

UNIVERSIDAD DE GUADALAJARA
Centro Universitario de Ciencias Exactas e Ingenierías
División de Tecnologías para la Integración Ciber-Humana



Análisis de algoritmos

Jennifer Patricia Valencia Ignacio, Código: 223991721

Elizabeth Arroyo Moreno, Código: 221453749

Karla Rebeca Hernández Elizarrarás, Código: 223991977

Ingeniería en computación

Reporte Divide y Vencerás

26 de Octubre de 2025

Introducción

La esteganografía digital, es una técnica que se usa para esconder información dentro de imágenes, sin que nadie lo note a simple vista. Una de las formas en que podemos hacer esto es con el método de LSB (Bit menos significativo) que cambia los bits menos importantes de cada pixel.

El problema de revisar cada píxel uno por uno puede tomar demasiado tiempo. Por eso, se usa una versión del método Divide y Vencerás, que separa la imagen en partes más pequeñas y analiza solo las que parecen sospechosas.

Así, se puede ahorrar mucho tiempo y encontrar con más precisión las partes donde podría estar escondida la información.

Es muy útil en seguridad digital y análisis forense, donde se busca detectar si una imagen fue alterada o si tiene información oculta.

Objetivo

Aplicamos el enfoque de divide y vencerás para que mejore la detección de información oculta en imágenes usando la técnica de esteganografía LSB.

El propósito es desarrollar un algoritmo que permita dividir las imágenes en subregiones para facilitar el análisis. También se analizará qué tan complicado es el método y lo comparamos con el método de fuerza bruta que hemos realizado anteriormente.

Desarrollo

El enfoque divide y vencerás divide un problema grande en partes más pequeñas. En este caso, la imagen se separa en subregiones, si una zona tiene datos sospechosos, se revisa con más detalle. Pero si una zona parece normal, se dejará de revisar. Así, el programa no pierde tiempo analizando todo, sino que se enfoca más en lo que realmente importa. No solo hace el procesamiento más rápido, sino que también permite que se pueda ejecutar en paralelo.

El programa recorre cada píxel de la imagen previamente cargada, o la que genera, extrayendo su bit menos significativo. Usa un acceso secuencial a la memoria que

es fácil de implementar y entender. Después, con divide y vencerás, se aplican pruebas estadísticas como chi-cuadrado o entropía a cada subregión para ver si hay cambios sospechosos.

Para que el algoritmo no pierda tiempo revisando zonas que están limpias. Se definieron reglas simples para decidir si una subregión es sospechosa o no. Por ejemplo, si la variación de los bits menos significativos es muy baja, se considera que la zona está limpia y se descarta. En cambio, si la variación supera cierto umbral, se marca como sospechosa y se vuelve a dividir para analizarla con mas detalle.

El umbral sólo se puede ajustar según el tipo de imagen: en imágenes con mucho ruido se necesita un umbral más alto, mientras que en imágenes más “planas” se puede usar uno más bajo. Esto permite que el análisis se enfoque solo en las zonas donde realmente hay más posibilidades de encontrar información oculta.

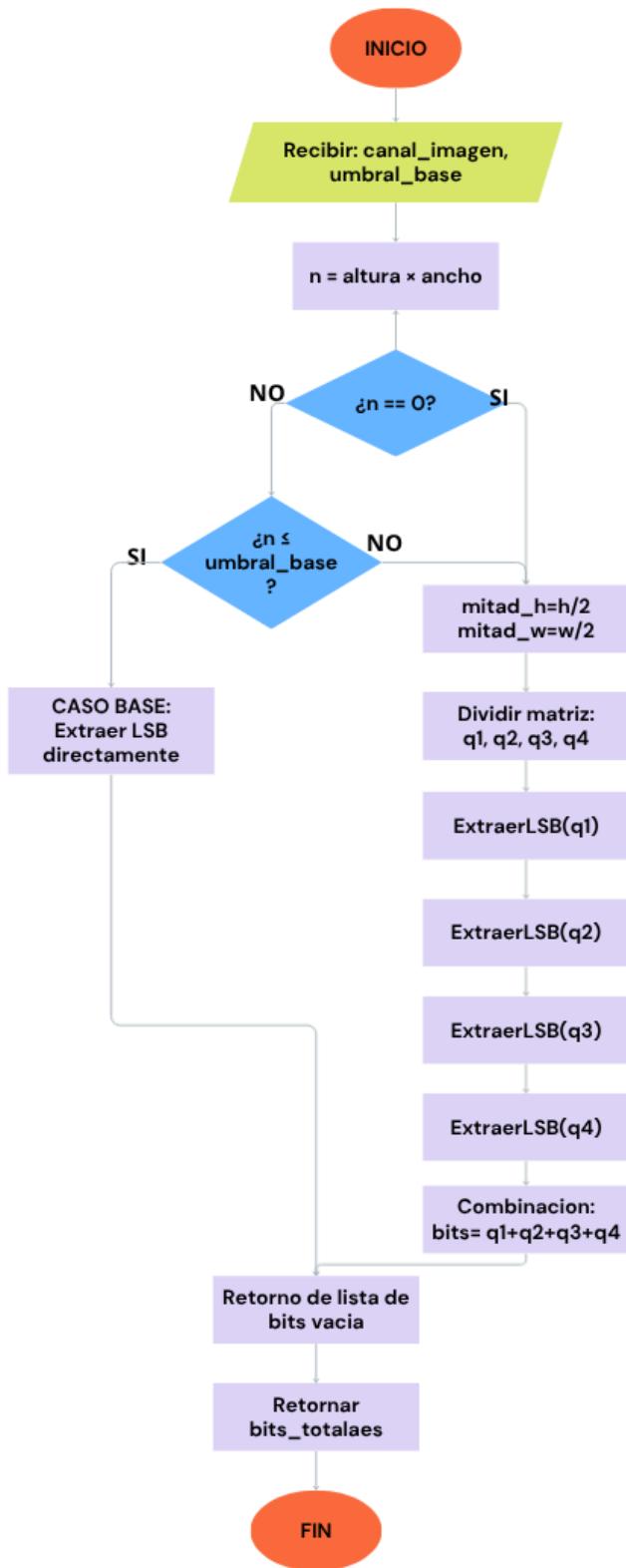
Pseudocódigo

```
ALGORITMO ExtraerLSB_FuerzaBruta(canal_imagen):
    bits_extraidos ← lista_vacia

    PARA cada fila i en altura_imagen:
        PARA cada columna j en ancho_imagen:
            pixel ← canal_imagen[i][j]
            lsb ← pixel AND 1
            bits_extraidos.agregar(lsb)
        FIN PARA
    FIN PARA

    RETORNAR bits_extraidos
FIN ALGORITMO
```

El algoritmo implementado de Divide y Vencerás divide la imagen en 4 secciones alcanzando un tamaño base, extrayendo los bits menos significativos de cada bloque combinando los resultados De tal manera la división de la imagen hace de un problema grande a problemas más pequeños lo que hace que se pueda ejecutar el programa en paralelo, creando diferentes núcleos en el procesador, esto reduce el tiempo total de ejecución del programa.



```

ALGORITMO ExtraerLSB_DyB(canal_imagen, umbral_base):
    alto, ancho ← dimensiones(canal_imagen)

```

```

n ← alto × ancho

// CASO BASE
SI n == 0:
    RETORNAR lista_vacia
FIN SI

SI n <= umbral_base:
    // Extraer directamente (fuerza bruta en bloque pequeño)
    bits ← lista_vacia
    PARA cada pixel en canal_imagen:
        bits.agregar(pixel AND 1)
    FIN PARA
    RETORNAR bits
FIN SI

// DIVIDIR
mitad_alto ← alto / 2
mitad_ancho ← ancho / 2

cuadrante1 ← canal_imagen[0:mitad_alto, 0:mitad_ancho]
cuadrante2 ← canal_imagen[0:mitad_alto, mitad_ancho:ancho]
cuadrante3 ← canal_imagen[mitad_alto:alto, 0:mitad_ancho]
cuadrante4 ← canal_imagen[mitad_alto:alto, mitad_ancho:ancho]

// CONQUISTAR (recursión)
bits1 ← ExtraerLSB_DyV(cuadrante1, umbral_base)
bits2 ← ExtraerLSB_DyV(cuadrante2, umbral_base)
bits3 ← ExtraerLSB_DyV(cuadrante3, umbral_base)
bits4 ← ExtraerLSB_DyV(cuadrante4, umbral_base)

// COMBINAR
bits_totales ← concatenar(bits1, bits2, bits3, bits4)

RETORNAR bits_totales
FIN ALGORITMO

```

En comparación con el método de fuerza bruta, divide y vencerás tiene una gran ventaja, En fuerza bruta, el análisis es completamente lineal ($O(n)$) y revisa cada píxel lo que puede tardar mucho en imágenes grandes.

Con divide y vencerás, la complejidad se reduce a $(O(n^*a))$, donde a es menor que 1, porque muchas regiones se descartan antes de ser analizadas por completo. Esto hace que el algoritmo sea más rápido, especialmente cuando las imágenes solo tienen pocas áreas alteradas.

Además, el proceso se puede parallelizar fácilmente, puede correr aun mas rapido.

Conclusión

Divide y vencerás nos permite detectar más rápido y localizar la presencia de datos ocultos en una imagen. Gracias al divide y vencerás, se evita hacer trabajo innecesario y se optimiza el tiempo de análisis sin perder precisión.

Esto muestra como una técnica de algoritmos puede aplicarse a un problema moderno de seguridad digital y obtener resultados eficientes y fáciles de escalar.