

# Esteganografía en Imágenes en un chat en Java mediante Sockets

Andrés Cruz Chipol

Facultad De Ciencias de la Computación - Criptografía  
Benemérita Universidad Autónoma de Puebla  
Veracruz, México  
[andres.cruz@alumno.buap.mx](mailto:andres.cruz@alumno.buap.mx)

**Abstract—** Creación de un chat con sockets en Java – NetBeans con interfaz gráfica, Aplicando esteganografía en imágenes.

**Keywords:** *chat, esteganografía, criptografía cuantica, sockets.java.*

## I. INTRODUCCIÓN

El uso de diversas técnicas para ocultar información es requerida para distintos propósitos, pasamos de la criptografía que cifraba la información que nosotros queríamos enviarle a alguien más. Pasando a esteganografía, que ahora no es nuestra necesidad cifrarla si no ocultarla de las demás personas.

## II. ESTEGANOGRAFÍA

Esta técnica nos sirve para ocultar la información en algo que parece a simple vista normal o sin ningún cambio aparente. Esta técnica es tan antigua como la escritura.

A diferencia de la criptografía, que cifra el mensaje para que no se pueda leer sin la clave, el objetivo es simplemente ocultar la existencia del mensaje para los demás. Existen distintos tipos así mismo como la esteganografía digital, que es un método que desarrollaremos aquí en el proyecto.

En cualquier objeto digital se puede ocultar un mensaje secreto, ya sea en un documento de texto, imágenes, claves, extensiones, canciones.

Una de las mejores formas de ocultar información puede ser en imágenes, ya que son formatos grandes donde es difícil a veces obtener esa información rayada.

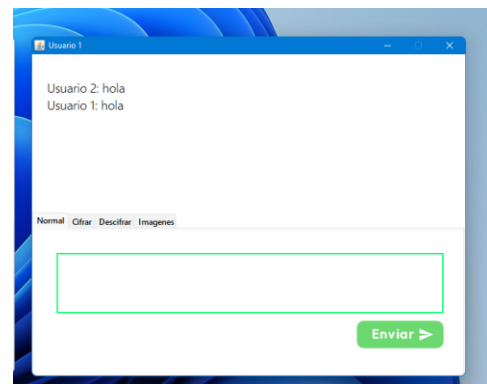
## III. DESAROLLO

Vamos a construir un chat en java, en donde podamos enviar mensajes normales, pero si se requiere enviar un

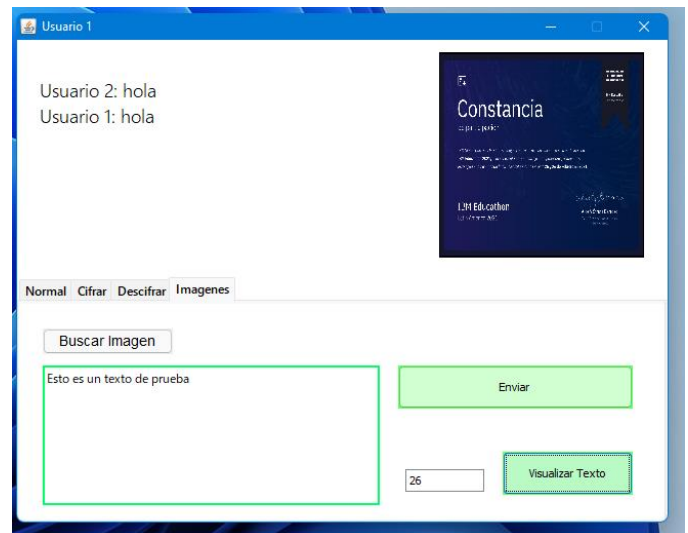
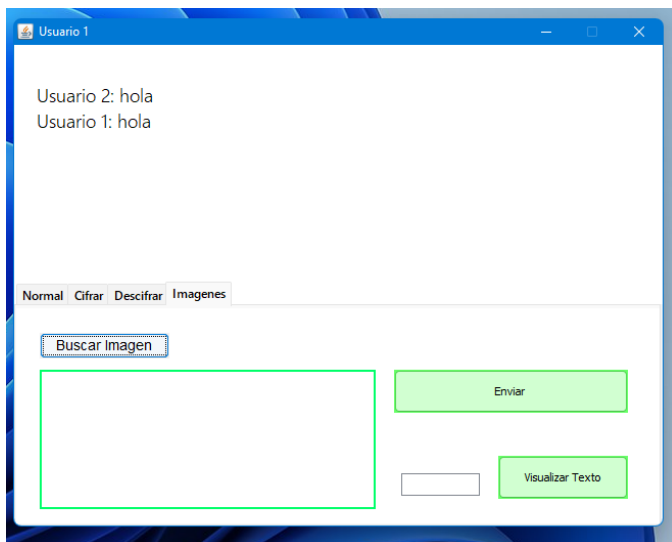
mensaje cifrado, podrá hacerlo sin problemas, además también podrás utilizar las técnicas de esteganografía digital, en la que podrás visualizar las imágenes y también enviarlas para que las otra persona pueda visualizar la imagen y obtener la información mediante un rango de palabras.



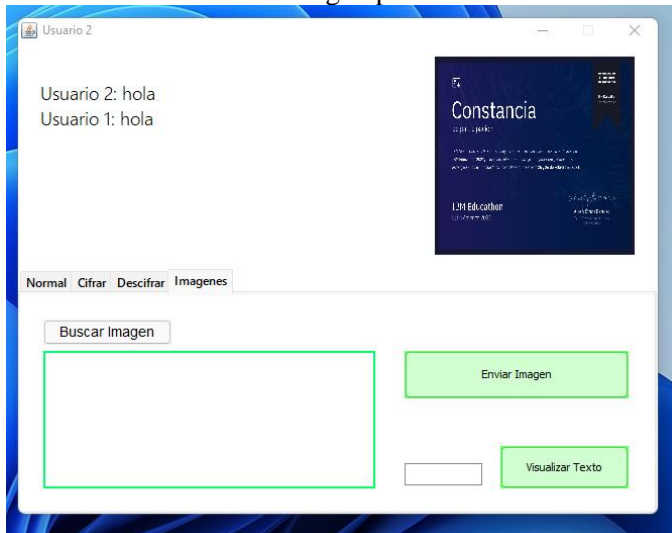
Utilizaremos los pixeles de cada imagen, cada imagen tiene 1 pixel, que contiene 3 bytes donde cada uno contiene la intensidad de colores, RGB, donde podemos modificar los bytes entre 0 y 255, solamente modificaremos el ultimo digito, ya que tiene menos peso, se modifica el bit y no hay una mayor alteración de la imagen.



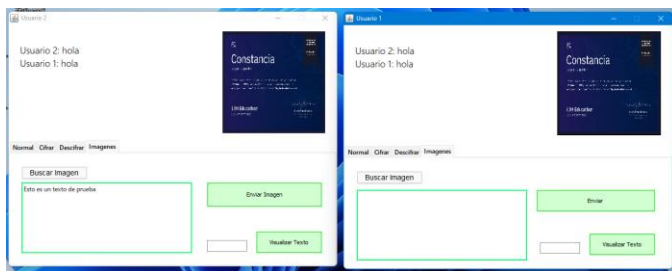
La interfaz se agrando un poco para meter las imágenes.



Buscas la imagen para subirla.



Puedes escribir el texto y enviarlo al destinatario.



Colocas la longitud de texto que quieres visualizar y listo, puedes descubrir el texto.

#### IV. CODIGO

```
public static String visualizar(int numLet){
    String binarioEstego = "";
    try{
        File imagen = new File("C:\\Users\\landy_\\Desktop\\UsuarioUno.png");
        BufferedImage img = ImageIO.read(imagen);
        //get image width and height
        int width = img.getWidth();
        int height = img.getHeight();
        String prueba = "Hola mis queridos amigos";
        String binarioTexto = "";
        binarioTexto = asciiBinario(prueba);
        //System.out.println("Texto Binario:" + binarioTexto);

        int contadorTexto = 0;
        int lenBinarioTexto = binarioTexto.length();
        int numerosLetra = numLet * 8; //50 letras

        for(int y = 0; y < height; y++){
            for(int x = 0; x < width; x++){
                int p = img.getRGB(x, y);
                Color c = new Color(p);

                int R = c.getRed();
                int G = c.getGreen();
                int B = c.getBlue();

                String binarioR = String.format("%8s", Integer.toBinaryString(R & 0xFF)).replace(' ', '0');
                String binarioG = String.format("%8s", Integer.toBinaryString(G & 0xFF)).replace(' ', '0');
                String binarioB = String.format("%8s", Integer.toBinaryString(B & 0xFF)).replace(' ', '0');

                binarioEstego += binarioR.charAt(binarioR.length() - 1);
                numerosLetra = numerosLetra - 1;
                if(numerosLetra == 0){
                    break;
                }
                binarioEstego += binarioG.charAt(binarioG.length() - 1);
                numerosLetra = numerosLetra - 1;
                if(numerosLetra == 0){
                    break;
                }
                binarioEstego += binarioB.charAt(binarioB.length() - 1);
                numerosLetra = numerosLetra - 1;
                if(numerosLetra == 0){
                    break;
                }
                System.out.println(numerosLetra);
            }
        }
        break;
    }
    catch(IOException e){}
```

```

public static void estego(String link,String prueba){
    try{
        File imagen = new File(link);
        System.out.println(link);
        BufferedImage img = ImageIO.read(imagen);
        //get image width and height
        int width = img.getWidth();
        int height = img.getHeight();
        //prueba = "Hola mis queridos amigos";
        //longitudDeTexto = prueba.length();
        String binarioTexto = "";
        binarioTexto = asociBinario(prueba);
        //System.out.println("Texto Binario:" + binarioTexto);
        int contadorTexto = 0;
        int lenBinarioTexto = binarioTexto.length();

        for(int y = 0; y < height; y++){
            for(int x = 0; x < width; x++){
                int p = img.getRGB(x, y);
                Color c = new Color (p);
                int R = c.getRed();
                int G = c.getGreen();
                int B = c.getBlue();
                String binarioR = String.format("%8s", Integer.toBinaryString(R & 0xFF)).replace(" ", '0');
                String binarioG = String.format("%8s", Integer.toBinaryString(G & 0xFF)).replace(" ", '0');
                String binarioB = String.format("%8s", Integer.toBinaryString(B & 0xFF)).replace(" ", '0');

                int lenR = binarioR.length();
                int lenG = binarioG.length();
                int lenB = binarioB.length();

                if((contadorTexto - lenBinarioTexto) != 0){
                    String a = ""+binarioTexto.charAt(contadorTexto);
                    binarioR = binarioR.replaceFirst(".?",a);
                    ++contadorTexto;
                }
                if((contadorTexto - lenBinarioTexto) != 0){
                    String a = ""+binarioTexto.charAt(contadorTexto);
                    binarioG = binarioG.replaceFirst(".?",a);
                    ++contadorTexto;
                }
                if((contadorTexto - lenBinarioTexto) != 0){
                    String a = ""+binarioTexto.charAt(contadorTexto);
                    binarioB = binarioB.replaceFirst(".?",a);
                    ++contadorTexto;
                }
                R = Integer.parseInt(binarioR, 2);
                G = Integer.parseInt(binarioG, 2);
                B = Integer.parseInt(binarioB, 2);

                Color d = new Color(R,G,B);
                int a = A + 0.5f;
                Color d = new Color(R,G,B);
                p = d.getRGB();
                //System.out.println(p);
                img.setRGB(x, y, p);
            }
        }
        ImageIO.write(img,"png",new File("C:\\Users\\landy_\\Desktop\\UsuarioDos.png"));
        System.out.println("Listo");
    }catch(IOException e){}
}

```

## V. CONCLUSIÓN

Este tipo de técnicas son tan útiles como el cifrado de mensajes, los dos tipos son difíciles de descubrir o al menos de obtener, aun que en este caso la esteganografía oculta muchísimo mejor el mensaje, ya que la interceptación del mensaje puede generar dudas al pensar como esta implementado.

Esto depende de la calidad de la técnica que se aplica, muy bien aplicada puede generar redes de comunicaciones tan seguras como son los algoritmos de cifrado. Incluso puede que en nuestro día a día, estas técnicas son empleadas mientras navegamos en internet, lo que puede suceder, es que no tenemos idea en donde esta aplicada esta técnica.

## VI. REFERENCIAS

- [1] [¿Qué es la esteganografía? | Blog oficial de Kaspersky](#)
- [2] [Esteganografía - Wikipedia, la enciclopedia libre](#)
- [3] [Esteganografía. Definición, técnicas y usos frecuentes | Ayuda Ley Protección Datos \(ayudaleyprotecciondatos.es\)](#)