GITUD

GRUPO DE INVESTIGACIÓN DE TELECOMUNICACIONES
DE LA UNIVERSIDAD DISTRITAL

AES-STEGANO:

Herramienta criptográfica y esteganografía.

Versión: 1.0.1

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

MANUAL DE USUARIO AES STEGANO

AUTORES

ELVIS EDUARDO GAONA
Docente Facultad Ingeniería
egaona@udistrital.edu.co

EDWAR JACINTO GÓMEZ.
Docente Facultad Tecnológica
ejacintog@udistrital.edu.co

THOMAS DANIEL AVILA
Ingeniero de Sistemas
tdavilab@correo.udistrital.edu.co

Versión: 1.0.1

2021

TABLA DE CONTENIDIO

INDICE DE FIGURAS

INDICE DE TABLAS

1. INTRODUCCIÓN

La herramienta de software AES-STEGANO realiza el cifrado y oculta un mensaje mezclando criptografía y esteganografía por medio de LSB. Emplea AES en su modo de operación CBC para el cifrado de la información usando también cifrados asimétricos e intercambio de clave con RSA.

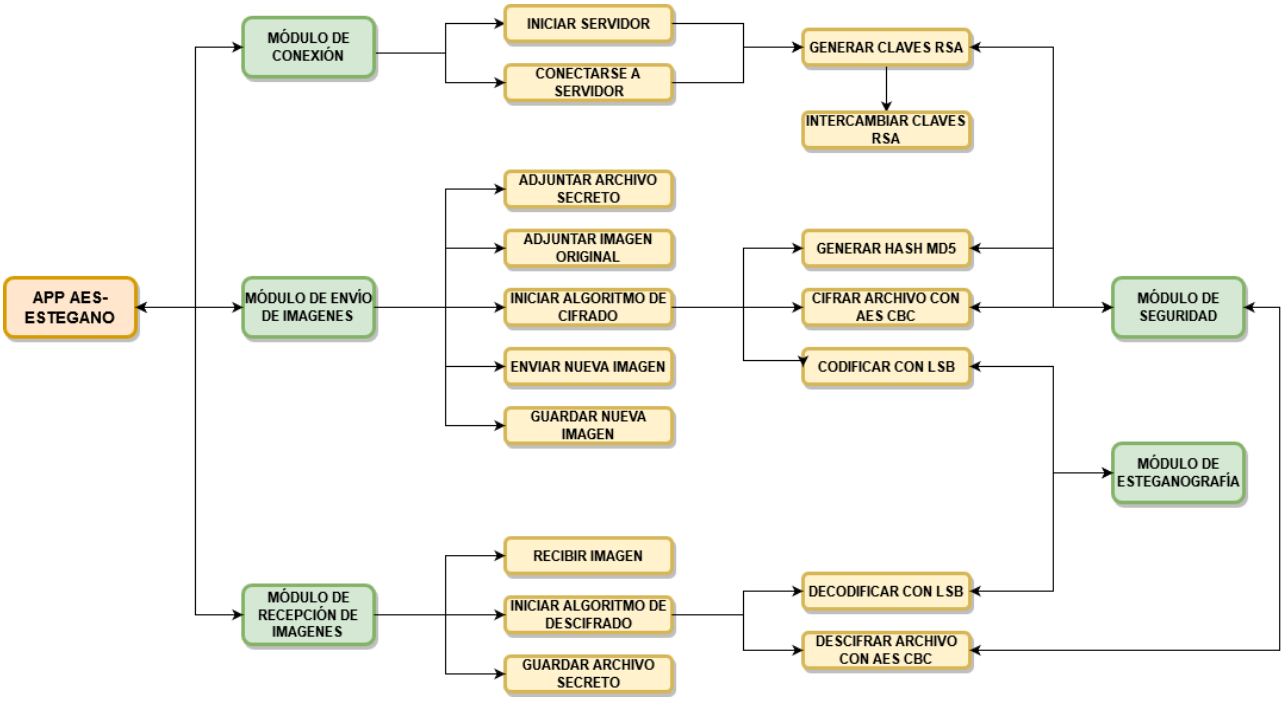

Figura 1 Diagrama jerárquico de la aplicación

2. Portabilidad Y Usabilidad

La aplicación *AES STEGANO*, cuenta con archivos ejecutables para los sistemas operativos en Windows o Linux únicamente, los cuales funcionan sin necesidad de instalar software adicional.

Si se desea compilar la aplicación a partir del código fuente, es necesario cumplir con los siguientes requerimientos de software y librerías; como también un sistema operativo Windows o Linux relacionado en la tabla 1.

Tabla 1. Requerimientos de software.

Tecnología	Versión
Python	>=3.8
Pip (python framework)	>=19.2.3
Virtualenv	>=20.0.25
opencv-python	Cualquiera

Pillow	Cualquiera
Numpy	Cualquiera
pickle-mixin	Cualquiera
pycryptodome	Cualquiera

Los archivos y carpetas del código fuente de la aplicación se muestran en la figura 2, los cuales hacen uso de un entorno virtual de python para su ejecución.

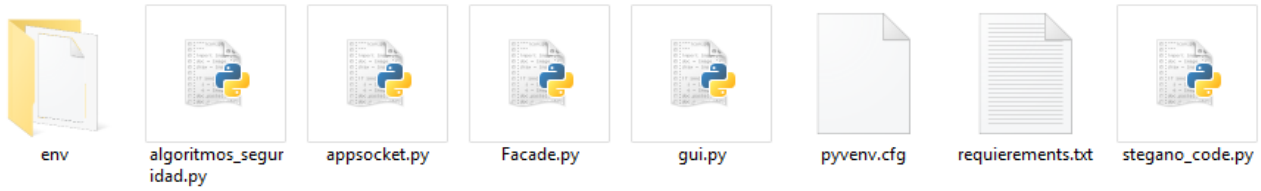

Figura 2: Distribución de la aplicación en archivos y carpetas

En cuanto a la usabilidad del sistema, visualmente se utiliza la librería tkinter de python como interfaz gráfica de usuario, y para crearla se emplea la aplicación PAGE, que facilita la creación dinámica de interfaces. La aplicación se divide en dos secciones para el cifrado y descifrado de las imágenes, como también paneles en donde se puede interactuar y visualizar información acerca de las imágenes como su tamaño en bytes. En la figura 3 se muestra la interfaz gráfica de la aplicación.

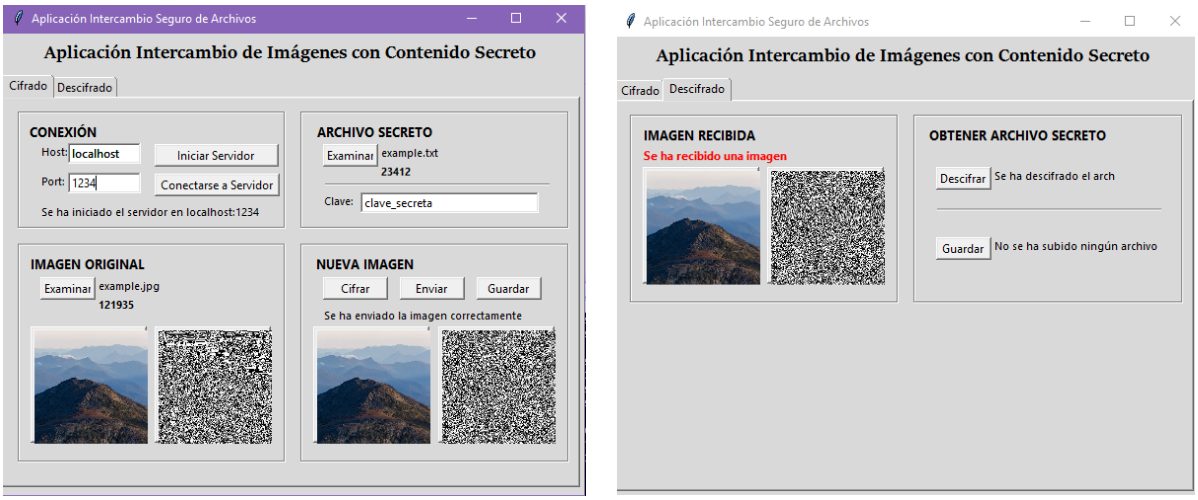

Figura 3: Interfaz gráfica de la aplicación.

A su vez, la aplicación muestra miniaturas de las imágenes utilizadas y su respectiva entropía; las cuales se pueden ampliar para realizar comparaciones, como muestra la figura 4.

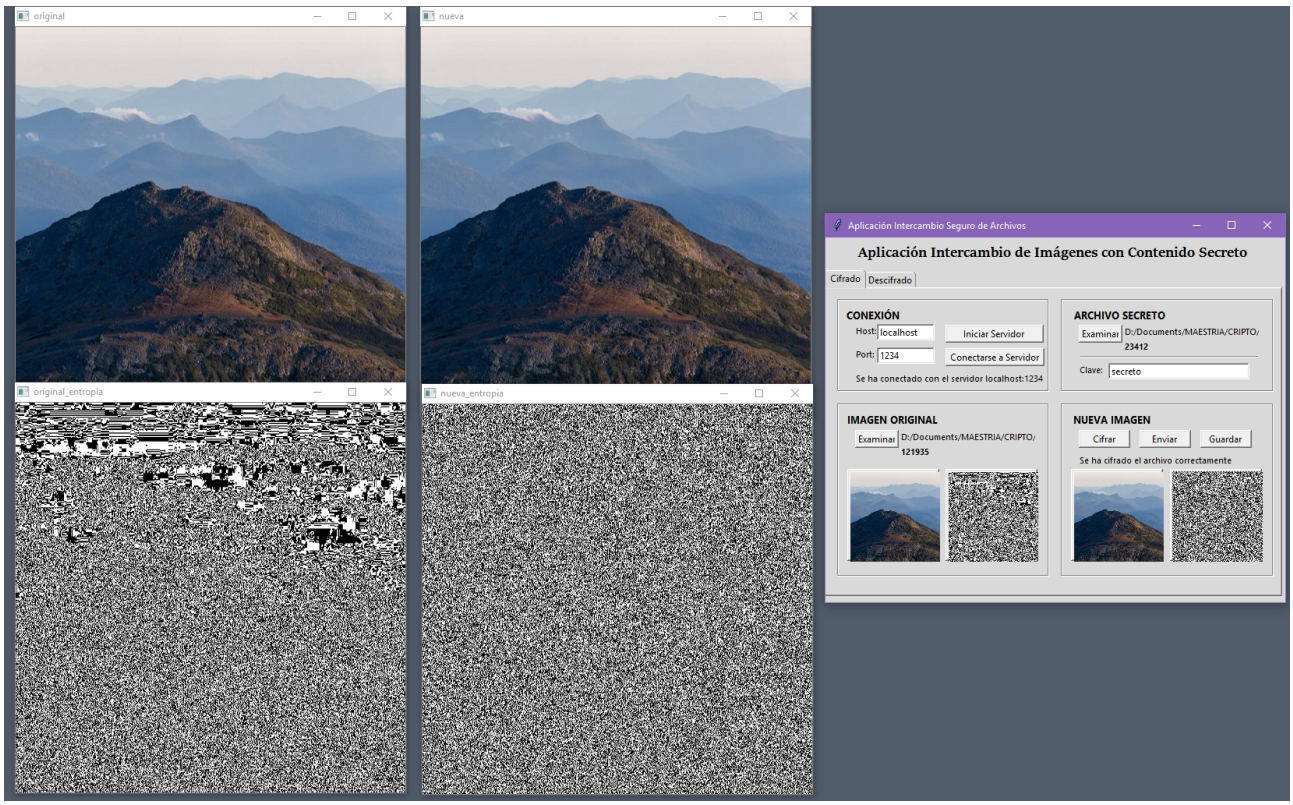

Figura 4 Visualización ampliada de las imágenes y su entropía

6. DESPLIEGUE Y CONFIGURACIÓN DEL ENTORNO DE LA APLICACIÓN.

6.1 Archivo ejecutable

La aplicación *Steganography Tool* se puede utilizar a partir del archivo ejecutable, el cual muestra la interfaz gráfica y la consola cuando se inicia. Si no se desea ver los resultados de la consola y sólo se quiere utilizar la funcionalidad con la interfaz gráfica, se debe abrir el otro archivo ejecutable que especifica el no uso de la consola.

6.2 Implementación de la aplicación a partir del código fuente

Si se desea construir el proyecto a partir del código fuente, se debe tener instalado python con una versión mayor a la 3.8, con la última versión de pip y de virtualenv.

Python	>=3.8
Pip (python framework)	>=19.2.3
Virtualenv	>=20.0.25

6.2.1 Instalación de Virtualenv

En la terminal, se debe dirigir al directorio donde se tiene instalado python e ingresar a la carpeta de Scripts. Posteriormente se debe escribir el siguiente comando:

```
python3 -m pip install virtualenv
```

6.2.2 Configuración del entorno virtual

Para configurar el entorno de virtual e instalar las librerías correspondientes, se debe dirigir al directorio del proyecto y ejecutar los siguientes comandos:

Windows	**Linux**

`python -m venv env`
 	
`.\env\Scripts\activate`
 	
`pip install requirements.txt`
	`virtualenv env`
	
	`source env/bin/activate`
	
	`pip install –r requierements.txt`

6.2.3 Ejecución de la aplicación

Para este paso el entorno virtual de python debe estar activado y la ejecución se inicia a partir del siguiente comando.

```
python gui.py
```

Este comando se realiza dos veces en varias terminales localmente o en varios computadores para que se puedan comunicar e intercambiar imágenes entre estos.

7. USUARIOS

En la tabla 2 se describe los distintos roles de usuarios.

Tabla 2. Descripción de los distintos roles de usuario.

ACTOR	DESCRIPCIÓN
Usuario	Este usuario tiene acceso a todas las funcionalidades de la aplicación.

8. REQUISITOS DE EJECUCIÓN

A continuación, se enlistan los requisitos y se detallan los requisitos.

8.1 Requisitos Funcionales

8.1.1 Interfaces Externas

- Interfaz gráfica de usuario generada a partir de la librería tkinter de python y el software PAGE.
- Terminal en donde se imprime información acerca de la ejecución.

8.1.2 Funciones

- **Por objeto:**

En la figura 17 se muestra el diagrama de funcionalidad por objeto

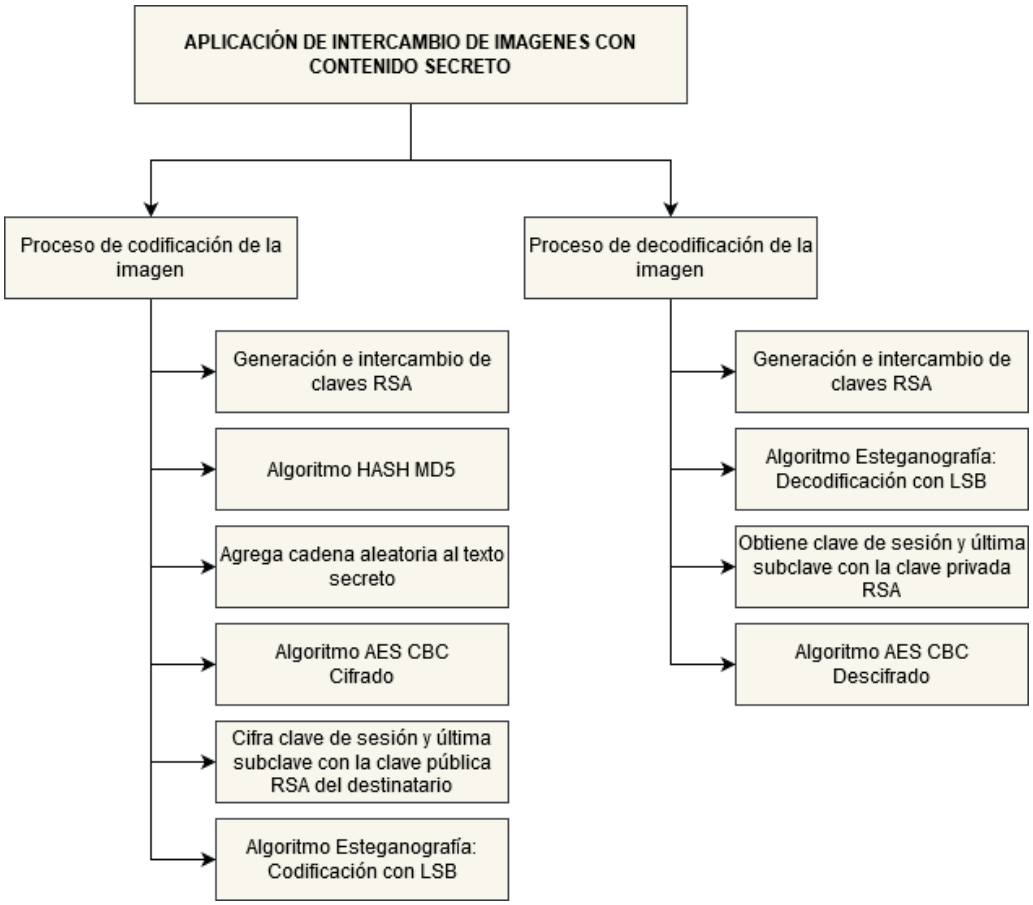

Figura 5 Diagrama de funcionalidades por objeto

- **Por objetivos:**

En la tabla 3 se muestra la funcionalidad por objetivos.

Tabla 3. Funcionalidades por objetivos

FUNCIONALIDAD	**ENTRADA**	**SALIDA**
Conexión	Host	Servidor iniciado en la dirección especificada
	Puerto	Cliente conectado al servidor en la dirección especificada
		Intercambio de claves
		Tiempo de ejecución del algoritmo RSA
Proceso de codificación de la imagen	Archivo secreto	Nueva imagen
	Clave de sesión	Reporte de tiempos de ejecución de los algoritmos
	Imagen original	Reporte del tamaño en bytes de las variables relevantes
Envío / Recepción de la imagen	Nueva imagen	Imagen recibida en el destinatario
Guardar Imagen	Nueva imagen	Imagen almacenada en disco
Proceso de decodificación de la imagen	Imagen recibida	Archivo secreto
		Reporte de tiempos de ejecución de los algoritmos
		Reporte del tamaño en bytes de las variables relevantes
Guardar archivo secreto	Archivo secreto	Archivo secreto almacenado en disco

- **Por jerarquía funcional:**

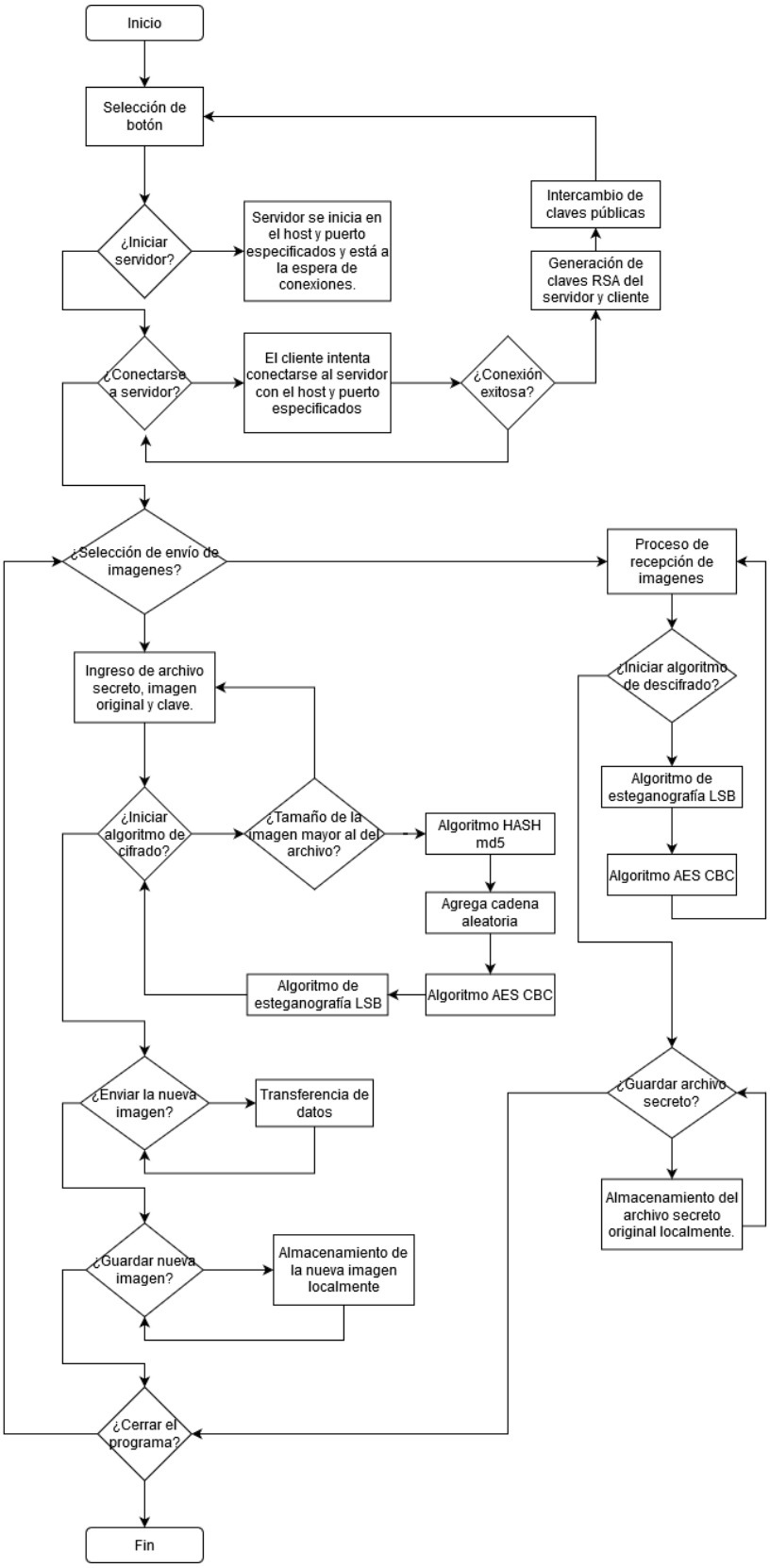

Figura 6. Funcionalidad por jerarquía

8.2 Requisitos No Funcionales

- Visualización a través de la interfaz gráfica de la aplicación y descripción detallada de la ejecución mediante la terminal.
- Almacenamiento de las imágenes y archivos resultantes en memoria persistente.
- Comunicación entre dos instancias de la aplicación ya sea en un entorno local o remoto.
- Portabilidad de la aplicación a partir del archivo ejecutable para los sistemas operativos Windows y Linux.
- Seguridad de la información intercambiada a partir de los algoritmos de criptografía y esteganografía utilizados.

8.3 Requisitos de software y Hardware.

8.3.1 Requisitos de software:

- Python 3.8 ó superior.
- Pip version 19.2.3 o superior.
- Sistema operative Windows 10 o Linux.

8.3.2 Requisitos de Hardware:

- Un computador o dos computadores con conexión LAN con al menos 8MB de ancho de banda.
- Memoria RAM de al menos 4GB
- 44MB de espacio de almacenamiento en el disco duro para la aplicación portable o 200MB para el código fuente y su entorno virtual.

9. VISTA FUNCIONAL

9.1 Módulo de Conexión

En las tablas 4 a 6 se muestran las diferentes opciones del aplicativo móvil.

Tabla 4. Iniciar servidor

Título:	Iniciar servidor
Actor:	Usuario
Escenario:	1. El usuario ingresa la información del host y el puerto. 2. El usuario presiona el botón “Iniciar Servidor”. 3. La aplicación servidor inicia el servidor y abre el puerto. 4. La aplicación servidor se queda esperando a la conexión del cliente.

Tabla 5. Iniciar cliente

Título:	Iniciar cliente
Actor:	Usuario
Escenario:	1. El usuario ingresa la información del host y el puerto. 2. El usuario presiona el botón “Conectarse a Servidor”. 3. La aplicación cliente se conecta con el servidor. 4. La aplicación servidor acepta la conexión del cliente. 5. La aplicación servidor inicia su “handler thread”, el cual va a estar escuchando constantemente a los mensajes del cliente.
Precondición:	• La otra instancia de la aplicación debió haberse iniciado como servidor. • La otra instancia de la aplicación debe estar esperando a la conexión del cliente.

Tabla 6. Generación e Intercambio de claves

Título:	Generar e Intercambiar claves públicas
Actor:	Usuario
Escenario:	1. La aplicación cliente genera las claves públicas y privadas mediante RSA. 2. La aplicación cliente envía su clave pública al servidor. 3. La aplicación cliente inicia su “handler thread”, el cual va a estar escuchando constantemente a los mensajes del servidor. 4. La aplicación servidor recibe y almacena la clave pública del cliente. 5. La aplicación servidor genera las claves públicas y privadas mediante RSA. 6. La aplicación servidor envía su clave pública al cliente. 7. La aplicación cliente recibe y almacena la clave pública del servidor.
Precondición:	• El servidor y el cliente deben estar conectados y el servidor debe estar escuchando los mensajes del cliente.

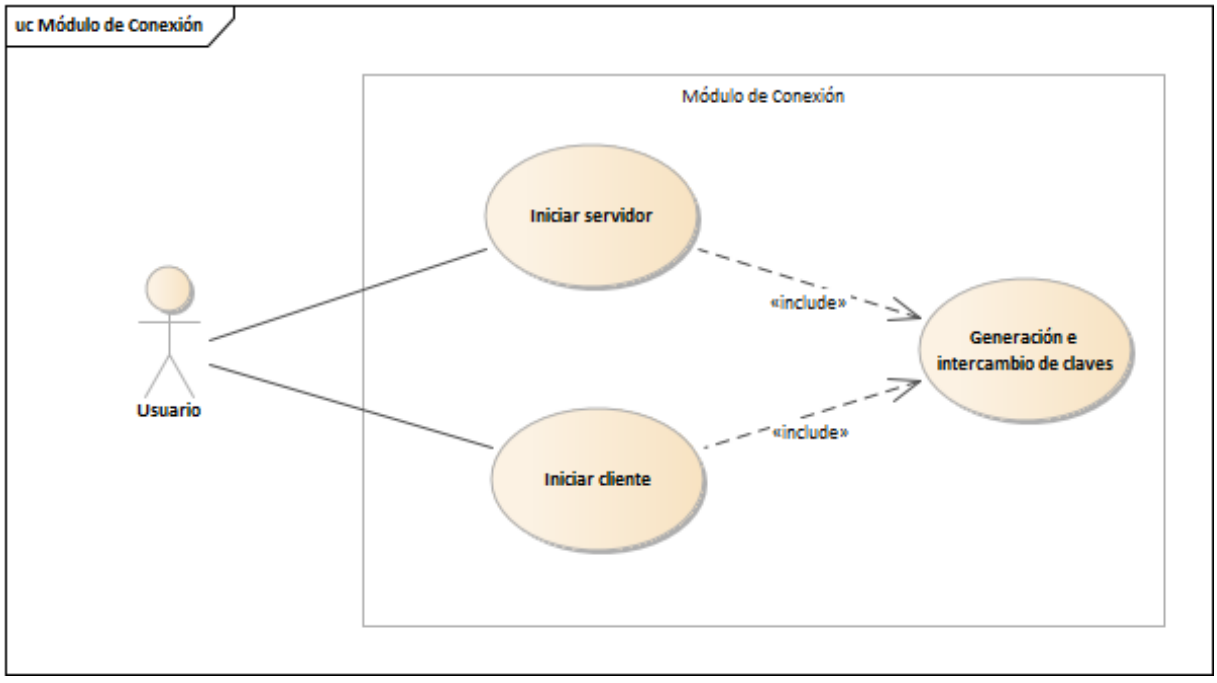

Figura 7 Casos de uso del módulo de conexión

9.2 Módulo de Cifrado y Envío de Imágenes

En las tablas 7 a 11 se muestran las diferentes opciones del aplicativo móvil.

Tabla 7. Adjuntar archivo secreto

Título:	Adjuntar archivo secreto
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Examinar” en la sección “Archivo Secreto”. 2. La aplicación lee el archivo en forma de bytes, lo que permite que este archivo pueda ser de cualquier formato.

Tabla 8. Adjuntar imagen original

Título:	Adjuntar imagen original
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Examinar” en la sección “Imagen Original”. 2. La aplicación almacena la imagen original como un arreglo mediante la librería cv2. 3. La aplicación obtiene la entropía de la imagen a partir del bit menos significativo de cada pixel en un canal. 4. La aplicación muestra en la interfaz gráfica la imagen y su entropía.

Tabla 9. Iniciar algoritmo de codificación de la imagen

Título:	Iniciar algoritmo de codificación de la imagen
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Cifrar” en la sección “Nueva Imagen”. 2. La aplicación verifica que el tamaño de la imagen original sea mayor al del archivo ingresado. 3. La aplicación obtiene la clave de sesión de la interfaz y obtiene su digest a partir del algoritmo hash md5. 4. La aplicación genera una cadena aleatoria para que el tamaño del archivo secreto sea el mismo que el de la imagen.

	5. La aplicación ejecuta el algoritmo de cifrado AES CBC. 6. La aplicación cifra mediante RSA la clave de sesión y última subclave a partir de la clave pública del destinatario. 7. La aplicación ejecuta el algoritmo de esteganografía LSB. 8. La aplicación muestra por consola los tiempos en cada paso de la ejecución y el tamaño en bytes de las variables relevantes. 9. La aplicación muestra en la interfaz gráfica la nueva imagen generada con su respectiva entropía.
Precondición:	• Se debe haber realizado el proceso de intercambio de claves públicas. • El usuario debió haber adjuntado el archivo secreto y la imagen original. • El usuario debió haber ingresado la clave en el campo correspondiente.

Tabla 10. Enviar la nueva imagen

Título:	Enviar la nueva imagen
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Enviar” en la sección “Nueva Imagen”. 2. La aplicación serializa el objeto de la imagen. 3. La aplicación envía los datos al destinatario. 4. La instancia de la aplicación destinataria recibe y almacena el objeto serializado. 5. La instancia de la aplicación destinataria obtiene la entropía de la imagen recibida.
Precondición:	• La nueva imagen debe haberse generado a partir del algoritmo de codificación de la imagen.

Tabla 11. Guardar la nueva imagen

Título:	Guardar la nueva imagen
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Guardar” en la sección “Nueva Imagen”.

	2. La aplicación almacena la nueva imagen localmente en el disco duro.
Precondición:	• La nueva imagen debe haberse generado a partir del algoritmo de codificación de la imagen.

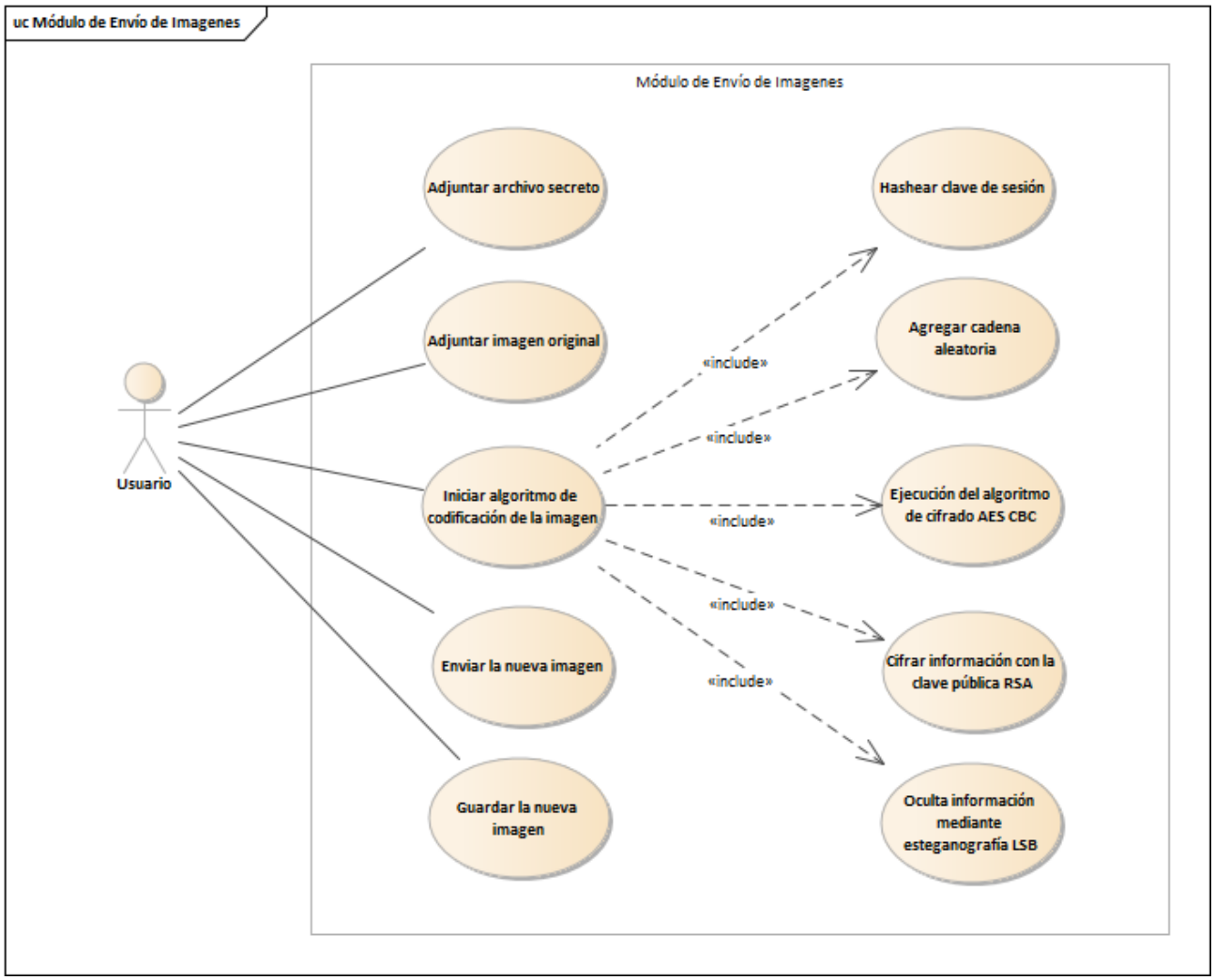

Figura 8 Casos de uso del módulo de envío de imágenes

9.3 Módulo de Recepción y Descifrado de Imágenes

En las tablas 12 a 14 se muestran las diferentes opciones del aplicativo móvil.

Tabla 12. Recibir imagen

Título:	Recibir imagen
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Actualizar” en la sección “Imagen Recibida”. 2. La aplicación muestra la imagen recibida y su respectiva entropía en la interfaz gráfica.
Precondición:	• La aplicación debe haber recibido la imagen desde la otra instancia.

Tabla 13. Iniciar algoritmo de decodificación de la imagen

Título:	*Iniciar algoritmo de decodificación de la imagen*
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Descifrar” en la sección “Obtener Archivo Secreto”. 2. La aplicación recupera la información mediante el algoritmo de esteganografía LSB. 3. La aplicación descifra mediante RSA la clave de sesión y última subclave a partir de la clave privada. 4. La aplicación ejecuta el algoritmo de descifrado AES CBC y obtiene el archivo secreto original. 5. La aplicación muestra por consola los tiempos en cada paso de la ejecución y el tamaño en bytes de las variables relevantes.
Precondición:	• La aplicación debe haber recibido la imagen desde la otra instancia.

Tabla 14. Guardar el archivo secreto

Título:	*Guardar el archivo secreto*
Actor:	Usuario
Escenario:	1. El usuario presiona el botón “Guardar” en la sección “Obtener Archivo Secreto”. 2. La aplicación almacena el archivo secreto localmente en el disco duro.
Precondición:	• La aplicación debe haber obtenido el archivo secreto original a partir de la imagen.

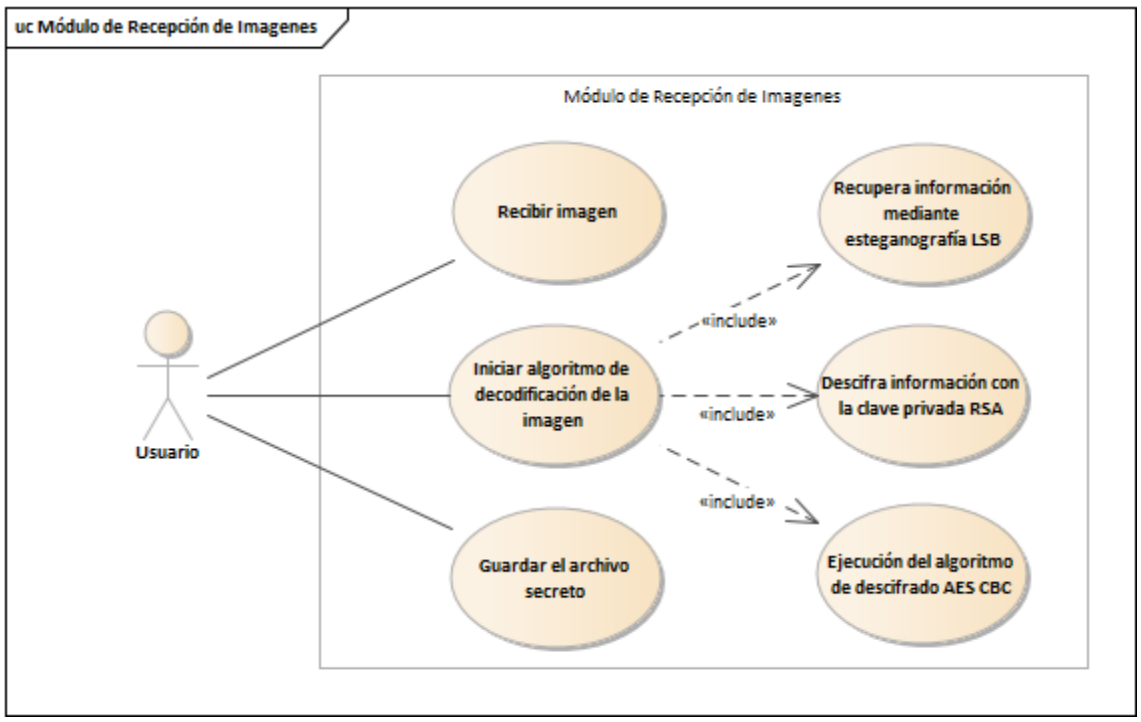

Figura 9 Casos de uso del módulo de recepción de imágenes

10. VISTA LÓGICA DEL SISTEMA

10.1 Modelo lógico de datos

En la figura 10 se muestra el modelo lógico de datos.

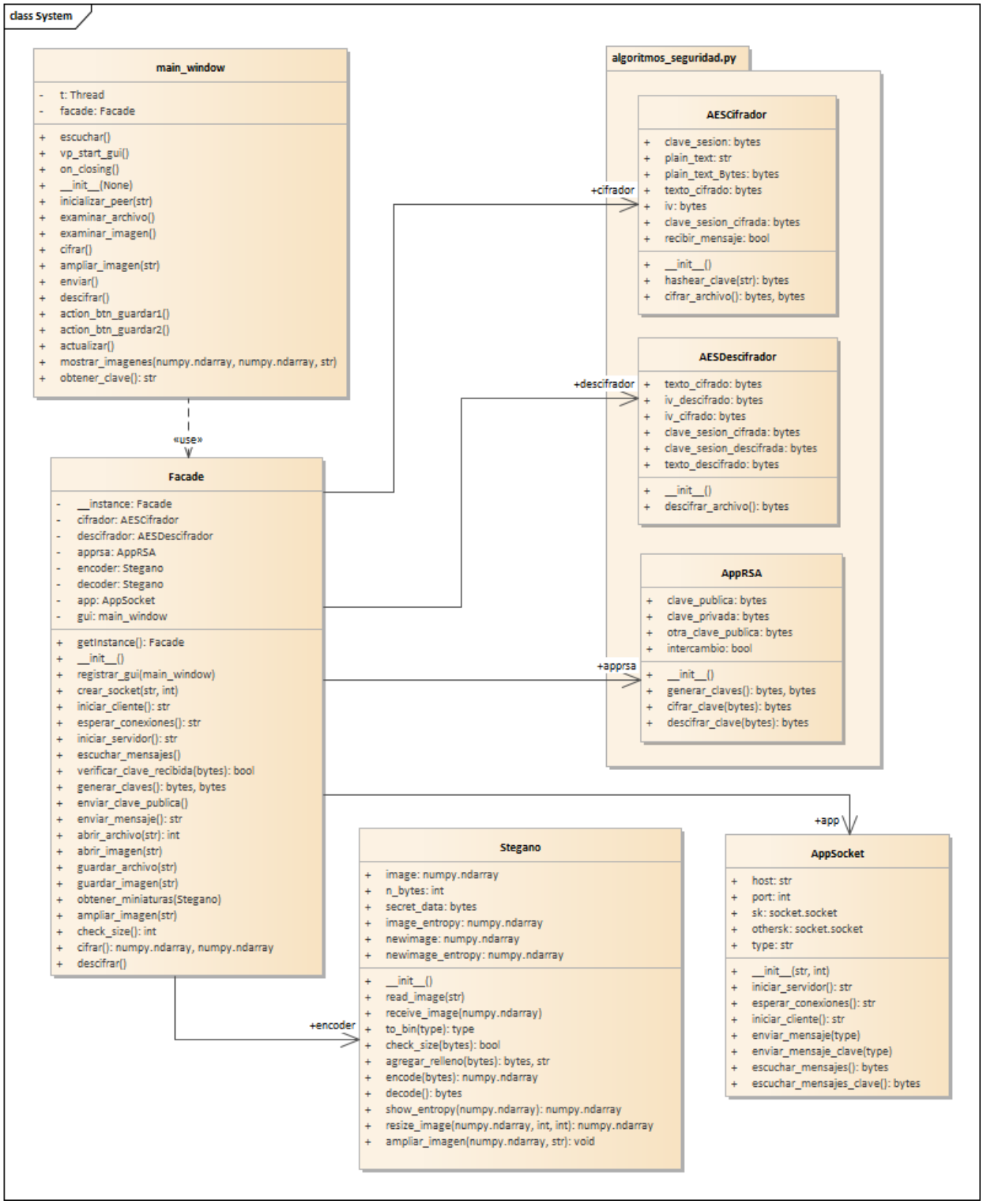

Figura 10 Modelo lógico de los datos

10.2 Diagrama de despliegue

En la figura 11 se muestra el diagrama de despliegue

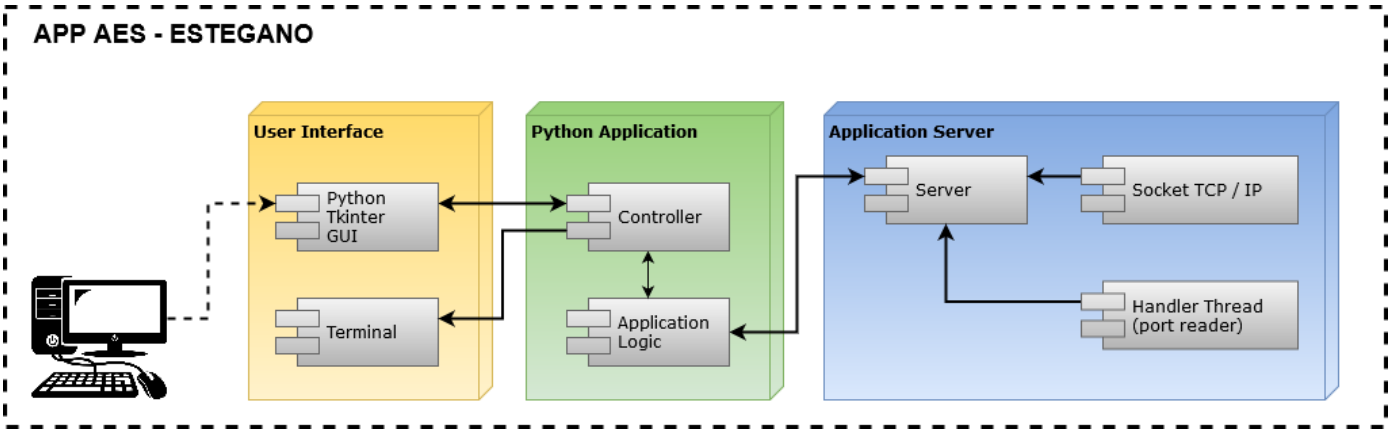

Figura 11 Diagrama de despliegue