio entre píxeles de la imagen original y la stego, donde valores bajos indican menor distorsión. La diferencia de entropía de Shannon mide el cambio en la distribución estadística de colores de la imagen. Valores cercanos a cero indican que la imagen modificada mantiene características estadísticas similares a la original, dificultando su detección. El porcentaje de bits modificados indica qué proporción de la imagen fue alterada, permitiendo comparar la eficiencia de distribución espacial entre métodos.

#### Robustez

La robustez, entendida como la dificultad de un tercero no autorizado para detectar o extraer el mensaje oculto, no puede medirse directamente sin implementar técnicas de estegananálisis específicas. Sin embargo, métricas como la entropía y la

distribución espacial de cambios proporcionan indicios sobre la detectabilidad estadística del método.

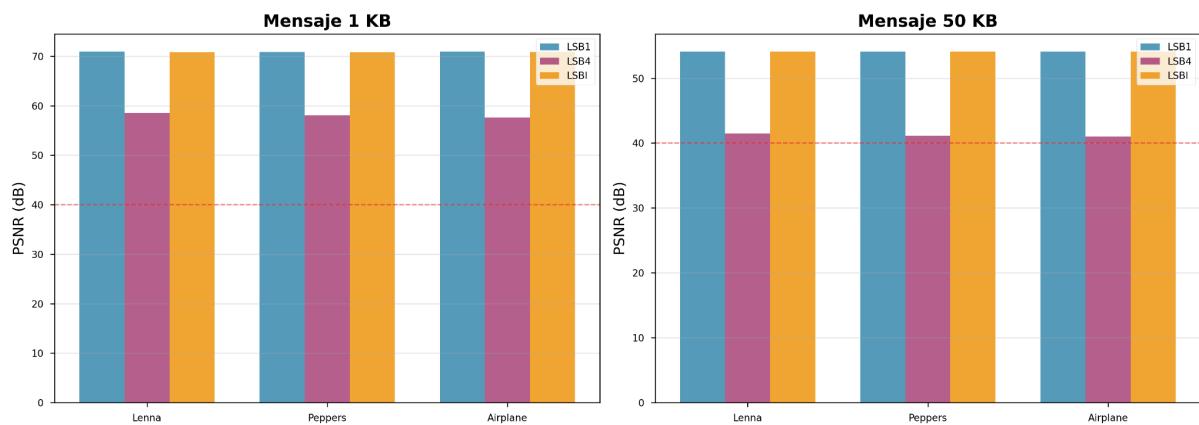
## 2.2. Resultados experimentales

Los resultados experimentales se resumen a continuación, analizando cada métrica en detalle, según la tabla completa de valores que encuentra en el [Apéndice](#) (Tablas A1 y A2).

## 2.3. Análisis de resultados

### 2.3.1. Calidad visual (PSNR y MSE)

La Figura 1 presenta la comparación de PSNR entre los tres métodos para ambos tamaños de mensajes evaluados (1 KB y 50 KB).

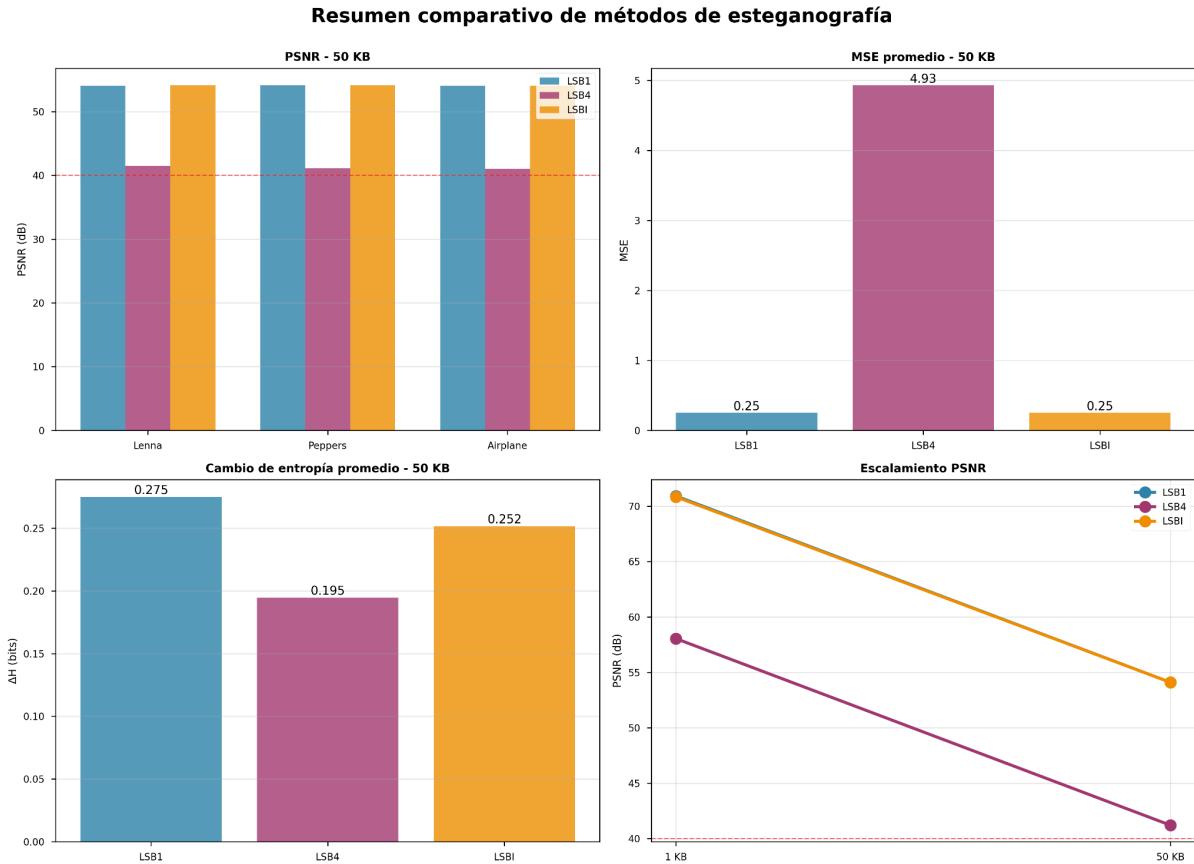


**Figura 1** - Comparación de PSNR entre métodos.

Con mensajes pequeños de 1 KB, los tres métodos mantienen excelente calidad visual. LSB1 lidera con valores superiores a 70 dB en todas las imágenes evaluadas, mientras que LSB4 presenta valores consistentemente menores, rondando los 58 dB. LSBI muestra un desempeño prácticamente idéntico a LSB1, confirmando que el *overhead* del mecanismo de inversión es despreciable con cargas pequeñas. La línea punteada en 40 dB marca el umbral de imperceptibilidad: todos los métodos permanecen muy por encima de este límite.

Con mensajes de 50 KB, las diferencias se acentúan significativamente. LSB1 y LSBI mantienen valores prácticamente idénticos alrededor de 54 dB, permaneciendo muy por encima del umbral de imperceptibilidad. En contraste, LSB4 llega al límite de 40 dB, el punto donde las modificaciones comienzan a ser potencialmente detectables por el ojo humano. Esta diferencia se explica por la estrategia de modificación: LSB4 altera cuatro bits por byte afectado, concentrando mayor distorsión en menos bytes, lo que resulta en cambios más abruptos en los valores de los píxeles.

El panel comparativo de la Figura 2 (abajo a la derecha) muestra el escalamiento del PSNR con el tamaño del mensaje. Las tres curvas descienden aproximadamente 17 dB al pasar de 1 KB a 50 KB, confirmando que la calidad disminuye proporcionalmente al tamaño del mensaje. Sin embargo, LSB4 parte de un valor inicial más bajo y termina peligrosamente cerca del umbral de perceptibilidad.

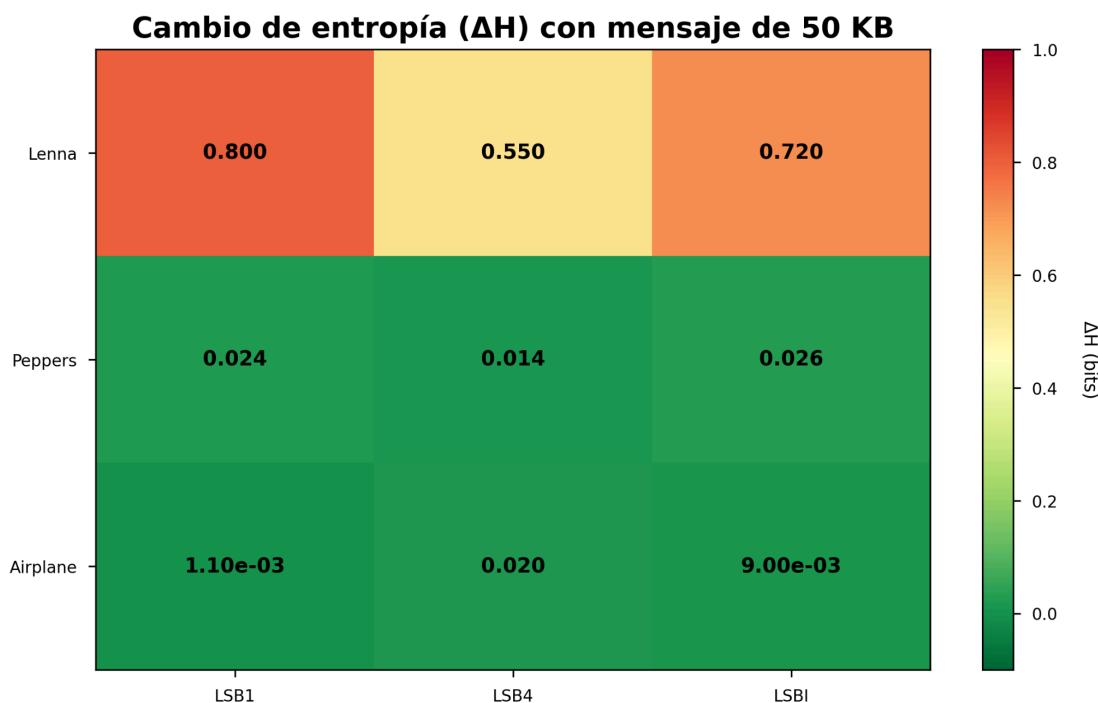


**Figura 2** – Resumen de métodos de esteganografía.

El panel superior derecho de la Figura 2 presenta los valores promedio de MSE para mensajes de 50 KB. Los resultados confirman la tendencia observada en PSNR: LSB1 y LSBI mantienen valores muy bajos e idénticos, mientras que LSB4 presenta un MSE aproximadamente veinte veces superior. Esta diferencia cuantifica la mayor distorsión introducida por modificar cuatro bits simultáneamente.

### 2.3.2. Naturalidad estadística (entropía)

La Figura 3 presenta un mapa de calor del cambio de entropía para mensajes de 50 KB, donde los colores más intensos indican mayores alteraciones estadísticas.



**Figura 3** – Mapa de calor del cambio de entropía.

El análisis de entropía revela que el impacto del *embedding* depende fuertemente de las características de la imagen portadora. *Lenna*, que tiene grandes áreas de color uniforme y tonos suaves, muestra cambios de entropía significativos con los tres métodos. Esta imagen es particularmente vulnerable a la detección estadística porque cualquier introducción de información aleatoria cambia notablemente su distribución de colores naturalmente uniforme.

En contraste, *Pepper*, con mayor variedad de colores y texturas complejas, mantiene cambios de entropía mínimos incluso con mensajes grandes. Estas imágenes naturalmente contienen más "desorden" o información, por lo que la adición del mensaje oculto queda estadísticamente enmascarada dentro de su complejidad natural. Este comportamiento confirma el principio teórico de que las imágenes con alta entropía inicial proporcionan mejor camuflaje estadístico.

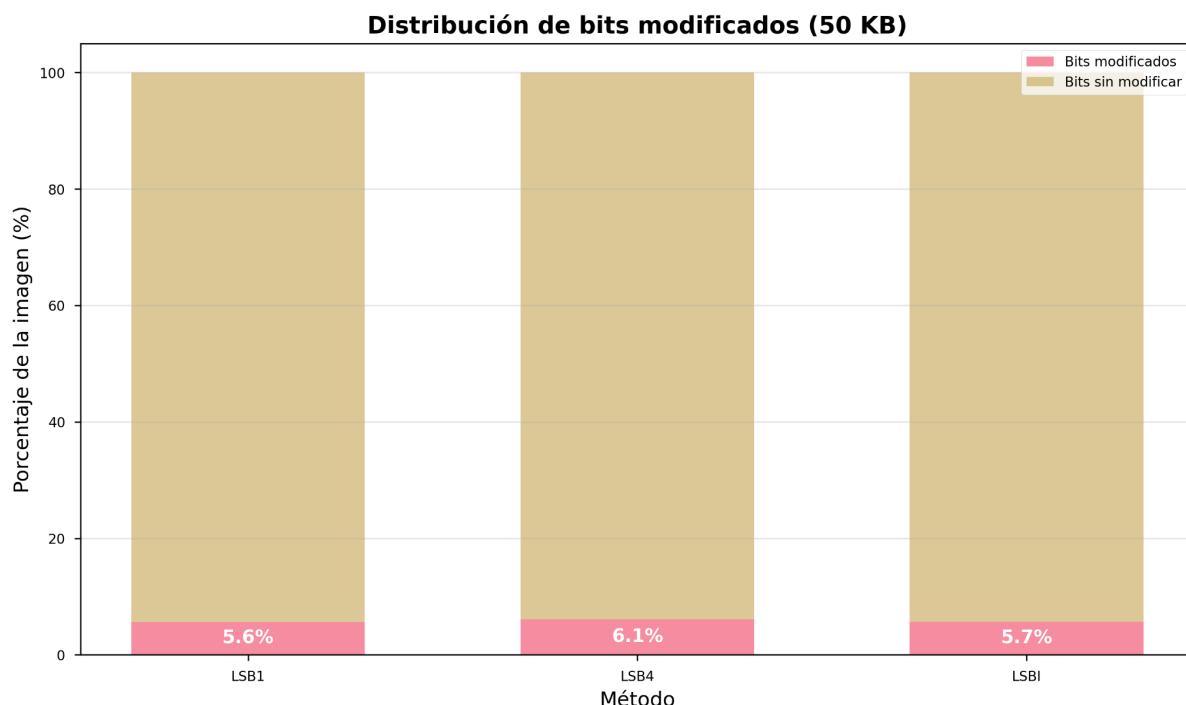
La comparación entre columnas (métodos) muestra que LSBI y LSB1 producen alteraciones de entropía muy similares en todos los casos evaluados. LSB4 tiende a producir cambios ligeramente menores en promedio, lo que podría parecer contradictorio dado su mayor impacto visual, pero se explica porque modifica menos bytes totales: al alterar cuatro bits simultáneamente por byte, afecta una cuarta parte de los bytes que LSB1, lo que puede resultar en menor dispersión estadística global.

El panel inferior izquierdo de la Figura 2 resume estos resultados mostrando el cambio de entropía promedio entre las tres imágenes. LSBI y LSB1 presentan valores

muy cercanos, mientras que LSB4 muestra el valor más bajo, confirmando su menor impacto estadístico global a costa de mayor concentración local.

### 2.3.3. Distribución espacial de modificaciones

La Figura 4 ilustra el porcentaje de bits modificados para mensajes de 50 KB.



**Figura 4** - Porcentaje de bits modificados para mensajes de 50 KB.

Los tres métodos modifican porcentajes similares de bits totales: LSB1 modifica aproximadamente 5.6% de los bits de la imagen, LSB4 alrededor de 6.1%, y LSBI cerca de 5.7%. Sin embargo, la diferencia fundamental está en cómo distribuyen estos cambios. LSB1 y LSBI esparcen modificaciones de 1 bit en muchos bytes, alterando un bit por vez, lo que produce un "ruido" uniforme y fino distribuido por toda la región afectada. LSB4 concentra modificaciones de 4 bits en menos bytes, alterando cuatro bits simultáneamente. Esta concentración explica por qué, a pesar de modificar porcentajes similares de bits totales, LSB4 genera mayor distorsión visual: los cambios son más abruptos y concentrados espacialmente.

La barra de color claro representa los bits que no requieren modificación. Aproximadamente el 94-95% de los bits de la imagen permanecen inalterados, lo que contribuye a la imperceptibilidad de los tres métodos.

#### **2.3.4. Síntesis comparativa**

LSBI demuestra desempeño consistente a través de todos los tamaños de mensaje, igualando prácticamente a LSB1 en PSNR. El mecanismo de inversión selectiva de patrones cumple su objetivo teórico de minimizar modificaciones redundantes: cuando la mayoría de los bits de un patrón específico ya requieren cambios, LSBI invierte todos los bits de ese patrón, reduciendo el número total de modificaciones necesarias. Este comportamiento es particularmente efectivo con mensajes grandes, donde la probabilidad de encontrar patrones que se beneficien de la inversión aumenta.

LSB4 ofrece el doble de capacidad teórica, pero a costa de una calidad visual significativamente menor. Con mensajes grandes se aproxima al límite de perceptibilidad, haciéndolo adecuado solo cuando maximizar el espacio de almacenamiento es absolutamente prioritario sobre la imperceptibilidad. Su ventaja radica en situaciones donde el tamaño del mensaje excede la capacidad de LSB1/LSBI, pero debe usarse con cautela debido a su mayor detectabilidad.

LSB1 mantiene la calidad más consistente y predecible en todo el rango evaluado. Su simplicidad algorítmica y comportamiento uniforme lo hacen la opción más equilibrada cuando no se requiere la capacidad adicional de LSB4. Es el estándar de facto en la literatura de esteganografía por su balance óptimo entre simplicidad de implementación, calidad visual y capacidad razonable.

### **2.4. Análisis visual de diferencias**

Además de las métricas cuantitativas, se realizó un análisis visual comparativo sobre la imagen Pepper con mensaje de 50 KB. Se calculó la diferencia absoluta píxel a píxel entre la imagen cover y su versión stego, amplificando el resultado mediante un factor de escala ( $\times 50$ ) para resaltar visualmente las alteraciones que normalmente serían imperceptibles al ojo humano.

Método	Mapa de diferencia amplificado ( $\times 50$ )	Imagen stego
LSB1		
El mapa de diferencias muestra modificaciones que ocupan aproximadamente la mitad inferior de la imagen. Este patrón refleja directamente la relación entre el tamaño del mensaje (50 KB y la capacidad de <i>embedding</i> . Como LSB1 embebe 1 bit por byte de imagen y los archivos BMP se leen de abajo hacia arriba, las modificaciones comienzan en la parte inferior y avanzan hacia arriba hasta agotar el mensaje. El ruido visible es fino, multicolor y uniformemente distribuido en la región afectada, característico de modificaciones de un solo bit que alteran mínimamente los valores originales.		
LSB4		
El mapa de diferencias muestra un patrón concentrado predominantemente en la parte inferior de la imagen. Al igual que LSB1, el embedding comienza desde abajo siguiendo el orden de lectura del BMP. Sin embargo, LSB4 embebe 4 bits por byte, por lo que requiere cuatro veces menos bytes para ocultar el mismo mensaje. Esto explica por qué la región afectada es notablemente más pequeña que en LSB1. El patrón visible presenta mayor intensidad y variabilidad de color, reflejando que las modificaciones de 4 bits simultáneos producen cambios más drásticos en los valores de los píxeles. Esta concentración espacial de		

modificaciones más intensas resulta en los valores de PSNR más bajos observados y potencialmente mayor detectabilidad por análisis estadísticos localizados.

LSBI	
------	------------------------------------------------------------------------------------

El mapa de diferencias de LSBI presenta características distintivas: se extiende más allá de la mitad de la imagen (más que LSB1), y exhibe una tonalidad predominantemente cyan-verdosa en lugar del multicolor observado en LSB1. La mayor extensión vertical se explica por el *overhead* del método: LSBI requiere 4 bytes adicionales al inicio de la imagen para almacenar los *flags* de inversión de patrones, desplazando el inicio efectivo del mensaje. Además, el mecanismo de inversión selectiva implica que algunos bytes se modifican para aplicar la inversión de patrón incluso si no contienen datos del mensaje directamente. La coloración distintiva sugiere que las modificaciones afectan de manera diferencial los canales verde y azul (recordemos que LSBI no usa el canal rojo). Este patrón visual confirma que LSBI, si bien logra valores de PSNR similares a LSB1, distribuye las modificaciones de manera cualitativamente diferente debido a su mecanismo de inversión de patrones basado en los bits segundo y tercero menos significativos.

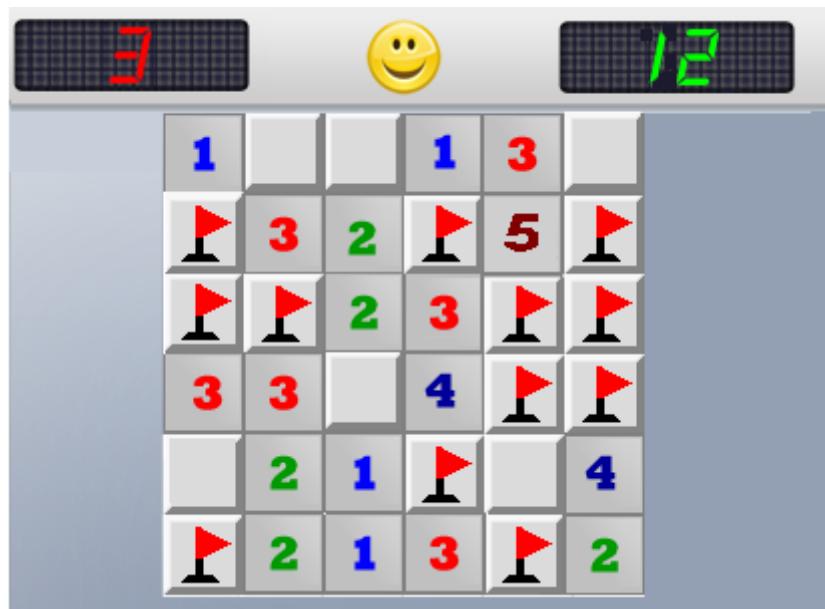
**Tabla 2** - Comparación visual de métodos de esteganografía sobre Pepper (50 KB).

Esta visualización confirma cuantitativamente lo observado en las métricas de PSNR y MSE. LSBI logra la mejor relación entre imperceptibilidad y preservación estadística, seguido de cerca por LSB1, mientras que LSB4 prioriza capacidad a costa de mayor distorsión visual y mayor detectabilidad potencial en análisis forenses.

### 3. Estegoanálisis

El estegoanálisis se llevó a cabo sobre cuatro archivos BMP provistos por la cátedra: **madridoso.bmp**, **montevideo.bmp**, **medianochе1.bmp** y **titanic1.bmp**. La estrategia general consistió en aplicar de forma sistemática los tres métodos implementados –LSB1, LSB4 y LBSI– sin encriptación sobre cada archivo para identificar sus contenidos ocultos.

El archivo **titanic1.bmp** reveló, mediante el método LSB4, una imagen PNG correspondiente a un tablero de Buscaminas (Figura 5).



Por su parte, el archivo **medianochе1.bmp** contenía un documento PDF extraído con LSB1, con la siguiente instrucción: “al .png cambiarle la extensión por .zip y descomprimir”. Siguiendo esta indicación sobre el archivo **buscaminas.png**, al descomprimirlo se obtuvo el archivo **sol8.txt** con las siguientes instrucciones:

“cada mina es un 1.  
cada fila forma una letra.  
Los ascii de las letras empiezan todos en 01.  
Así encontrarás el algoritmo que tiene clave de 128 bits y el modo  
La password está en otro archivo  
Con algoritmo, modo y password hay un .wmv encriptado y oculto.”

#### Extracto 1 - Contenido del archivo sol8.txt

Para obtener la contraseña mencionada, se analizó el archivo **montevideo.bmp**, el cual utilizaba un método de ocultamiento distinto a los dos anteriores: una simple concatenación de datos al final del archivo BMP sin modificar la imagen. Mediante el

comando `strings` se identificó el texto “*h4la password es camuflado*” revelando que la contraseña era “camuflado”. Este método, aunque simple de implementar, presenta una debilidad evidente: el mensaje secreto es fácilmente detectable mediante herramientas básicas de análisis textual.

Para descifrar el algoritmo y el modo de encriptación, fue necesario resolver correctamente el juego de Buscaminas. Las posiciones finales de las minas obtenidas se muestran en la Tabla 3.

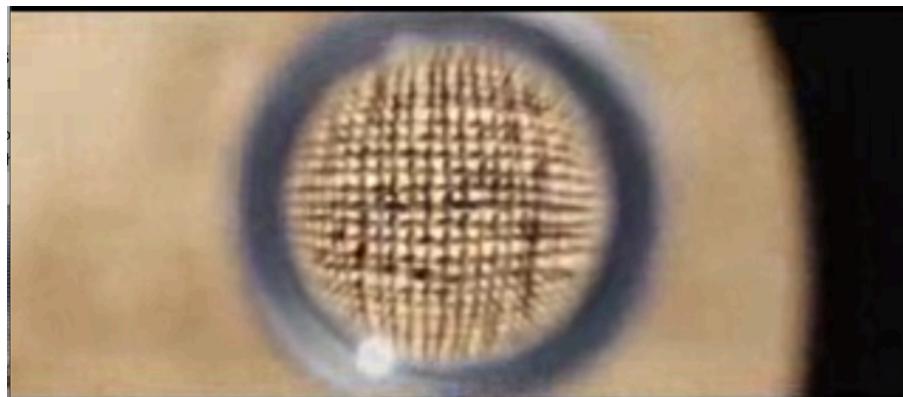
1	1	1	1	3	F
F	3	2	F	5	F
F	F	2	3	F	F
3	3	2	4	F	F
F	2	1	F	F	4
F	2	1	3	F	2

**Tabla 3** - Tablero de buscaminas resuelto (F = mina)

Interpretando cada fila según las instrucciones del archivo `sol8.txt` –donde las mina representan un 1 y el resto de las celdas un 0, anteponiendo el prefijo “01”– se obtuvieron los siguientes caracteres ASCII: “AesCfb”.

Hasta este punto, puede observarse cómo **titanic1.bmp** integraba tres niveles de ocultamiento: primero, una imagen PNG del tablero de Buscaminas, oculta con LSB4; segundo, un archivo ZIP contenido dentro del PNG, aprovechando que ambos formatos pueden coexistir en el mismo archivo; y tercero, información codificada en el propio tablero que, una vez resuelto, revelaba el algoritmo y el modo de encriptación.

Finalmente, con todos los elementos reunidos –algoritmo AES-128, modo CFB, contraseña “camuflado”– y utilizando el método LSB1, se procedió a extraer el contenido de **madridoso.bmp**. El resultado fue una escena de la película Wanted (2008) (Figura 6).



**Figura 6** - Escena de la película “Wanted” (2008)

En esta escena, el personaje Sloan (interpretado por Morgan Freeman) explica una técnica de esteganografía antigua: un tejido de hilos en el que la posición de cada hilo vertical sobre o bajo el horizontal representa un bit (1 o 0). Esta idea se vincula directamente con los métodos de esteganografía utilizados en este trabajo. Así como en el tejido, las “imperfecciones” ocultan un mensaje entre hilos que a simple vista parecen normales, en los métodos LSB cada bit menos significativo de la imagen transporta información invisible al ojo humano, revelando el mensaje solo a quienes conocen el método de ocultamiento.

## 4. Análisis de LSBI

### 4.1. Comparación con LSB1

La propuesta del documento de Majeed y Sulaiman representa una mejora respecto al método LSB estándar porque, en teoría, reduce el número total de modificaciones necesarias para ocultar información. Sin embargo, esta mejora no se cumple de forma uniforme en todas las imágenes.

En imágenes con patrones de bits que favorecen la inversión (por ejemplo, aquellas con regiones de color uniforme o correlación alta entre píxeles vecinos), LSBI efectivamente disminuye la cantidad de modificaciones y mejora la calidad visual. En cambio, en imágenes donde los patrones naturales ya se oponen al criterio de inversión, el algoritmo puede incluso generar más cambios que el LSB tradicional, como se observó en algunos de nuestros resultados experimentales.

Esto demuestra que la eficacia de LSBI depende fuertemente de la estructura estadística de la imagen portadora, lo que constituye tanto una ventaja (por su adaptabilidad) como una limitación (por su comportamiento no determinista en todos los casos).

### 4.2. Almacenamiento de patrones

En nuestra implementación, se decidió guardar los 4 bits de *flags* de inversión al inicio de la imagen, utilizando los primeros 4 bytes con el método LSB1 estándar (sin aplicar inversión).

Esta decisión simplifica la lectura: los *flags* se extraen primero, permitiendo configurar correctamente la desinversión antes de procesar el resto del mensaje. Además, separa de forma lógica el encabezado (*flags*) del contenido oculto. Sin embargo, esta elección introduce un pequeño *overhead* visible y un punto fijo predecible que podría ser explotado por un atacante.

Otras alternativas posibles incluyen:

- Guardar los *flags* al final del mensaje, lo que requeriría primero extraer todo el mensaje y luego aplicar las desinversiones retroactivamente.
- Distribuir los *flags* de forma pseudoaleatoria en distintas posiciones de la imagen, según una clave compartida entre emisor y receptor. Esta opción mejora la seguridad, pero aumenta la complejidad y requiere sincronización entre ambas partes.

### 4.3. Dificultades de implementación

La implementación del algoritmo LSBI presentó varios desafíos, tanto por ambigüedades del paper original como por la complejidad inherente del método.

Uno de los principales problemas fue el manejo del canal rojo, que el algoritmo indica omitir durante el proceso de *embedding* y extracción. En una primera versión, se implementó un recorrido secuencial sobre todos los bytes de la imagen, lo que produjo un desalineamiento del mensaje, y en consecuencia, la generación de datos corruptos al decodificar.

La solución consistió en implementar funciones auxiliares específicas -`lsb1_embed_no_red()` y `lsb1_extract_no_red()`- que encapsulan la lógica de salteo del canal rojo y mantienen la alineación correcta entre bytes de mensaje e imagen.

Otro desafío fue interpretar correctamente el mecanismo de inversión de patrones, dado que el paper no especifica con claridad cómo deben agruparse los bits ni en qué momento aplicar la inversión. Fue necesario realizar pruebas empíricas para confirmar que los patrones debían calcularse sobre la imagen original antes del *embedding*.

## 5. Mejoras o futuras extensiones

Si bien los métodos implementados cumplen su función de ocultar información en BMP de forma efectiva, pueden explorarse mejoras puntuales. En primer lugar, optimizar LSBI con decisiones locales por bloques –basadas en métricas como entropía o varianza– permitiría reducir modificaciones innecesarias y elevar el PSNR. Además, incorporar un módulo de estegoanálisis automático (histograma, varianza local, entropía, etc.) brindaría una evaluación objetiva de la detectabilidad. Por último, una interfaz gráfica facilitaría la comparación interactiva de LSB1, LSB4 y LSBI, mostrando en tiempo real mapas de diferencia y métricas (PSNR/MSE/entropía).

## 6. Bibliografía

- [1] M. A. Majeed and R. Sulaiman, "An Improved LSB Image Steganography Technique using bit-inverse in 24 bit colour image," Journal of Theoretical and Applied Information Technology, vol. 80, no. 2, pp. 342-348, Oct. 2015.

## 7. Apéndice

Imagen	Método	PSNR (dB)	MSE	$\Delta H$ (bits)	% Bits mod.
Lenna	LSB1	70.95	0.0052	0.046	0.065
Lenna	LSB4	58.50	0.092	0.023	0.071
Lenna	LSBI	70.83	0.053	0.047	0.054
Pepper	LSB1	70.87	0.0053	4.39e-05	0.067
Pepper	LSB4	58.02	0.10	0.00017	0.067
Pepper	LSBI	70.83	0.0053	7.42e-05	0.054
Airplane	LSB1	70.92	0.0053	0.00011	0.066
Airplane	LSB4	57.62	0.11	0.00043	0.065
Airplane	LSBI	70.85	0.0053	8.30e-05	0.067

Tabla A1 - Resultados experimentales con mensaje de 1 KB.

Imagen	Método	PSNR (dB)	MSE	$\Delta H$ (bits)	% Bits mod.
Lenna	LSB1	54.09	0.25	0.80	3.16
Lenna	LSB4	41.48	4.62	0.55	3.39
Lenna	LSBI	54.11	0.25	0.72	3.16
Pepper	LSB1	54.10	0.25	0.024	3.16
Pepper	LSB4	41.15	4.99	0.014	3.17
Pepper	LSBI	54.11	0.25	0.026	3.16
Airplane	LSB1	54.09	0.25	0.0011	3.17
Airplane	LSB4	40.99	5.18	0.020	3.17
Airplane	LSBI	54.07	0.25	0.009	3.18

Tabla A2 - Resultados experimentales con mensaje de 50 KB.