MANUAL DEL SOFTWARE: PROYECTO O2, ESTEGANOGRAFÍA.

SOBRE EL PROGRAMA

El programa del “Proyecto 02, Esteganografía” fue concebido para permitirle al usuario ocultar o revelar mensajes dentro de imágenes mediante el método de esteganografía de LSB (Least Significant Bit). Este programa permite a los usuarios incorporar y extraer información oculta de manera discreta, sin alterar visiblemente la calidad de la imagen original. Además, dado que LSB-Encode maneja información que puede ser sensible y podría tener implicaciones en privacidad y seguridad, el uso responsable del software requiere adherirse a marcos legales y normativas aplicables en México.

Este manual proporciona una descripción general de las funciones de LSB-Encode y orientaciones específicas para cumplir con dichos marcos, junto con instrucciones paso a paso para realizar tareas como la inserción y recuperación de mensajes ocultos en imágenes digitales.

LEGISLACIÓN Y LINEAMIENTOS APLICABLES

Tratados internacionales

El proyecto se alinea con tratados internacionales relevantes para la protección de datos y la ciberseguridad. En este sentido, el Convenio 108 del Consejo de Europa es un referente central al que México se ha adherido, estableciendo estándares de protección de datos aplicables a sistemas que procesan información personal. Si bien el proyecto actual no recoge grandes volúmenes de datos personales, la implementación de funcionalidades que impliquen almacenamiento o uso de información de los usuarios debe seguir estas pautas. La GDPR (Reglamento General de Protección de Datos de la Unión Europea) es igualmente importante en caso de que el proyecto sea accesible por ciudadanos de la Unión Europea, estableciendo requerimientos rigurosos para el manejo de datos personales. El T-MEC, por su parte, es clave para asegurar que el proyecto cumpla con las disposiciones de protección de datos de los Estados miembros. El Tratado de Comercio Digital del T-MEC también fomenta la seguridad y el uso responsable de las tecnologías digitales, garantizando que las plataformas online respeten la privacidad de los usuarios.

A continuación, se destacan algunos marcos legales y normativos a los que nos apegamos que también son de relevancia:

1. **Convención de Berna para la Protección de las Obras Literarias y Artísticas**
   Este tratado establece derechos de autor para obras creativas, incluidas las imágenes digitales. Cualquier modificación en una imagen, incluso a nivel de los bits menos significativos, puede considerarse una derivación de la obra original, lo que requeriría la autorización explícita del titular de los derechos. Por lo tanto, es esencial asegurarse de que el uso de imágenes en el proceso de esteganografía no infrinja estos derechos.
2. **Acuerdo sobre los Aspectos de los Derechos de Propiedad Intelectual relacionados con el Comercio (TRIPS)**
   Complementando la Convención de Berna, el acuerdo TRIPS establece estándares mínimos de protección para la propiedad intelectual a nivel internacional. Este acuerdo exige que los países miembros adopten medidas que protejan la integridad de las obras y prevengan el uso no autorizado de las mismas, aplicable a cualquier modificación o uso derivado de una imagen original en esteganografía.
3. **Convención sobre la Ciberdelincuencia**
   Este tratado, también conocido como Convenio de Budapest, abarca delitos relacionados con la informática, incluida la violación de la privacidad y la falsificación de documentos. La esteganografía puede caer bajo esta regulación si se utiliza con fines ilícitos, como la ocultación de datos en actividades criminales. De ahí que el software de esteganografía deba usarse bajo un marco ético y legal, y debe prohibirse para fines ilegales.

Leyes
Además de la Ley Federal de Protección de Datos Personales en Posesión de los Particulares (LFPDPPP), existen otras leyes mexicanas y estándares internacionales que son pertinentes:

- **Ley de Protección al Consumidor**: Es esencial para asegurar que el software sea claro y transparente en la prestación de servicios. Esto implica informar de manera comprensible al usuario sobre los términos de uso, políticas de privacidad y el manejo de la información, así como ofrecer un servicio que cumpla con los estándares de calidad y protección al consumidor.
- **Ley de Firmas Electrónicas**: Si el proyecto llegara a incorporar un sistema de autenticación o firma digital para el manejo seguro de datos, debe cumplir con la Ley de Firma Electrónica Avanzada en México. Esta ley garantiza que las firmas digitales tengan el mismo valor que las firmas manuscritas y establece la autenticidad e integridad de las transacciones electrónicas.
- **Ley de Seguridad Nacional**: Dado que el proyecto puede implicar el cifrado de información, es importante que se adhiera a las disposiciones relacionadas con la ciberseguridad y el uso de tecnologías de cifrado,

asegurando que no se emplee de forma que pueda vulnerar la seguridad nacional.

Códigos
El proyecto sigue principios éticos y de responsabilidad profesional. Los códigos de ética relevantes incluyen:

- **Código de Ética Profesional del Ingeniero en Sistemas Computacionales**: Subraya la importancia de la responsabilidad en el desarrollo de software seguro, que protege la privacidad y garantiza la seguridad de los usuarios. En este proyecto, se han implementado buenas prácticas de programación, como la validación de entradas, la protección de rutas de archivos y el uso de bibliotecas seguras para evitar la exposición de datos.
- **Código de Conducta de la Industria de Software**: Recomienda el desarrollo de software que cumpla con estándares éticos y de seguridad, asegurando la transparencia en el manejo de datos y el cumplimiento de las normativas legales.

El proyecto también sigue principios de programación segura, incluyendo el uso de cifrado para proteger los datos durante los procesos de codificación y decodificación.

Reglamentos
Además del Reglamento de la Ley Federal de Protección de Datos Personales en Posesión de los Particulares, el proyecto debe cumplir con:

- **Reglamento del Instituto Federal de Telecomunicaciones (IFT)**: Establece normas para la protección de la confidencialidad de los datos en las telecomunicaciones. La implementación de HTTPS y otros métodos de cifrado es esencial para garantizar la integridad de la información transmitida entre el usuario y el sistema.
- **Reglamento de la Ley de Protección al Consumidor**: Detalla la obligación de las plataformas de proporcionar información clara y accesible sobre los servicios y la protección al usuario. Debe haber un compromiso de transparencia en la prestación de servicios digitales y la garantía de la seguridad de la información.

GENERALIDADES

Estructura y arquitectura del proyecto.

Estructura del paquete del proyecto:

- Proyecto2_Esteganografia/
    - include/
        - Codificador.h
        - Decodificador.h
        - Esteganografia.h
        - ImagenPNG.h
        - TextoABits.h
    - src/
        - images/
        - Codificador.cpp
        - Decodificador.cpp
        - Esteganografia.cpp
        - ImagenPNG.cpp
        - TextoABits.cpp
        - main.cpp
    - test/
        - File_test/
            - test.jpeg
            - test1.txt
            - test10.txt
            - test2.txt
            - test3.jpeg
            - test3.txt
            - test4.png
            - test4.txt
            - test5.png
            - test5.txt
            - test6.txt
            - test7.txt
            - test8.txt
            - test9.txt
        - Test.cpp
    - CMakeLists.txt
    - README.md
    - run.sh
    - uninstall.sh

Arquitectura general:

La aplicación, desarrollada íntegramente en C++, presenta una arquitectura de backend con una lógica sólida que sigue el patrón de Esteganografía en imágenes por el método LSB, el cual es bastante eficiente. El backend, en general,

se encarga de todas las operaciones que realiza el programa (esconder y revelar el texto).

Opera directamente desde la terminal o línea de comandos. Esta elección tecnológica permite un control preciso sobre el flujo de ejecución y facilita la integración con otros sistemas a través de interfaces de línea de comandos.

Dependencias externas:

El programa utiliza dependencias que son escenciales para su correcto funcionamiento. La primera, “OpenCV”, se encarga de manejar imágenes que le sean proporcionadas para manipularlas por medio de matrices de bytes, haciendo que sea posible sobreescribir bits de esta matriz para el correcto funcionamiento de la Esteganografía. La segunda es CMake, una herramienta necesaria para generar los archivos de configuración adecuados para construir el proyecto y gestionar de forma eficiente las dependencias externas. CMake simplifica el proceso de configuración, ya que permite definir las rutas y parámetros de compilación para OpenCV y otras bibliotecas adicionales de manera flexible.

La tercera dependencia es Google Test (GTest), que se utiliza para realizar pruebas unitarias. GTest proporciona un marco de pruebas robusto para verificar el correcto funcionamiento de las diferentes partes del código y garantizar que cualquier cambio o mejora en el programa no afecte el comportamiento esperado. Al incluir GTest, el proyecto permite implementar pruebas automatizadas que ayudan a validar funciones clave, especialmente en los módulos de manipulación de imágenes y en cualquier algoritmo de procesamiento de datos.

Para el fácil uso del programa

El archivo `run.sh` ha sido programado para facilitar la ejecución del programa con un solo comando. Este script automatiza el proceso de configuración y ejecución, eliminando la necesidad de ejecutar múltiples pasos manualmente. Para instalar las dependencias necesarias para el funcionamiento del programa, simplemente hay que ejecutar un comando en la terminal, asegurándose previamente de tener **CMake** instalado en el sistema.

Si se desea desinstalar las dependencias previamente instaladas, puede hacerse también al alcance de un sólo comando. Esto asegura que el usuario no encuentre complicaciones al momento de usar el programa.

Responsabilidad del Usuario

El uso del programa de esteganografía es responsabilidad exclusiva del usuario. Al utilizar este software, el usuario se compromete a hacerlo de acuerdo con las leyes y regulaciones locales e internacionales relacionadas con la protección de la propiedad intelectual, la privacidad de los datos y la prevención de actividades ilegales (véase todo el apartado de Legislación y Lineamientos Aplicables).

El usuario es responsable de cualquier uso del programa que viole leyes nacionales o internacionales. Este software está diseñado para la manipulación de imágenes y datos de manera ética y legal. El uso del programa con fines ilícitos, como la ocultación de información en actividades ilegales, está estrictamente prohibido.

El desarrollador no se hace responsable de los actos ilegales que puedan derivarse del uso inapropiado del software. Se recomienda utilizar el programa de manera responsable y ética, respetando siempre las leyes y marcos legales aplicables.

¿Cuál es el público objetivo del software de esteganografía LSB?

El software está diseñado para ser utilizado por diferentes tipos de usuarios, cada uno con roles específicos y objetivos relacionados con la esteganografía de imágenes mediante el método LSB (Least Significant Bit). A continuación, se describe a los principales grupos a quienes está dirigido:

- **Usuarios finales:**

Son los usuarios generales que interactúan con el programa para realizar tareas de esteganografía en imágenes.

○ Objetivos: Los usuarios finales buscan ocultar o extraer información de imágenes de manera sencilla, como textos o archivos dentro de las imágenes, utilizando el método LSB.

○ Experiencia de usuario: El software está diseñado para ser fácil de usar, con una "interfaz" de terminal intuitiva que permite tanto a principiantes como a usuarios más experimentados realizar tareas de esteganografía sin complicaciones.

- **Desarrolladores o profesionales de seguridad informática:**

Son los usuarios con conocimientos técnicos que utilizan el software para implementar o probar técnicas de esteganografía, o incluso mejorar y personalizar el programa según sus necesidades.

○ Objetivos: Los desarrolladores buscan explorar, modificar y analizar la seguridad de los métodos de esteganografía, especialmente en términos de robustez y eficiencia.

○ Experiencia de usuario: El software proporciona herramientas avanzadas, documentación y scripts que permiten a los desarrolladores adaptar el programa para usos más específicos o realizar pruebas en diversos entornos.

● **Administradores o equipo técnico:** Aunque no son usuarios finales del software, los desarrolladores están encargados de crear, mantener y mejorar el sistema. Esto incluye la implementación de nuevas características, corrección de errores y optimización de rendimiento.

○ Objetivos: Asegurar que el software cumpla con los requerimientos funcionales y no funcionales del proyecto, mientras mantienen una base de código eficiente y bien estructurada.

Información de software

Tipo de software:
El software de esteganografía LSB es una herramienta diseñada para ocultar información dentro de imágenes utilizando el método de menor significancia de bits (LSB). El programa está implementado en C++ y hace uso de bibliotecas como OpenCV para el manejo de imágenes. El usuario puede encriptar un archivo de texto (.txt) dentro de una imagen o, si lo desea, desencriptar la información oculta a partir de una imagen. El proceso es sencillo: para encriptar, se necesita una imagen y un archivo de texto, y para desencriptar, solo se requiere la imagen que contiene los datos ocultos.

Requerimientos:

Requisitos Funcionales:

- **Encriptación y Desencriptación:**
  El programa permite encriptar un archivo .txt en una imagen, ocultando los datos en los bits menos significativos de los píxeles de la imagen. También es capaz de extraer los datos ocultos en una imagen previamente modificada.
- **Entrada de Datos:**
  Para encriptar, se proporcionan dos entradas: una imagen de entrada y un archivo de texto a encriptar. Para desencriptar, solo es necesaria la imagen que contiene la información oculta.
- **Validación de Datos:**
  El sistema verifica que las imágenes sean compatibles con el programa, que el texto a encriptar quepa en la imagen y que el archivo de texto tenga un formato adecuado para la encriptación. Si se proporciona una imagen para desencriptar, el programa también valida que la imagen contenga datos válidos.

Requisitos No funcionales:

- **Rendimiento:**
  El programa está diseñado para ser rápido y eficiente, utilizando las funciones de OpenCV para manipular imágenes y asegurando que las operaciones de encriptación y desencriptación sean rápidas, incluso con imágenes de tamaño considerable.
- **Compatibilidad y Portabilidad:**
  Dado que el software está escrito en C++ y depende de CMake y OpenCV, es compatible con múltiples plataformas como Linux, Windows y macOS, siempre que se tenga configurado CMake y las dependencias necesarias.
- **Seguridad:**
  El sistema asegura que los datos ocultos en las imágenes solo sean accesibles mediante el proceso de desencriptación. Sin embargo, al ser un sistema basado en la manipulación de imágenes, la seguridad no es infalible frente a ataques sofisticados. Se recomienda utilizar imágenes de tamaño considerable para dificultar la extracción de datos ocultos.

Enfoque

El enfoque de este programa de esteganografía LSB es permitir la encriptación y desencriptación de datos dentro de imágenes, utilizando el método de menor significancia de bits (LSB). Este software está diseñado para usuarios que necesitan ocultar información sensible de forma segura dentro de una imagen, como una forma de asegurar la privacidad de los datos, o para aquellos que buscan recuperar información oculta en imágenes ya modificadas.

El sistema está dirigido principalmente a usuarios comunes que buscan una herramienta simple y eficiente para ocultar o recuperar datos, sin necesidad de conocimientos técnicos avanzados. El programa ofrece una “interfaz” sencilla y clara, que permite a los usuarios encriptar un archivo de texto en una imagen con solo cargar la imagen y el archivo a encriptar. Para desencriptar, solo es necesario proporcionar la imagen que contiene los datos ocultos.

Sin embargo, también se tiene en cuenta a usuarios más avanzados, como técnicos o desarrolladores, ya que el programa está diseñado en C++ utilizando bibliotecas como OpenCV y CMake, lo que permite que los usuarios más técnicos puedan realizar ajustes o integraciones adicionales según sus necesidades.

El objetivo principal es ofrecer una herramienta confiable y accesible para la encriptación y desencriptación de mensajes, mientras se mantiene la simplicidad en la experiencia del usuario, haciendo que la manipulación de imágenes y el manejo de archivos de texto sean procesos fáciles y seguros.

Posibles mejoras

- **Interfaz de usuario mejorada**: Implementar una interfaz gráfica de usuario (GUI) amigable que facilite la interacción, ya sea a través de una aplicación de escritorio con bibliotecas como PyQt o Tkinter, o una versión web sencilla utilizando Flask o Django.
- **Compatibilidad con múltiples formatos de imagen**: Ampliar la compatibilidad para soportar diferentes formatos de imagen (como BMP, TIFF) además de PNG, incrementando así las opciones de uso y la versatilidad del proyecto.
- **Opciones de cifrado avanzado**: Incorporar algoritmos de cifrado más robustos como AES (Advanced Encryption Standard) para la codificación de datos antes de ocultarlos en las imágenes, lo que añadiría una capa de seguridad adicional.
- **Modo de prueba y análisis de errores**: Implementar un modo de prueba que permita a los usuarios verificar la calidad del mensaje decodificado y detectar posibles errores en el proceso de codificación/decodificación.
- **Soporte multilingüe**: Añadir soporte para múltiples idiomas en la interfaz y en los mensajes de ayuda para atraer a una audiencia más diversa.
- **Optimización del rendimiento**: Revisar y optimizar el código para reducir el tiempo de procesamiento en la codificación y decodificación de mensajes, lo cual es importante al manejar imágenes de mayor tamaño.
- **Funcionalidad de previsualización**: Incluir una opción de previsualización del archivo resultante antes de confirmar la codificación, permitiendo a los usuarios verificar si el archivo está intacto y si la información está adecuadamente escondida.
- **Módulo de verificación de integridad**: Agregar una función que valide la integridad del archivo de imagen y del mensaje oculto para confirmar que no ha habido alteraciones.

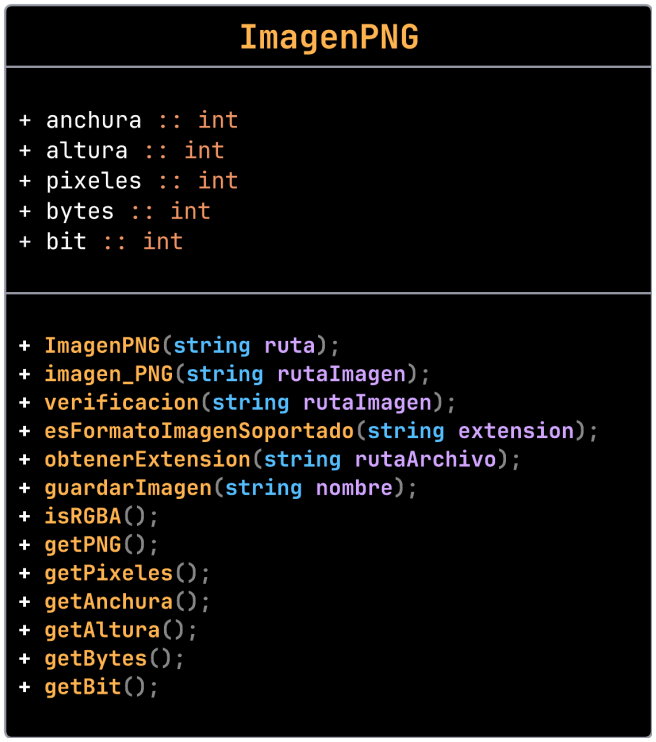
Mantenimiento

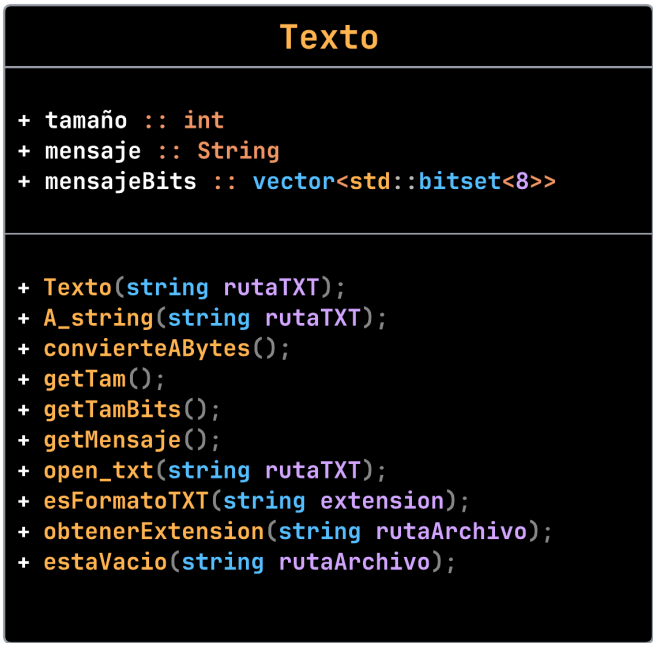
- **Actualización de bibliotecas**: Verificar periódicamente si las bibliotecas y dependencias utilizadas en el proyecto tienen actualizaciones disponibles para mantener la seguridad y la funcionalidad del software.
- **Documentación técnica**: Asegurarse de que la documentación esté actualizada y detallada, incluyendo guías de instalación, manual de usuario y descripciones del código. Esto facilitará el mantenimiento y la incorporación de nuevos desarrolladores al proyecto.
- **Pruebas unitarias y de integración**: Desarrollar y mantener un conjunto completo de pruebas unitarias y de integración para garantizar que las nuevas actualizaciones o funcionalidades no rompan el código existente.

- **Monitoreo de seguridad**: Implementar revisiones de seguridad periódicas para identificar y corregir posibles vulnerabilidades, como filtraciones de datos y fallos en los protocolos de cifrado.
- **Gestión de logs**: Integrar un sistema de registro de eventos (logging) que permita rastrear la actividad del programa, identificar errores y facilitar la depuración.
- **Mantenimiento del menú de opciones**: Revisar y mejorar el menú de opciones de la clase principal Esteganografia para mantenerlo funcional y adaptable a las nuevas características añadidas.
- **Soporte y resolución de problemas**: Mantener un canal de comunicación abierto con los usuarios del proyecto para recibir retroalimentación y resolver problemas rápidamente. Esto puede incluir un correo de soporte, un foro o una sección de FAQ en línea.
- **Modularización del código**: Verificar que el código esté bien modularizado para facilitar futuras expansiones o modificaciones, como la adición de nuevos métodos de esteganografía o la mejora de los existentes.
- **Refactorización periódica**: Realizar revisiones del código para identificar áreas que podrían beneficiarse de una mejor estructura o simplificación, mejorando así la eficiencia y la legibilidad del proyecto.
- **Soporte a nuevas tecnologías**: Mantenerse al tanto de nuevas bibliotecas o técnicas de esteganografía que podrían ser integradas para mantener el proyecto actualizado con los últimos avances en la materia.

Diagramas del proyecto

- **Clases**:





Decodificador

```
+ decode(ImagenPNG imagen);
+ guardatxt(string nombreFile, string mensaje);
+ generaMensaje(vector<std::bitset<8>> bytes);
+ bytesCaracter(bitset<8> byte);
+ verificaFinal(std::bitset<8> byte);
+ generaByte(int i, Mat PNG);
```

Esteganografia

```
+ cargarImagen(string ruta);
+ cargarTexto(string ruta);
+ menu();
```

Codificador

```
+ encode(ImagenPNG imagen, Texto mensaje, string nombrePNG);
+ mensajeaBytes(Texto &mensaje);
+ insertaByte(bitset<8> byte, int i, Mat PNG);
+ verificaEspacio(ImagenPNG imagen, Texto texto);
```

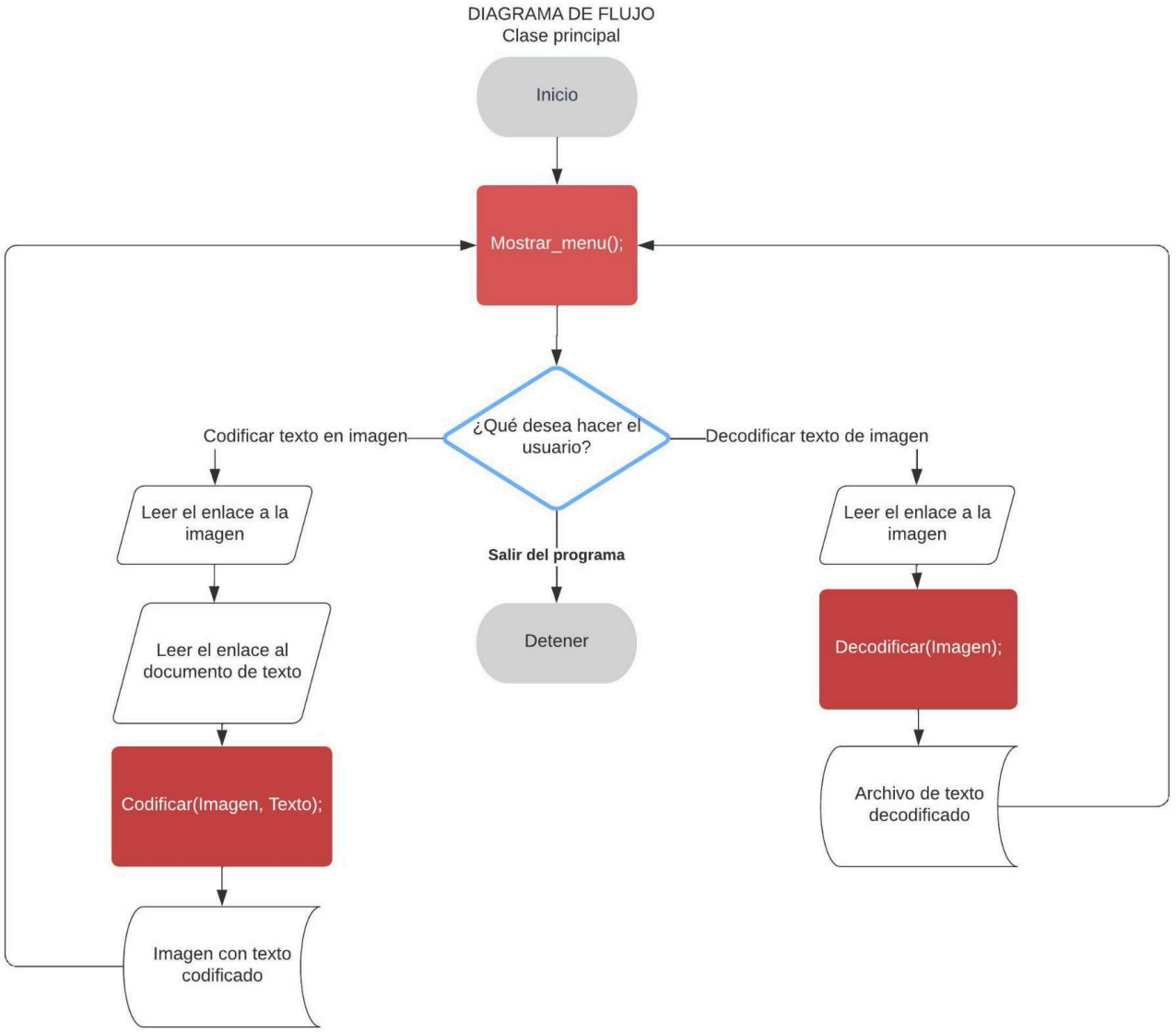


DIAGRAMA DE FLUJO
Clase principal
Inicio
Mostrar_menu();
¿Qué desea hacer el usuario?
Codificar texto en imagen
Decodificar texto de imagen
Salir del programa
Leer el enlace a la imagen
Leer el enlace al documento de texto
Codificar(Imagen, Texto);
Imagen con texto codificado
Detener
Leer el enlace a la imagen
Decodificar(Imagen);
Archivo de texto decodificado