



{ Algoritmo de análisis de archivos cifrados ocultos en imágenes }

Análisis de Algoritmos

Prof.:Jorge Ernesto Lopez Arce Delgado

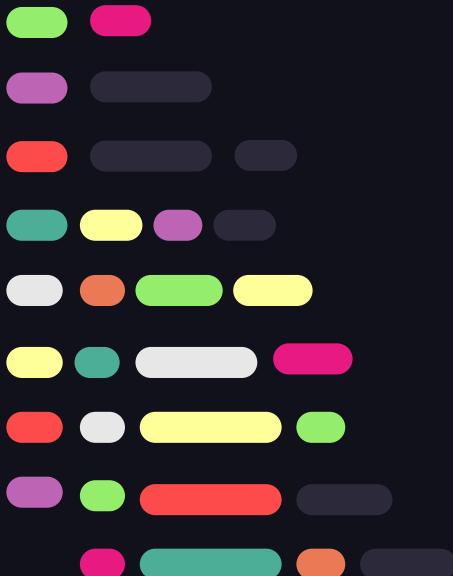
Arroyo Moreno Elizabeth 221453749

Hernández Elizarrarás Karla Rebeca 223991977

Valencia Ignacio Jennifer Patricia 223991721



Algoritmo



Este algoritmo mete información en los bits menos importantes de las pixeles de una imagen y luego revisa si hay mensajes escondidos usando pruebas estadísticas en el canal rojo

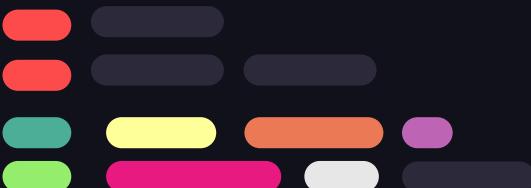
Permite esconder mensajes en imágenes sin que se note y también detectar cambios o datos ocultos de manera rápida. Se puede usar en seguridad digital y en análisis forense de imágenes.

Explicación del código



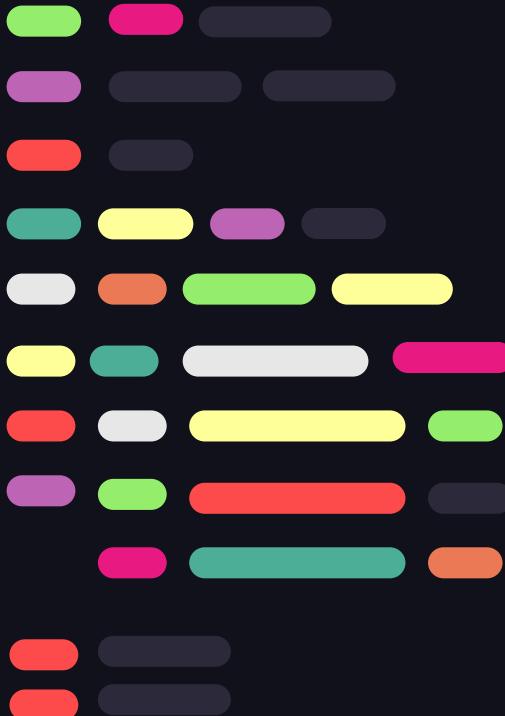
{

El código tiene dos clases principales: EsteganografiaLSB y LSBDetectos. Ademas, incluye un menú interactivo que permite crear imágenes nuevas, esconder mensajes en imágenes existentes y analizarlos fácilmente



}

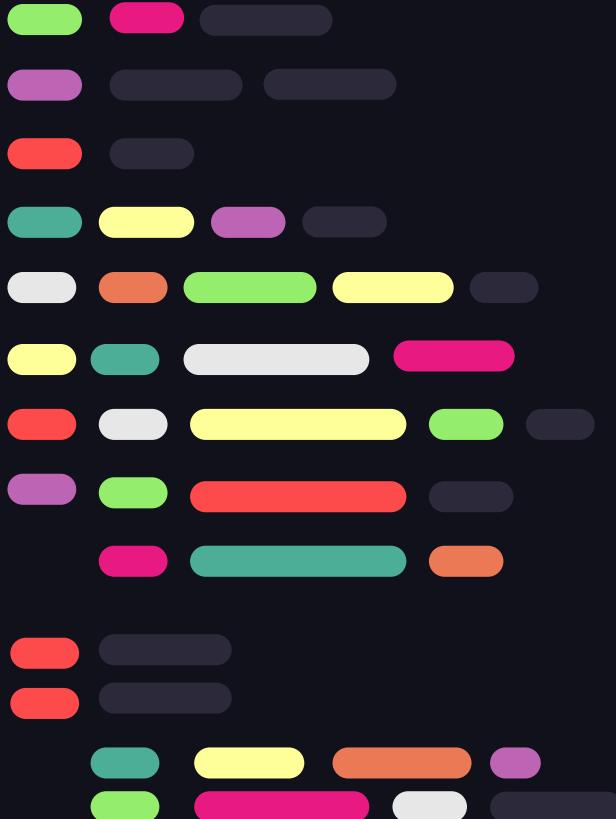
Bit menos significativo (LSB)



Es una técnica que agarra el último bit de cada pixel y lo cambia para esconder información. Funciona en imágenes con compresión sin pérdida, como PNG, JPG, TIFF, BMP.



Ocultando la información en los bits menos significativos.



Esteganografía



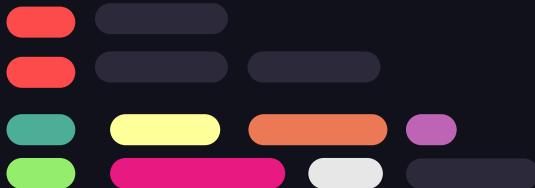
< Es una técnica que consiste en ocultar información dentro de otro archivo digital imagen, audio, video, texto, de forma que el mensaje secreto pase desapercibido. >





01 { ..

Clase LSBDetector

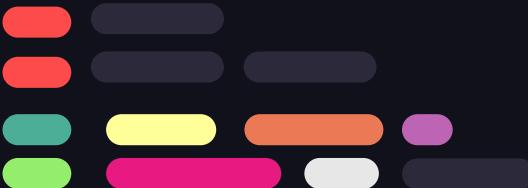


}

..

Clase LSBDetector

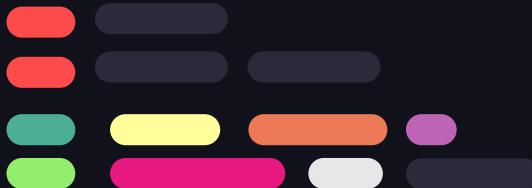
- `load_image()`: carga la imagen y la convierte a RGB.
- `extraer_mensaje_lsb()`: extrae un mensaje oculto en el canal rojo usando los bits menos significativos.
- `chi_square_test()`: prueba estadística para detectar anomalías en los valores de píxel.
- `lsb_analysis()`: calcula media, varianza, entropía y rachas de los bits LSB.
- `spatial_correlation_analysis()`: correlación horizontal y vertical entre píxeles vecinos.
- `calculate_suspicion_score()`: combina pruebas estadísticas para generar una puntuación de sospecha (0-1).
- `analizar_imagen_completo()`: ejecuta todos los análisis y muestra resultados.





02 { ..

Clase
EsteganografiaLSB

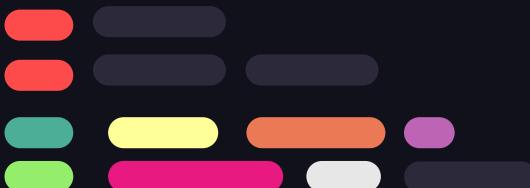


}

..

Clase EsteganografiaLSB

- `crear_imagen_con_mensaje()`: genera una imagen nueva y oculta un mensaje en LSB del canal rojo.
- `ocultar_mensaje_en_imagen_existente()`: oculta un mensaje en una imagen existente.



{}

..



Menú interactivo

Permite al usuario elegir entre analizar, crear o modificar imágenes con mensajes ocultos.

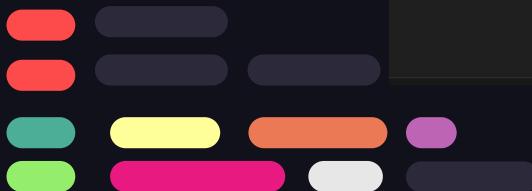
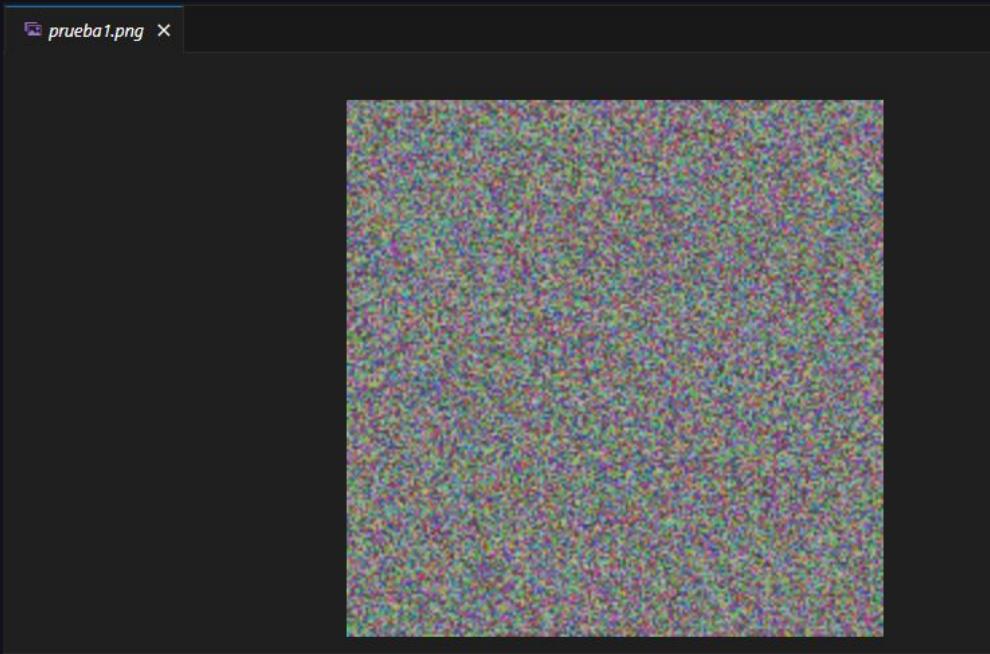
```
=====
🔍 DETECTOR DE MSJS OCULTOS (LSB) 🔎
=====
1. Analizar imagen
2. Crear imagen con esteganografía (nueva)
3. Ocultar mensaje en imagen existente
4. Salir
=====
Selecciona una opción:
```



Creación de imagen con esteganografía



{



}



Ocultar mensaje en imagen

{

```
=====
  DETECTOR DE MSJS OCULTOS (LSB)
=====

1. Analizar imagen
2. Crear imagen con esteganografía (nueva)
3. Ocultar mensaje en imagen existente
4. Salir
=====

Selecciona una opción: 3

  Imágenes encontradas:
1. perrito.jpg
2. prueba2.png
3. hola.png
4. nise.png

Selecciona una imagen (1-4): 3

Ingresa el mensaje que desea ocultar: mensaje secreto
  Nombre del archivo de salida (default: imagen_estego.png): alo
Mensaje oculto en: alo.png

Presiona Enter para continuar...
```

}

The screenshot shows a terminal window with the following content:

```
detector_lsb.py x shh.png perrito.jpg
detector_lsb.py > LSBDetector > analizar_imagen_completo
279 def seleccionar_imagen():
289     print("\n■ Imágenes encontradas:")
290     for i, img in enumerate(imagenes, 1):
291         print(f" {i}: {img}")
292     try:
293         opcion = int(input("\nSelecciona una imagen (1-{len(imagenes)}): "))
294         if 1 <= opcion <= len(imagenes):
295             return imagenes[opcion - 1]
296         else:
297             print("Opción no válida")
298             return None
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

Imágenes encontradas:

1. perrito.jpg
2. perrito_impostor.png
3. prueba1.png
4. prueba2.png
5. shh.png

Selecciona una imagen (1-5): 3

ANÁLISIS COMPLETO DE: prueba1.png

Imagen cargada: (200, 200, 3)

EXTRACCIÓN DE MENSAJE:
MENSAJE ENCONTRADO: 'HOLI'

ANÁLISIS (Canal Rojo):

RESULTADO DEL ANÁLISIS:
Puntuación de sospecha: 0.40/1.0
Estado: ESTEGANOGRAFÍA DETECTADA
Mensaje oculto: 'HOLI'

Presiona Enter para continuar... █

* Resultados de Análisis

Aplicación en la vida real



En el área de ciberseguridad, comunicaciones puede descifrar mensajes confidenciales ocultos en las imágenes como periodistas o activistas enviando información de manera segura.

En el área forense puede detectar la manipulación de imágenes digitales como encontrar si hay información alterada.

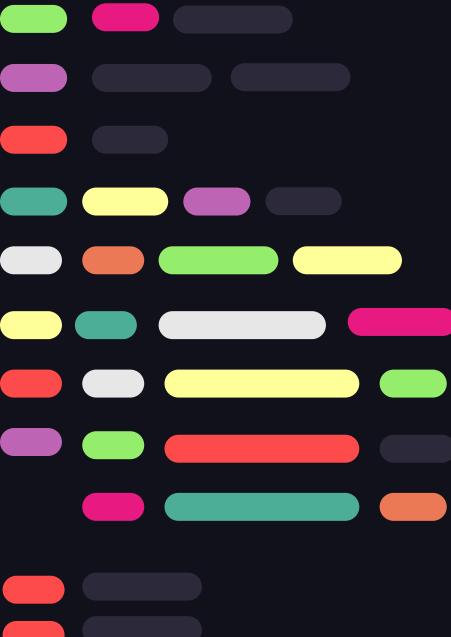




Complejidad computacional

$O(n \cdot m)$ en tiempo y espacio, ya que analiza cada píxel de la imagen. Es rápido y simple, pero menos seguro que métodos en dominios de frecuencia o con enmascaramiento aleatorio, que son más resistentes pero más lentos.

Conclusión



El algoritmo LSB es útil para detectar mensajes ocultos en imágenes pequeñas o medianas y en formatos sin compresión, siendo eficiente para mensajes cortos en el canal rojo. No es recomendable para imágenes muy grandes, mensajes extensos o formatos comprimidos, ya que puede generar errores o pérdida de información. Se podría mejorar analizando los tres canales, manejando mejor tipos de datos grandes y soportando mensajes más largos o imágenes comprimidas.

}

..



Gracias por su
atención!

< Esperamos que les haya gustado >

