ENCRIPTACIÓN DE IMÁGENES DIGITALES

JUAN CAMILO MUÑOZ TRUJILLO
DANIEL SEBASTIÁN PEÑA GARNICA

UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD DE CIENCIA Y TECNOLOGÍA
DEPARTAMENTO DE MATEMÁTICAS
BOGOTÁ, D.C.
2023

ENCRIPTACIÓN DE IMÁGENES DIGITALES

Juan Camilo Muñoz Trujillo
Daniel Sebastián Peña Garnica

Trabajo de grado presentado como
Requisito parcial para optar por el título de
Licenciado en Matemáticas

Asesor:
Mg. William Alfredo Jiménez Gómez
Prof. Departamento de Matemáticas UPN

UNIVERSIDAD PEDAGÓGICA NACIONAL
FACULTAD DE CIENCIA Y TECNOLOGÍA
DEPARTAMENTO DE MATEMÁTICAS
BOGOTÁ, D.C.
2023.

Dedicatoria

Quiero dedicar este trabajo a mis padres, Jairo Muñoz y Soraya Trujillo, por ser mi principal fuente de inspiración y por acompañarme en cada paso que doy en la búsqueda de ser una mejor persona y profesional. Gracias por su amor incondicional, su apoyo constante y por enseñarme el valor del esfuerzo y la dedicación.

A mi hermana, Cindy Muñoz, le dedico este logro. Gracias por ser mi apoyo incondicional, por creer en mí y alentarme en todo momento. Espero que podamos seguir compartiendo y apoyándonos mutuamente en cada uno de nuestros proyectos futuros.

A Elliot, mi sobrino peludo y fiel compañero, le dedico un lugar especial en esta dedicatoria. Gracias por llenar cada día de alegría, amor incondicional y lealtad. Tu presencia ha sido un bálsamo en los momentos de estudio intenso, recordándome la importancia de la felicidad y la gratitud en cada etapa de la vida.

Agradecimientos

Quiero expresar mi más sincera gratitud a todas las personas que contribuyeron de diversas formas a este trabajo. En especial, agradezco a mis profesores del DMA y a mi asesor, William Jiménez, por su invaluable guía, conocimientos y paciencia en la orientación de este proyecto. Gracias por compartir su sabiduría y experiencias, que han sido pilares fundamentales en mi desarrollo académico y personal.

A mi compañero de tesis, Sebastián Peña, le dedico un profundo agradecimiento. Su colaboración, compromiso y dedicación han sido fundamentales para el éxito de este trabajo. Juntos hemos superado desafíos y logrados metas importantes. Su amistad y colaboración en este equipo de trabajo han enriquecido enormemente mi experiencia académica.

Finalmente, deseo expresar mi profundo agradecimiento a la Universidad Pedagógica Nacional, por ofrecerme la oportunidad de formarme académicamente y por facilitar los recursos necesarios para llevar a cabo este y otros proyectos. Este logro representa también el resultado del respaldo constante de la institución y su dedicación a la excelencia educativa en la formación de educadores.

Juan Muñoz

Dedicatoria

Primero, quiero dedicar este trabajo a mis padres, Sandra Garnica y Mario Peña. Sin su inquebrantable apoyo, no habría sido posible llegar hasta donde estoy en este momento. Agradezco especialmente por enseñarme que no existen obstáculos insuperables cuando se tiene un objetivo claro.

Además, quiero dedicar esta tesis a mis queridos hermanos. Su constante apoyo y solidaridad han sido un faro en mi camino, sin importar la distancia o las circunstancias en las que me encuentre. Siempre sé que puedo contar con ellos.

Agradecimientos

En primer lugar, me gustaría expresar mi sincero agradecimiento a mi excelente compañero de trabajo, Juan Muñoz, por permitirme colaborar a su lado. Sus brillantes ideas han contribuido significativamente al desarrollo de este y otros proyectos en los que hemos trabajado juntos. Su entrega constante ha sido imprescindible para alcanzar nuestros objetivos.

Asimismo, quiero agradecer de corazón a todos los profesores de la Licenciatura de Matemáticas por sus valiosas enseñanzas a lo largo de mi carrera. Gracias a ellos, me he formado como una persona integral y he aprendido la importancia de nunca conformarse y estar en constante búsqueda de mejoras y evolución.

Por último, pero no menos importante, deseo expresar mi agradecimiento a mi tutor William Jiménez. Su ayuda, grandes ideas, disposición y guía han sido fundamentales para el éxito y culminación de este trabajo. Sin su apoyo no habría sido posible alcanzar los resultados obtenidos

Sebastián Peña

TABLA DE CONTENIDO

Lista de Tablas

Lista de Figuras

Resumen

El objetivo fundamental de este estudio fue crear un algoritmo destinado a cifrar imágenes digitales utilizando la técnica de Descomposición en Valores Singulares (SVD (Singular Valúes Decomposition) a través de la plataforma Octave Online. En paralelo se desarrolló un sitio web en Google Sites cuyo enfoque divulgativo presenta conceptos fundamentales de encriptación de imágenes digitales, además de ofrecer diagramas de flujo y manuales de usuario para las aplicaciones desarrolladas.

Además, el sitio da acceso a los códigos necesarios para encriptar y cargar las capas RGB de las imágenes originales y cifradas. El enfoque principal de este proyecto se centró en desarrollar y presentar los algoritmos creados y en hacer pruebas exhaustivas para detectar problemas en la carga de imágenes en Octave Online. Finalmente, se lleva a cabo un análisis experimental centrado en la distribución de intensidades en las imágenes, utilizando para ello histogramas.

Palabras Claves: Cifrado de imágenes digitales, Octave Online, Descomposición en Valores Singulares.

Abstract

This study aimed to create an algorithm for encrypting digital images using the Singular Value Decomposition (SVD) technique via Octave Online. A Google Sites website was developed with a focus on dissemination, presenting fundamental concepts of digital image encryption, offering flowcharts, user manuals, and access to necessary codes for encrypting and uploading RGB layers of original and encrypted images. The project primarily focused on developing and showcasing the created algorithms, conducting exhaustive tests to detect issues in image uploading on Octave Online. Lastly, an experimental analysis centered on intensity distribution in images was conducted using histograms.

Keywords: Digital image encryption, Octave Online, Singular Value Decomposition,

Introducción

Desde hace varias décadas la avalancha de datos, y salvaguardar la integridad de estos, constituye uno de los principales desafíos en los ámbitos de la informática, las matemáticas y la telemática en general (Arboledas, 2017). Este desafío ineludible ha dado origen a una nueva rama de estudio en el ámbito informático que se conoce como “Seguridad de la Información”. A medida que avanza el tiempo, se despliegan nuevos enfoques y métodos para asegurar los datos. Sin embargo, persiste la firme amenaza de hackers y actores no autorizados que constantemente intentan socavar los métodos y protocolos criptográficos, con la intención de acceder a información valiosa y delicada. Por esta realidad, los expertos en ciencias informáticas y en criptografía buscan soluciones sustentables y duraderas a esta problemática creciente (Dey, 2012). Según los expertos Mit-nick, Simon y Wozniak (2003 citados por Cárdenas S. et al. 2016) han planteado la siguiente premisa: "*Nunca debe depositar su confianza en los sistemas de seguridad en línea para resguardar su información. En su lugar, es fundamental evaluar el eslabón más débil. En la mayoría de las situaciones, identificará que este punto vulnerable radica en las acciones de las personas*" (p. 79).

En respuesta a estos riesgos latentes, una de las estrategias es proteger nuestros datos a través de técnicas criptográficas. Este enfoque garantiza, al menos, la confidencialidad de la información y la integridad de los mensajes. Por eso, la cifra de imágenes se ha propuesto como una solución contra posibles intrusiones de piratas informáticos y personas no autorizadas que intentan vulnerar los sistemas de seguridad (Dey, 2012).

Pero ¿qué es exactamente una imagen digital? No es necesario consultar a expertos en fotografía o computación, o expertos en sociología, para saber que las imágenes digitales juegan un papel importante en la vida cotidiana, ya sea de índole personal, político, militar o económico. Su definición y origen se encuentran arraigados en el campo de las matemáticas, lo que puede resultar un concepto algo alejado para quienes la utilizan sin entender su base teórica.

La percepción visual humana se basa en la combinación de tres colores primarios: rojo, verde y azul, según lo planteado por Alonso (2009). Estos colores fundamentales son clave en la representación de imágenes, ya que la construcción de estas se logra mediante la combinación de estas tres tonalidades. Esta concepción ha impulsado avances significativos en la tecnología y se encuentra ampliamente presente en computadoras, teléfonos celulares y otros dispositivos tecnológicos

Desde el punto de vista de las matemáticas, la imagen digital es una matriz de píxeles. Esto conduce a pensar que, al ser un elemento de la teoría del álgebra lineal, se pueden aplicar métodos como el de Hill o la Descomposición en Valores Singulares (SVD). Estos procedimientos se pueden utilizar para alterar la posición de los píxeles y hacer modificaciones en la imagen (Molina et al., 2019), lo que evidencia cómo las matemáticas son importantes en la manipulación y mejora de imágenes digitales.

Finalmente, este trabajo busca desarrollar un espacio web, en el que los usuarios logren asimilar la forma de encriptar una imagen digital, su concepto y su relación con las matemáticas, además de ofrecer un espacio para encriptar imágenes, accesible para todas las personas.

Basándonos en lo mencionado anteriormente, la estructura de esta investigación se divide en cinco secciones principales:

En esta primera sección, se realiza la *Investigación Documental* relacionadas con la cifra de imágenes a nivel local, nacional e internacional. Posteriormente abordando los *Antecedentes Teóricos*, que ofrecen una visión más amplia del campo de investigación que fundamenta este proyecto. A continuación, se presenta el desarrollo del programa principal denominado "Encriptación por SVD", el cual descompone una imagen en canales de colores, realiza la conversión a una matriz de doble precisión y emplea la descomposición en valores singulares para el cifrado. Además, se introduce otro algoritmo de encriptación, la "Creación Propia", que cifra la imagen mediante la sustitución con una clave secreta aleatoria, generando una nueva imagen cifrada. La siguiente sección se centra en la *Creación del Sitio Web* en Google Sites, que exhibe la aplicación para que los usuarios encripten y carguen las capas RGB de las imágenes original y cifrada. Esta sección también proporciona una explicación del código y fundamentos teóricos que respaldan la encriptación de imágenes, junto con ejemplos ilustrativos. Finalmente, se lleva a cabo un *Análisis de Resultados* que contrasta la imagen original en escala de grises con la imagen cifrada, presentando las conclusiones del trabajo basadas en los resultados obtenidos. Esta estructura organizativa ofrece una comprensión detallada y completa del trabajo realizado en cada fase, brindando una visión global de la investigación en encriptación de imágenes y su aplicabilidad práctica.

CAPÍTULO 1. Justificación

El rápido progreso tecnológico de las últimas décadas ha impulsado a la sociedad a desarrollar nuevos enfoques destinados a salvaguardar información de suma relevancia, que puede ser de índole personal, social, política o militar (Mamani C, 2018). La criptografía según establece Matus (2015), se puede entender como una ciencia que está relacionada con la teoría de números, la probabilidad y la teoría de la complejidad computacional, que busca proteger la información que se comparte en un medio de comunicación inseguro.

En concordancia con Viguer (2019), la encriptación de imágenes agrega un nivel adicional de seguridad al transformar la información visual en un formato ilegible para quienes no poseen la clave de descifrado correspondiente. Esta medida no solo dificulta los intentos de acceso no autorizado, sino que también proporciona protección a la información sensible.

Abbadi (2014) argumenta que la adopción de enfoques *innovadores* como la encriptación de imágenes digitales se ha convertido en una estrategia esencial para preservar la integridad y la privacidad de los datos en entornos digitales en constante evolución. Este enfoque innovador, al que Abbadi hace referencia, encuentra su aplicación en la enseñanza de las matemáticas y en el uso de las Tecnologías de la Información y las Comunicaciones (TIC). La diversidad de entornos educativos y la presencia de asignaturas con un énfasis en contenidos predominantemente descriptivos demandan el desarrollo de habilidades perceptivas en los estudiantes. Por esta razón, la selección y el eficaz manejo de los recursos educativos adquieren una relevancia particular (Marrero et al., 2016).

En el campo de la salud y la medicina, la encriptación de imágenes juega un papel crucial en la protección de la privacidad y la confidencialidad de los pacientes. Las imágenes médicas, como radiografías, resonancias magnéticas (RM)[1], tomografías computarizadas (CT)[2] y ecografías, contienen información sensible y personal, como detalles anatómicos, diagnósticos y datos biométricos.

Estas imágenes se encriptan durante su almacenamiento, transmisión y acceso para prevenir la intrusión no autorizada o el uso indebido de la información médica (Velandia et al., 2020). Al aplicar técnicas de encriptación, se garantiza que solo personal autorizado, como médicos, radiólogos o profesionales de la salud designados, puedan acceder a estos archivos protegidos.

Como señala Cidón et al. (2011), la encriptación se emplea para codificar estas imágenes, utilizando algoritmos y claves de seguridad, lo que significa que incluso si los datos se interceptan o se accede a ellos de manera no autorizada, permanecen ilegibles e incomprensibles sin la clave de desencriptación correspondiente.

En este contexto, tal como señalan Marrero et al (2016), estas imágenes tienen la capacidad de reflejar la realidad y facilitar la asimilación del conocimiento, ya que se convierten en un nuevo objeto material destinado a la comunicación. De esta manera, las representaciones visuales se erigen como signos que reemplazan al objeto original y presentan las cualidades necesarias para una comunicación efectiva.

[1] (RM). Abreviatura médica para referirse a Resonancias Magnéticas

[2] (CT). Abreviatura médica para referirse a Tomografías Computarizada

Objetivos

Objetivo General

Desarrollar una solución integral para la encriptación de imágenes digitales basada en métodos convencionales de cifrado.

Objetivos Específicos

- Consultar y apropiar los métodos convencionales de cifrado que hacen uso del álgebra lineal, seleccionando aquellos que presenten mayor accesibilidad.
- Mostrar la fundamentación matemática subyacente en los métodos de Hill y SVD, así como la aplicabilidad en el contexto de la encriptación de imágenes digitales.
- Examinar algunos lenguajes de programación para buscar idoneidad en el método de cifrado de imágenes digitales.
- Desarrollar una página web interactiva que brinde a los usuarios la capacidad de cargar imágenes, aplicar la encriptación mediante los algoritmos desarrollados, visualizar las capas RGB resultantes y acceder a manuales de uso detallados.

CAPÍTULO 2. Investigación Documental

Es fundamental revisar la información en orden cronológico poder tener presente los acontecimientos anteriores o que están aconteciendo para posibilitar una reflexión de los aspectos que alude a instrumentos para evaluar las categorías de análisis que se estén trabajando (Reyes y Carmona,2020). Este documento, sustancialmente presenta una revisión documental con fuentes primarias de información que estén relacionadas directamente con el tema de indagación. Con base en lo anterior, Reyes y Carmona (2020) denominan a este primer paso como ***Arqueo de fuentes.*** Para esto, si no se es experto en el tema, se debe extraer palabras claves, términos de búsqueda del problema o ideas de investigación de modo que permita realizar una indagación suficiente de la información en diversas bases de datos. Calle (2016) advierte la necesidad de que dichas bases de datos sean confiables, puesto que hay algunas que contienen documentos de manera fraudulenta, sin permiso de la editorial o del autor.

Después de recopilar las fuentes primarias de información, es esencial pasar a revisar y analizar su contenido mediante una consulta detallada se debe consultar. La ***Revisión*** se realiza con el fin de descartar documentos que no sean relevantes para el desarrollo de la investigación. Posteriormente, se debe hacer un ***Cotejo*** y organización de las fuentes consultadas con el fin de generar citas y referencias que sustenten los planteamientos del investigador. Por último, se debe ***Interpretar*** y analizar el material cotejado y establecer ***Conclusiones*** de todo el proceso de revisión documental (Reyes y Carmona, 2020).

Teniendo en cuenta la estructura planteada por Reyes y Carmona (2020) se presenta a continuación las fases o metodologías a seguir de la siguiente mamera:

La selección del material que podría servir de ayuda para el desarrollo del tema de investigación de Encriptación de imágenes, se centró en buscadores académicos y bases de datos *Springer, Google Scholar, Dialnet, Eureka* y repositorios de universidades, en su mayoría colombianas, como la Universidad Pedagógica Nacional de Colombia , Universidad Pedagógica y Tecnológica de Colombia, Universidad Distrital de Colombia , Universidad Nacional de Colombia , Universidad de la Sabana y en particular la Universidad Pontifica de Valencia. Como resultado de esta búsqueda se encontraron 3 artículos de revista, 2 libros o capítulos de libros, cinco tesis (una de doctorado, una de pregrado, tres de maestría) y una videoconferencia en la plataforma de YouTube.

Una vez obtenida toda la información primaria en formato impreso o electrónico, se procedió a descartar aquellos “poco” útiles para la elaboración de este trabajo. Para esto se analizó el índice de contenido o el índice analítico de los libros o tesis adquiridas, los cuales proporcionaron una idea de los temas, incluidos en la elaboración de dichas obras. Con respecto a los artículos en la Tabla 1, se examinó el resumen, palabras claves y las conclusiones.

Palabras Claves

Criptografía, Encriptación, Encriptación de imágenes, Cifrado, Tipos de Cifrados, Cifrado de Hill, Aritmética modular, Métodos numéricos, Algoritmos, Álgebra matricial, Programación, Lenguajes de Programación, Descomposición en Valores Singulares (SVD).

Tabla 1. Matriz de descripción de fuentes consultadas

Tipo de fuente	Autor (es)	Año(s)	Título de la fuente	Lugar de desarrollo	Institución
Artículos	Vilardy et al	2011	*Encriptación en Fase Aplicado a Imágenes Digitales a Color*	Colombia	*Revista Colombiana de Física. Vol. 43. No. 2*
	Arguello et al	2015	*Encriptación de Imágenes Aplicando el Método De Hill*	Colombia	*Tercer encuentro nacional de Matemáticas, Estadística y Educación Matemática*
	Rodríguez et al	2017	*Algoritmo de Encriptación de Imágenes Utilizando el Atractor Caótico de Lorenz*	Colombia	*Universidad Distrital Francisco José de Caldas*
	Abd El-Latif et al	2012	*A New Image Encryption Based on Chaotic Systems and Singular Value Decomposition*	China - Egipto	*School of Computer Science and Technology, Harbin Institute of Technology, 150080 Harbin China.* *Department of Mathematics, Faculty of Science, Menoufia University, Shebin El-Koom 32511, Egipto.* *School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate*

					School, 518055 Shenzhen China
	El Abbadi et al	2014	*Image encryption based on singular value decomposition*	*Iraq*	*Department of Computer Science, University of Kufa*
Tesis de Pregrado	Viguer	2020	*Desarrollo e implementación de una aplicación informática con Matlab para la encriptación fractal de las imágenes basadas en el Cifrado de Hill.*	España	Universidad Politécnica de Valencia
Tesis de Maestría	Durán	2017	*Propuesta de enseñanza para la comprensión del lenguaje algebraico, a partir de la aplicación de los enteros módulo n y el álgebra matricial en el campo de la criptografía*	Colombia	*Universidad Nacional de Colombia*
	Espinoza	2015	*Cifrado de imágenes digitales basada en teoría del caos: mapas logísticos*	España	*Universidad Politécnica de Madrid*

Nota: Esta tabla muestra los artículos, tesis de grado y de maestría, que sirvieron como referentes para este estudio.

Se inició la indagación en el repositorio de la Biblioteca de la Universidad Pedagógica Nacional, libros, tesis en físico o digital del tema de Encriptación, Cifrado de Hill u otro tipo de cifrados. El resultado de esta búsqueda arroja la ausencia de trabajos sobre dichos temas; o sea, hasta que no hay evidencia de tesis relacionadas con Encriptación o cifrados en esta Universidad, por lo que se descarta esta fuente de datos.

Por lo tanto, la búsqueda inicia en Google Académico. Se dedicó averiguar trabajos de Encriptación de imágenes. Inicialmente encuentran un trabajo de Vilardy et al. (2011), denominado "*Encriptación de Imágenes Digitales vía Transformada Fraccional De Fourier Discreta Y*

Transformada Jigsaw"; este documento presenta un nuevo método de encriptación de imágenes. La propuesta se basaba en que la imagen encriptada es real y tiene el mismo tamaño que la imagen original (en comparación con otros sistemas de seguridad basados en Transformada Fraccional de Fourier - FrFT), siendo esta imagen encriptada mucho más conveniente para su almacenamiento o transmisión por redes de comunicación digital. El uso de La Transformada Jigsaw incrementaba la seguridad de la imagen encriptada y eliminaba las "máscaras" de fases aleatorias utilizadas en muchos criptosistemas de imágenes basados en FrFT. Por último, para este método de encriptación propuesto, fue desarrollado un algoritmo digital de encriptación de imágenes. En el algoritmo criptográfico implementado doce llaves son usadas, constituidas por los seis órdenes fraccionales de las DFrFTs y las seis permutaciones aleatorias utilizadas en las transformadas Jigsaw, todas esas llaves eran necesarias para una correcta desencriptación, obteniendo un alto nivel de seguridad en la protección de las imágenes digitales para una determinada aplicación.

En seguida se acude al repositorio institucional de la Universidad Pedagógica y Tecnológica de Colombia (UPTC) con el trabajo titulado "*Encriptación de Imágenes Aplicando el Método De Hill*" realizado por Este documento muestra un grupo de Énfasis en Matemática del Instituto Nacional de Educación y está dirigido por un profesor. en el campo de las matemáticas, tratando de acercar a los alumnos a las tareas matemáticas. Este trabajo se presenta como un proyecto estudiantil que pretendía emplear un algoritmo que permitiera cifrar las imágenes digitalmente. Para ello se señala la necesidad de estudiar conceptos matemáticos relacionados con el álgebra

modular y para la creación del algoritmo, el método de cifrado es Hill, adaptándose a las imágenes digitales en escala de grises.

Posteriormente se observa el trabajo realizado por Duran (2017) localizado en el repositorio institucional de la Universidad Nacional de Colombia titulado "*Propuesta de enseñanza para la comprensión del lenguaje algebraico, a partir de la aplicación de los enteros módulo n y el álgebra matricial en el campo de la criptografía*". En este trabajo se presenta el diseño y aplicación de programas instruccionales en el marco instruccional de comprensión. La propuesta buscaba potenciar el proceso de comunicación matemática de los estudiantes situándolos en el contexto de la criptografía, específicamente la criptografía de Lester Hill. Para ello, Duran (2017) vio necesario abordar algunos temas de álgebra matricial y aritmética de módulo y se aplicó a un grupo de 9 alumnos de décimo grado del Colegio San Ignacio de Loyola de Medellín. El análisis de los resultados mostró que los estudiantes tenían niveles más altos de comprensión de las pruebas diagnósticas preliminares trabajados por Duran (2017), y estos niveles más altos de comprensión se percibían a partir del progreso que mostraban durante la comunicación matemática.

Con respecto a la investigación del repositorio de Universidad Distrital de Colombia se encuentra el trabajo realizado por Rodríguez et al (2020): en donde se propone un algoritmo simétrico empleado el sistema caótico Cat de Arnold para la permutación y para la difusión el sistema hipercaótico de Chen o el sistema hipercaótico de Lorenz. En la implementación del estudio de Rodríguez et al (2020) se utiliza programación paralela para reducir los tiempos de ejecución. Se aplican métricas de desempeño para evaluar la seguridad del modelo criptográfico propuesto. Se encuentra que los indicadores obtenidos se enmarcan en artículos recientes, que

abordan el problema de la seguridad a través del caos. Los resultados permitieron confirmar que la teoría del caos, para fortalecer los esquemas de seguridad en comunicaciones, es una buena alternativa, especialmente al referirse a la transferencia de imágenes.

En esta etapa de la revisión documental, se accedió a la base de datos Eureka y el repositorio de la universidad de la sabana, el primer documento que se presenta es Algoritmo de Encriptación para Imágenes a color basado en Sistemas Caóticos (Santos D., Amaya I., Parra C.) Esta investigación es un artículo de la revista científica Ingeniería, en la que se propone establecer un modelo de encriptación de imágenes digitales, basándose en los sistemas caóticos, presentan una serie de investigaciones relacionadas con la encriptación de imágenes digitales a color y a blanco y negro.

Los autores presentan los sistemas dinámicos como un proceso que varía con el tiempo asociado a una regla de tipo o continuo o discreto, el sistema que se utiliza en el algoritmo de la investigación es de Arnold, en las secciones posteriores se presenta el paso a paso para la configuración del algoritmo, y su posterior análisis con un ejemplo planteado.

La segunda investigación que se encuentra en esta base datos se desarrolla en España. Desarrollo e implementación de una aplicación informática con Matlab para la encriptación fractal de imágenes basada en el cifrado de Hill (Viguer A.,2019), esta investigación presenta en una primera parte un breve recuento de la encriptación a lo largo de la historia, presenta contenidos como aritmética modular y algunos comandos de operaciones en el software Matlab, hace una relación entre el método de Hill y como se aplicará al software. En una segunda parte presenta el concepto de fractal y los que se usarán para el desarrollo de la investigación, muestra

conceptos que serán claves en la puesta en práctica como el desarrollo de matrices pseudoaleatorias en Matlab y algunos algoritmos de encriptación como el de Diffie Helman.

Finalmente, presenta la investigación, muestra el paso a paso de la encriptación con dos imágenes y los códigos utilizados para encriptarlas en los métodos estudiados, expone un análisis detallado de la confiabilidad de los métodos utilizados y los resultados obtenidos.

El autor diseñó dos versiones ejecutables del repositorio de la universidad de valencia, en las que cualquier usuario inexperto puede encriptar imágenes digitales, tras la investigación.

Con base en el cifrado de imágenes digitales basado en teoría del caos – mapas logísticos, (Espinosa M., 2015) esta investigación se desarrolla en España, más exactamente en la Universidad Politécnica de Madrid, el autor divide en ocho capítulos el desarrollo de su trabajo que tiene por objetivo, el desarrollo de una propuesta criptográfica que minimice los riesgos de hackeo de la información y que además optimice el tiempo de ejecución del proceso de encriptación. En el capítulo tres presenta la fundamentación teórica utilizada, en el cual presenta un breve recuento histórico de la evolución de las imágenes digitales, información sobre la teoría del caos, sus propiedades y los diferentes sistemas caóticos, también presenta los mapas logísticos, los cuales son utilizados en la creación posterior del algoritmo.

El autor dedica el capítulo cuatro, a la explicitación de las tecnologías que utilizaran para el desarrollo del algoritmo, OpenCv y las diferentes librerías y procesos que se llevaran a cabo, en los siguientes capítulos se esboza y se presenta el algoritmo de encriptación, se presenta de manera detallada cada procedimiento utilizado, incluyendo los procesos de encriptación y desencriptación, finalmente presenta un análisis de los resultados obtenidos en comparación con

otros sistemas de encriptado en relación al tiempo de ejecución, y presenta algunas posibles líneas de trabajo a seguir posteriores a esa investigación**.**

Asimismo, en el artículo de Abd El-Latif et al (2012), introduce un eficaz plan de encriptación de imágenes basado en sistemas caóticos y la descomposición de valores singulares. En este método, las posiciones de los píxeles en la imagen se mezclan mediante sistemas caóticos con parámetros de control cambiantes. Para incrementar aún más la seguridad, los valores de nivel de gris de los píxeles se ajustan usando una combinación de la descomposición de valores singulares (SVD) y un mapa polinómico caótico. Los resultados de simulación respaldan la viabilidad de este plan propuesto para el propósito de encriptación de imágenes.

Además, se presenta un innovador plan de encriptación de imágenes que se basa en mapas caóticos y SVD. En este enfoque, se trastoca la imagen original utilizando un mapa caótico bidimensional con parámetros de control variables, y los valores de nivel de gris de los píxeles de la imagen se cifran empleando una combinación de la descomposición de valores singulares (SVD) de la imagen y un mapa caótico unidimensional. Los parámetros de control del mapa caótico bidimensional se generan al azar mediante un mapa caótico unidimensional para asegurar que la clave de trastorno dependa de la clave. Además, la secuencia clave usada para encriptar una imagen trastornada se extrae de los vectores singulares izquierdo y derecho, así como del mapa caótico.

En esta investigación El Abbadi et al (2014), presenta un método innovador y seguro para la encriptación de imágenes. El algoritmo propuesto se basa en la utilización de la descomposición de valores singulares (SVD). En este proceso, se inicia mezclando los datos de la imagen

mediante claves sugeridas, generando así dos matrices diferentes. La matriz diagonal resultante de la SVD se intercambia con las matrices obtenidas. Adicionalmente, se realiza otro proceso de mezcla e intercambio de matrices diagonales para aumentar la complejidad. Posteriormente, las dos matrices resultantes se combinan en una sola matriz según un procedimiento establecido. El resultado es una imagen encriptada con contenido significativo. La eficacia de este método se ha demostrado a través de pruebas con diversas imágenes.

En la misma línea, el artículo de El Abbadi et al (2014), introduce un método sólido, simple e innovador para asegurar la protección de imágenes, haciendo uso de la descomposición de valores singulares (SVD). Esta aproximación innovadora para encriptar imágenes mediante SVD desorganiza los píxeles en la imagen, reduciendo la correlación entre ellos y generando así una imagen cifrada con una correlación de píxeles menor. Durante la investigación, se logró descifrar una imagen que guarda similitud con la original. Una característica destacada de este enfoque es que las imágenes cifradas mantienen su contenido representativo. Además, los tiempos de cifrado y descifrado resultan competitivos en comparación con otras técnicas de encriptación. La inclusión de SVD en el proceso de cifrado brinda una perspectiva nueva, y se sugiere, como posibilidad futura, la aplicación de SVD en el cifrado de texto.

En la Tabla 2, se enunciarán las fuentes que se investigaron y que no se reseñaron, para compararlas y contrastarlas con las detalladas anteriormente. Por consiguiente, la intensión es, por un lado, delimitar aquellos documentos que se enfocan principalmente en la historia de la encriptación y los diversos métodos empleados por diversas culturas y, por otro lado, los métodos que se desarrollaron para realizar específicamente el cifrado en imágenes son:

Tabla 2. Fuentes no reseñadas

Título	Autor(es)
Algoritmos de encriptación de clave asimétrica.	Franchi (2012).
Técnicas de encriptación para mejorar la seguridad en la transferencia de archivos en un entorno fiable.	Sánchez. (2017)
Proyecto para la técnica de transparencia y encriptación de información.	Sáenz (2015)
Análisis de algoritmos de encriptación de datos de texto, una revisión de la literatura científica.	Cabanillas et al. (2022)
Una propuesta de práctica informática: aritmética modular y encriptación de imágenes.	Palomares et al (2023)

La comparación y contrastación elaborada permitió establecer aspectos por indagar respecto a la encriptación de imágenes en específico. Se observó que hay mucha información sobre el uso de encriptación de archivos, datos e información y descripciones de los algoritmos de clave pública más utilizados actualmente. Cabe mencionar que esta es un área de la ciencia con constantes avances, no obstante, los algoritmos más seguros son siempre los que más tiempo han estado expuestos al dominio público y que aún no han podido ser atacados exitosamente.

De acuerdo con el análisis realizado a las fuentes consultadas en la fase de revisión, se resaltan aquellas que tienen una mayor afinidad en relación a los objetivos planteados para esta investigación. Según estas fuentes, se perfiló y decidió las directrices para fundamentar y acotaría el trabajo, presentadas en capítulos posteriores.

Gracias al contraste realizado en la fase de cotejo, se descartan algunas de las fuentes no reseñadas, ya que no están relacionadas con el tema de investigación, por lo tanto, no son idóneas para establecer una orientación con las ideas principales que se planean trabajar. No obstante, son de utilidad y apoyo para el sustento teórico de este estudio.

Estas investigaciones permitieron hacer una clasificación acerca de los pilares teóricos que componen el tema de investigación; las categorías que parcialmente se eligieron son: aritmética modular, algebra lineal e imagen digital. Sin embargo, a lo largo de la investigación se verá la suficiencia y conveniencia, de establecer dichas categorías, ya sea para agregar otra que no se tuvo en cuenta, o suprimir alguna que no haya tenido relevancia sustancial para la investigación.

La investigación inicialmente se planteó realizando la siguiente Tabla 3 de clasificación:

Tabla 3. Forma de la tabla diseñada como propuesta de clasificación.

Autor	Aritmética Modular	Algebra lineal	Imagen digital	…
Palomares et al (2023)	x			
Duran (2017)		x		
…			x	

Inicialmente, se destaca la carencia de fuentes locales relacionadas con el ámbito de la encriptación de imágenes, especialmente en lo que concierne a repositorios provenientes de las principales universidades públicas nacionales. Esta observación resalta una marcada ausencia de investigación y documentación en el contexto nacional en un tema tan relevante como la encriptación de imágenes.

La culminación del estudio se orientará hacia la elaboración de conclusiones sustanciales. Estas abordarán una serie de reflexiones generales en relación con la información extraída de las fuentes analizadas. Se llevará a cabo un análisis sobre el logro de los objetivos establecidos, tanto los generales como los específicos.

Dentro de esta etapa conclusiva, también se señalarán áreas que podrían explorarse con mayor profundidad, ofreciendo posibilidades para investigaciones futuras en la misma línea. Se destacará la continuidad posible para dar seguimiento a la labor realizada hasta el momento y expandir la exploración en aspectos relacionados con la encriptación de imágenes y su integración en los contextos educativos, así como en el ámbito de la investigación local y nacional.

CAPÍTULO 3. Marco Teórico

Este capítulo inicia con un breve repaso histórico de las primeras nociones de encriptación, además de presentar algunos de los métodos clásicos utilizados en este proceso. Se aborda el diverso propósito con el que se empleaban estos métodos, que iba desde ocultar secretos militares y políticos, hasta cuestiones de índole sentimental. Es importante destacar que esta investigación es un panorama general y no pretende profundizar en la historia y evolución de la encriptación, sino contextualizar la relevancia de este proceso en el tiempo.

Mas adelante, se exponen dos de los métodos más empleados en la encriptación y procesamiento de imágenes digitales. Es importante señalar que esta explicación no se adentrará en la definición de conceptos básicos del álgebra lineal, sino que se enfocará en resaltar la esencia fundamental de cada método.

Finalmente, se ofrece un análisis concluyente que respalda la elección del software destinado al desarrollo de esta propuesta. Se detallan los parámetros considerados en el proceso de selección, fundamentando la decisión final. Con este software seleccionado, se sienta una base sólida para encriptar imágenes digitales, que se mostrará en los capítulos posteriores.

3.1 Hitos del Desarrollo de la Encriptación a lo Largo de la Historia

Las primeras apariciones relacionadas con la criptografía pueden situarse en el año 3000 A.C., con el desarrollo de las civilizaciones de Mesopotamia; algunos autores señalan que las primeras nociones de la encriptación, se dieron gracias a la escritura jeroglífica, en un grabado que aparece en la llamada piedra Rosetta (Figura 1). En esta, se plasmó la vida, y los aspectos más relevantes de un noble de esa época. Este texto, tallado en la piedra, presenta algunos

semagramas[3] que se utilizaban en la época. No se tallaron con el objetivo de ocultar algún mensaje, sino con el fin de resaltar o dotar de interés al escrito (Xifre, 2009).

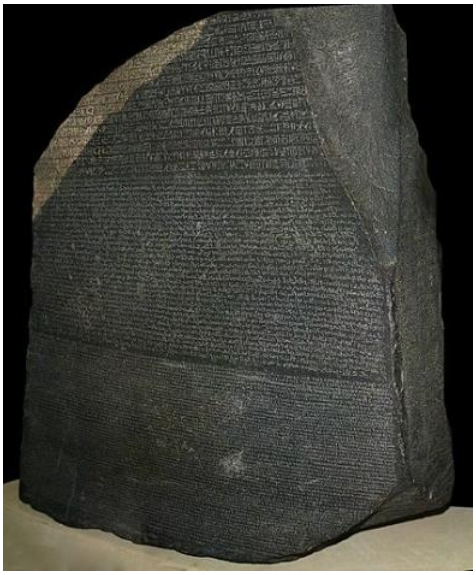


Figura 1. Piedra Rosetta[4]

Esta misma civilización, de acuerdo con lo que expresa Xifre (2009), desarrollaron la escritura cuneiforme[5] con el fin de ocultar mensajes. Un ejemplo de ello se encuentra en la Figura 2, tablilla de arcilla que se conserva hasta la época, en la que se escribió secretamente la fórmula para el barniz, el cual se concibe como un recurso muy valioso para la época.

[3] Semagrama: los determinantes o semagaramas se ponen al final de la palabra, son caracteres mudos que sirven para indicar el campo semántico de la palabra.

[4] Imagen tomada de: Fatás G. (2005) Historia Antigua, La piedra Rosetta, Universidad de Zaragoza

[5] Cuneiforme: Es como se designa al tipo de escritura tallado por medio de "cuñas" generalmente en arcilla donde cada cuña representa una palabra.

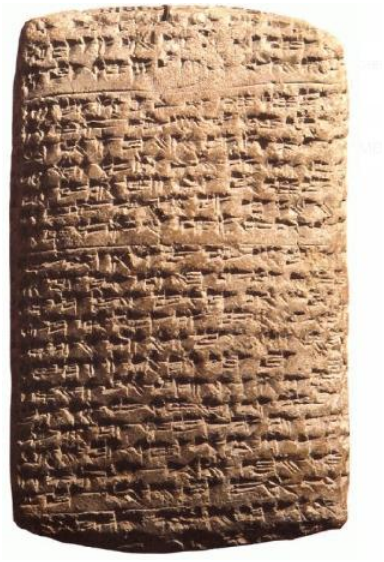


Figura 2. Escritura Cuneiforme[6]

Otra cultura que desarrolló técnicas para ocultar información fue la cultura China. Utilizaban la estenografía en seda o papiros, un método que implicaba la sustitución de letras por diferentes signos. Lo peculiar del método (Figura 3), es que los mensajeros ingerían el mensaje o lo ocultaban en sus propios cuerpos.

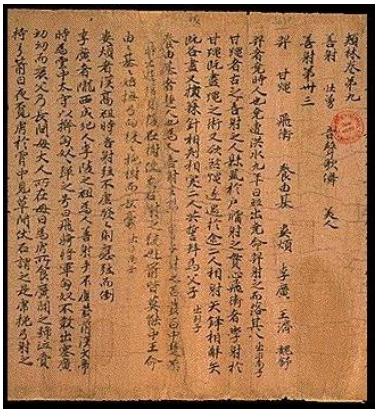


Figura 3. Estenografía China[7].

[6] Imagen tomada de: Seri A (2014). El uso de la escritura Cuneiforme para escribir el acadio, Universidad Nacional del Rosario.

[7] Imagen sin autor tomada de: http://www.cop.es/cabezas/chino.htm.

Es importante destacar que tanto las culturas mesopotámicas como la china, ocultaban información con propósitos políticos, militares, religiosos y económicos. Sin embargo, fue en la India donde se desarrolló un sistema para salvaguardar información con propósitos distintos.

La civilización india, hizo uno de los aportes más importantes al desarrollo de la criptografía: diseñó un sistema de encriptación para las mujeres, contenido en el Kama Sutra. Este libro abarca las 64 artes que las mujeres deben dominar, desde el arte de vestirse y cocinar, hasta habilidades como caminar con gracia, entre otros comportamientos fundamentales que forman parte de la vida cotidiana. Para efectos de este texto, interesa describir el número 45. En él se aludía a la escritura secreta; para ello, se presenta el alfabeto dividido en dos filas, omitiendo letras como la ñ y la ll. Cada fila contiene 13 letras las cuales están aparentemente en desorden. Cuando la mujer quería escribir algún mensaje, debía intercambiar cada letra del mensaje, por la letra que aparecía bien sea en la columna superior o inferior de donde aparecía la letra original. Se presenta en la Figura 4 una adaptación de este sistema, relacionado con nuestro alfabeto, tomado de la investigación realizada por Xifré (2009).

A	D	H	I	K	M	O	R	S	U	W	Y	Z
V	X	B	G	J	C	Q	L	N	E	F	P	T

Figura 4. Adaptación del cifrado encontrado en el Kama Sutra.

A manera de ejemplo se ilustra el siguiente texto:

Texto original: UNIVERSIDAD PEDAGÓGICA

Texto encriptado con el cifrado de Kama Sutra: ESGAULNGXVX YUXVIQIMV

Se puede decir que los sistemas desarrollados por estas culturas representan los primeros y más significativos avances en el ámbito del ocultamiento de información. Su contribución

inicial, sentó las bases para el desarrollo posterior de técnicas criptográficas. Estos métodos antiguos, ingeniosamente concebidos, forman una parte integral de la evolución de la criptografía a lo largo de la historia.

3.1.2 *Sistematización de los Métodos de Cifrado*

Con el paso del tiempo, se produjo una notable evolución en la sistematización de los métodos de cifrado. Entre las culturas que se destacaron en este proceso se encuentran la griega, la romana y la hebrea. Estos pueblos no solo refinaron las técnicas de ocultamiento de información, sino que también contribuyeron significativamente a la creación de sistemas más elaborados y robustos para la protección de datos sensibles.

Los espartanos, en el siglo *v* A.C., desarrollaron la denominada lacedemonia, llamada así en la obra vidas paralelas de Plutarco. Este método era un sistema de encriptación militar que consistía de una vara o un bastón redondo, en el cual se enrollaba una cinta de pergamino muy larga y estrecha. En este sistema (Figura 5), el mensaje se escribía de manera longitudinal en una cinta que luego se desenrollaba y enviaba. El receptor, necesitaba un bastón con el mismo diámetro y longitud para enrollar la cinta y poder leer el mensaje que había sido enviado (Xifré S., 2009).

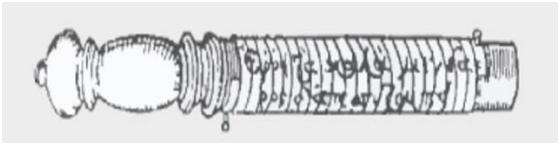


Figura 5. Lacedemonia Siglo V [8]

[8] Imagen sin autor tomada de: *Línea del tiempo-Criptografía Timeline.* Timetoast timelines. https://www.timetoast.com/timelines/criptografia-del-tiempo

En este sistema, el bastón –y en específico, el diámetro del bastón– jugaba un papel fundamental, ya que este era la clave para encriptar y desencriptar el mensaje; esta "llave" debía conocerla tanto el emisor como el receptor.

De igual manera, en Roma surgió el famoso algoritmo de cifrado de "César" para fines militares; este recibe su nombre gracias al emperador Julio César, quien en un intento por proteger sus secretos y estrategias tácticas desarrollo este sistema. Este método de encriptación consistía en que cada letra del mensaje original era remplazada por otra letra del mismo abecedario, pero 3 posiciones más adelante según el orden del abecedario (Maiorano, 2009).

Este algoritmo representó uno de los primeros en emplear una sustitución sencilla, utilizando su propio conjunto de símbolos y siguiendo un orden específico. Actualmente, estos procedimientos se clasifican como sustitución simple (Maiorano, 2009).

Mediante la Figura 3, se presenta una adaptación de la forma en la que se utilizaba este método de encriptación usando el alfabeto español.

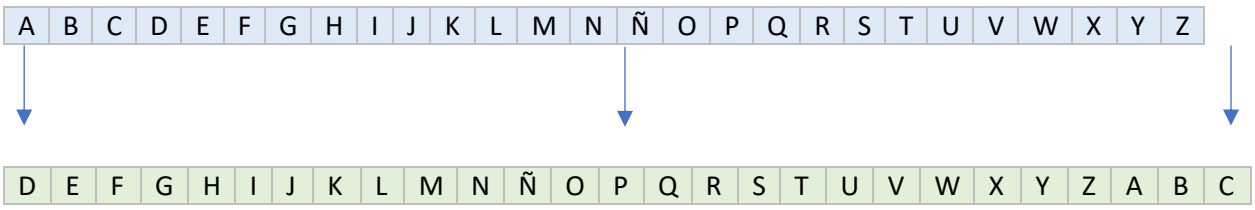
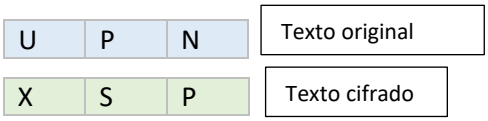


Figura 6. Adaptación del cifrado de César



Los hebreos, por su parte, desarrollaron sistemas de cifrado mono alfabético, que refiere a un sistema de encriptación donde cada letra del alfabeto se reemplaza por otra en específica de manera constante. En este tipo de cifrado, una letra determinada siempre se sustituye por la misma letra cifrada, lo que lo hace más susceptible al análisis de frecuencia y a técnicas de criptoanálisis. Los hebreos utilizaron este tipo de sistema para codificar y proteger la información, denominado como el Atbash, implementado en el siglo VI A. C. Este método era frecuentemente empleado en textos antiguos para ocultar palabras o mensajes de relevancia. Sin embargo, no se sabe con exactitud la razón por la cual tal encriptación, teniendo en cuenta que la mayoría de las personas de la época no sabían leer ni escribir.

El cifrado Atbash como se muestra en la Figura 7, implicaba permutar las letras del alfabeto hebreo mediante una sustitución específica. En este proceso, la primera letra del abecedario era reemplazada por la última, la segunda letra por la antepenúltima y así sucesivamente. Similar a este método se encontró el cifrado Albam, en el que la primera letra se intercambiaba con la duodécima, la segunda con la decimotercera, y así hasta la última siguiendo el mismo patrón.

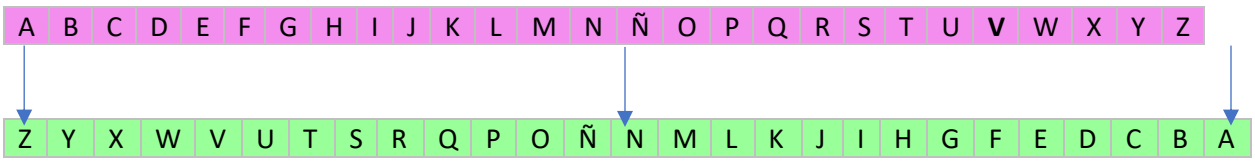


Figura 7. Adaptación del cifrado Atbash

Estos métodos usados por estas civilizaciones son conocidos como cifrados por sustitución o transposición, en donde los métodos por sustitución establecen una correspondencia entre los caracteres del alfabeto original en el que está escrito el mensaje y los elementos de otro conjunto que puede estar contenido dentro el mismo alfabeto. La transposición consiste en barajar los símbolos del mensaje original, alterando el orden en el que están dispuestos, de modo que el criptograma contiene los elementos del mismo sistema alfabético pero incomprensibles para quienes no conocen la permutación realizada.

A lo largo de la historia, estos métodos han perdurado y evolucionado, experimentando diversas modificaciones y adaptaciones a medida que el tiempo avanzaba. Así, han llegado hasta nuestros días, formando una parte integral del legado criptográfico que ha ido transformándose y perfeccionándose a lo largo de los siglos.

3.1.3 *Criptoanálisis Edad Media*

Durante los siglos VIII y XIII, el islam experimentó un periodo de esplendor conocido como su 'Época de Oro' o 'Renacimiento Islámico'. Durante este tiempo, se produjo un florecimiento significativo en áreas como la ciencia, la cultura, el arte y la filosofía en el mundo islámico. Gracias al desarrollo social y la buena comunicación entre gobernantes y el pueblo, los impuestos de esta sociedad eran relativamente bajos, todo esto estaba relacionado con la comunicación segura que se daba entre funcionarios y gobernantes.

Se protegían los archivos importantes y los registros de los impuestos mediante un sistema de transposición, en el que hacían variar las letras del alfabeto por símbolos tomados de

otros alfabetos; por ejemplo, la letra A se podía cambiar por el signo +, o la letra B podía ser representada con el símbolo #.

Durante este periodo, lo más destacado no fue simplemente la forma en que se cifraban los mensajes, sino la habilidad de los árabes para "destruir" los mismos, refiriéndose a la capacidad de descifrarlos sin conocer la clave. Este proceso dio paso a lo que se conoce como criptoanálisis, impulsando a diversas culturas a evolucionar en sus sistemas de encriptación (Xifré S., 2009).

Yusuf Yaqub ibn Ishaq al-Sabbah Al-Kindi, un sabio árabe del siglo IX publicó, en su manuscrito, una de las técnicas más usadas en esa época para descifrar mensajes; este método es conocido como análisis de frecuencias, y consistió en analizar la frecuencia con la que aparecen los símbolos en cada palabra, para predecir las posibles letras originales del mensaje encriptado. Así, si el alfabeto original es el español, se sabe que las letras que mayormente aparecen en cada palabra son las vocales ocupando un 45% del texto, y con más frecuencia las vocales a y e; por otro lado, las letras f, z, j, x, w, k, son las que menor porcentaje tienen alrededor de un 2%.

Basándose en suposiciones como estas y en los conocimientos lingüísticos de cada sistema de alfabeto, se podría en un tiempo determinado lograr descifrar el mensaje y más importante aún, descifrar la clave con la que se encriptan los mensajes, es una técnica basada en la estadística (Xifre, 2009).

En la Figura 8, se exhibe un análisis de frecuencia de las obras "Don Quijote de la Mancha" y "La Sombra del Viento", destacando que las vocales e, a y o son las letras de mayor aparición en ambos textos.

Don Quijote de la Mancha Miguel de Cervantes			*La sombra del viento* Carlos Ruiz Zafón			Análisis de frecuencias		
Letra	Veces	Frecuencia	Letra	Veces	Frecuencia	Letra	Veces	Frecuencia
e	33 757	13,89%	e	96 317	13,58%	e	130 074	13,66%
a	30 581	12,58%	a	95 718	13,50%	a	126 299	13,27%
o	24 276	9,99%	o	62 727	8,85%	o	87 003	9,14%
s	18 352	7,55%	s	47 231	6,66%	s	65 583	6,89%
n	15 899	6,54%	n	46 960	6,62%	n	62 859	6,60%
r	14 860	6,12%	r	46 559	6,57%	r	61 419	6,45%
l	13 662	5,62%	i	45 715	6,45%	i	58 442	6,14%
d	12 797	5,27%	l	41 142	5,80%	l	54 804	5,76%
i	12 727	5,24%	d	35 625	5,02%	d	48 422	5,09%
u	11 966	4,92%	u	32 417	4,57%	u	44 383	4,66%
t	9149	3,77%	c	27 280	3,85%	t	36 317	3,81%
c	8829	3,63%	t	27 168	3,83%	c	36 109	3,79%
m	6652	2,74%	m	21 937	3,09%	m	28 589	3,00%
p	5171	2,13%	p	17 342	2,45%	p	22 513	2,36%
q	4840	1,99%	b	13 302	1,88%	b	17 231	1,81%
b	3929	1,62%	q	9205	1,30%	q	14 045	1,48%
y	3572	1,47%	v	7767	1,10%	y	10 648	1,12%
h	2993	1,23%	h	7338	1,03%	v	10 417	1,09%
v	2650	1,09%	g	7144	1,01%	h	10 331	1,09%
g	2522	1,04%	y	7076	1,00%	g	9666	1,02%
j	1593	0,66%	f	4790	0,68%	j	6233	0,65%
f	1162	0,48%	j	4640	0,65%	f	5952	0,63%
z	1001	0,41%	z	2701	0,38%	z	3702	0,39%
x	57	0,02%	x	918	0,13%	x	975	0,10%
k	1	0,00%	k	48	0,01%	k	49	0,01%
w	0	0,00%	w	16	0,00%	w	16	0,00%

Figura 8. Tabla de frecuencias Relativas, en la lengua castellana[9].

Este enfoque de análisis de frecuencias se denomina criptoanálisis básico o clásico, y se fundamenta en el conocimiento del sistema lingüístico y la habilidad para llevar a cabo análisis estadísticos, con el fin de identificar las propiedades, frecuencias y comportamientos de los componentes en cada alfabético.

Durante varios años, el criptoanálisis básico fue la técnica predominante. Sin embargo, su efectividad disminuyó drásticamente durante la Segunda Guerra Mundial. Esto se debió a la implementación de sistemas criptográficos considerablemente más complejos y avanzados en comparación con los métodos básicos de criptoanálisis previamente utilizados. Fue en este contexto, que emergió el criptoanálisis moderno, aprovechando herramientas computacionales y algoritmos específicos para identificar y explotar las vulnerabilidades presentes en los sistemas criptográficos.

[9] Imagen tomada de: Arboledas D. (2017) Criptografía sin secretos con Python, pág. 33.

Algunas formas en las que funciona este sistema es el ataque de fuerza bruta y diccionario, que consiste en utilizar el potencial computacional para probar todas las combinaciones de posibles claves hasta encontrar la clave contraseña correcta, o el análisis de lado secreto, que se basa en información obtenida durante el proceso de cifrado, como tiempos de respuesta o consumos de energía, para inferir detalles sobre la clave.

Aunque el análisis por frecuencias fue un método eficaz para descifrar la información encriptada en esta época, en el renacimiento se diseñaron sistemas complejos de encriptación para hacerle frente al criptoanálisis básico.

3.1.4 *Criptografía Durante el Renacimiento*

En esta época y años posteriores surgen métodos de cifrado conocidos como silabarios, basado en un catálogo de silabas asociadas a símbolos, voces u otras silabas. También se desarrollaron los nomenclátores, catálogos de nombres en los que cada uno aparece asociado a la palabra que le sustituye en un texto cifrado.

Estos nomenclátores se convirtieron en herramientas ampliamente empleadas en épocas posteriores debido a su capacidad para generar ambigüedad en los análisis por frecuencia. Esta ambigüedad surge del hecho de que, en el proceso de sustitución homofónica, no se establece una correspondencia directa y única entre las letras del texto original y los múltiples símbolos o números utilizados para el reemplazo.

Esta característica añade una capa adicional de complejidad al cifrado, lo que dificulta significativamente los esfuerzos de criptoanálisis. Los análisis que dependen de la frecuencia de

letras o patrones en el texto original se ven obstaculizados, ya que no pueden identificar una relación clara entre las letras originales y sus representaciones cifradas (Xifre S., 2019).

Se presenta un ejemplo de nomenclátor en la Figura 9, adaptado al alfabeto español, para mostrar mejor como funcionó este método durante casi 450 años.

1. Se van a asignar un conjunto de números aleatorios a cada letra de nuestro reducido sistema lingüístico.
2. Se sustituyen las letras del mensaje en este caso la palabra “estudiante” por alguno de sus correspondientes números.
3. Nótese que a cada letra se le asigna una pareja de dígitos, la cual solo corresponde a esa letra.

Figura 9. Adaptación cifrada de un Nomenclator.

Estos sistemas, aunque innovadores, poseían varias desventajas:

- Si el catálogo de claves era demasiado pequeño, con el tiempo los analistas podrían descifrar la clave utilizada en el sistema de encriptación.
- Al establecer un sistema con un gran número de caracteres para mejorar la seguridad, se necesitaban libretas con claves para cada letra o combinación, lo que volvía el sistema menos práctico a la hora de descifrar mensajes.
- La posibilidad de que las libretas con las claves fueran robadas por atacantes representaba una vulnerabilidad seria, lo que podía comprometer por completo la seguridad del sistema.

Estos sistemas históricos, a pesar de sus desventajas, sentaron las bases para el desarrollo posterior de la criptografía y la búsqueda continua de soluciones más sólidas y efectivas en la protección de datos.

3.1.5 *Criptografía Moderna*

Con el primer mensaje por telégrafo en 1844 de Samuel F.B. Morse, surgieron las primeras compañías telegráficas que mostraban al mundo la posibilidad de enviar comunicaciones desde largas distancias, por lo que estas compañías decidieron implementar códigos para que menos caracteres. Cabe señalar que el código Morse en sí mismo no se considera una técnica de criptografía, ya que no implica ocultar el mensaje. Los puntos y las rayas en el código Morse simplemente representan un alfabeto alternativo utilizado en las comunicaciones telegráficas (Arboledas D., 2017). Así pues, los mensajes se convirtieron en nomenclátores, pero con códigos públicos.

La implementación del telégrafo, en el ámbito militar, trajo consigo inconvenientes para proteger la información que viajaba a través de las líneas telegráficas, implementar códigos y poseer manuales que contuvieran estas nomenclaturas, no era una opción viable ya que, si alguna base se rendía o era interceptada, el libro iba a caer en manos enemigas y comprometer la totalidad de las comunicaciones (Xifre S. 2019). Como consecuencia de esto surgieron sistemas de cifrado que usaban claves, y reglas de cifrado, como por ejemplo el cifrado Playfair, desarrollado por Charles Wheatstone, en 1854, el cual fue usado durante la primera y segunda guerra mundial, este sistema primero implementaba una forma de ordenar los componentes o símbolos del sistema lingüístico, la forma en que se ordenaban era en forma matricial, en la que se omitían signos de puntuación y espacios.

El Playfair consiste en primero, establecer una palabra clave, luego de esto se elige el tamaño de la matriz con la que se va a encriptar el mensaje, esta matriz debe ser de un tamaño tal que, sus elementos puedan ser completados con todas las letras del alfabeto a usar. Luego de esto se elige la posición en la que se va a ubicar la clave dentro de la matriz, generalmente se ubicaba en las primeras filas, seguido se completan todos los espacios con las letras del abecedario sin que se repitan teniendo en cuenta las letras de la clave. Una vez este diseñada la matriz, se dividen las palabras a encriptar en grupos de dos letras; si alguna palabra no es par, se completa con una letra elegida como nula; luego, se proceden a establecer las reglas que conforman el sistema. En el caso del Playfair son:

- Si las dos letras no están en la misma fila ni columna, se cambia cada letra por la que está en su misma fila, pero en la columna de la otra letra.

- Si las dos letras están en la misma fila, se sustituyen cada una por la que está a su derecha. En caso de ser la última de su fila, se reemplaza por la primera de dicha fila.
- Si las letras se localizan en la misma columna, se reemplaza cada una de ellas por la que está debajo. En caso de ser la última de su fila, se reemplaza por la primera de dicha fila.

La Figura 10, se presenta un pequeño ejemplo del funcionamiento de este sistema *Playfair*:

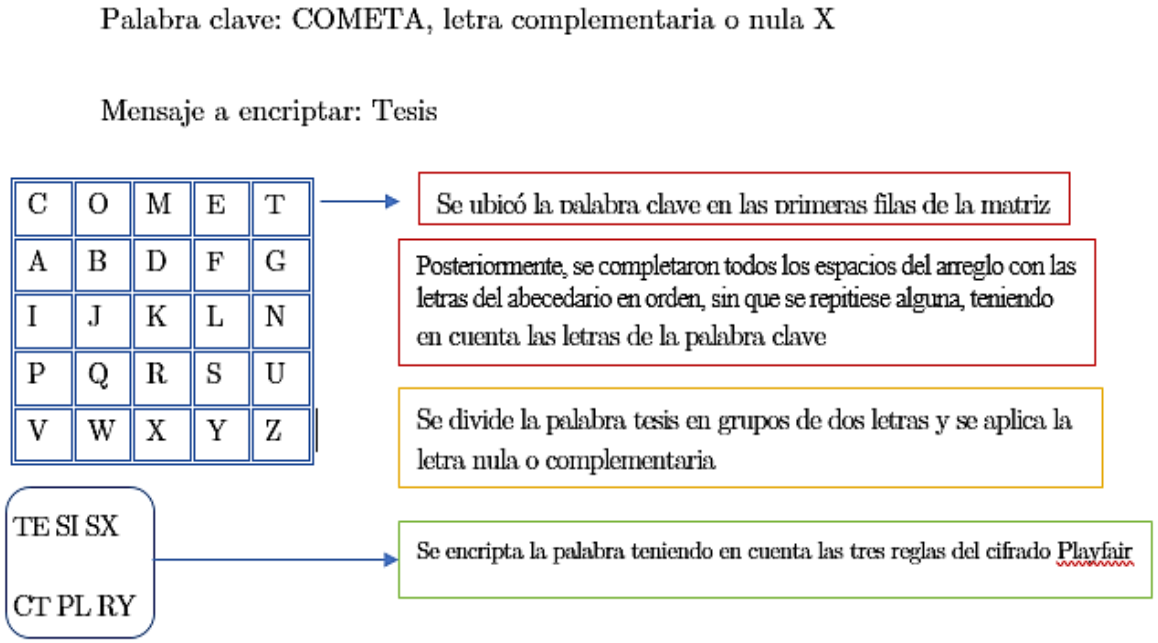


Figura 10. Adaptación del sistema Playfair.

Para descifrar el mensaje, únicamente se deben modificar las reglas dos y tres, en este caso se reemplazan las letras por las que están a la izquierda o arriba respectivamente.

Esta fue la base para que surgieran las primeras máquinas de cifrado como la maquina Enigma o la maquina Sigaba. Sin embargo, durante la segunda guerra mundial, Enigma constituyo

un papel fundamental no solo para el desarrollo y culminación de la guerra si no para la evolución de la criptografía y los ordenadores.

La máquina Enigma funcionaba por medio de un teclado y rotores, inicialmente tres, aunque se conocen algunas máquinas de hasta 20 rotores; el texto original pasaba por estos rotores que alteraban las letras del mensaje; luego, por un reflector que agregaba otra capa de encriptación; finalmente, se mostraba el mensaje encriptado en el papel de visualización de la máquina. Podría decirse que se hacía una sustitución poli alfabética ya que usaba distintos alfabetos, su "clave" era la posición inicial con la que se configuraban los diferentes rotores, ya que estos debían ser iguales tanto para encriptar y desencriptar el mensaje (Xifre S. 2019).

Durante la guerra, existieron varios intentos por descifrar los mensajes de enigma, y resultado de esto fue la evolución de la computación gracias a Alan Turing. Junto con sus colegas diseñó una maquina capaz de descifrar los códigos de enigma, y destruir el sistema de encriptación proporcionado por esta máquina.

Con el avance de la computación, se volvió imperativo adaptar la criptografía para su uso en los modernos ordenadores, garantizando mayor eficacia y seguridad. Esto se debió a que la información sujeta a encriptación empezó a abarcar un espectro más amplio, incluyendo no solo secretos de índole militar o político, sino también información personal. La creciente sistematización de empresas, sectores de salud pública, ámbitos económicos y otros, impulsó la necesidad de encriptar la información gestionada de diversas personas.

Es así como en 1976, se hace la primera publicación escrita por Whitfield Diffie y Martin Hellman, en la que se presentan las primaras nociones de la criptografía de clave pública o

criptografía asimétrica. La criptografía Asimétrica, básicamente consiste en que se encripta el mensaje utilizando el algoritmo de encriptación y una clave pública; para desencriptar el mensaje o la información, se usa esa clave privada que únicamente conoce el receptor.

Estas dos claves poseen características matemáticas especiales, y deben estar ligadas intrínsicamente, lo que quiere decir que si dos claves públicas son diferentes entonces sus claves privadas también los son (Moran et al., 2003).

Este método de cifrado con claves presenta dos inconvenientes presentes desde épocas de antaño, el primero es que al poseer dos claves que están ligadas intrínsicamente, y una de estas es de carácter "público", se puede obtener algún tipo de información a partir de esta para lograr encontrar la clave privada.

El segundo problema radica en que, para cambiar una de las claves se debe cambiar la otra también, esto desencadena en un problema de distribución de claves, Afortunadamente, existen estrategias probadas para superar este obstáculo. Una solución efectiva es la transmisión de claves a través de canales distintos y seguros, lo que añade una capa adicional de protección contra posibles interceptaciones. Además, la implementación de técnicas como el doble cifrado de claves proporciona una capa adicional de seguridad al proceso de distribución.

En este corto viaje a través de la historia de la encriptación, se ha explorado el ingenio humano en la búsqueda de formas de proteger la información valiosa. Desde las primeras técnicas de sustitución hasta la complejidad de la criptografía moderna, cada avance ha sido un hito en la evolución de la seguridad de la información. Estos logros no solo reflejan el afán por mantener la confidencialidad, sino también la constante adaptación a los desafíos tecnológicos y

estratégicos de cada época. Al comprender este recorrido histórico, se puede apreciar el papel vital que desempeña la encriptación en la protección de nuestros datos en el mundo digital actual y anticipar las innovaciones futuras en este campo.

Se ha obtenido una primera visión sobre la encriptación, y es hora de adentrarse en el elemento central de este trabajo: la imagen digital. Este componente esencial permitirá explorar de manera más detallada cómo se aplican los métodos de encriptación en el contexto de las imágenes digitales y qué implicaciones tiene en términos de seguridad y confidencialidad de la información visual. En la Figura 11 presenta la *Infografía: línea del tiempo - Historia de la Encriptación,* que resumen el apartado anterior*:*

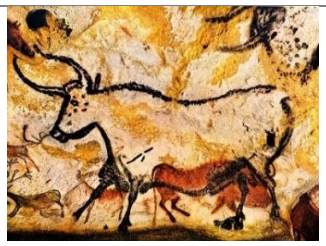
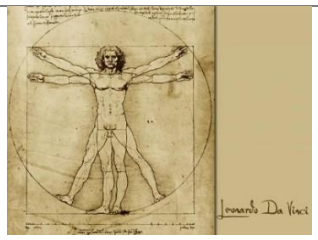
Figura 11. Infografía: línea del tiempo - Historia de la Encriptación[10].

[10] *Infografía: línea del tiempo- Historia de la Encriptación,* Elaboración propia.

3.2 Imagen Digital

Las primeras nociones sobre el concepto de imágenes en la historia de la humanidad se pueden remontar hasta hace 3000 años atrás, en el que el hombre hacía sus primeras impresiones en piedras de animales, comunidades, historias, comida, y otros tantos aspectos. Las imágenes de la Tabla 4, han sido esenciales en diversos ámbitos sociales, económicos, culturales, educativos y por supuesto artísticos (Saavedra, 2003).

Tabla 4. Figuras ejemplificadoras[11]

Pintura Rupestre	***El hombre viturbio***	***11 de septiembre 2001***
		

La fotografía, desarrollada a principios del siglo XIX, marcó una transformación en la percepción de las sociedades; el hecho de poder materializar una imagen en papel permitió que las personas observaran diferentes mundos. Las impresiones gráficas se popularizaron alrededor del

[11] Imágenes tomadas de: *" Las etapas del arte a lo largo de nuestra historia humana" Carolina Borrero Timeline. Timetoast timelines. https://www.timetoast.com/timelines/las-etapas-del-arte-a-lo-largo-de-nuestra-historia-humana*

planeta y se volvieron tan comunes como los libros o las películas cinematográficas (Saavedra, 2003).

Por su parte, la imagen digital tiene su origen hacia el año 1957, gracias a la creación del llamado píxel por parte del informático estadounidense Russel Kirsch (1929-2020). En la Figura 12, se muestra la primera imagen que Kirsch produjo fue la de su hijo cuando era un bebé, décadas antes de que existieran las cámaras digitales (Fabro, 2020).

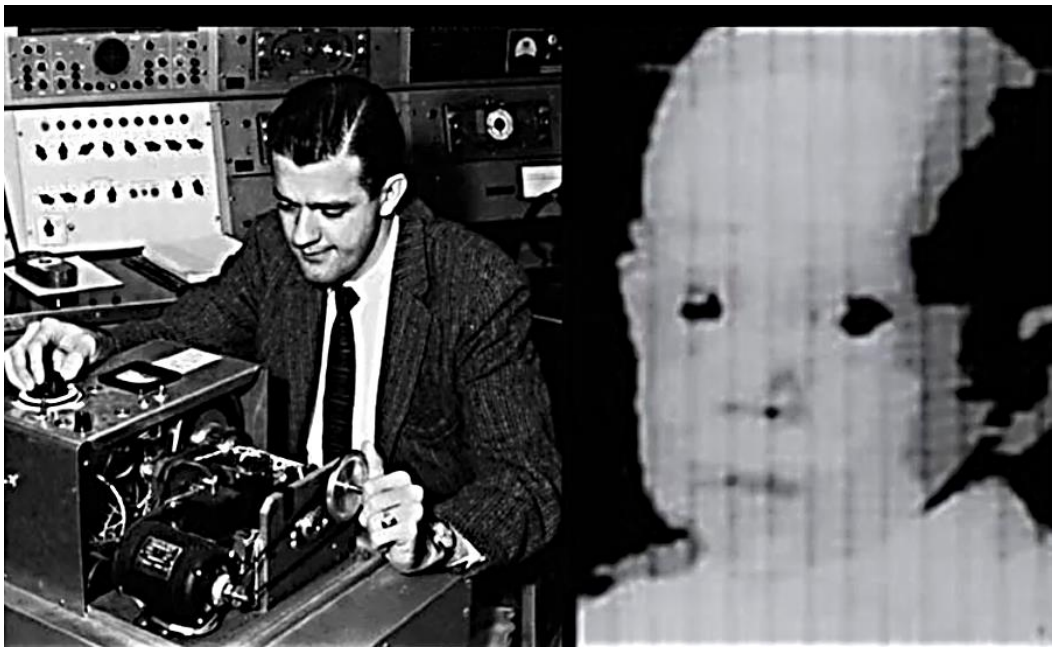


Figura 12. Primera imagen digital[12]

Este logro en la historia de la imagen digital marcó el inicio de una revolución visual que transformó la manera en que se captura, comparte y almacena imágenes en la era moderna. La creación del píxel allanó el camino para la tecnología de imágenes digitales que se usan hoy en

[12] Imagen tomada de Fabro. 2020, "*muere Russel Kirsch, creador del píxel y la primera imagen digital*"

día. Por esta razón, resulta crucial definir qué es exactamente un pixel y como este define lo que es una imagen digital.

3.2.1 *Píxel*

Proveniente del inglés *picture element*. Es una unidad de color que forma parte de una imagen digital; el píxel no posee una medida concreta es simplemente la división por medio de casillas que se hace a una imagen, una matriz. Cuando una imagen tiene x cantidad de pixeles, significa que tiene a pixeles de ancho multiplicado por b pixeles de alto ($x = a * b$). Podría decirse que un píxel posee dos componentes a y b; sin embargo, el pixel posee una profundidad que es donde se almacena el color de la imagen, por lo tanto, a mayor cantidad de pixeles mayor resolución de la imagen (De la fraga, 2001).

Teniendo como base esta definición de píxel, se puede decir que una **imagen digital** se define como una función bidimensional de intensidad de luz, $f(a, b)$ donde a y b (a es el ancho y b el alto de la imagen) denotan las coordenadas del valor de intensidad de luz en cualquier punto; es decir, una imagen digital puede considerarse como una matriz cuyos componentes son los pixeles de coordenadas (a, b). (De la fraga, 2001).

Con el fin de aclarar la definición anterior, se pone en consideración los siguientes aspectos:

- Dominio: El dominio de la función $f(a, b)$ está dado por el conjunto de todas las posibles combinaciones de valores para a (ancho) y b (alto) que representan las coordenadas dentro de la imagen digital. Es decir, el dominio estaría determinado por todos los pares de valores enteros no negativos que representan las posiciones dentro de la imagen.

- Conjunto imagen: El conjunto imagen de la función $f(a, b)$ se refiere al conjunto de todos los posibles valores de intensidad de luz que pueden tomar los píxeles en la imagen digital. Esto generalmente está limitado por la resolución de la imagen y el rango de valores que pueden ser representados, como por ejemplo, en una imagen en escala de grises serían valores entre 0 (negro) y 255 (blanco).
- Relación de correspondencia: La función $f(a, b)$ asigna a cada par de coordenadas (a, b) un valor específico que representa la intensidad de luz en ese punto de la imagen. Establece la relación entre las coordenadas (ancho y alto) y los valores de intensidad lumínica para cada píxel en la imagen digital, permitiendo su representación como una matriz de píxeles.

Con base en esta definición, es relevante comprender cómo los procesadores reciben e interpretan las imágenes digitales y el proceso con el que se hacen modificaciones en estas. Estos procedimientos son conocidos como la digitalización y procesamiento de imágenes digitales.

3.2.2. Digitalización de una imagen. Se entenderá como el proceso mediante el cual una imagen digital se recibe y comprende en un ordenador, bajo cierto formato. La imagen digital se puede obtener por medio de diferentes sistemas como cámaras digitales, diseños en softwares, etc.

3.2.3. El procesamiento digital de imágenes. Consiste en la manipulación, por medio de diferentes tipos de software, en el que se toma la imagen almacenada bajo el tipo de formato, y se divide en un arreglo matricial rectangular, compuesto por los pixeles; dependiendo del objetivo que se quiera desarrollar, los pixeles son manipulados. El procesamiento digital de imágenes

ofrece utilidades diversas y significativas en distintos campos. En medicina, facilita la visualización y análisis de radiografías, mientras que en la exploración del espacio, permite la interpretación de imágenes transmitidas por sondas espaciales. Además, en áreas como la topografía y la cartografía, resulta esencial para la medición y evaluación precisa de terrenos. Asimismo, despliega un rol crucial en la encriptación y seguridad de imágenes, entre otras aplicaciones relevantes (Hidalgo et al., 2009).

Uno de los factores más importantes con los que se realiza el procesamiento de imágenes digitales es el color, ya que este atributo es fundamental para la representación y percepción de la información visual. El manejo preciso y la manipulación efectiva del color en una imagen pueden tener un impacto significativo en su calidad, legibilidad y estética. Por tanto, comprender cómo se gestionan y ajustan los colores en el procesamiento de imágenes digitales es esencial para obtener resultados óptimos.

3.2.4. *El Color*

De acuerdo con el diccionario de la real academia española, color es una sensación producida por los rayos luminosos que impresionan los órganos visuales y que depende de la longitud de onda. Otra definición de color está en la investigación de Pérez (2009) en la que se describe que la sensación de color, se entiende como un fenómeno fisicoquímico asociado a las infinitas combinaciones de la luz, que se entrelazan mediante ondas atrapadas por los órganos visuales de humanos y animales.

Estas definiciones convergen en la Teoría Tricromática, la cual postula que el sistema visual humano se fundamenta en tres receptores primarios de color: rojo, verde y azul. Esta teoría

sostiene que el ojo humano interpreta los estímulos visuales a partir de estos tres colores, independientemente de la cantidad de ondas o radiaciones electromagnéticas. Sin embargo, la razón por la cual el ojo humano percibe más “colores”, tiene que ver con el tono, la saturación o intensidad, y la luminosidad.

El tono se refiere a lo que, en sí se percibe como el color, es la combinación de los colores primarios rojo, verde y azul; dependiendo del grado de combinación de estos se forman los distintos tonos. La saturación se refiere a la pureza del color, el nivel de gris presente en cada tono. La luminosidad es el nivel de luz existente en cada tono, dependiendo del nivel de luminosidad se dice que el color está más cerca del blanco o del negro (Pérez,2009).

El sustento de le teoría de colimetría es la adición de estos tres colores, llamados colores primarios, en lo que respecta al tono blanco es la combinación de todos los colores y, el tono negro es la ausencia de luminosidad y por lo tanto ausencia de tonos.

3.2.5. Espacio de Color RGB

Un espacio de color se define como la representación de un tono específico, permitiendo la comprensión y la interpretación de la información visual contenida en una imagen. El espacio de color RGB, derivado de la teoría tricromática, recibe su nombre de las iniciales en inglés de sus tres colores primarios: rojo (Red), verde (Green) y azul (Blue).

Este espacio de color es uno de los más utilizados para ordenadores, televisores y cámaras digitales, ya que está modelado por un cubo unitario, en el cual por adición de los colores primarios se forman las diferentes tonalidades. En la Figura 13 se visualiza este concepto:

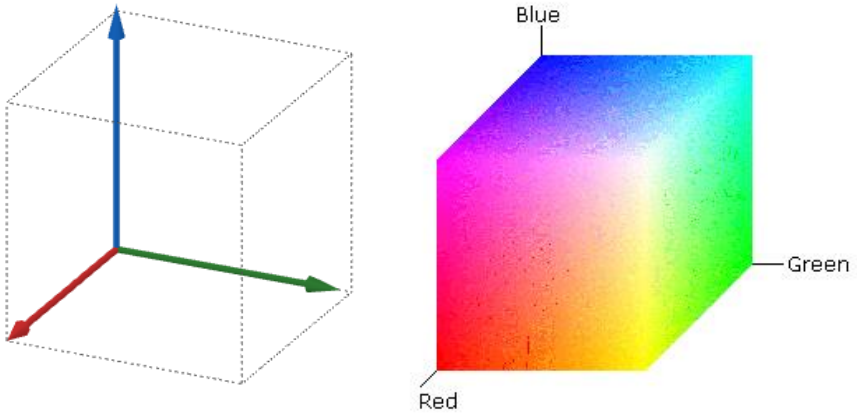


Figura 13. Cubo unitario RGB[13]

En este espacio, todos los valores se encuentran restringidos al intervalo de cero a uno, por lo tanto, el color negro es (0,0,0) ausencia de luminosidad o de color y el tono blanco será (1.0,1.0,1.0), la escala de grises se encuentra en la diagonal con origen en el color negro y extremo el color blanco. En las imágenes digitales estos valores de R, G y B son números enteros que van de 0 a 255.

3.2.6 Imagen de color RGB. De acuerdo con la investigación de Alonso M. (2009), este tipo de imágenes digitales son un arreglo de tres imágenes monocromáticas independientes, de tamaño $m * n$ correspondientes a la escala de rojos, verdes y azules.

Esta definición empalma completamente con la definición de imagen digital, presentada en la sección 3.2.1, por ende, una imagen monocromática es una función bidimensional de la

[13] *Imagen tomada de sitio oficial Microsoft Online*

intensidad de luz, es decir una matriz de tamaño $m * n$ cuyos elementos $f(x, y)$ son los pixeles. En la Figura 14 se visualiza el Logo Licenciatura en Matemáticas en versión RGB.

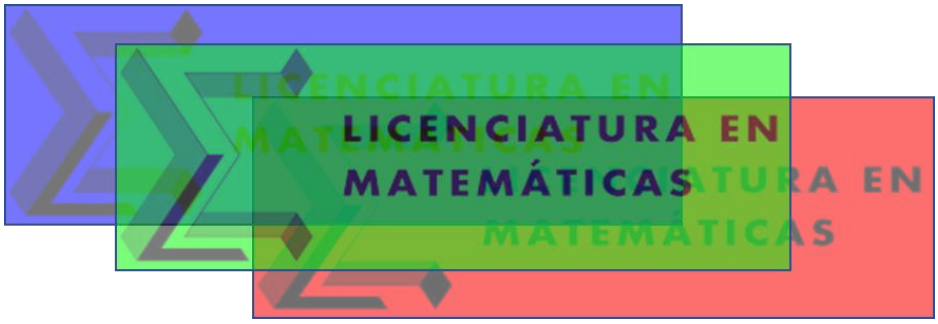


Figura 14. Logo Licenciatura en Matemáticas RGB[14]

Los elementos previamente mencionados conforman la noción de imagen digital. Hasta este punto, hemos abordado dos de los tres pilares esenciales que constituyen el fundamento de este trabajo: encriptación e imagen digital. El tercer pilar persigue establecer una conexión significativa entre estas bases, que, se centra en los métodos de encriptación. Para llevar a cabo este proceso de manera efectiva, es crucial adquirir conocimiento profundo de algunos métodos desarrollados y las herramientas apropiadas para su implementación.

3.3 Métodos de encriptación usuales

En esta sección, se expondrán dos de los métodos de encriptación más utilizados. Se brindará una explicación detallada de cada método, incluyendo las bases matemáticas subyacentes. Es importante destacar que el núcleo de esta técnica radica en la conversión de una imagen en

[14] Imagen adaptada de: licenciatura en matemáticas Universidad pedagógica nacional, sitio oficial: http://cienciay-tecnologia.upn.edu.co/wp-content/uploads/2023/04/Encabezadi-Licenciatura1-1.png

una matriz numérica, sobre la cual se aplican estos métodos con el propósito de cifrar el contenido visual.

3.3.1 *Método de Hill*

El matemático y educador estadounidense Lester Hill (1891-1961), quien se interesaba en la aplicación de las matemáticas a las comunicaciones, lo llamó este método basado en el algebra lineal y la aritmética modular.

Este avance tecnológico, fue patentado y publicado sin éxito en la revista *The American Mathematical Monthly, volumen 36.* Cabe señalar que este sistema, se pensó para encriptar mensajes textuales, por eso se dice que es un sistema de sustitución simple, poli alfabético, involucrando varios alfabetos y sustituyendo los caracteres del alfabeto original.

El cifrado consiste en, primeramente, establecer una correspondencia entre el alfabeto y los números enteros, y mediante una organización matricial $n * n$ del texto a encriptar, cada carácter es reemplazado por su correspondiente dígito; luego se efectúa una multiplicación entre matrices, donde una de estas es el texto a encriptar y la otra es la llamada “llave” o “clave” del sistema. Para desencriptar el texto se utiliza la misma llave y la clave es la matriz inversa de la clave con que se encriptó.

A continuación, se presenta una ilustración del método creado por Hill, el cual será la base para la elaboración de los algoritmos de encriptación:

1. Se realiza una asociación entre los caracteres a encriptar y los números enteros, en nuestro caso los caracteres a encriptar son los pixeles de la imagen digital.

2. Suponer que en la siguiente Figura 15, se representa una imagen digital RGB de $4 * 3$ pixeles, en las que la profundidad de los pixeles para este ejemplo no se tendrá en cuenta; se supondrá que se está representando una imagen en escala de grises. Cada pixel se representa con la letra p y su respectiva posición en la imagen.

$p_{1,1}$	$p_{1,2}$	$p_{1,3}$
$p_{2,1}$	$p_{2,2}$	$p_{2,3}$
$p_{3,1}$	$p_{3,2}$	$p_{3,3}$
$p_{4,1}$	$p_{4,2}$	$p_{4,3}$

Figura 15. Representación imagen digital en matriz

Estableciendo la relación entre los pixeles de la imagen y los números enteros en forma matricial se obtiene:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

1. Cabe resaltar que cada valor de esta matriz resultante no representa en sí los pixeles que componen la imagen, es solo una suposición que se hace para explicitar el funcionamiento del método, ya que una imagen en RGB podría decirse que es una matriz en tres dimensiones

donde varían los colores del RGB.Se procede a realizar un cambio de base en cada componente de la matriz. Para este ejemplo, se supondrá que el módulo es $m = 10$; sin embargo, para el encriptado de imágenes RGB, se utiliza modulo 256 ya que es el valor que posee cada tono rojo, verde y azul. Se elije la llave con la cual se va a realizar la encriptación, esta llave debe ser una matriz B que sea invertible en $\mathbb{Z}$ módulo m, y que además se pueda multiplicar con A.

Para este caso utilizaremos la matriz B_m

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 5 & 4 \\ 3 & 6 & 5 \end{pmatrix}$$

2. Se efectúa el producto $A * B = C$, donde C es la matriz encriptada, es el mensaje encriptado, esta matriz C debe se modulo m

$$C_m = \begin{pmatrix} 16 & 29 & 24 \\ 37 & 65 & 54 \\ 58 & 101 & 84 \\ 79 & 137 & 114 \end{pmatrix}$$

Nota: En algunos casos, dependiendo del módulo que se esté manejando y el tipo de mensaje que se quiera encriptar (texto o imagen) se trabaja a B, B^{-1} y C, en términos de unidades y con componentes positivos.

Desencriptación del mensaje: Básicamente es el mismo proceso explicitado anteriormente, aquí se resalta la importancia de que la matriz B sea invertible en $\mathbb{Z}$ modulo m, ya que la lleva de desencriptado es B^{-1} (matriz inversa de B), es decir que $C * B^{-1} = A$

$$B^{-1} = \begin{pmatrix} 1 & 1 & -1 \\ -3 & 2 & -1 \\ 3 & 3 & 2 \end{pmatrix}$$

Este método resulta bastante efectivo y de fácil aplicación cuando se trabaja con matrices de un orden relativamente pequeño, pero en las imágenes digitales las matrices y vectores que se forman con el RGB son de dimensiones grandes, este problema se estudiará en posteriores capítulos donde se analice y se desarrolle la propuesta.

Otra dificultad de aplicación de este método que parece surgir a simple vista recae con la clave, ya que no puede ser cualquier matriz y para el trabajo con matrices de tamaños mucho más grandes que las del ejemplo mostrado, el cálculo de estas multiplicaciones y el cálculo de su inversa se vuelve tedioso de efectuar. Una posible solución sería generar un algoritmo que permita generar matrices pseudoaleatorias para la clave, en las que se cumplan las condiciones que se estipulan en este método.

A modo de ejemplo del proceso previamente descrito, en la Figura 16 se presenta la encriptación de la imagen "Lena", comúnmente empleada en pruebas de cifrado de imágenes. Esta acción fue llevada a cabo por Rojas A. & Cano A. en su investigación titulada "Cifrado de imágenes y matemáticas"

Figura 1: Imagen original

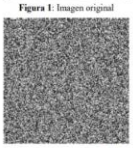


Figura 2. Imagen cifrada con el método de Hill

Figura 16 . Cifrado de Hill, Lena.[15]

Nota. Una vez entendido este método, se propone al lector la siguiente la pregunta la pregunta: ¿Toda matriz sirve como llave para encriptar y desencriptar la imagen original?

3.3.2 Descomposición en Valores Singulares (SVD)

Es un proceso de factorización matricial, más conocida por sus siglas del inglés SVD (singular valúes decomposition). El primer matemático en trabajar con este procedimiento fue el británico Alan Turing, durante la segunda guerra mundial, sin embargo, no nombro formalmente este método, más adelante matemáticos como Eugenio Beltrami, Irving Segal y William Karush descubrieron el método de forma independiente y lo formalizaron bajo este nombre.

El SVD es una técnica que consiste en descomponer una matriz de dimensiones $m \times n$ en el producto de tres componentes principales: una matriz $U_{m\times m}$, una matriz diagonal Σ de dimensiones $m \times n$ y una matriz $V_{n\times n}^{T}$

$$A = U\Sigma V^{T}$$

De donde U y V son matrices ortogonales, Σ es una matriz diagonal conformada por los valores singulares no nulos de A, de aquí el nombre del método.

Recordemos que los valores singulares de una matriz A son las raíces cuadradas de los valores propios (autovalores) de la matriz $M = A^{T}A$, y usualmente se denotan estos valores con $\sigma_1 \dots \sigma_n$ en donde por convención se organizan los valores singulares de modo que $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_n$ (Burden j., Douglas J, 1985).

[15] Imagen tomada de: Rojas A., & Cano A., Cifrado de imágenes y matemáticas, pag 2.

Para calcular los autovalores se utiliza la definición de autovalor de una matriz, sea una matriz A cuadrada, se dice que λ es un autovalor de la matriz si existe un vector $v \neq 0$ tal que $Av = \lambda v$, v se conoce como el auto vector. A tendrá tantos autovalores como dimensiones tenga A. Para calcular los valores de λ se utiliza lo que se denomina polinomio característico $|A - \lambda I| = 0$ (Molina et at., 2019).

Para obtener los componentes de la matriz V, se deben obtener primero los subespacios propios asociados a los autovalores de M que se pueden encontrar fácilmente usando $(M - \lambda I)v = 0$ y eliminación Gaussiana. Luego se determina el módulo de cada subespacio, el cual va a dividir a cada componente del subespacio propio, para obtener las columnas de la matriz V.

Los valores de la matriz U se obtienen a partir de la expresión $\frac{1}{\sigma_1} A \begin{pmatrix} a \\ b \\ c \end{pmatrix}$, donde $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ representa las columnas de la matriz V, es decir los componentes de la matriz U van a variar a medida que σ y $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$ varíen respectivamente.

El método SVD es una técnica matemática utilizada para descomponer una matriz en tres componentes: $U\Sigma V^T$ que abarca diferentes conceptos relacionados con el álgebra matricial. Aunque es un proceso complejo, algunos programas y plataformas contienen esta función ya incorporada, lo que facilita su uso sin necesidad de calcular cada matriz $U\Sigma V^T$ de manera manual.

Es importante señalar que, aunque algunas plataformas contienen la función SVD de manera incorporada, es esencial entender los conceptos subyacentes para poder aplicarla correctamente en contextos específicos y evaluar su robustez en cada aplicación en particular.

En el encriptado de imágenes digitales usualmente se utiliza este método descomponiendo la imagen en los factores y manipulando las matrices U, Σ, V^T. Para encriptar se usa una clave de carácter matricial que mediante una operación con una o unas de estas matrices oculte la información de forma adecuada. Para desencriptar se usa la misma clave y el mismo proceso SVD sobre la imagen encriptada. Actualmente, el cifrado de imágenes es importante para asegurar la seguridad digital, y la encriptación basada en SVD ha demostrado ser una técnica útil y efectiva en muchos casos (Molina et at., 2019).

Ya se han presentado dos de los métodos más usuales usados en la encriptación y el tratamiento de imágenes digitales. Ahora compete presentar un análisis sobre la herramienta más idónea para el desarrollo de este trabajo, analizando diferentes factores como la facilidad de uso, la accesibilidad y la vinculación con los objetivos propuestos.

3.4 Elección de Software de Programación

En esta sección, se presenta un análisis para determinar la herramienta más adecuada para encriptar imágenes digitales, con los métodos de encriptación mencionados. La elección del software se fundamenta en un documento guía tomado de la Agencia de Selección de Gobierno Electrónico y Sociedad de la Información y del Conocimiento (AGESIC) del Gobierno de Uruguay.

3.4.1 *Identificación de los posibles candidatos*

Para identificar a posibles candidatos, se realizó una búsqueda en internet de software que permitiera procesar imágenes digitales con el álgebra lineal y el desarrollo de algoritmos, los posibles candidatos arrojados por la búsqueda se enumeran a continuación, donde el orden de listado fue aleatorio y no representan ventajas.

a. Matlab
b. Mathematica
c. Open cv
d. Octave
e. Python

Es evidente que se omiten muchos más programas y lenguajes en este primer barrido, sin embrago se sitúan los más conocidos y de los que se tiene algún conocimiento o que poseen amplia información sobre su uso.

3.4.2 *Comparación de Atributos*

Este proceso de comparación de software implica la evaluación de varios criterios, como la confiabilidad, funcionalidad, costos y beneficios, funcionalidad en línea y usabilidad de cada software seleccionado. Para evaluarlo, se hace una investigación detallada en los sitios web oficiales de cada software, sus comunidades, blogs y foros de discusión de su página oficial. De esta manera, se obtiene información sobre cada software, lo que permite una comparación detallada basada en criterios específicos.

Matlab. Es uno de los programas más completos, ya que abarca una amplia gama de funciones encaminadas hacia el diseño de sistema de control, simulación y despliegue de redes neuronales, procesamiento de imágenes y visión artificial, desarrollo e implementación de software predictivo, robótica, procesamiento de señales, entre otras aplicaciones más, además maneja un lenguaje computacional basado en matrices. Actualmente este programa cuenta con varias licencias que se pueden adquirir online, y su costo para uso estándar ronda los USD940, $4.350.000 por año.

Al adquirir una licencia bien sea por un año o indefinidamente, Matlab ofrece un servicio de soporte técnico, servicio de mantenimiento de software, tutoriales, comunidad online y servicio de soporte.

Matlab tiene una versión online que ofrece un espacio de almacenamiento en la nube, con posibilidad de colaborar en tiempo real y facilidad para compartir archivos, esta versión se puede adquirir comprando la licencia, sea estándar o corporativo.

La usabilidad de Matlab resulta simple de entender, gracias a la abundante información tanto en su sitio oficial, como en diferentes canales de YouTube dedicados a enseñar las diferentes herramientas del sistema.

Wolfram Mathematica**.** Es un software con múltiples funcionalidades, que se desarrolla en diferentes áreas como las matemáticas, la física e ingeniería, bilogía y medicina, las finanzas y la economía. Entre sus variadas herramientas permite la resolución de ecuaciones, el cálculo simbólico, la creación de simulaciones, la dinámica de fluidos, la exploración de lenguajes formales y teoría de autómatas, el procesamiento de imágenes digitales, y otras más.

Wólfram es un software comercial, al cual todas las personas pueden acceder por un costo de USD295 aproximadamente, claro que existen descuentos para estudiantes vinculados con entidades asociadas a Wólfram Research. Cuenta con una versión gratuita online, la cual no posee una interfaz gráfica y para algunos procesos debe integrarse con otros programas.

Wólfram cuanta con una sólida reputación, lo que permite al usuario tener fiabilidad y seguridad de los productos de la compañía, por otro lado, la usabilidad es un poco menor en comparación con otros softwares, ya que la gran mayoría de recursos relacionados con la funcionalidad del programa se adquieren con la compra de la licencia, sin embargo, la comunidad "Wólfram community" es gratuita y abierta a cualquier persona.

Open CV**.** este es un software de código abierto y totalmente gratis, su mayor funcionalidad es el procesamiento de imágenes digitales en tiempo real, como por ejemplo la visión por computadora en la detección de anomalías, o la navegación autónoma de robots, sin embargo, se usa también en medicina para el análisis de imágenes médicas, en seguridad para la detección de incendios, el monitoreo de cámaras de seguridad, entre muchos otros usos encaminados en el desarrollo de aplicaciones de IA (Inteligencia Artificial) y el procesamiento de imágenes digitales.

Open CV, no cuenta con un sitio web en línea para su ejecución, ya que debe descargarse e instalarse en el dispositivo local para su uso. Aunque puede ser compatible con diferentes versiones de sistemas operativos, no ofrece soporte comercial directo debido a la falta de una presencia en línea estable. Sin embargo, existen comunidades en línea dedicadas a Open CV, que ofrecen soluciones rápidas a los usuarios, especialmente en caso de problemas técnicos. Además,

existe una documentación detallada en línea disponible en el sitio web oficial que brinda información completa sobre la instalación, configuración y uso del software.

Octave. Es un programa y lenguaje de programación alto nivel, está enfocado hacia el procesamiento de señales y la realización de cálculos numéricos. Su uso es compatible con Matlab y cuenta con una interfaz cómoda para resolver problemas lineales y no lineales.

Por esta razón es muy usado en áreas como la ingeniería en el diseño y análisis de sistemas y la modelación de objetos, se usa para la creación de modelos financieros y de inversión, en la educación se usa para el aprendizaje de las matemáticas desde la educación básica hasta la superior.

Octave es una herramienta de código abierto y gratis, el cual es mantenido y actualizado por la comunidad bajo ciertos términos establecidos en su licencia, octave cuenta con una versión online (Octave Online), la cual es de acceso gratis, esta permite el desarrollo de algoritmos sin necesidad de descargar librerías, al contrario de su versión de escritorio GNU octave, sin embargo, su versión online requiere una conexión estable a internet, existen algunas variaciones mínimas en cuanto al lenguaje de programación en sus dos versiones.

El sitio oficial de GNU Octave ofrece información detallada sobre las características y funcionalidades del software, manuales y documentación técnica sobre cómo usarlo y aprovechar sus capacidades, así como actualizaciones y noticias relacionadas con el proyecto GNU.

Python**.** Es uno de los programas más conocidos y usados en las últimas décadas; cuenta con una sintaxis relativamente sencilla de usar, con un lenguaje de programación interpretado, lo que significa que el código se ejecuta línea por línea identificando posibles errores.

Es muy usado en el desarrollo de software y aplicaciones web; también se usa en la ciencia, gracias a sus múltiples bibliotecas enfocadas en el análisis y procesamiento de datos, es uno de los programas que se usan en el desarrollo de juegos y gráficos, ya que permite crear simulaciones por medio de bibliotecas como Pygame.

Este software es de código abierto y es completamente gratuito; sin embargo, no cuenta con una versión online directamente, aunque existen blogs y otros tipos de páginas en los que se puede copiar y ejecutar el código desarrollado en la versión de escritorio de Python.

El sitio oficial ofrece acceso gratuito a sus grupos de discusión y comunidad Python, en la que se ofrecen la mayor parte de las soluciones a problemas de soporte técnico de los usuarios; también cuenta con serie de conferencias y talleres que ayudan a la comunidad al aprendizaje del programa, además de videos tutoriales enlazados con la plataforma YouTube, que contribuyen a la enseñanza de las distintas herramientas de este programa.

3.4.3 ***Análisis de Idoneidad del Software de Programación***

Después de haber examinado con detenimiento las características principales de cada software, se ha determinado que la usabilidad y confiabilidad son similares entre las opciones disponibles. Por lo tanto, en la elección del software, estos atributos no se considerarán como factores decisivos. Sin embargo, el costo y la funcionalidad en línea son dos características que tienen un impacto significativo en la toma de decisiones en cualquier punto de este proyecto. Por lo tanto, se concederá una atención especial a cómo el costo y la funcionalidad en línea de cada opción afectan a la elección del software.

La funcionalidad en línea del software permitirá, enfocar el desarrollo de los objetivos hacia la creación de un sitio web, que permita la encriptación y desencriptación de imágenes digitales. En relación con el costo del software, se considera elevado y no se justifica el valor para el desarrollo de esta primera investigación, de licenciatura.

En la Tabla 5 se califica cada software investigado. Se puntúa cada programa de 1 a 5 siendo 1 lo menos favorable para la investigación y 5 lo más favorable.

Tabla 5. Calificación del software de programación

Software	Costo	V. Online	Usabilidad	Confiabilidad	Total
Matlab	1	3	5	5	14
Mathematica	1	2	4	5	12
Open cv	5	1	4	3	13
Octave	5	4	4	4	17
Python	5	1	4	4	14

Nota. Cabe aclarar que la información de esta tabla es de carácter subjetivo y enfocado con los objetivos de esta investigación.

3.4.4 *Conclusiones*

De acuerdo con los datos proporcionados en las secciones anteriores, es notable que el software Octave con la puntación más alta es el más recomendado para el desarrollo de la investigación, en su versión online.

Es importante destacar que, aunque Matlab y Python obtuvieron calificaciones similares durante la evaluación, se recomienda Python como segunda opción debido a que se considera

una alternativa más accesible en comparación con Matlab, ya que Python no requiere una licencia costosa para su uso.

La accesibilidad de una versión ejecutable en línea permite el desarrollo del sitio web. Por esta razón, se concluyó que Octave Online era el software más viable para la propuesta, dado que integra bibliotecas en su versión en línea, lo cual representa una ventaja significativa, al no requerir la descarga de extensiones adicionales.

CAPÍTULO 4. Desarrollo y Descripción de la Programación.

Tras explicar los fundamentos teóricos que respaldan la funcionalidad de los programas, se abordará cómo se implementan los algoritmos en Octave Online.

En primer lugar, el comando `imread`, importa una imagen y la convierte en un arreglo tridimensional compuesto por tres matrices correspondientes a los colores primarios: rojo, verde y azul (RGB). Estos colores son esenciales en la composición de todas las tonalidades presentadas en una pantalla de las imágenes digitales. Los valores en estas matrices son números escalares que van desde 0 hasta 255, representando la intensidad de cada color en el píxel correspondiente en la imagen. El valor de 255 denota la intensidad máxima en el píxel, mientras que 0 representa la ausencia total de color. En el caso del color negro, las tres matrices tienen un valor de 0, mientras que para el blanco todas las matrices tienen un valor de 255.

Con base en lo anterior, la función *double* se empleó en los códigos *'Encriptación por SVD*' y '*Creación Propia*' para convertir una imagen de tipo de dato *uint8* a tipo de dato *double*. Esto es común en el procesamiento de imágenes y tiene una implicación importante en la representación de los valores de píxeles de la imagen. Aquí hay una explicación de lo que sucede:

Tipo de dato uint8. Las imágenes en la mayoría de los casos se almacenan en matrices donde cada elemento de ella representa un valor de píxel en la imagen. En el caso de una imagen con formato uint8, cada valor de píxel se representa utilizando 8 bits (1 byte). Esto significa que los valores de píxeles están en el rango de 0 a 255, ya que 8 bits pueden representar 2^8 = 256 valores diferentes (Elizondo & Maestre, 2005).

Tipo de dato double. Por otro lado, *double* es un tipo de dato de punto flotante de doble precisión que utiliza 64 bits para representar valores numéricos. Cuando se convierte una imagen de *uint8* a *double*, los valores de píxeles se convierten en números de punto flotante con mayor precisión y un rango mucho mayor (Manna, 2016).

Según Elizondo & Maestre (2005), la razón para convertir la imagen a tipo de dato *double* se debe a las operaciones de procesamiento de imágenes que se realizarán más adelante en los códigos. Algunas operaciones con imágenes pueden requerir valores en punto flotante para mantener una alta precisión en los cálculos, especialmente cuando se realizan operaciones matemáticas complejas o cuando se trabaja con valores fuera del rango 0-255.

Teniendo en cuenta lo anterior, se crearon tres programas esenciales junto con un programa auxiliar dedicado al análisis de resultados: `Encriptación por SVD`, `Creación Propia` y `Carga de capas`; mientras que: `Histo_Analisis`, complementan a los dos primeros.

La `Encriptación por SVD` se destaca como el algoritmo principal de codificación de imágenes. En esencia el programa inicia al 'leer' la imagen y la convierte en una matriz. Por consiguiente, esta matriz se divide en tres componentes RGB y se somete a la función SVD para calcular los factores matriciales. Con base en esto, se aplican operaciones en cada matriz y la imagen se recompone con los valores modificados.

Por otro lado, "`Creación Propia`" sigue un proceso similar al 'leer' imágenes, pero con la particularidad que genera vectores a partir de las matrices y aplica un cifrado de sustitución

con una clave secreta aleatoria. Estos vectores cifrados se convierten nuevamente en matrices de píxeles, formando una imagen cifrada.

En "`Carga de capas`", el código comienza con un menú condicional a través de un bucle '`while`', permitiendo la elección del usuario según los datos disponibles. En el primer caso, muestra la imagen cuando las capas no están encriptadas, cargando las capas `R, G` y `B` secuencialmente, y luego las combina para reconstruir la imagen. En el caso 2, una vez cargadas las capas de la imagen encriptada, se verifica su completa carga y se procede a reconstruir la imagen cifrada. La opción 3 permite salir del menú.

Finalmente, "`Histo_Analisis`" se encarga del cálculo de histogramas para una imagen original y una encriptada en escala de grises. Así pues, visualiza estos histogramas junto con las imágenes correspondientes, representado los niveles de intensidad del color de f con respecto al número de píxeles presentes en f con cada intensidad de color (Manna, 2016). El eje x (horizontal) representa los diferentes tonos de gris desde el negro puro (a la izquierda) al blanco puro (a la derecha). Mientras que el eje y (vertical) representa el número de pixeles que contiene la imagen para cada tono representado en el eje horizontal.

4.1 Programación de Encriptación por SVD

A continuación, se explica el código del programa `Encriptación por SVD` comentado. Las primeras líneas del código utilizan la función “`disp`” para mostrar mensajes informativos en la consola. "`Encriptar por SVD`" se presenta como un título, seguido de una línea en blanco. Posteriormente, se proporciona una nota la cual aconseja al usuario descargar las capas de las imágenes, lo que sugiere que el proceso afectará la imagen original y producirá varias

capas de resultados. La imagen "imagen.jpg" se carga utilizando la función imread y se almacena en la variable im. Esto carga la imagen original en el código para su posterior procesamiento.

Seguidamente, la imagen se convierte de su formato original a tipo de dato double utilizando la función double(im) y se almacena en la variable I. Esto es necesario para realizar operaciones matemáticas en la imagen, ya que como afirma Elizondo & Maestre (2005), el formato double permite trabajar con valores decimales en lugar de enteros. Esta imagen que se guardó en I se divide en sus tres canales de color: R (rojo), G (verde) y B (azul) con la finalidad de procesar cada canal por separado utilizando SVD. Los canales de color se almacenan en las variables R, G y B. Se realiza la descomposición de valores singulares (SVD) para cada uno de los canales de color (R, G, B) utilizando las funciones svd cuya sintaxis es [U1 S1 V1] = svd(R) (Para el caso del canal de color Rojo). Esto resulta en tres conjuntos de matrices de descomposición: U1, S1 y V1 para el canal R, U2, S2 y V2 para el canal G, y U3, S3 y V3 para el canal B.

Consecutivamente se crean dos matrices denominadas llave1 y llave2. Estas matrices se utilizan como componentes de seguridad para encriptar los valores singulares. llave1 se calcula multiplicando una matriz de unos por valores numéricos específicos, y llave2 se calcula multiplicando la matriz de identidad por los mismos valores. Esto introduce un factor de encriptación a los valores singulares, se suman las matrices de llaves llave1 y llave2 a las matrices de valores singulares (S1, S2, S3) de los canales de color R, G y B, respectivamente. Esto encripta los valores singulares y se almacenan en las matrices Sf1, Sf2 y Sf3.

Después, se lleva a cabo la 'desencriptación' de los valores singulares encriptados restando las matrices de llaves a los valores singulares encriptados (`Sf1, Sf2, Sf3`). Los valores desencriptados se almacenan en las matrices deimcR, deimcG y deimcB. Se utilizan las matrices desencriptadas de valores singulares, para reconstruir los canales de color desencriptados (`deimcR, deimcG, deimcB`) mediante multiplicación matricial y se emplean las funciones `dlmwrite` para guardar las capas de la imagen originales y desencriptada en archivos separados en formatos .png y .jpg. Esto permite al usuario acceder a las capas de color de las imágenes por separado.

Utilizando las matrices encriptadas de valores singulares (`Sf1, Sf2, Sf3`) y las matrices de descomposición SVD (`U1, U2, U3 y V1, V2, V3`), se reconstruyen las imágenes 'encriptada' (final) y la imagen original (dese) mediante multiplicación matricial. Finalmente se emplea la función `imshow` para mostrar la imagen encriptada y la imagen desencriptada en dos figuras separadas.

En definitiva, este código realiza una encriptación y desencriptación de una imagen mediante la descomposición de valores singulares (SVD) y añade una capa de seguridad mediante la adición de llaves numéricas a los valores singulares. Posteriormente, muestra las imágenes resultantes y guarda sus capas de color en archivos separados para su posterior uso.

En seguida, se presenta el código de la propuesta de encriptación de imágenes utilizando el algoritmo Encriptación por SVD en la plataforma Octave Online.

4.1.1 *Programación código encriptación por SVD (Versión Comentada)*

```
% Paso 1: Título y Nota Informativa
```

```
disp ("Encriptar por SVD"); % Mostrar mensaje informativo
disp ("") % Línea en blanco
disp ("Nota importante: Recuerda descargar las capas de las imágenes."); % Mostrar nota
% Paso 2: Cargar la Imagen
im = imread("imagen.jpg"); % Cargar la imagen original en 'im'
% Paso 3: Conversión a Tipo de Dato 'double'
I = double(im); % Convertir la imagen a tipo de dato 'double' para cálculos precisos
% Paso 4: Separación de Canales de Color
R = I(:,:,1); % Extraer el canal de color rojo
G = I(:,:,2); % Extraer el canal de color verde
B = I(:,:,3); % Extraer el canal de color azul

% Paso 5: Descomposición SVD de los Canales de Color
[U1 S1 V1] = svd(R); % Descomposición SVD del canal R
[U2 S2 V2] = svd(G); % Descomposición SVD del canal G
[U3 S3 V3] = svd(B); % Descomposición SVD del canal B
% Paso 6: Obtener Dimensiones de las Matrices de Valores Singulares
[f,c]= size(S1); % Obtener las dimensiones de una de las matrices de valores singulares (S1)
% Paso 7: Generación de Llaves para Encriptación
llave1 = (ones(f,c).*(100000)).*(2).*(3).*(5).*(7).*(11); % Generar una llave de encriptación
llave2 = (eye(f,c).*(100000)).*(2).*(3).*(5).*(7).*(11);  % Generar otra llave de encriptación
```

```
% Paso 8: Encriptación de Valores Singulares
Sf1= S1+llave1+llave2; % Encriptar valores singulares del canal R
Sf2= S2+llave1+llave2; % Encriptar valores singulares del canal G
Sf3= S3+llave1+llave2; % Encriptar valores singulares del canal B
% Paso 9: Desencriptación de Valores Singulares y Reconstrucción de Ca-
nales
deimcR = U1*(Sf1-llave1-llave2)*V1'; % Desencriptar y reconstruir el ca-
nal R
deimcG = U2*(Sf2-llave1-llave2)*V2'; % Desencriptar y reconstruir el ca-
nal G
deimcB = U3*(Sf3-llave1-llave2)*V3'; % Desencriptar y reconstruir el ca-
nal B
% Paso 10: Almacenar las Capas de la Imagen Desencriptada en Archivos
dese(:,:,1)=deimcR; % Almacenar el canal R desencriptado
dese(:,:,2)=deimcG; % Almacenar el canal G desencriptado
dese(:,:,3)=deimcB; % Almacenar el canal B desencriptado
dlmwrite('deseR.png',deimcR) % Guardar el canal R desencriptado en PNG
dlmwrite('deseG.png',deimcG) % Guardar el canal G desencriptado en PNG
dlmwrite('deseB.png',deimcB) % Guardar el canal B desencriptado en PNG
dlmwrite('deseR.jpg',deimcR) % Guardar el canal R desencriptado en JPG
dlmwrite('deseG.jpg',deimcG) % Guardar el canal G desencriptado en JPG
dlmwrite('deseB.jpg',deimcB) % Guardar el canal B desencriptado en JPG
% Paso 11: Reconstrucción de la Imagen Encriptada
imcR = U1*Sf1*V1; % Reconstruir el canal R encriptado
imcG = U2*Sf2*V2; % Reconstruir el canal G encriptado
imcB = U3*Sf3*V3; % Reconstruir el canal B encriptado
```

```
% Paso 12: Almacenar las Capas de la Imagen Encriptada en Archivos
final(:,:,1)=imcR; % Almacenar el canal R encriptado
final(:,:,2)=imcG; % Almacenar el canal G encriptado
final(:,:,3)=imcB; % Almacenar el canal B encriptado
dlmwrite('finalR.jpg',imcR) % Guardar el canal R encriptado en JPG
dlmwrite('finalG.jpg',imcG) % Guardar el canal G encriptado en JPG
dlmwrite('finalB.jpg',imcB) % Guardar el canal B encriptado en JPG
dlmwrite('finalR.png',imcR) % Guardar el canal R encriptado en PNG
dlmwrite('finalG.png',imcG) % Guardar el canal G encriptado en PNG
dlmwrite('finalB.png',imcB) % Guardar el canal B encriptado en PNG
% Paso 13: Visualización de las Imágenes Resultantes
figure, imshow(uint8(final)); % Mostrar la imagen encriptada
title('Imagen Encriptada')
figure, imshow(dese/255); % Mostrar la imagen desencriptada
title('Imagen Desencriptada')
```

4.2 Programación de “Creación Propia”

Este código lleva a cabo un proceso de encriptación de una imagen utilizando un cifrado de sustitución simple. A continuación, se describirá paso a paso las acciones que realiza.

Primero, el archivo "`imagen.jpg`" se carga en la variable `'f'` utilizando la función `imread`. Esto carga la imagen en su formato original para luego se convertirla en una matriz de doble precisión llamada `X` utilizando `double(f)`. Como se mencionó anteriormente para el código de `Encriptar por SVD`, la conversión a `double` es importante para realizar operaciones matemáticas precisas en los valores de píxeles de la imagen.

La matriz de la imagen en doble precisión `X`, se divide en sus tres canales de color: rojo `(XR)`, verde `(XG)` y azul `(XB)`. Esto tiene como finalidad procesar cada canal de color por separado y así se crear tres vectores vacíos: `vr` para el canal rojo, `vg` para el canal verde y `vb` para el canal azul. Después, se recorren las matrices de píxeles de cada canal de color y se transforman en vectores unidimensionales. Esto significa que cada píxel se convierte en un elemento del vector y los vectores resultantes contienen todos los valores de píxeles de sus respectivos canales de color.

En este contexto, se genera una clave con la expresión `clave = randperm(256)`. La función `randperm(256)` produce una permutación aleatoria de los números del 1 al 256. Esto implica que la clave contendrá todos los números del 1 al 256, pero en un orden único y aleatorio. La clave consta de 256 elementos y se utiliza para realizar un cifrado de sustitución en los vectores de valores de píxeles (`vr, vg, vb`). Por ejemplo, `vr_cifrado = clave(vr+1)`, cifra el vector `vr`. Es importante destacar que se suma 1 a cada elemento del vector `vr (vr+1)` antes de utilizarlo como índice en la clave. Esto se debe a que los índices de la matriz clave comienzan en 1, mientras que los valores de píxeles varían de 0 a 255 y la adición de 1 a los valores de `vr` asegura que estén dentro del rango adecuado para acceder a la clave.

Se realizó un proceso similar para los vectores `vg` y `vb`; `vg_cifrado = clave(vg+1); vb_cifrado = clave(vb+1);` aplican el cifrado de sustitución a los vectores `vg` y `vb`, respectivamente.

Cada elemento de los vectores `vr`, `vg` y `vb` se sustituye por el valor correspondiente en la clave. Esto significa que cada valor de píxel original se reemplaza por un valor cifrado basado en su posición en la clave secreta.

En sí, se crea una clave secreta aleatoria que consiste en una permutación aleatoria de los números del 1 al 256. Luego, se aplica un cifrado de sustitución a los valores de píxeles de los vectores `vr`, `vg` y `vb` utilizando esta clave secreta. Cada valor de píxel en los vectores se reemplaza por un valor cifrado basado en su posición en la clave, lo que resulta en la encriptación de la imagen.

Posteriormente se reconstruyen las matrices de píxeles cifrados `(XR_cifrado, XG_cifrado, XB_cifrado)` a partir de los vectores cifrados. Esto restaura la estructura de matriz bidimensional, lo que te permite obtener imágenes cifradas en lugar de vectores. La imagen cifrada se genera al combinar los tres canales de color encriptados `(XR_cifrado, XG_cifrado, XB_cifrado)` en una matriz tridimensional denominada `X_cifrado`, que representa la imagen encriptada en su totalidad.

Seguidamente, las capas de color de la imagen original `(XR, XG, XB)` y las capas de color de la imagen cifrada `(XR_cifrado, XG_cifrado, XB_cifrado)`, se guardan en archivos separados utilizando las funciones dlmwrite. Los archivos se guardan en formatos .png y .jpg. Esto te permite mantener una copia de cada canal de color para referencia o posteriores manipulaciones.

Para finalizar, se emplean las funciones `imshow` para mostrar la imagen original `(X)` y la imagen `cifrada (X_cifrado)` en dos figuras separadas. Esto te permite visualizar y comparar las dos imágenes, una antes de la encriptación y otra después del cifrado de sustitución.

4.2.1 *Programación código encriptación por "Creación Propia" (Versión Comentada)*

```
% Cargar la imagen y convertirla en una matriz de doble precisión
f = imread("imagen.jpg"); % Carga la imagen desde un archivo llamado "imagen.jpg"
X = double(f); % Convierte la imagen en una matriz de doble precisión
XR = X(:,:,1); % Extrae el canal de color rojo de la imagen
XG = X(:,:,2); % Extrae el canal de color verde de la imagen
XB = X(:,:,3); % Extrae el canal de color azul de la imagen

% Guardar las capas de la imagen original en archivos separados
dlmwrite ('oriR.png', XR) % Guarda el canal rojo como "oriR.png"
dlmwrite ('oriG.png', XG) % Guarda el canal verde como "oriG.png"
dlmwrite ('oriB.png', XB) % Guarda el canal azul como "oriB.png"
dlmwrite ('oriR.jpg', XR) % Guarda el canal rojo como "oriR.jpg"
dlmwrite ('oriG.jpg', XG) % Guarda el canal verde como "oriG.jpg"
dlmwrite ('oriB.jpg', XB) % Guarda el canal azul como "oriB.jpg"

% Obtener dimensiones de la imagen
[n1, n2] = size (XR);
% Inicializar vectores para los canales de color
vr = [];
```

```
vg = [];
vb = [];
% Transformar las matrices de píxeles en vectores
for i = 1: n1
    for j = 1: n2
        vr = [vr; XR (i, j)]; % Vector de intensidad de rojo
        vg = [vg; XG (i, j)]; % Vector de intensidad de verde
        vb = [vb; XB (i, j)]; % Vector de intensidad de azul
    end
end
% Crear una clave secreta aleatoria
clave = randperm (256); % Genera una permutación aleatoria de números del 1 al 256

% Aplicar el cifrado de sustitución a cada elemento de los vectores
vr_cifrado = clave (vr + 1); % Cifra el canal rojo
vg_cifrado = clave (vg + 1); % Cifra el canal verde
vb_cifrado = clave (vb + 1); % Cifra el canal azul
% Transformar los vectores cifrados en matrices de píxeles
XR_cifrado = reshape(vr_cifrado, n1, n2); % Reconstruye el canal rojo cifrado
XG_cifrado = reshape(vg_cifrado, n1, n2); % Reconstruye el canal verde cifrado
XB_cifrado = reshape(vb_cifrado, n1, n2); % Reconstruye el canal azul cifrado
% Crear la imagen cifrada combinando los canales cifrados
```

```
X_cifrado(:,:,1) = XR_cifrado; % Canal rojo cifrado
X_cifrado(:,:,2) = XG_cifrado; % Canal verde cifrado
X_cifrado(:,:,3) = XB_cifrado; % Canal azul cifrado
% Guardar las capas de la imagen cifrada en archivos separados
dlmwrite ('capaR.png', XR_cifrado)  % Guarda el canal rojo cifrado como "capaR.png"
dlmwrite('capaG.png', XG_cifrado) % Guarda el canal verde cifrado como "capaG.png"
dlmwrite('capaB.png', XB_cifrado) % Guarda el canal azul cifrado como "capaB.png"
dlmwrite ('capaR.jpg', XR_cifrado)  % Guarda el canal rojo cifrado como "capaR.jpg"
dlmwrite('capaG.jpg', XG_cifrado) % Guarda el canal verde cifrado como "capaG.jpg"
dlmwrite('capaB.jpg', XB_cifrado) % Guarda el canal azul cifrado como "capaB.jpg"
% Mostrar la imagen original y la imagen cifrada
figure, imshow(uint8(X)); % Muestra la imagen original
title ('Imagen original');
figure, imshow(uint8(X_cifrado)); % Muestra la imagen cifrada
title ('Imagen cifrada');
```

Finalmente, el código realiza las siguientes operaciones: carga una imagen, la divide en sus canales de color (rojo, verde y azul), encripta la imagen mediante un cifrado de sustitución utilizando una clave secreta aleatoria y muestra tanto la imagen original como la encriptada.

Además, guarda por separado las capas de color de ambas imágenes en archivos .PNG Y .JPG, facilitando su uso en el código relacionado con la carga de estas capas RGB.

4.3 Programación de Carga de Capas

Para la programación de este código se estableció un bucle ‘while’ que se ejecutará indefinidamente mientras la condición true sea verdadera. En otras palabras, este bucle se efectuará continuamente hasta que se rompa explícitamente con la instrucción break.

Después, muestra mensajes informativos y las opciones disponibles para el usuario:

"Bienvenido al programa de carga de imágenes": Este mensaje da la bienvenida al usuario al programa.

"Por favor seleccione una opción:": Indica al usuario que debe elegir una de las opciones proporcionadas.

"1. Cargar imagen desencriptada" y "2. Cargar imagen encriptada": Estas líneas presentan las dos primeras opciones disponibles para el usuario, que son cargar una imagen desencriptada o cargar una imagen encriptada.

"3. Salir": Esta línea muestra la opción de salir del programa.

Se hace la aclaración las dos situaciones específicas en las que el ciclo `while` se rompe:

1. Cuando el usuario selecciona la opción "3" (Salir): En el caso 3 (case 3) del `switch`, el programa muestra un mensaje de despedida con `disp("¡Gracias por usar el programa, esperamos volvernos a ver!")`. En seguida, utiliza la instrucción `return` para finalizar la ejecución del programa. Esto significa que, si el usuario elige la opción "3", el programa se detendrá y no se ejecutará más.

2. Después de realizar una operación (cargar y visualizar una imagen o salir del programa), el programa pregunta si el usuario desea cargar otra imagen. Si el usuario ingresa "no", el programa muestra un mensaje de agradecimiento con `disp("Gracias por usar el programa.");` y posteriormente utiliza la instrucción `break` para salir del bucle `while`. Esto detiene la ejecución del programa.

En resumen, el ciclo `while` se romperá cuando el usuario seleccione la opción "3" para salir del programa o cuando el usuario decida no cargar más imágenes al responder "no" a la pregunta final. En ambos casos, el programa se detendrá y la ejecución finalizará.

Continuando con la explicación del código, la línea `opcion = input ("Seleccione una opción: ");` implica que el programa solicita al usuario que introduzca un número que represente la opción que desean seleccionar. Este número ingresado se guarda en la variable "`opcion`".

El código utiliza una estructura `switch` para manejar la elección del usuario:

case 1: Si el usuario selecciona la opción 1, el programa le pide que ingrese los nombres de archivo de las capas de la imagen desencriptada (rojo, verde y azul). Por consiguiente, el programa utiliza `dlmread` para cargar las matrices de las capas R, G y B desde los archivos cuyos nombres proporcionó el usuario. Combina estas matrices para formar la imagen original `I` utilizando `cat (3, R, G, B)`. Finalmente, muestra la imagen original al usuario utilizando imshow y establece un título para la ventana de visualización.

case 2: seleccionando la opción 2, se le pide que ingrese los nombres de archivo de las capas de la imagen encriptada (rojo, verde y azul). El código verifica si se han ingresado las tres

capas antes de intentar combinarlas y mostrar la imagen encriptada. Si faltan capas, se muestra un mensaje de advertencia.

case 3: Si el usuario selecciona la opción 3, el programa muestra un mensaje de despedida y utiliza `return` para salir del programa.

En el caso que el usuario ingrese una opción que no es 1, 2 ni 3, se muestra un mensaje de error indicando que la opción no es válida y se ocupó la función `otherwise` que se utiliza para manejar entradas no válidas o no reconocidas del usuario, proporcionando un mensaje de error amigable para mantener la interacción con el programa lo más comprensible posible.

Finalmente se explica el porqué se utilizó el doble igual "==" en lugar de uno solo en la línea `if respuesta == "no"` pues se está realizando una comparación en lugar de una asignación

== (doble igual): Se utiliza para comparar dos valores o expresiones y verificar si son iguales en términos de valor. En este contexto, "respuesta" se compara con la cadena de caracteres "no" para verificar si son iguales. (MATLAB, s.f.)

= (un solo igual): Se utiliza para asignar un valor a una variable o una expresión. Por ejemplo, respuesta = "no" asignaría el valor "no" a la variable respuesta, pero no estaría realizando una comparación. (MATLAB, s.f.)

En la línea `if respuesta == "no"`, la intención fue verificar si la respuesta del usuario es igual a la cadena "no". Por lo tanto, se utiliza el doble igual == para realizar esta comparación. Si la respuesta del usuario es exactamente igual a "no", la condición se considera verdadera y el programa ejecuta las instrucciones dentro del bloque `if`. Si la respuesta no es igual

a "no" (por ejemplo, si el usuario ingresa "sí" u otra cosa), la condición se considera falsa y el programa no ejecuta las instrucciones dentro del bloque `if`.

En conclusión, este código es un programa interactivo que permite al usuario cargar y visualizar imágenes desencriptadas o encriptadas, con opciones para salir o cargar más imágenes según sea necesario.

4.3.1 *Programación código "Carga de Capas" (Versión Comentada)*

```
disp("Carga de capas");
disp("")
      while true

          % Mostrar un mensaje de bienvenida al programa

          disp("Bienvenido al programa de carga de imágenes");

          disp("")

          % Mostrar las opciones disponibles para el usuario

          disp("Por favor seleccione una opción:");

          disp("")

          disp("1. Cargar imagen desencriptada");

          disp("")

          disp("2. Cargar imagen encriptada");

          disp("")

          disp("3. Salir");

          disp("")

          % Solicitar al usuario que seleccione una opción

          opcion = input("Seleccione una opción: ")

          % Usar una estructura de control switch para manejar la opción se-
leccionada
```

```
        switch opcion
            case 1
          % Cargar una imagen desencriptada
                disp("Ingrese las capas de la imagen desencriptada");
                disp("")
      % Solicitar al usuario los nombres de los archivos de las capas de color
R, G y B
                nombre_R = input("Ingresa el nombre del archivo de la capa R
(la extensión debe ser .png o .jpg): ", 's');
                disp("")
                nombre_G = input("Ingresa el nombre del archivo de la capa G
(la extensión debe ser .png o .jpg): ", 's');
                disp("")
                nombre_B = input("Ingresa el nombre del archivo de la capa B
(la extensión debe ser .png o .jpg): ", 's');
                disp("")
      % Leer las matrices de las capas de color R, G y B a partir de los archivos
                R = dlmread(nombre_R);
                G = dlmread(nombre_G);
                B = dlmread(nombre_B);
      % Combinar las matrices de las capas de color R, G y B para formar la
imagen original
                I = cat(3, R, G, B);
      % Mostrar la imagen original cargada
                imshow(uint8(I));
                title('Imagen desencriptada cargada')
```

```
            case 2
        % Cargar una imagen encriptada
                disp("Ingrese las capas de la imagen encriptada");
      % Solicitar al usuario los nombres de los archivos de las capas de color
R, G y B
                nombre_R = input("Ingresa el nombre del archivo de la capa R
(la extensión debe ser .png o .jpg): ", 's');
                disp("")
                nombre_G = input("Ingresa el nombre del archivo de la capa G
(la extensión debe ser .png o .jpg): ", 's');
                disp("")
                nombre_B = input("Ingresa el nombre del archivo de la capa B
(la extensión debe ser .png o .jpg): ", 's');
                disp("")
      % Leer las matrices de las capas de color R, G y B a partir de los archivos
                R = dlmread(nombre_R);
                G = dlmread(nombre_G);
                B = dlmread(nombre_B);
      % Verificar si se tienen las tres capas de color
                if isempty(R) || isempty(G) || isempty(B)
                    disp("Faltan capas para obtener la imagen completa.");
                else
      % Combinar o concatenar las matrices de las capas de color R, G y B para
formar la imagen encriptada
                    I = cat(3, R, G, B);
```

```
        % Mostrar la imagen encriptada cargada
                        imshow(uint8(I));
                        title('Imagen encriptada cargada')
                    end
                case 3
             % Salir del programa si el usuario selecciona la opción 3
              disp("¡Gracias por usar el programa, esperamos volvernos a ver!");
                        return;
                otherwise
        % Mostrar un mensaje de opción no válida si el usuario ingresa una opción
inválida
                    disp("Opción no válida.");
            end
        % Preguntar al usuario si desea cargar otra imagen
            respuesta = input("¿Desea cargar otra imagen? (si/no): ", 's');
        % Salir del bucle while si la respuesta es "no"
            if respuesta == "no"
                disp("Gracias por usar el programa.");
                break; % Salir del bucle while
            end
        end
```

4.4 Programación del Código “Histo Análisis”

Este código se encarga de realizar operaciones relacionadas con los histogramas de imágenes en escala de grises. En primer lugar, la imagen original `I` se convierte a escala de grises y luego se asegura de que los valores de píxeles estén dentro del rango válido de 0 a 255, utilizando

el tipo uint8 (Manna, 2016). La conversión a escala de grises se realiza mediante `rgb2gray` y el resultado se almacena en `I_gris`. El mismo proceso se aplica a una imagen encriptada en escala de grises. Se calcula el histograma de la imagen `I_gris` utilizando la función `imhist`. El histograma representará la distribución de intensidades de píxeles en la imagen (Elizondo & Maestre, 2005), y se guarda en la variable `original_hist`.

A continuación, se crean dos ventanas de figura. En la primera ventana, se muestra la imagen en escala de grises original en un `subplot(2,1,1)`, enfocándose en el primer subtrazado. Para mostrar la imagen, se utiliza el comando `imshow(I_gris, [])` que ajusta automáticamente el mapeo de colores para adaptarse al rango de intensidad de la imagen. En la segunda ventana de figura, se repite el proceso para la imagen encriptada en escala de grises, mostrándola en el `subplot(2,1,2)` y creando el segundo subtrazado en la misma ventana. En este segundo subtrazado, se traza el histograma correspondiente utilizando el comando `plot(encrip_hist)`. Esta representación gráfica permite una comparación visual de los histogramas de ambas imágenes.

En síntesis, este código se empleó para analizar las diferencias en las distribuciones de intensidades de píxeles entre una imagen original y su versión encriptada en escala de grises. Esto puede ser útil para evaluar la calidad de la encriptación y entender cómo afecta a la imagen en términos de contenido de intensidad de píxeles.

4.4.1 *Programación código "Histo_Análisis"(Versión Comentada)*

```
% Calcular histograma de la imagen original en escala de grises
I_gris = rgb2gray(uint8(I));
```

```
original_hist = imhist(I_gris);
% Calcular histograma de la imagen encriptada en escala de grises
final_gris = uint8(final);
final_gris = rgb2gray(final_gris);
encrip_hist = imhist(final_gris);
% Mostrar histograma de la imagen original en escala de grises
figure;
subplot(2,1,1);
imshow(I_gris, []);
title('Imagen Original (Escala de Grises)')
subplot(2,1,2);
plot(original_hist);
title('Histograma de la Imagen Original (Escala de Grises)'
% Mostrar histograma de la imagen encriptada en escala de grises
figure;
subplot(2,1,1);
imshow(final_gris, []);
title('Imagen Encriptada (Escala de Grises)')
subplot(2,1,2);
plot(encrip_hist);
title('Histograma de la Imagen Encriptada (Escala de Grises)')
```

4.5 Diagrama de flujo

El diagrama de procesos, también conocido como diagrama de flujo, en el contexto de los sistemas dirigidos por procesamiento electrónico de datos, sigue un algoritmo, es decir, una secuencia lógica de pasos para resolver un problema según Hernández et al. (2014). Manene L. (2011) explica que esta representación es una herramienta altamente beneficiosa para cualquier organización que busca administrar diversos tipos de proyectos, tanto internos como externos. Tal como afirman Rojas & García (2020), facilita desglosar y comprender los procesos empresariales en un orden lógico que se plasma en un solo documento, junto con sus interconexiones. Además, permite identificar áreas de mejora y otorgar relevancia, estableciendo un control minucioso sobre todos los procesos de una empresa, incluso aquellos que puedan parecer insignificantes (Manene L., 2011). Por ende, el esquema de procesos se convierte en una herramienta esencial para realizar un análisis exhaustivo y determinar en qué etapas del flujo de trabajo es posible implementar métricas, controles o identificar oportunidades para mejorar.

Por esta razón, se representó gráficamente en la Figura 17, las distintas etapas de los códigos anteriormente descritos y sus interacciones, para facilitar la comprensión de su funcionamiento (Hernández et al., 2014).

En la Tabla 6, *Sistema de Encriptación de imágenes y Carga de Capas*, se delinearán los elementos claves que participan en la página web "Encriptación de Imágenes Digitales". Esta plataforma posibilita la codificación de imágenes mediante dos técnicas principales: la Descomposición en Valores Singulares (SVD) y el uso de vectores de sustitución. Posteriormente el usuario deberá cargar las capas al programa denominado como “Carga de Capas”.

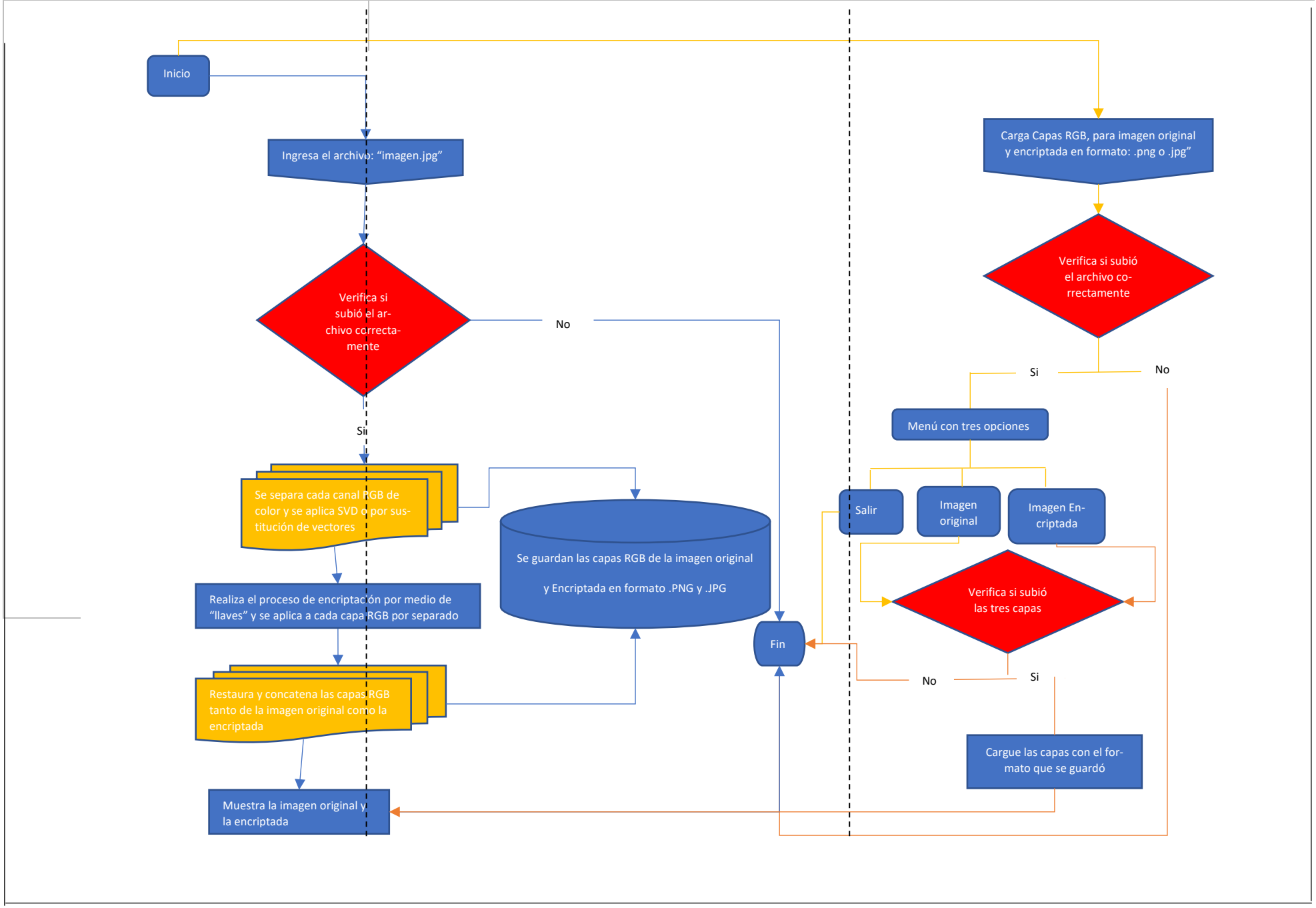
Tabla 6. Sistema de Encriptación de imágenes y Carga de Capas.

Entradas	***Subproceso o Actividad***	***Salidas***
El usuario carga su documento visual en los archivos de Octave Online con el nombre "imagen.jpg".	Encriptación de la imagen mediante descomposición en Valores Singulares o Sustitución de vectores.	Se almacenan las capas de los canales RGB de la imagen original y la encriptada
Verificación del usuario para descargar las capas necesarias y cargarlas en la aplicación "Carga de Capas".	Carga de las capas en los archivos de Octave Online	Almacenamiento de las capas seleccionadas por el usuario
El usuario, en la aplicación de carga de capas, elige entre cargar la imagen original o la encriptada.	Concatenación de las capas unidimensionales RGB de la imagen original o encriptada en un solo arreglo matricial tridimensional	Imagen original o encriptada

Tabla 7. Relaciones y actores en el proceso de Encriptación de Imágenes.

Relaciones	**Actores**
• El usuario accede a la página web y se dirige a la sección de "Encriptación por SVD" para el caso de este método de cifrado. Si está de acuerdo con su elección, carga una imagen de su biblioteca personal en formato .JPG a través de Octave Online. • En la página, se encuentra el código de encriptación seleccionado. El usuario simplemente verifica que la imagen se cargue con el tamaño adecuado y el nombre predeterminado de lectura que tiene por defecto el código, que debe ser "imagen.jpg". • Una vez que la imagen se carga de manera satisfactoria, Octave Online lleva a cabo el procedimiento de cifrado, lo cual puede tomar algunos segundos. En ocasiones, puede requerirse un tiempo adicional para la carga, ya que el software puede requerirlo para completar el procedimiento de manera satisfactoria. • El usuario debe descargar las capas de los canales RGB que el programa guarda en archivos .PNG y .JPG. • El software genera tanto las capas RGB de la imagen original como las capas de la imagen encriptada.	• Usuario que carga la imagen. • Software encargado de generar las encriptaciones de imágenes digitales. • Descarga de capas RGB de las imágenes originales y encriptadas

Figura 17. Diagrama de flujo Encriptación de Imágenes Digitales.


Encriptación por SVD
Encriptación Creación Propia
Carga de Capas
Inicio
Ingresa el archivo: "imagen.jpg"
Verifica si subió el archivo correctamente
No
Si
Se separa cada canal RGB de color y se aplica SVD o por sustitución de vectores
Realiza el proceso de encriptación por medio de "llaves" y se aplica a cada capa RGB por separado
Restaura y concatena las capas RGB tanto de la imagen original como la encriptada
Muestra la imagen original y la encriptada
Se guardan las capas RGB de la imagen original y Encriptada en formato .PNG y .JPG
Fin
Carga Capas RGB, para imagen original y encriptada en formato: .png o .jpg"
Verifica si subió el archivo correctamente
Si
No
Menú con tres opciones
Salir
Imagen original
Imagen Encriptada
Verifica si subió las tres capas
No
Si
Cargue las capas con el formato que se guardó

4.6 Diagramas de Casos de Uso

Para definir claramente los requisitos funcionales de estos programas, se utilizarán "Diagramas de Casos de Uso". Estos diagramas proporcionarán una representación visual precisa de cómo los diferentes actores interactuarán con el sistema y qué funcionalidades podrán utilizar. Al emplear "Diagramas de Casos de Uso", se busca comprender de manera detallada las acciones que los usuarios podrán llevar a cabo en el programa, permitiendo así identificar y definir los requisitos funcionales esenciales (Gutiérrez, 2011). A través de esta técnica, se establecerá una base sólida para el diseño e implementación del software, garantizando de cierta manera, que cumpla con las expectativas y necesidades de los usuarios.

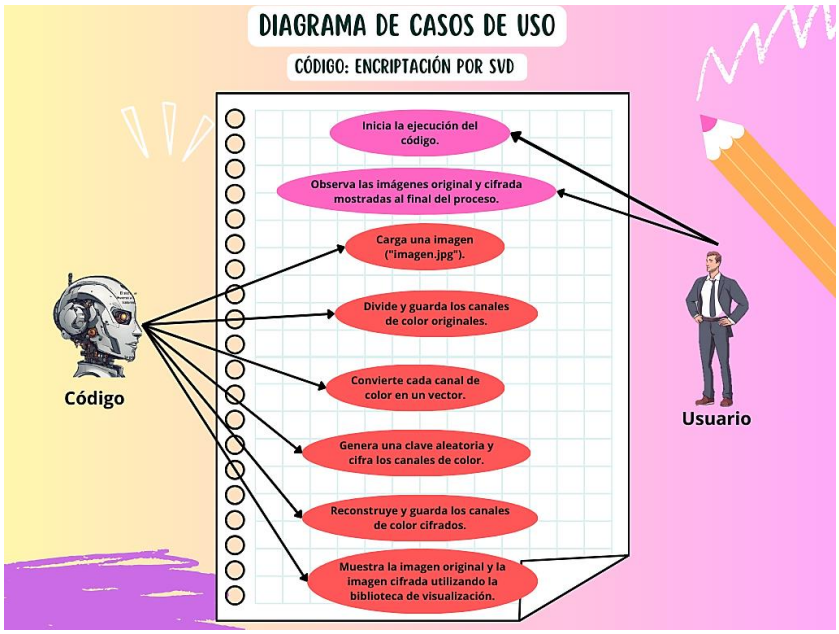


Figura 18. Diagrama de Casos de Uso SVD.

Nota. El “Diagrama de Casos de Usos” para el código de SVD, (elaboración propia).

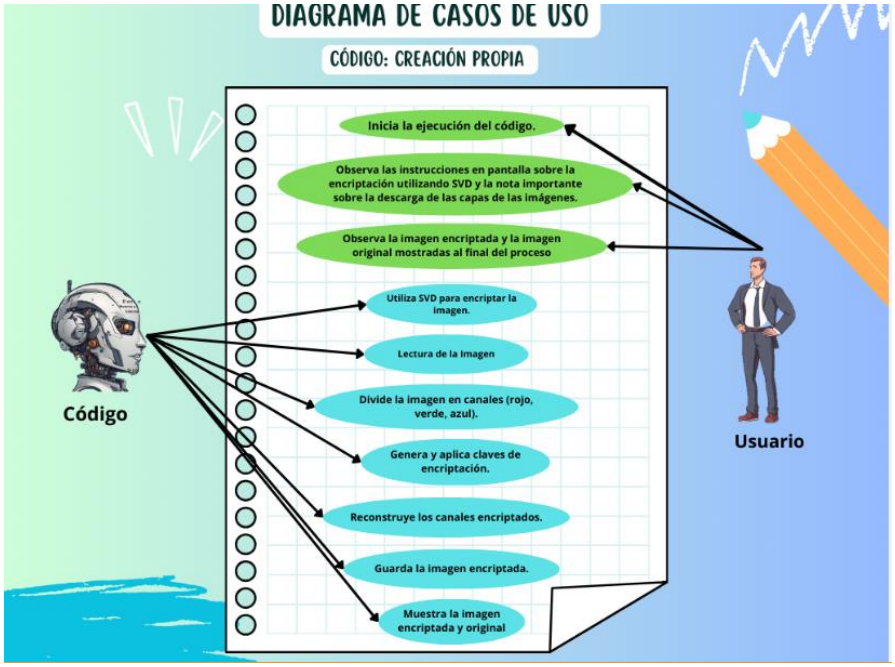


Figura 19. Diagrama de Casos de Uso "Creación Propia".

Nota. El "Diagrama de Casos de Usos" para el código de Creación Propia (elaboración propia).

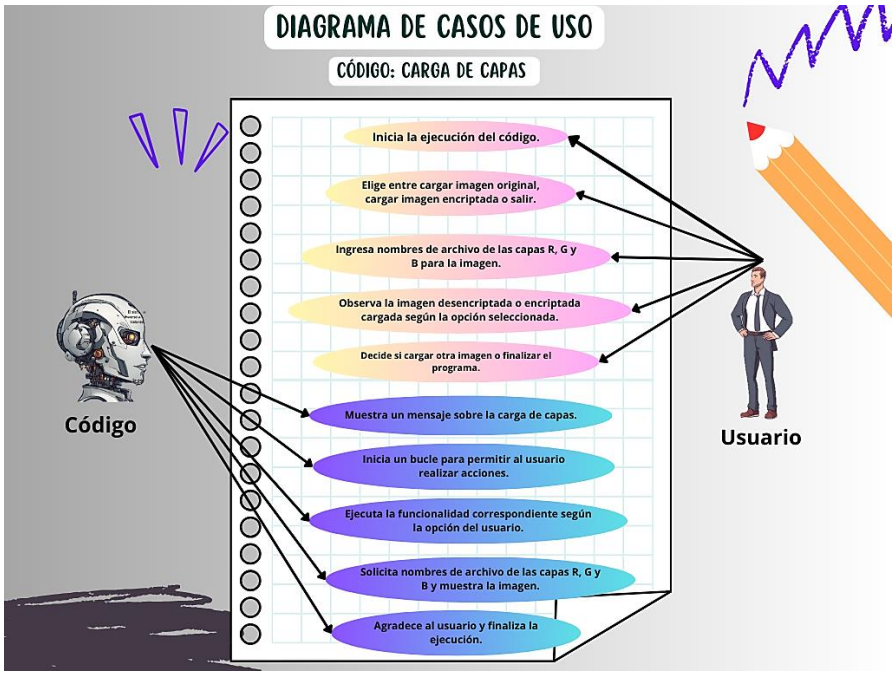


Figura 20. Diagrama de Casos de Uso Carga de Capas.

Nota. El "Diagrama de Casos de Usos" para el código de Carga de Capas (elaboración propia).

CAPÍTULO 5. Descripción del Sitio Web

La navegación en un sitio web debe ser intuitiva y predecible, es más, debe ser de fácil recordación. Tanto para usuarios nuevos como para quienes vuelven a usarlo, debe ser simple de descubrir y recordar cómo moverse en las distintas secciones o vistas con comodidad. Lograr que la navegación sea descubrible y accesible puede ser un reto por las limitaciones de diseño en las pantallas y, a su vez, se requiere priorizar el contenido sobre los componentes de la interfaz. En la actualidad, existen distintos patrones de navegación que apuntan a resolver este mismo problema de diferentes modos, pero todos tienen, en algún aspecto, problemas de usabilidad (Ferraris, 2018).

Por esta razón, es esencial llevar a cabo al menos una técnica de prototipado rápido, ya que esto simplificará considerablemente la transición entre las diferentes pantallas del software y mejorará la interacción. De esta manera, se puede apreciar una conexión entre la navegación entre las diversas vistas de un sitio web o aplicación y las acciones que los usuarios deben realizar, lo que se conoce principalmente como sistemas de interacción.

5.1 Patrón Tradicional para el Diseño de Navegación de un Sitio Web

Es esencial pensar que la forma en que se estructura la navegación influye mucho en cómo los usuarios interactúan con un sitio. Por un lado, los diseñadores deben invertir tiempo en comprender a fondo cómo los usuarios buscan información, sus emociones y sus reacciones. Esto a su vez, les proporcionará una base sólida para tomar decisiones informadas sobre el tipo de navegación más adecuado para el sitio y elegir entre diversas opciones disponibles (Webcion, 2022). De esta manera, se exploró una similitud en el diseño de la página web representada en la imagen siguiente.

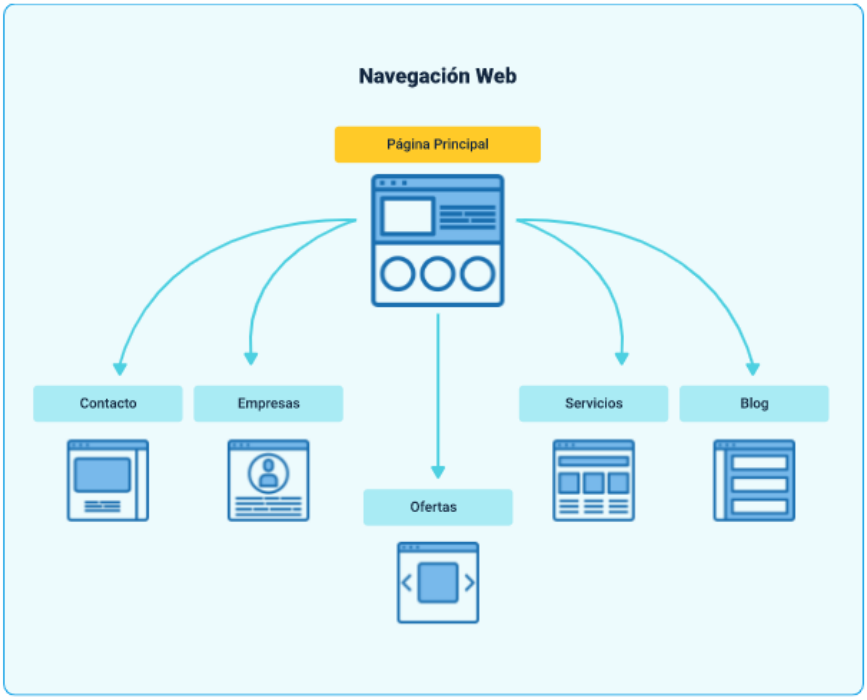


Figura 21. Modelo de diseño de página web[16]

A continuación, se proporciona una descripción detallada del sitio web "Encriptación de Imágenes Digitales", que se ha creado utilizando la aplicación Google Sites. El siguiente enlace proporciona acceso directo a la página: https://sites.google.com/view/encriptacion-de-imagendi/inicio.

[16] Diseño de página web (2023) imagen sin autor tomada de: https://lh3.googleusercontent.com/YCEy0eW-PTq6K_6hhK72MkURk8GP3IvpvCmP4rUKABHGbwjZ0-2wvwmSVTG9AWrTezQ6iyR0k5tNy4CbXox8f9Y0Hmi--q8hs4u-jqFT-

5.1.1 *Pestaña: Inicio*

Se presentó el título de la página en la Figura 22: 'Encriptación de Imágenes mediante SVD: Protege tus Datos Visuales'. Posteriormente, había un mensaje de bienvenida que ofrecía una visión de las pestañas disponibles en la barra de navegación. Al ingresar, se presentaba la barra de navegación que incluía las siguientes pestañas: Inicio, Marco de Referencia, Manuales de Uso, Encriptación por SVD y Ejemplos de Imágenes Encriptadas utilizando los códigos propuestos.

Figura 22. Título de la Página y Mensaje de Bienvenida.

Después, se insertó un botón (Figura 23), con la etiqueta 'Empezar', el cual conduce a la pestaña de 'Manual de Uso'. Justo debajo de este enlace, se proporcionó una breve descripción del software Octave, detallando su función en el procesamiento de imágenes. Además, si se hace clic en la imagen del logo de esta plataforma, esta enlazará con la página oficial del sitio web correspondiente.

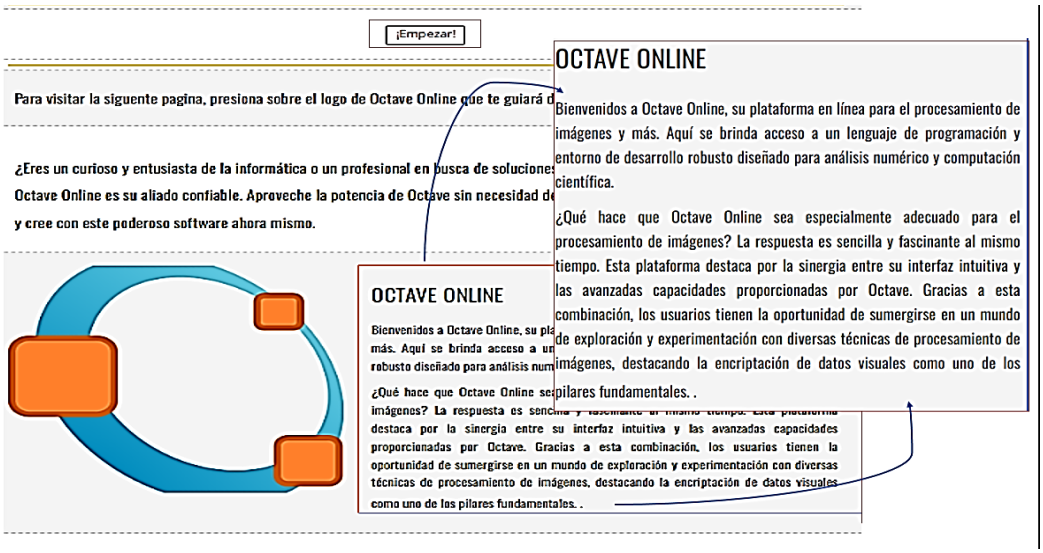


Figura 23. Botón ‘Empezar’ y breve descripción de Octave Online.

5.1.2 *Pestaña: Marco de referencia*

Como se puede observar en la Figura 24, la introducción contextualizó el proyecto de tesis sobre la encriptación de imágenes digitales mediante la técnica SVD, desarrollado por estudiantes de la Universidad Pedagógica Nacional de Colombia. En este marco, se proporcionó una breve reseña histórica de la encriptación y se enfatiza su relevancia en la actual seguridad de la información.

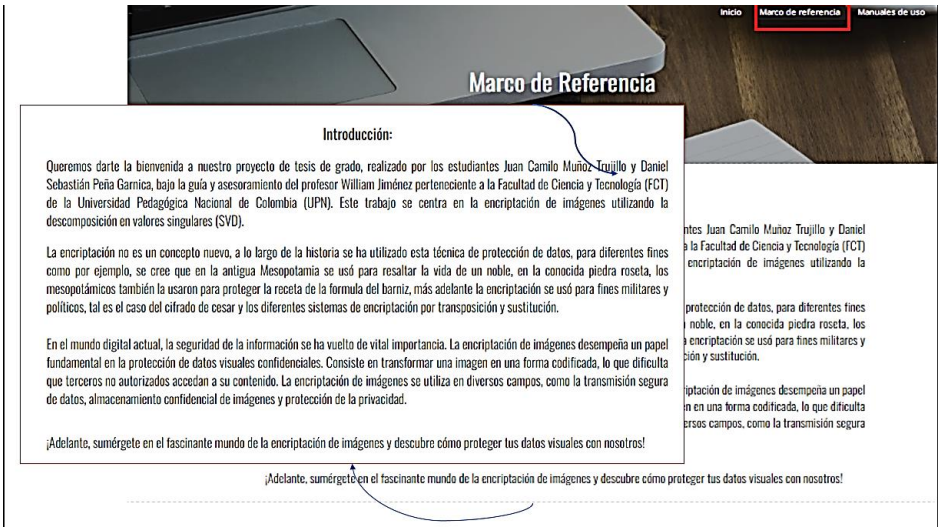


Figura 24 Marco referencia e introducción.

En la Figura 25, se ofreció una breve definición de conceptos clave como "Imagen Digital" y "Descomposición en Valores Singulares (SVD)". También se resaltaron tanto las ventajas como los desafíos asociados con la encriptación de imágenes utilizando esta técnica. El propósito de esta sección fue brindar una visión general, clara y concisa de estos conceptos fundamentales que servirían como base teórica antes de explorar las aplicaciones que permiten realizar la encriptación de imágenes.

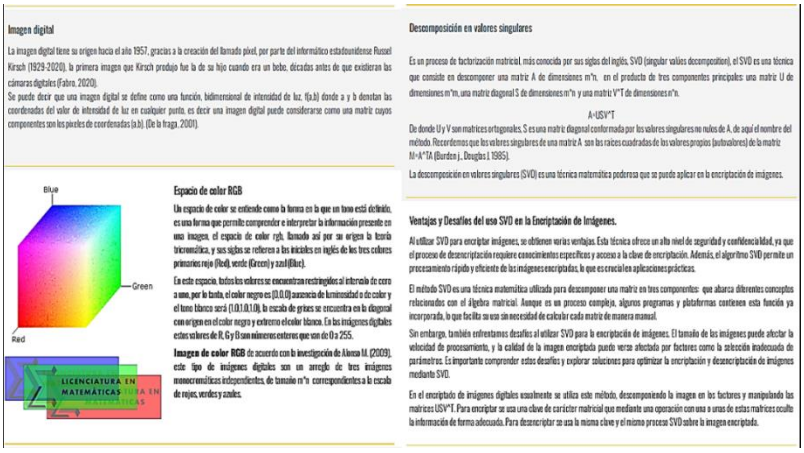


Figura 25. Imagen digital y Descomposición en Valores Singulares.

Después de concluir la sección de Marco Teórico, se incluyeron diagramas de flujo que representan las aplicaciones de encriptación por SVD, carga de capas y "Creación Propia" (Figura 26). La razón de esta inclusión fue proporcionar a los usuarios una visión general del funcionamiento de los códigos propuestos. Estos diagramas ayudaron a comprender de manera visual cómo se implementaban las soluciones encriptadas y facilitaron su interpretación antes de su utilización.

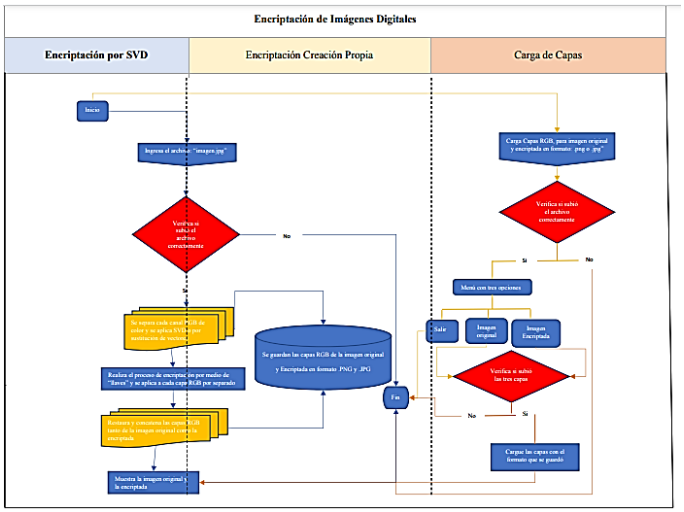


Figura 26. Diagramas de flujo: SVD, Carga de Capas y Creación Propia.

5.1.3 *Pestaña: Manual de Uso*

De la misma manera, se realizó una introducción que incluyó videos explicativos sobre el procedimiento de carga de imágenes en los aplicativos (Figura 27). Estos videos desempeñaron un papel esencial al ofrecer explicaciones detalladas y aclarar cualquier aspecto relacionado con el uso de las aplicaciones. Su objetivo principal fue garantizar una comprensión completa y facilitar una experiencia fluida para los usuarios al utilizar las herramientas proporcionadas.

Figura 27. Introducción Manual de Uso.

Se proporcionaron videos instructivos en la Figura 28, que explicaban en detalle el proceso de carga de imágenes en las aplicaciones desarrolladas para la encriptación en Octave Online. Estos videos incluían recomendaciones, como la consideración de redimensionar la imagen previamente en caso de demoras en la carga en Octave Online. Para realizar este ajuste, se sugería acceder a la pestaña "Manuales de uso". En la parte inferior izquierda de dicha pestaña, había una imagen que redirigía al usuario a la página iLoveIMG, donde podía modificar cómodamente el tamaño de la imagen. También se destacaba la importancia de agregar el tiempo necesario que Octave Online requería para procesar la imagen y así evitar errores de carga.

Posteriormente, se aconsejaba almacenar la imagen en la computadora con el nombre "imagen.jpg" en una ubicación específica para poder cargarla nuevamente en los archivos de Octave. Era fundamental mencionar que el código reconocía únicamente imágenes con extensión .jpg, por lo que se enfatizaba la necesidad de asegurarse de que las imágenes estuvieran en dicho formato. Finalmente, se instaba a descargar las capas RGB deseadas para cargarlas en la aplicación "Carga de Capas".

La organización se llevó a cabo de manera estructurada, presentando tutoriales específicos sobre la encriptación mediante la técnica SVD, “Creación Propia” y Carga de Capas. Además, se incluyó una breve descripción de su contenido. Adicionalmente, se agregaron botones que redirigían directamente a las pestañas correspondientes, ya sea las de encriptación o las de carga de capas RGB. Estos recursos se diseñaron con la intención de proporcionar a los usuarios una comprensión completa y facilitar su interacción con las herramientas disponibles.

Figura 28. Videos de códigos de encriptación y Carga de Capas RGB.

Se agregaron enlaces que dirigían a la página Iloveimg (Figura 29). Una de las imágenes proporcionaba acceso directo a la página principal de este sitio web, facilitando así la navegación. Además, se incorporaron dos imágenes adicionales, cada una con una función específica: la primera posibilitaba la redimensión de imágenes, mientras que la segunda simplificaba la conversión de formatos de PNG a JPG. Estos accesos directos se activaban con un simple clic en las imágenes correspondientes, agilizando el proceso de edición y mejorando la experiencia del usuario al ofrecer vías rápidas a las funciones más comunes.

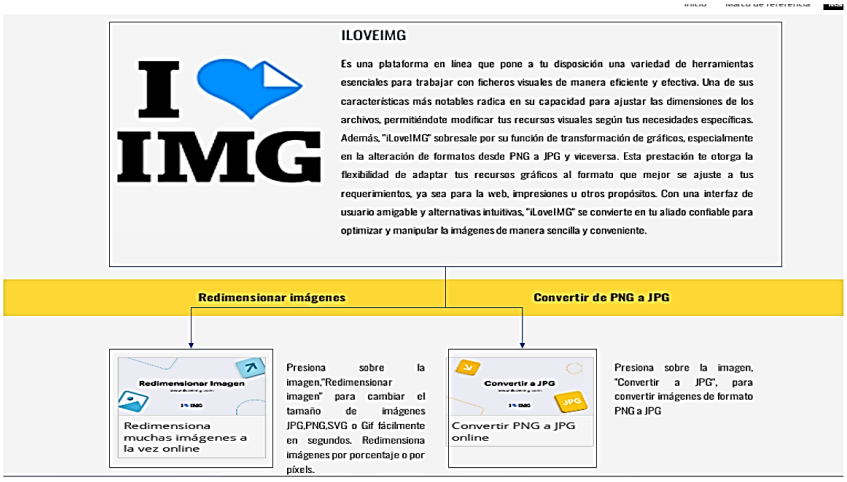


Figura 29. Enlaces sitio Web iLOVEIMG.

5.1.4 *Pestaña: Encriptación por SVD, Creación Propia y Carga de Capas RGB.*

Esta pestaña presentaba cuatro subpáginas, las cuales dirigían a aplicativos específicos. Se encontraba la sección dedicada a la Encriptación por SVD, junto con su funcionalidad de Carga de Capas RGB. También se incluía la sección de Propuesta de Encriptación por Vectores, que se complementaba con su función de Carga de Capas RGB correspondiente.

Además, en la descripción de cada pestaña, se resumían las observaciones destacadas en los videos tutoriales disponibles en la sección de Manuales de Uso. Algunas de ellas, como se visualiza en la Figura 30, aconsejaban a los usuarios redimensionar las imágenes según fuera necesario y se recomendaba tener en cuenta el tiempo requerido por Octave Online para procesar las imágenes, junto con otras sugerencias útiles.

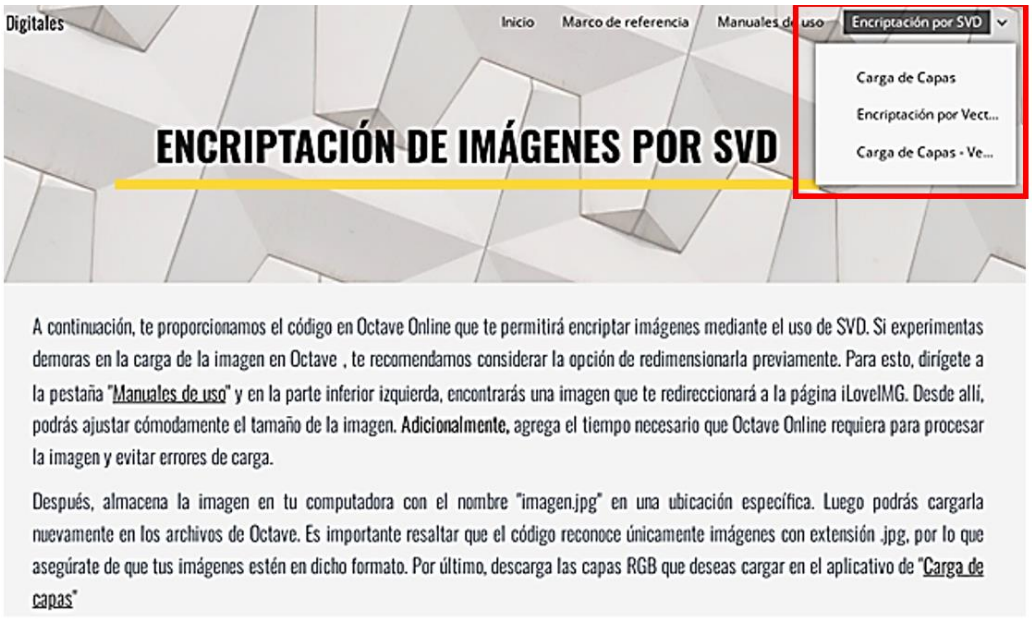


Figura 30. Descripción de páginas y subpáginas.

En las subpáginas mencionadas, se encontraban directamente en Google Sites aplicativos integrados de encriptación y carga de capas en Octave Online (Figura 31), lo que permitía a los usuarios realizar todo el proceso de manera eficiente sin salir del sitio web.

Figura 31. Códigos: Encriptación SVD, Creación Propia y Carga de Capas.

5.1.5 *Pestaña: Ejemplos de imágenes encriptadas.*

Se presentaron ejemplos de prueba en la Figura 32, que ilustraron el proceso de encriptación mediante SVD, junto con la alternativa de código basada en "vectores". Estos ejemplos utilizaron imágenes seleccionadas para demostrar el cifrado. Cada imagen original fue mostrada junto a su versión encriptada, y se incluyó un carrusel debajo de ellas para permitir una inspección más detallada de las imágenes. Esta sección proporcionó una visualización completa de los resultados de la encriptación, facilitando la comprensión y evaluación de la técnica utilizada**.**

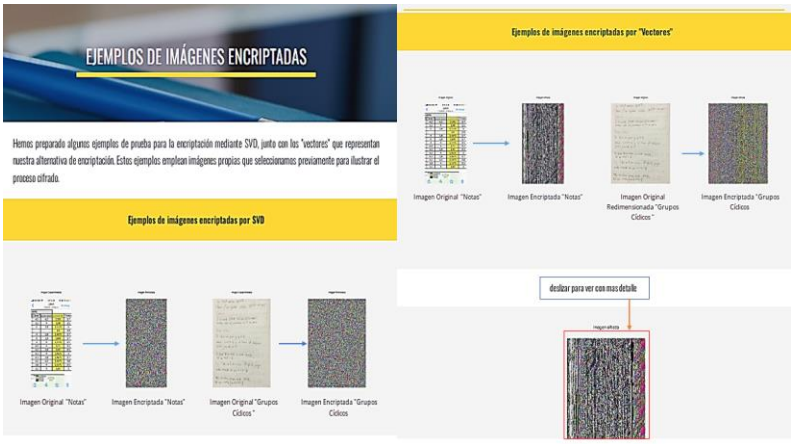


Figura 32. Ejemplos de imágenes encriptadas.

Nota. Es importante destacar que la página web está sujeta a actualizaciones y correcciones de errores que puedan surgir a medida que los usuarios interactúen y ofrezcan sugerencias, mejorando así la navegación y la experiencia de los visitantes.

Capítulo 6. Análisis de Resultados

De acuerdo con Taquía (2017), un histograma de una imagen proporciona una representación de cómo se distribuyen las intensidades de los píxeles, ya sea en color o en escala de grises, dentro de una imagen. Este gráfico permite observar el nivel de intensidad o valor de píxel asociado al color. En el espacio de color RGB, los valores de los píxeles se encuentran en el rango de 0 a 255. Al trazar el histograma con un eje X que tiene 256 divisiones (bins), se está contando de manera precisa cuántas veces aparece cada valor de píxel.

Con base en esto, se realizará el análisis experimental con histogramas para comparar imagen original y encriptada. El objetivo es obtener información sobre la distribución de los

valores de intensidad de los píxeles en ambas imágenes (Gómez et al., 2012), permitiendo así evaluar los efectos del proceso de encriptación en la estructura de las mismas.

En la Tabla 8, *Convenciones de los ejes para Histogramas Matlab*, se presentan las convenciones establecidas por Manna (2016), para la interpretación de los ejes en los histogramas de imágenes en escala de grises y a color:

Tabla 8. Convenciones de los ejes para Histogramas Matlab.

Histograma de una Imagen en Escala de Grises	**Histograma de una Imagen a Color (por cada canal RGB)**
Eje x (valor de intensidad): representa los posibles valores de intensidad en una escala de grises, que van desde 0 (color negro) hasta 255 (color blanco). esta escala define la variación de tonos de grises presentes en la imagen. Eje y (frecuencia de ocurrencia): indica cuántas veces aparece cada valor de intensidad en la imagen; es decir, representa la cantidad de píxeles en la imagen que tienen un valor de intensidad específico. cuanto mayor sea la frecuencia de ocurrencia de un valor de intensidad, mayor será la altura del pico en el histograma. El histograma de una imagen en escala de grises proporciona una visualización de la distribución de intensidades en la imagen, permitiendo identificar rápidamente si la imagen es oscura, clara o si posee un amplio rango dinámico de intensidades (Gómez et al., 2012).	Eje x (Valor de Intensidad): Al igual que en el caso de escala de grises, representa los posibles valores de intensidad para un canal de color específico: rojo, verde o azul. Estos valores varían de 0 a 255 y determinan la intensidad de cada color en el canal respectivo. Eje y (Frecuencia de Ocurrencia): Muestra cuántas veces aparece cada valor de intensidad para ese canal de color en particular. La altura de los picos en el histograma refleja la frecuencia con la que se encuentra una determinada intensidad de color en el canal correspondiente. Según Gómez et al. (2012), un histograma de una imagen a color, se obtienen tres histogramas distintos, uno para cada canal de color (rojo, verde y azul), revelando la distribución de intensidades en cada uno de estos canales individuales. Estos histogramas son esenciales para entender

	cómo están distribuidas las intensidades de color y contribuyen al análisis del balance de color en la imagen.

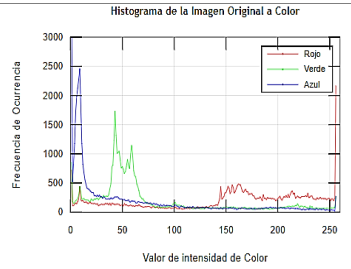
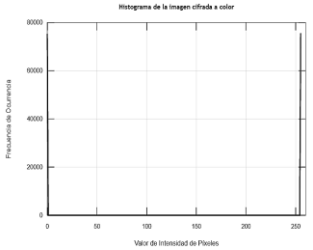
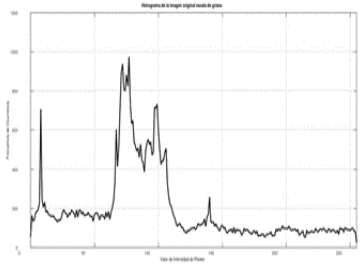
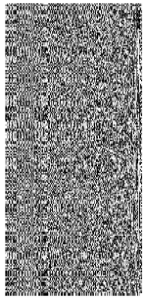
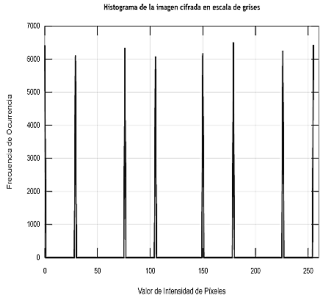
Para analizar el nivel de distorsión (Ruido) de los algoritmos propuestos (“Encriptación por SVD” y Creación Propia), se seleccionó una imagen, la cual se analizó en función de los algoritmos de encriptación. La imagen seleccionada (Figura 33) se analizó en versión color y en escala de grises; en el caso de la imagen a color: se presenta un histograma mostrando los tres canales de color RGB, luego su versión cifrada a color y su respectivo histograma. De la misma manera, se realizó el mismo procedimiento para la imagen en escala de grises

Figura 33. Imagen de prueba para realizar la encriptación

Nota: Imagen Dragón Ball Z a color, Resolución de 183 x 275 pixeles con tamaño 14.834 de bytes. Tomado de (Toei Animation, 2014).

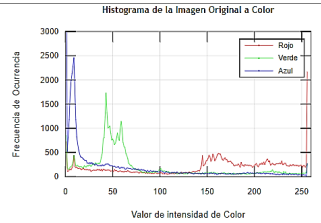
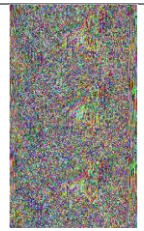
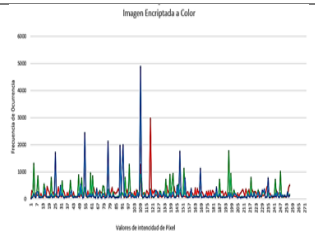
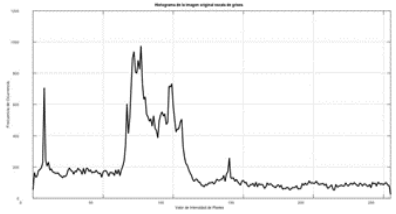
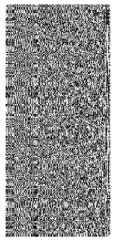
La Figura 27, fue seleccionada como la muestra para llevar a cabo diversas pruebas de encriptación y análisis de histogramas utilizando los dos códigos que han sido desarrollados. A continuación, en la Tabla 9 se detalla el resultado de la encriptación de esta imagen.

Tabla 9. Comparación de histogramas Encriptación por SVD.

Encriptación por SVD						
Tipo	**Imagen Original**	**Histograma de imagen a color**	**Imagen Encriptada**	**Histograma de imagen cifrada a color**	**Dimensiones**	**Tamaño (Bytes)**
Dragon Ball (Color)					183 x 275	14.834
Tipo	**Imagen Original (Escala de gris)**	**Histograma de imagen original (Escala de Grises)**	**Imagen Encriptada (Escala de Gris)**	**Histograma de imagen cifrada en escala de grises.**	**Dimensiones**	**Tamaño (Bytes)**
Dragon Ball (Escala de Gris)					183 x 275	13.759

Nota. Imagen original y encriptada (A color y escala de grises).

Tabla 10. Comparación de histogramas Creación Propia.

Creación Propia					
Tipo	**Imagen Original**	**Histograma de imagen a color**	**Imagen Encriptada**	**Histograma de imagen cifrada a color**	**Tamaño (Bytes)**
Dragon Ball (Color)					14.834
Tipo	**Imagen Original (Escala de gris)**	**Histograma de imagen original (Escala de Grises)**	**Imagen Encriptada (Escala de Gris)**	**Histograma de imagen cifrada en escala de grises.**	**Tamaño (Bytes)**
Dragon Ball (Escala de Gris)					13.759

Nota. Imagen original y encriptada (A color y escala de grises)

6.1 Resultados

Como resultado del análisis de los histogramas, se procedió a comparar los métodos de cifrado de imágenes denominados "Encriptación por SVD" y "Creación Propia", presentados en las Tablas 9 y 10, respectivamente. Este análisis proporcionó una comprensión detallada de cómo se distribuían las intensidades en la imagen cifrada, tanto en escala de grises como en su versión a color. En la Tabla 11 se consideraron dos categorías de análisis para esta comparación:

Tabla 11. Descripciones de categorías de Análisis de Histograma de imagen.

Categoría	Descripción
Picos en el Histograma	Identificar los picos de intensidades que indican áreas con alta concentración de píxeles en la imagen.
Áreas de Baja Intensidad	Encontrar regiones en la imagen donde las intensidades son bajas, indicando zonas oscuras o de sombras.
Áreas de Alta Intensidad	Identificar regiones en la imagen con intensidades altas, que pueden representar áreas brillantes o resaltadas.

Figura 34. Convenciones de histograma de una imagen[17]

[17] Imagen sin autor Tomada de: ¿Qué Es el Histograma en Fotografía? Frecuencia de pixeles en una imagen digital, 2021 capturethatlas.com. Recuperado 5 de julio de 2023.

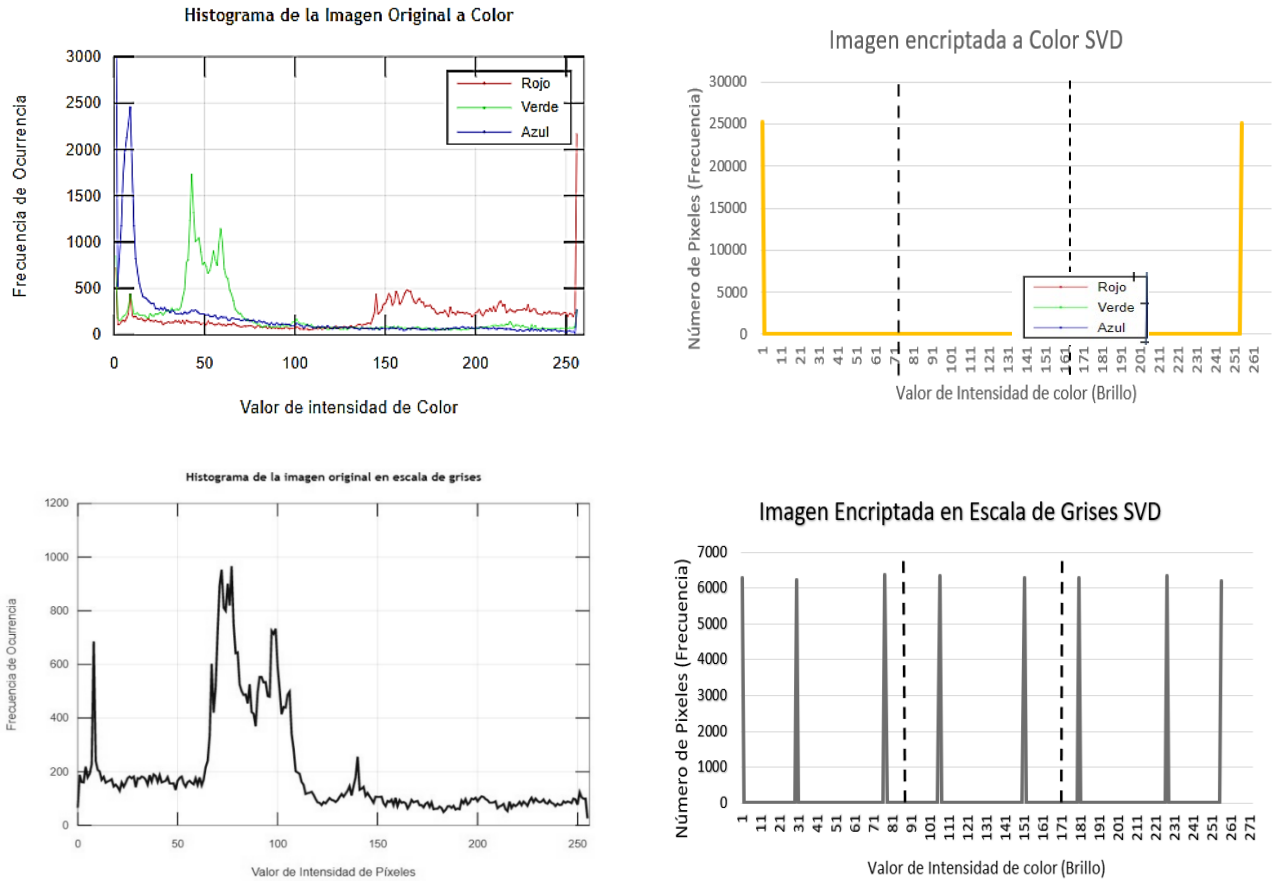


Figura 35. Histogramas: Imagen original y encriptada “SVD”.

Nota. Histogramas de la imagen a color y en escala de grises *Encriptación por SVD*

Tabla 12. Categorías método de cifrado de imágenes "Encriptación por SVD"

Categorías	Imagen color original	Imagen encriptada a color	Imagen original en escala de grises	Imagen encriptada en escala de grises
Picos en el Histograma	**Capa Roja**: Se observan dos picos prominentes en el histograma de la capa roja. El primero se encuentra en las tonalidades oscuras, con coordenadas $(x, y) = (1, 1272)$, mientras que el segundo pico está en las tonalidades claras, con coordenadas $(x, y) = (255, 2975)$. **Capa Verde**: Se distinguen cinco picos en el histograma de la capa verde. El primero se ubica en $(1, 907)$, el segundo en $(9, 423)$, el tercero en $(43, 1775)$, el cuarto en $(59, 1126)$ y el quinto en $(255, 359)$. **Capa Azul**: Presenta dos picos significativos en su histograma. El primero se registra en $(1, 4887)$ y el segundo en $(9, 2447)$.	Al analizar las capas RGB de esta imagen, se observa un comportamiento uniforme en todas ellas. Por lo tanto, se presenta el análisis de un único histograma, entendiendo que es representativo para las tres capas. Se identifican dos picos en el histograma: el primero con coordenadas $(1, 25.205)$ y el segundo con coordenadas $(255, 25120)$.	Se observan cuatro picos significativos en la imagen. El primero se registra en coordenadas $(x, y) = (9, 706)$, el segundo en $(78, 974)$, el tercero en $(100,733)$ y el último en $(141, 257)$.	Se identifican ocho picos significativos en la imagen. El primero se ubica en coordenadas $(x, y) = (1, 6280)$, el segundo en $(30, 6230)$, el tercero en $(77, 6359)$, el cuarto en $(106, 6336)$, el quinto en $(151, 6282)$, el sexto en $(180, 6292)$, el séptimo en $(227, 6347)$ y el octavo en $(255, 6199)$.
Áreas de baja y alta intensidad	1. En el primer intervalo $[\mathbf{0}, \mathbf{85}]$ de valores de intensidades de píxeles (eje x), se observa una alta frecuencia de intensidad en los tres canales RGB, indicando una tonalidad oscura. 2. En el segundo intervalo $[\mathbf{85}, \mathbf{170}]$ de valores de intensidades de píxeles (eje x), las frecuencias son bajas y no superan los 1000, mostrando una tendencia	1. En el primer intervalo $[\mathbf{0}, \mathbf{85}]$ de valores de intensidades de píxeles (eje x), se observa una alta concentración de tonalidades oscuras. 2. En el segundo intervalo $[\mathbf{85}, \mathbf{170}]$, se nota una ausencia de intensidades en tonos medios, lo que lleva la frecuencia de valores a cero.	1. En el primer intervalo $[\mathbf{0}, \mathbf{85}]$, se observa la mayor concentración de sombras en la imagen. 2. En el segundo intervalo $[\mathbf{85}, \mathbf{170}]$, se aprecia la segunda mayor concentración de tonos medios en la imagen. 3. En el tercer intervalo $[\mathbf{170}, \mathbf{255}]$, la concentración de	1. En el primer intervalo $[\mathbf{0}, \mathbf{85}]$, se observa un pico de tonalidad completamente negra y dos picos correspondientes a sombras. 2. En el segundo intervalo $[\mathbf{85}, \mathbf{170}]$, se presentan dos picos de tonalidades medias. 3. En el tercer intervalo $[\mathbf{170}, \mathbf{255}]$, se identifican dos picos de tonalidades altas y

	hacia tonalidades en escala de grises. 3. En el tercer intervalo $[\mathbf{170}, \mathbf{255}]$, las intensidades son bajas y muestran una tendencia hacia el blanco, presentando un aumento en el contraste especialmente en el canal rojo. .	3. En el tercer intervalo $[\mathbf{170}, \mathbf{255}]$, se observa una concentración en tonos blancos.	altas luces y blancos es baja.	un pico de tonalidad blanca.

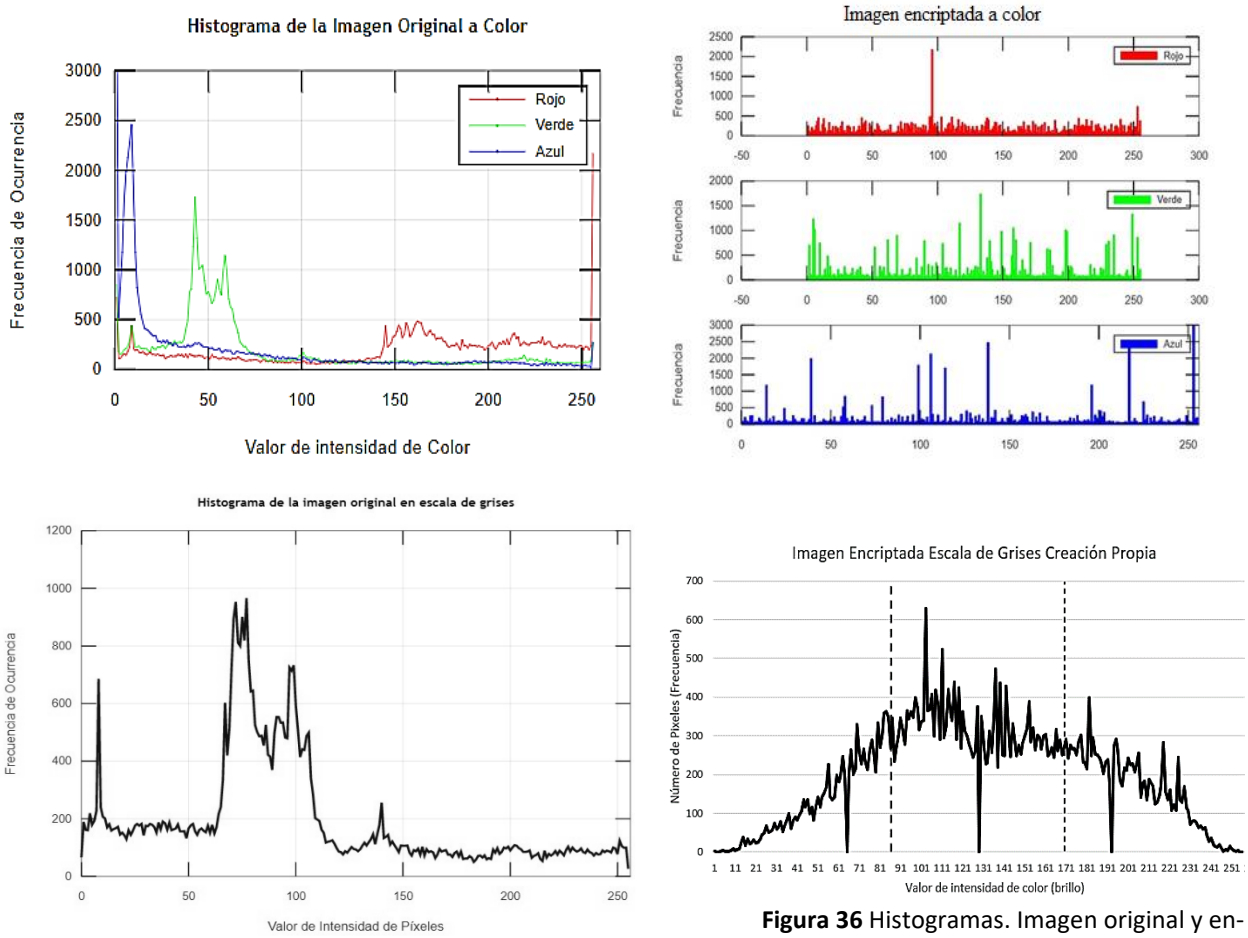


Figura 36 Histogramas. Imagen original y encriptada “Creación Propia”

Nota. Histogramas de la imagen a color y en escala de grises.

Tabla 13. Categorías método de cifrado de imágenes "Creación Propia "

Categorías	Imagen color original	Imagen encriptada a color	Imagen original en escala de grises	Imagen encriptada en escala de grises
Picos en el Histograma	**Capa Roja**: Se identifican dos picos notables en el histograma de la capa roja. El primero está presente en las tonalidades oscuras y tiene coordenadas $(x, y) = (1, 1272)$, mientras que el segundo pico se halla en las tonalidades claras con coordenadas $(x, y) = (255, 2975)$. **Capa Verde**: Muestra cinco picos. El primero tiene coordenadas $(1, 907)$, el segundo $(9, 423)$, el tercero $(43, 1775)$, el cuarto $(59, 1126)$ y el quinto $(255, 359)$. **Capa Azul**: Exhibe 2 picos significativos, el primero con coordenadas $(1, 4887)$ y el segundo $(9, 2447)$.	**Capa Roja**: Se observan dos picos significativos en el histograma, el primero con coordenadas $(109, 1272)$ y el segundo con coordenadas $(119, 2986)$. **Capa Verde**: Aunque muestra un comportamiento oscilante en el histograma, se identifican tres picos destacados en las coordenadas $(4, 1306)$, $(98, 1273)$ y $(196, 1775)$, respectivamente. **Capa Azul**: En el histograma de esta capa sobresalen tres picos: el primero con coordenadas $(54, 2447)$, el segundo en $(77, 2127)$ y el tercero en $(109, 4887)$.	Se identifican cuatro picos significativos en la imagen. El primero está en $(9, 706)$, el segundo en $(78, 974)$, el tercero en $(100, 733)$ y el último en $(141, 257)$.	El histograma muestra una acumulación hacia los colores medios, donde se encuentran los picos más altos. El pico más prominente se ubica en $(103, 629)$.
Áreas de baja y alta Intensidad	1. En el primer intervalo $[\mathbf{0}, \mathbf{85}]$ de valores de intensidades de pixel (eje x), se observa una alta frecuencia de intensidad en los tres canales RGB, reflejando una tonalidad oscura. 2. En el segundo intervalo **[85,170],** las intensidades son bajas, no superando los 1000, y tienden hacia tonalidades en escala de grises.	1. En el primer intervalo $[\mathbf{0}, \mathbf{85}]$ de valores de intensidades de píxeles (eje x), se observa un predominio del color azul en tonalidades oscuras, mientras que el rojo muestra la menor intensidad en los tonos negros. 2. En el segundo intervalo $[\mathbf{85}, \mathbf{170}]$**,** se mantiene este predominio del azul en tonalidades oscuras, pero con una frecuencia mayor. El rojo, en comparación con el primer intervalo, muestra una mayor frecuencia indicando	1. En el primer intervalo $[\mathbf{0}, \mathbf{85}]$, se observa una alta concentración de tonalidades oscuras, representando las sombras predominantes en la imagen. 2. En el segundo intervalo $[\mathbf{85}, \mathbf{170}]$, se aprecia la segunda mayor concentración de tonos medios en la imagen. 3. En el tercer intervalo	En general, el histograma abarca los tonos medios de la imagen, indicando que se capturan todos los tonos desde los oscuros hasta los blancos. Se observa una acumulación en la escala de medios tonos, representando un comportamiento de exposición neutral.

	3. En el tercer intervalo **[170,255],** la intensidad es aún más baja, similar al intervalo anterior, inclinándose hacia tonalidades blancas y mostrando un aumento de contraste en el color rojo.	una tendencia hacia los medios tonos, mientras que el verde se comporta de manera similar a los tonos oscuros. 3. En el tercer intervalo $[170,255]$, tanto el rojo como el azul reducen su frecuencia, tendiendo hacia las altas luces y blancos. El verde mantiene un comportamiento similar a los intervalos anteriores, con una frecuencia ligeramente mayor.	$[\mathbf{170}, \mathbf{255}]$, la concentración de tonalidades correspondientes a altas luces y blancos es baja.

6.2 Comentarios

Al hacer la encriptación por SVD, se evidencia que el histograma de la imagen encriptada a color muestra asimetría debido a la presencia de picos claramente definidos en diferentes rangos de intensidades (Figura 37). Este fenómeno puede estar relacionado con las características particulares de la encriptación mediante SVD, donde la transformación de la imagen original en componentes singulares puede generar una distribución desigual de intensidades. La encriptación realizada a la imagen a color en capas, se comporta de manera similar en capas separadas, es decir se altera casi de la misma forma cada capa de color RGB.

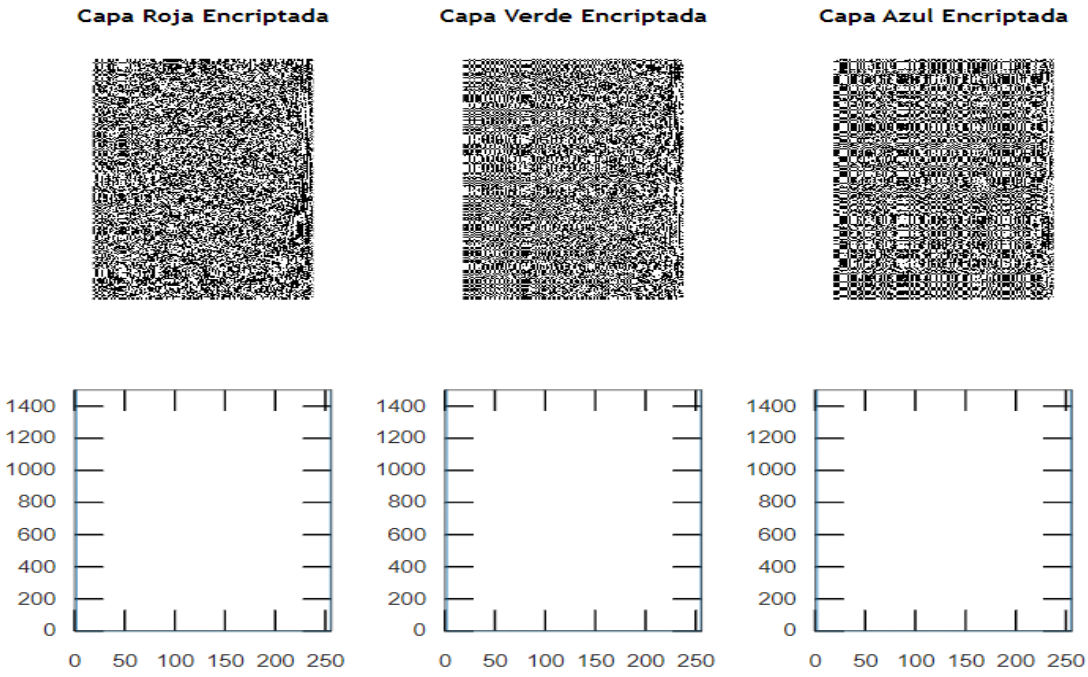


Figura 37. Imagen Encriptada por capas RGB con histograma.

Es posible inferir que la encriptación por SVD da lugar a un fenómeno en el que cada capa de color RGB de la imagen encriptada, se comporta tendiendo a escala de niveles de grises.

Al encriptar la imagen a color por el método de “creación propia”, se evidencia en la Figura 38, mayor distribución de los pixeles en las diferentes capas de tonalidad, a diferencia de la encriptación por SVD, la encriptación de cada canal RGB es diferente, esto se debe al cifrado por sustitución simple, en el que solo se desplazan, los valores que componen los vectores de cada canal, como se muestra en la Figura 38.

histograma de imagen cifrada a color por canales RGB, “Creación Propia

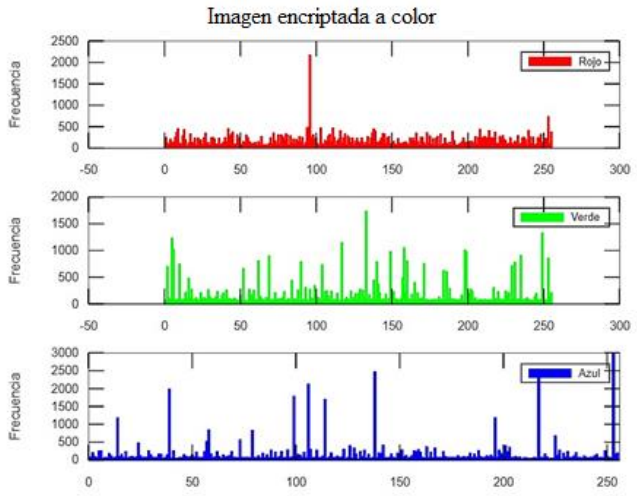
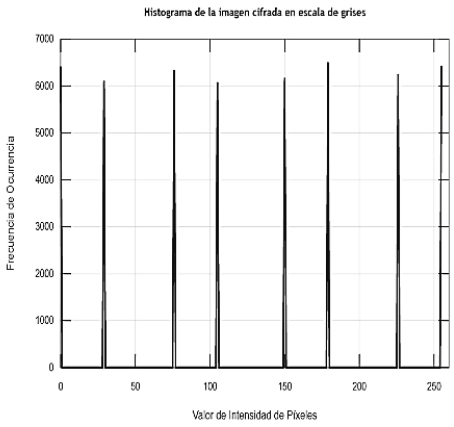


Figura 38 Histograma de imagen cifrada a color RGB.

Nota. Capas de canales RGB, “Creación Propia”

A continuacion, en la Tabla 14, se presenta la comparacion entre los dos metodos de encriptacion, aplicados a la imagen en grises, para efectos de esta compracion las escalas en x e y de los histogramas son las mismas.

Tabla 14. Comparación de métodos de cifrado: imagen en escala de grises.

Imagen cifrada por SVD grises	**Imagen cifrada por creación propia grises**
	
El comportamiento de este tipo de cifrado tiende a generar simetria entre las tonalidades de grises.	Presenta mayor concentracion en la escala de tonos medios de grises, sin embargo la frecuencia de pixeles es menor que la encriptacion por SVD.

Conclusiones

Para finalizar el estudio que hemos expuesto a lo largo del documento, las conclusiones abordan tres aspectos fundamentales: el grado de cumplimiento de los objetivos planteados en el trabajo, las consideraciones finales derivadas de los análisis de los resultados obtenidos y los aportes que este trabajo de grado proporciona para la formación futura como docentes.

Relativas a los Objetivos

A continuación, en la Tabla 15 se presenta el nivel de desarrollo del objetivo general y de los objetivos específicos.

Tabla 15. Nivel de desarrollo de los objetivos

Objetivo General	
Desarrollar una solución integral para la encriptación de imágenes digitales basada en métodos convencionales de cifrado.	Se desarrollaron dos soluciones para la encriptación de imágenes digitales; la primera basada en la descomposición de valores singulares SVD y la segunda por sustitución de vectores, la cual se denominó “Creación Propia”.
Objetivos Específicos	
Consultar y apropiar los métodos convencionales de cifrado que hacen uso del álgebra lineal, seleccionando aquellos que presenten mayor accesibilidad	El estudio del desarrollo del concepto de encriptación permitió comprender algunos de los métodos de cifrado más destacados a lo largo de la historia y su impacto en la sociedad. Más allá de esto, se ha hecho evidente que siempre perdurará la necesidad de ocultar o restringir información, dado que la comunicación es primordial para cualquier tipo de interacción en la vida humana. Sin esta seguridad, tanto individuos como sociedades se vuelven vulnerables a una amplia gama de amenazas, que pueden ser de índole económico, religiosa, personal, político, militar, entre otras. Se puede decir que su relevancia perdurará en el tiempo, asegurando que la comunicación y el intercambio de información se realicen de manera segura y protegida.

Mostrar la fundamentación matemática subyacente en los métodos de Hill y SVD, así como la aplicabilidad en el contexto de la encriptación de imágenes digitales.	La técnica de Descomposición en Valores Singulares (SVD), emergió como una herramienta para llevar a cabo la encriptación de imágenes. La SVD permitió entonces transformar la representación matricial de una imagen en componentes singulares, añadiendo así una capa adicional de seguridad que hace más difícil el acceso no autorizado y la manipulación indebida de la información visual. Este método de factorización algebraico posibilitó llevar un objeto como la imagen digital, al área de las matemáticas.
Examinar algunos lenguajes de programación para buscar idoneidad en el método de cifrado de imágenes digitales	Durante el análisis de los distintos softwares, se identifican dos inconvenientes. El primero se refiere a la dificultad en cuanto a la accesibilidad y la usabilidad, ya que muchos de ellos no están disponibles en versión gratuita. El segundo inconveniente está relacionado con la accesibilidad de una versión ejecutable en línea que permita el desarrollo del sitio web. Por esta razón, se concluyó que Octave Online era el software más viable para la propuesta, dado que integra bibliotecas en su versión en línea, lo cual representa una ventaja significativa, al no requerir la descarga de extensiones adicionales.
Desarrollar una página web interactiva que brinde a los usuarios la capacidad de cargar imágenes, aplicar la encriptación mediante los algoritmos desarrollados, visualizar las capas RGB resultantes y acceder a manuales de uso detallados.	La creación de un sitio web especializado en la encriptación de imágenes digitales se diseñó pensando en la accesibilidad y la comprensión del usuario. La estructura de navegación intuitiva pretendió que los usuarios puedan explorar fácilmente las distintas secciones, desde los conceptos básicos hasta las aplicaciones prácticas. Se adjunta enlace del Sitio Web *Encriptación de Imágenes Digitales*: https://sites.google.com/view/encriptacion-de-imagendi/inicio

Relativas a los resultados del análisis de los histogramas

El análisis de los histogramas brindó una comprensión profunda de cómo se distribuyen las intensidades de píxeles en la imagen original y encriptada. La representación gráfica de la frecuencia de ocurrencia de intensidades, reveló patrones y características específicas de la encriptación, permitiendo inferir su influencia en la estructura y contenido de las imagen. El fenómeno observado en la asimetría y distribución de intensidades en las imágenes encriptadas, especialmente en la técnica de encriptación por SVD, señala una transformación significativa

hacia valores de escala de grises. Este resultado tiene implicaciones sustanciales en la percepción visual y el análisis de las imágenes cifradas, demostrando que el proceso de encriptación modifica profundamente la representación de colores originales. Esta comprensión es esencial para abordar de manera efectiva la seguridad de la información en el contexto de imágenes digitales

Relativas a la formación docente

La exploración del tema sobre la encriptación de imágenes digitales nos sumergió en un proceso de indagación profunda. Al principio, al buscar cómo desarrollar códigos de encriptación específicamente para imágenes, nos adentramos en un territorio inicialmente desconocido y algo confuso. Para nosotros, era la primera vez que nos adentrábamos en temas de desarrollo de software. La complejidad intrínseca y la necesidad de comprender los principios subyacentes de la programación de códigos de cifrado nos desafiaron a explorar, aprender y experimentar conceptos de álgebra lineal hasta entonces desconocidos, aunque no complejos, como la descomposición en valores singulares (SVD).

Esta inmersión nos instó a adquirir un profundo entendimiento de los algoritmos criptográficos, entender cómo se aplican a imágenes digitales y discernir sus implicaciones en términos de viabilidad. Nos vimos impulsados a consultar y comprender las estructuras complejas de datos y los métodos criptográficos para poder diseñar soluciones efectivas y robustas. Este desafío nos motivó a profundizar en el dominio técnico y a perfeccionar nuestras habilidades y conocimientos previos para abordar con éxito la encriptación de imágenes digitales.

A medida que avanzábamos en este estudio, pudimos aplicar y adaptar estos conocimientos de manera creativa, para abordar los desafíos específicos que se nos presentaban. Esta experiencia nos enseñó a navegar por la incertidumbre inicial y a transformarla en un conocimiento

aplicable. Además, nos permitió desarrollar habilidades valiosas para enfrentar problemas complejos y desconocidos en el ámbito de la encriptación. Estamos convencidos que esta experiencia ha fortalecido nuestra capacidad para abordar venideros desafíos en nuestra trayectoria como futuros profesores de matemáticas e investigadores.

El desarrollo de este proyecto de encriptación no solo ha fortalecido nuestra formación tanto profesional como académica, sino que también ha representado una puerta hacia un enfoque de aplicabilidad del álgebra lineal que puede conectar de manera significativa la enseñanza y aprendizaje de las matemáticas con la educación. En particular, el método de encriptación basado en la Descomposición de Valores Singulares (SVD), abre la oportunidad de proponer una estrategia de enseñanza del álgebra lineal. Este enfoque, no solo podría ampliar la comprensión de los estudiantes sobre conceptos matemáticos abstractos, sino que a priori, brindaría una perspectiva práctica sobre la aplicabilidad de tales conceptos en la seguridad de la información. Abordar una propuesta de integración de la encriptación de imágenes como recurso dinámico y atractivo para promover habilidades y competencias matemáticas en los estudiantes, podría ser sumamente enriquecedor. Esta iniciativa no solo ilustraría cómo la teoría algebraica se traduce en soluciones prácticas, sino que también podría estimular el interés de los estudiantes al mostrar la utilidad directa de las matemáticas en contextos del mundo real.

Para finalizar, reconocemos que estas propuestas de códigos asociados a la encriptación de imágenes están abiertas a discusión y mejora constante. Animamos a futuros profesores de matemáticas a retomar este trabajo con el objetivo de ajustar y perfeccionar este ámbito de la programación de los códigos expuestos en este trabajo. Invitamos a explorar nuevas

perspectivas, enriquecer las técnicas existentes y buscar soluciones innovadoras que contribuyan a fortalecer la eficacia de la encriptación de imágenes en el contexto tecnológico actual.

Referencias

Al-Husainy M. (2012). *Un nuevo método de cifrado para la seguridad de imágenes*. En t. J. Aplicación de seguridad, cap 6 pag 1-8.

Alonso M. (2009) "*Espacios de color RGB, HSI y sus generalizaciones a n- dimensiones*" [Tesis doctoral Instituto Nacional de Astrofísica, Óptica y Electrónica], Puebla, México. https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/362/1/AlonsoPeMA.pdf

Arboledas, D. (2017). *Criptografía sin secretos con Python*. Editorial RA-MA.

Burden R., Douglas J., (1985.) *Análisis Numérico*, Grupo Editorial Iberoamericana, México.

Cárdenas L. & Becerra L. (2016). *Gestión de seguridad de la información: revisión bibliográfica. Profesional De La información*, 25(6), 931–948.

Cidón A., De la torre I., Cidón E., (2011) El estándar DICOM y su nivel de implantación en Europa, Revista de salud, vol 7,N° 27, Valladolid, España.

Recuperado de: https://dialnet.unirioja.es/servlet/articulo?codigo=4204205.

De la Fraga L. (2001) Curso: *Procesamiento Digital de Imágenes*, [Centro de investigación y de Estudios Avanzados, Instituto politécnico Nacional], México.

Dey S., (2012). *Un método de cifrado de imágenes: Estándar de cifrado de imágenes avanzado* SD: SD-AIES. En t. J. Forenses digitales de ciberseguridad., 1: 82-88.

El Abbadi, N. K., Mohamad, A. & Abdul-Hameed, M. (2014). *Image encryption based on singular value decomposition.* Journal of Computer Science, 10(7), 1222-1230. https://doi.org/10.3844/jcssp.2014.1222.1230

Elizondo, JE y Maestre, LP (2005). *Fundamentos de procesamiento de imágenes*. [Mexicali: Universidad Autónoma de Baja California.] https://fundamentos-de-procesamiento-de-imagenes-digitales.pdf

Enayatifar, R. y Abdullah A., (2011). *Seguridad de la imagen mediante algoritmo genético*. [Actas de la Conferencia Internacional sobre Modelado de Computadoras y Software], (CFM' 11), Press, Singapur, págs: 198-203.

Fabro M., (2020), *muere Russel Kirsch, creador del píxel y la primera imagen digital* Revista Gaceta UNAM, México

Ferraris, J. (2018). Patrones básicos de Navegación en Apps Móviles. Medium.

Gobierno electrónico de Uruguay (2019) *¿Cómo evaluar y comparar software?*, Agesic. https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/comunicacion/publicaciones/evaluar-comparar-software

Gómez C., Moreno E, Chávez P. (2012). *Aplicación de tresholding sobre histogramas y filtros para croma keying usando Matlab.* https://www.researchgate.net/publication/277996995_Aplicacion_de_tresholding_sobre_histogramas_y_filtros_para_chroma_keying_usando_matlab.

Gutiérrez, D. (2011). Casos de uso Diagramas de Casos de Uso. Gutiérrez, Demián, 1, 45.

Hernández L. Torrero P. Hernández V. (2014). Mapas Mentales–Mapas conceptuales diagramas de flujo y esquemas. [Archivo PDF]

Hill L. (1929), *Criptografía en un Alfabeto Algebráico*, The American Mathematical Monthly, vol. 36, núm. 6. págs. 306-312. http://www.upd.edu.mx/PDF/Libros/Mapas.pdf

https://www.gaceta.unam.mx/muere-russel-kirsch-creador-del-pixel -y-la-primera-imagen-digital/#

Mamani C., (2018) *Imagen Diseño de página: Información oculta en imágenes* [Tesis doctoral] http://repositorio.umsa.bo/xmlui/handle/123456789/17487

Manene L., (2011). *Los diagramas de flujo: su definición, objetivo, ventajas, elaboración, fases, reglas y ejemplos de aplicaciones* https://actualidadempresa.com/diagramas-de-flujo-definicion-objetivo-ventajas/

Manna, A. (2016). Introducción al procesamiento de imágenes con Matlab 1era Parte. Buenos Aires [Archivo PDF] Introduccion_al_Procesamiento_de_Imagenes_con_Matlab-libre.pdf

Mitnick K.; Simon W.; Wozniak S. (2003). *The art of deception: Controlling the human ele-ment of security. Indianapolis* Wiley Publishing. ISBN: 978 0764542800 http://sbisc.ut.ac.ir/wp-content/uploads/2015/10/mitnick.pdf

Molina G., Palacios O., Torres G, (2019). Compresión y encriptación de información utilizando SVD, Universidad de Guanajuato, UG veranos de la ciencia, México.

Nota. Adaptado de: *¿Qué Es el Histograma en Fotografía? Frecuencia de pixeles en una imagen digital*, (2021), capturethatlas.com. (https://capturetheatlas.com/es/histograma-fotografia/) CC BY 2.0

Operadores y caracteres especiales de MATLAB - MATLAB & Simulink - MathWorks América Latina. (s. f.). https://la.mathworks.com/help/matlab/matlab_prog/matlab-operators-and-special-characters.html.

Rojas A., García F. (2020). *Evaluación del pensamiento computacional para el aprendizaje de programación de computadoras en educación superior.* Revista de Educación a Distancia, 20(63) https://doi.org/10.6018/red.409991.

Saavedra L. (2003). *La historia de la imagen o una imagen para la historia. Cuicuilco Revista De Ciencias Antropológicas*, 10 (29), 197–205. Recuperado a partir de https://revistas.inah.gob.mx/index.php/cuicuilco/article/view/434

Taquía G. (2017). *El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital*. Interfaces, 10 (0-10) https://doi.org/10.26439/interfases2017.n10.1767

Toei Animation (Ed.) (2014). *Dragon Ball Z. Crunchyroll*. Recuperado el 11 de mayo de 2022, de https://www.crunchyroll.com/imgsrv/display/thumbnail/480x720/catalog/crunchyroll/35e4ac6339f5fdcc164160a5755790cd.jpe.

Velandia L., Rivera S., Giraldo F,. (2020) sistema de cifrado para seguridad en el manejo de imágenes médicas tipo Dicom, Universidad Distrital Francisco José de Caldas, Bogotá D.C. Recuperado de: https://repository.udistrital.edu.co/bitstream/handle/11349/28082/VelandiaMenesesLeonardo.pdf?sequence=1&isAllowed=

Viguer Güémez, A. (2020, 13 de noviembre). *Desarrollo e implementación de una aplicación informática con Matlab para la encriptación fractal de las imágenes basada en el Cifrado de Hill*. [Tesis de maestría] https://webcion.com/paginas-web/proceso-de-diseno-de-sitios-web/ .

Anexos

Anexo 1. Códigos asociados a la Encriptación de Imágenes

Programación código Encriptación por SVD. (versión ejecutable)

```
disp("Encriptar por SVD");
disp("")

disp("Nota importante: Recuerda descargar las capas de las imágenes.");

im = imread("imagen.jpg");

I = double(im);

R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);

[U1 S1 V1] = svd(R);
[U2 S2 V2] = svd(G);
[U3 S3 V3] = svd(B);

[f,c]= size(S1);

llave1 = (ones(f,c).*(100000)).*(2).*(3).*(5).*(7).*(11);
llave2 = (eye(f,c).*(100000)).*(2).*(3).*(5).*(7).*(11);

Sf1= S1+llave1+llave2;
Sf2= S2+llave1+llave2;
Sf3= S3+llave1+llave2;

deimcR = U1*(Sf1-llave1-llave2)*V1';
deimcG = U2*(Sf2-llave1-llave2)*V2';
deimcB = U3*(Sf3-llave1-llave2)*V3';

dese(:,:,1)=deimcR;
dese(:,:,2)=deimcG;
dese(:,:,3)=deimcB;

dlmwrite('deseR.png',deimcR)
dlmwrite('deseG.png',deimcG)
dlmwrite('deseB.png',deimcB)
dlmwrite('deseR.jpg',deimcR)
```

```
dlmwrite('deseG.jpg',deimcG)
dlmwrite('deseB.jpg',deimcB)

imcR = U1*Sf1*V1;
imcG = U2*Sf2*V2;
imcB = U3*Sf3*V3;

final(:,:,1)=imcR;
final(:,:,2)=imcG;
final(:,:,3)=imcB;

dlmwrite('finalR.jpg',imcR)
dlmwrite('finalG.jpg',imcG)
dlmwrite('finalB.jpg',imcB)
dlmwrite('finalR.png',imcR)
dlmwrite('finalG.png',imcG)
dlmwrite('finalB.png',imcB)

figure, imshow(uint8(final));
title('Imagen Encriptada')
figure, imshow(dese/255);
title('Imagen Desencriptada')
```

Programación código Creación Propia. (versión ejecutable)

```
disp("Creación Propia");
disp("")
f = imread("imagen.jpg");
X = double(f);
XR = X(:,:,1);
XG = X(:,:,2);
XB = X(:,:,3);

dlmwrite('oriR.png',XR)
dlmwrite('oriG.png',XG)
dlmwrite('oriB.png',XB)
dlmwrite('oriR.jpg',XR)
dlmwrite('oriG.jpg',XG)
dlmwrite('oriB.jpg',XB)

[n1,n2] = size(XR);

vr = [];
vg = [];
vb = [];
```

```
for i=1:n1
    for j=1:n2
        vr = [vr;XR(i,j)];
        vg = [vg;XG(i,j)];
        vb = [vb;XB(i,j)];
    end
end

clave = randperm(256);

vr_cifrado = clave(vr+1);
vg_cifrado = clave(vg+1);
vb_cifrado = clave(vb+1);

XR_cifrado = reshape(vr_cifrado,n1,n2);
XG_cifrado = reshape(vg_cifrado,n1,n2);
XB_cifrado = reshape(vb_cifrado,n1,n2);

X_cifrado(:,:,1) = XR_cifrado;
X_cifrado(:,:,2) = XG_cifrado;
X_cifrado(:,:,3) = XB_cifrado;

dlmwrite('capaR.png',XR_cifrado)
dlmwrite('capaG.png',XG_cifrado)
dlmwrite('capaB.png',XB_cifrado)
dlmwrite('capaR.jpg',XR_cifrado)
dlmwrite('capaG.jpg',XG_cifrado)
dlmwrite('capaB.jpg',XB_cifrado)

figure, imshow(uint8(X));
title('Imagen original');
figure, imshow(uint8(X_cifrado));
title('Imagen cifrada');
```

Anexo 2. Código asociado al Procesamiento de Imágenes

Programación código Carga de Capas. (versión ejecutable)

```
disp("Carga de capas");
disp("")
while true
    disp("Bienvenido al programa de carga de imágenes");
    disp("")
     disp("Por favor seleccione una opción:");
    disp("")
    disp("1. Cargar imagen original");
    disp("")
    disp("2. Cargar imagen encriptada");
     disp("")
    disp("3. Salir");
    disp("")
    opcion = input("Seleccione una opción: ");

    switch opcion
        case 1
            disp("Ingrese las capas de la imagen desencriptada");

            nombre_R = input("Ingresa el nombre del archivo de la capa R (la
extensión debe ser .png o .jpg): ", 's');
            disp("")
            nombre_G = input("Ingresa el nombre del archivo de la capa G (la
extensión debe ser .png o .jpg): ", 's');
            disp("")
            nombre_B = input("Ingresa el nombre del archivo de la capa B (la
extensión debe ser .png o .jpg): ", 's');
            disp("")

            R = dlmread(nombre_R);
            G = dlmread(nombre_G);
            B = dlmread(nombre_B);

            I = cat(3, R, G, B);


            imshow(uint8(I));
            title('Imagen desencriptada cargada')

        case 2
            disp("Ingrese las capas de la imagen encriptada");

            nombre_R = input("Ingresa el nombre del archivo de la capa R (la
extensión debe ser .png o .jpg): ", 's');
```

```
            disp("")
            nombre_G = input("Ingresa el nombre del archivo de la capa G (la
extensión debe ser .png o .jpg): ", 's');
            disp("")
            nombre_B = input("Ingresa el nombre del archivo de la capa B (la
extensión debe ser .png o .jpg): ", 's');
            disp("")

            % Leer las matrices de las capas R, G y B a partir de los archi-
vos
            R = dlmread(nombre_R);
            G = dlmread(nombre_G);
            B = dlmread(nombre_B);

            % Verificar si se tienen las tres capas
            if isempty(R) || isempty(G) || isempty(B)
                disp("Faltan capas para obtener la imagen completa.");
            else
                % Combinar las matrices de las capas R, G y B para formar la
imagen encriptada
                I = cat(3, R, G, B);

                % Mostrar la imagen encriptada
                imshow(uint8(I));
                title('Imagen encriptada cargada')
            end

        case 3
            disp("¡Gracias por usar el programa, esperamos volvernos a
ver!");
            return;

        otherwise
            disp("Opción no válida.");
    end

    respuesta = input("¿Desea cargar otra imagen? (si/no): ", 's');
    if respuesta == "no"
        disp("Gracias por usar el programa.");
        break; % Salir del bucle while
    end
end
```

Anexo 3. Código asociado a la creación de histogramas de imágenes

Programación código Histo_Análisis. (versión ejecutable)

```
I_gris = rgb2gray(uint8(I));
original_hist = imhist(I_gris);
```

```
final_gris = uint8(final);
final_gris = rgb2gray(final_gris);
encrip_hist = imhist(final_gris);

figure;
subplot(2,1,1);
imshow(I_gris, []);
title('Imagen Original (Escala de Grises)')

subplot(2,1,2);
plot(original_hist);
title('Histograma de la Imagen Original (Escala de Grises)')

figure;
subplot(2,1,1);
imshow(final_gris, []);
title('Imagen Encriptada (Escala de Grises)')

subplot(2,1,2);
plot(encrip_hist);
title('Histograma de la Imagen Encriptada (Escala de Grises)')
```

Anexo 4. Enlace Sitio Web Encriptación de Imágenes Digitales

Sitio Web: ***https://sites.google.com/view/encriptacion-de-imagendi/inicio***

Código QR asociado al Sitio Web Encriptación de Imágenes Digitales.