

**Autores:**

Diego Camarena Gutiérrez

Juan Pablo Hernández Ponce de León

Iván Vivas García

Jared Yahir Hernández Delgado

**Encriptación de imágenes**

Modelado y procesamiento de imágenes  
Tercer parcial  
Proyecto final  
512

Contenido

[Introducción 3](#_Toc183383484)

[Objetivo principal 3](#_Toc183383485)

[Objetivos secundarios 3](#_Toc183383486)

# Introducción

Este proyecto consiste en un programa desarrollado en Python con la funcionalidad de encriptar una imagen dentro de otra, es decir, encriptar una imagen oculta dentro de una imagen “señuelo”, permitiendo almacenar la información de una imagen dentro de otra, modificando los bits menos significativos con los mas significativos de la imagen oculta.

# Objetivo principal

El objetivo de este proyecto es generar una solución que permita ocultar y mostrar imágenes en función de los bits utilizados.

# Objetivos secundarios

1. Permitir aplicar la esteganografía a imágenes de diferentes dimensiones.

2. Permitir aplicar la esteganografía a imágenes con diferentes modos de color.

3. Mostrar las métricas de error (MSE y PSNR) sobre la imagen.

# Glosario

**(LSB) Bits menos significativos:**

Los bits menos significativos son los últimos n bits de un numero binario leyéndolo de derecha a izquierda.

**(MSB) Bits mas significativos:**

Los bits más significativos son los primeros n bits de un numero binario leyéndolo de derecha a izquierda.

# Explicación del modelo utilizado

El modelo utilizado para esta aplicación se basa en el manejo de [los bits menos significativos](#Bit_menos_significativo) de la imagen de portada y los [bits más significativos](#Bit_mas_significativo) de la imagen oculta, es decir, toma los valores de los bits menos significativos de cada píxel y los sustituye con los valores más significativos de la imagen a ocultar.

## Ejemplo

*Considerando como valor un píxel de una imagen a color con 3 canales.*

**n = 4**

**Imagen portada (128, 25, 239):**Binario(10000000, 00011001, 11101111)  
[Bits menos significativos:](#Bit_menos_significativo) (0000, 1001, 1111)

**Imagen oculta (23, 95, 115):**Binario(00010111, 01011111, 01110011)  
[Bits más significativos:](#Bit_mas_significativo) (0001, 0101, 0111)

## Ocultar imagen

### Sustitución de valores:

En la imagen de portada se sustituyen los bits mas relevantes en las posiciones de los bits menos relevantes.

**Imagen de portada** (10000000, 00011001, 11101111) **Imagen resultante** (10000001, 00010101, 11100111)

### Creación de máscaras para sustituir los valores correspondientes:

    #Crea la mascara para los bits menos significativos elegidos

    mask\_cover = 256 - (2\*\*bits)

    #Crea la mascara para los bits mas significativos

    mask\_secret = 2\*\*bits - 1

Básicamente crea una mascara basada en los bits a utilizar para los bits menos significativos y para los más significativos.

nBits = 4

**Mascara para los bits menos significativos.**  
11111111 - (2^nBits) = 11110000

**Mascara para los bits mas significativos.**

2^nBits – 1= 00001111

### Limpieza de los bits menos significativos de la portada