

Luis Alberto Vargas González

Práctica AES-Reporte.

Ciberseguridad.

23/09/2022.

1. **Introducción.**

El algoritmo aes es el actual y más usado algoritmo de cifrado para la protección y manutención de datos personales o privados a través de una red, envíos por correo o plataformas públicas, por lo que su comprensión , uso y actualización es muy importante para la criptografía actualmente.

Cabe destacar que; este algoritmo es el sucesor de otros ya previamente utilizados y estructurados , pero que por fortuna han sido invalidados debido a sus debilidades en software y código, por lo que no es de extrañarse que ; en un futuro cercano , se pueda quedar igualmente inválido este algoritmo.

1. **Descripción de la implementación.**

Empezamos por importar las clases necesarias de las librerías necesarias para poder hacer la correcta implementación de este programa; entre ellas están ; Security, útil,Crypto( siendo la mas importante de java ) y por último usamos swing que es la que contiene las opciones gráficas para poder implementar los gráficos de entrada y salida de datos.

Tenemos por siguiente nuestra función main , en donde se correrá el código java principalmente, en ella creamos un objeto llamado llave que será nuestra llave dinámica, dos strings para guardar las cadenas de texto, vacías, un objeto main para poder invocar , y mandamos llamar nuestros JOptionpane para poder desplegar los paneles tanto de entrada como de salida de datos, así como los showMessageDialog para que el programa nos pueda mostrar la cadena de texto descifrada.

Después tenemos nuestra función generadora de llave de encriptación y desencriptación en donde le diremos al programa la longitud de la llave que siempre será usada por el programa , la misma no cambia debido a la estructura del mismo, creándose una llave e instanciación de tipo aes con el SecretKeySpec.

En la función encriptar inicializamos y usamos un objeto de tipo cipher usando una instancia de aes de la biblioteca de cipher en donde le indicaremos al programa el tipo de codificación de caracteres que hará al texto , en este caso UTF-8,y en el cipher.init se hace todo el proceso de encriptación de la o las cadenas de texto .

Finalmente en la función de desencriptación usamos una llave simétrica la cual viene siendo la misma llave que creamos en la función generadora de llave para poder desencriptar nuestras cadenas de texto inicializando un objeto de tipo cipher , pero ahora con el método DECRYPT, de la misma biblioteca de cipher e, cual al ser instanciado en otro objeto que llenara un string , nos devolverá la cadena o las cadenas desencriptadas.

1. **Código indentado.**

package aes**;**

**import** java**.**security**.**MessageDigest**;**

**import** java**.**util**.**Arrays**;**

**import** javax**.**crypto**.**Cipher**;**

**import** javax**.**crypto**.**spec**.**SecretKeySpec**;**

**import** javax**.**swing**.**JOptionPane**;**

**import** com**.**sun**.**org**.**apache**.**xml**.**internal**.**security**.**utils**.**Base64**;**

public class Main **{**

String LLAVE **=** "Luis Vargas Glez "**;**

public static void main**(**String**[]** args**)** **{**

String encriptada **=** ""**;**

String aEnccriptar **=** ""**;**

Main main **=** **new** Main**();**

aEnccriptar **=** JOptionPane**.**showInputDialog**(**"Ingresa la cadena a encriptar:"**);**

encriptada **=** main**.**Encriptar**(**aEnccriptar**);**

JOptionPane**.**showMessageDialog**(null,** encriptada**);**

JOptionPane**.**showMessageDialog**(null,** main**.**Desencriptar**(**encriptada**));**

**}**

//Clave de encriptación / desencriptación

public SecretKeySpec CrearCalve**(**String llave**)** **{**

**try** **{**

byte**[]** cadena **=** llave**.**getBytes**(**"UTF-8"**);**

MessageDigest md **=** MessageDigest**.**getInstance**(**"SHA-1"**);**

cadena **=** md**.**digest**(**cadena**);**

cadena **=** Arrays**.**copyOf**(**cadena**,** 16**);**

SecretKeySpec secretKeySpec **=** **new** SecretKeySpec**(**cadena**,** "AES"**);**

**return** secretKeySpec**;**

**}** **catch** **(**Exception e**)** **{**

**return** **null;**

**}**

**}**

// Encriptar

public String Encriptar**(**String encriptar**)** **{**

**try** **{**

SecretKeySpec secretKeySpec **=** CrearCalve**(**LLAVE**);**

Cipher cipher **=** Cipher**.**getInstance**(**"AES"**);**

cipher**.**init**(**Cipher**.**ENCRYPT\_MODE**,** secretKeySpec**);**

byte **[]** cadena **=** encriptar**.**getBytes**(**"UTF-8"**);**

byte **[]** encriptada **=** cipher**.**doFinal**(**cadena**);**

String cadena\_encriptada **=** Base64**.**encode**(**encriptada**);**

**return** cadena\_encriptada**;**

**}** **catch** **(**Exception e**)** **{**

**return** ""**;**

**}**

**}**

// Des-encriptación

public String Desencriptar**(**String desencriptar**)** **{**

**try** **{**

SecretKeySpec secretKeySpec **=** CrearCalve**(**LLAVE**);**

Cipher cipher **=** Cipher**.**getInstance**(**"AES"**);**

cipher**.**init**(**Cipher**.**DECRYPT\_MODE**,** secretKeySpec**);**

byte **[]** cadena **=** Base64**.**decode**(**desencriptar**);**

byte **[]** desencriptacioon **=** cipher**.**doFinal**(**cadena**);**

String cadena\_desencriptada **=** **new** String**(**desencriptacioon**);**

**return** cadena\_desencriptada**;**

**}** **catch** **(**Exception e**)** **{**

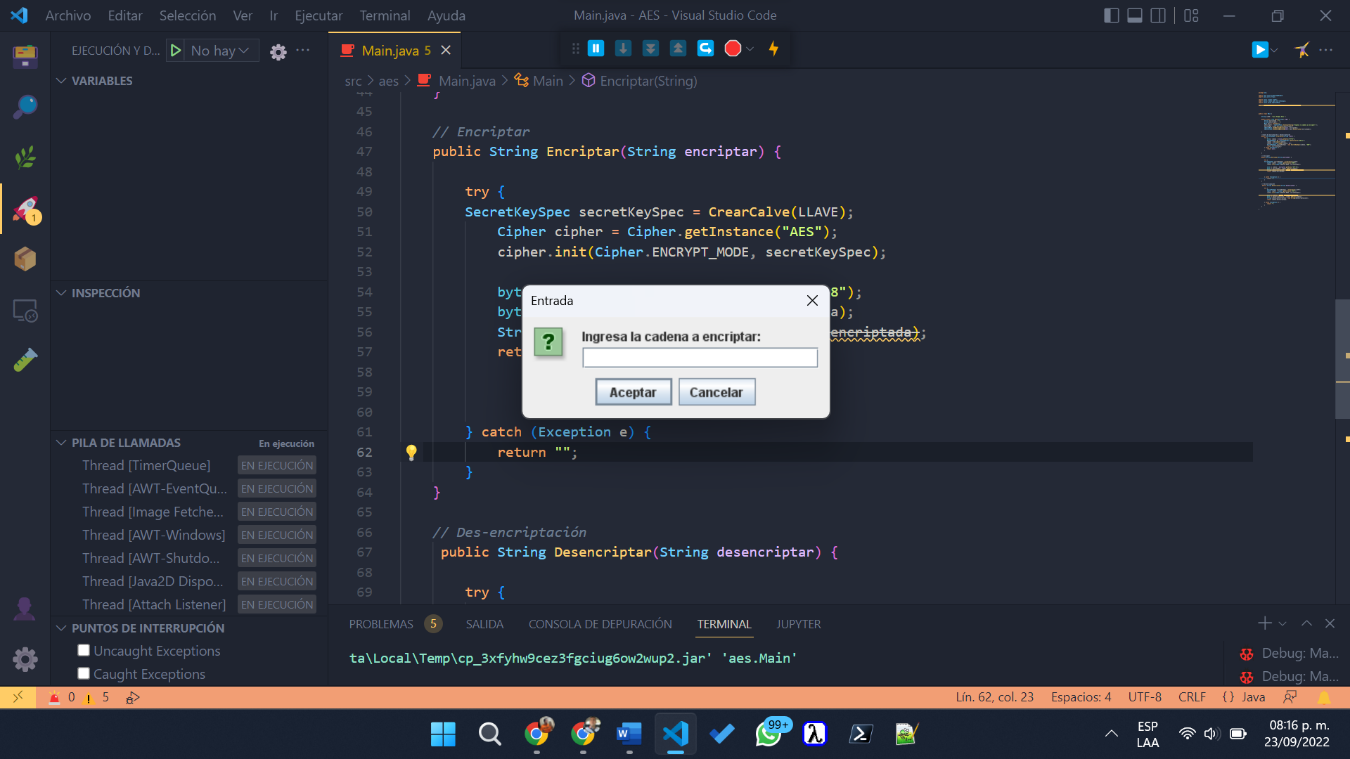
**return** ""**;**

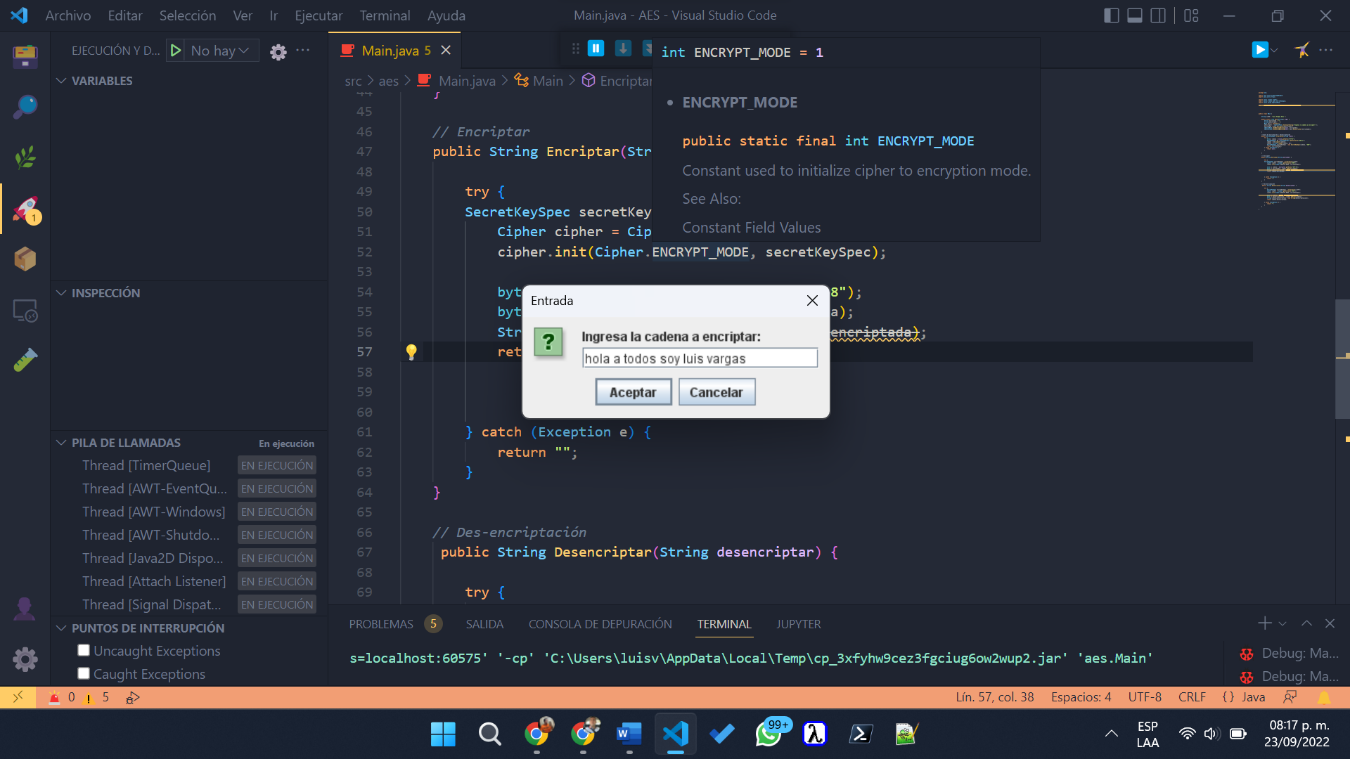
**}**

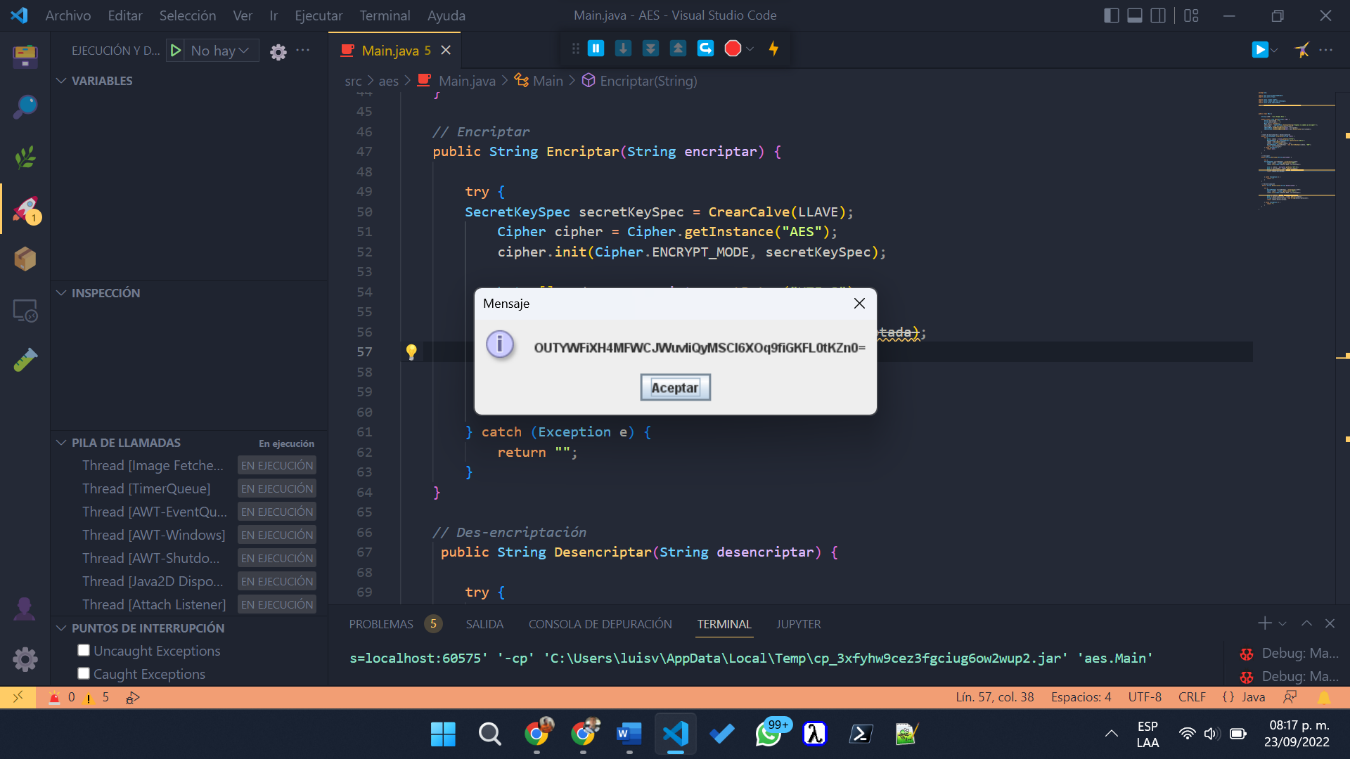
**}**

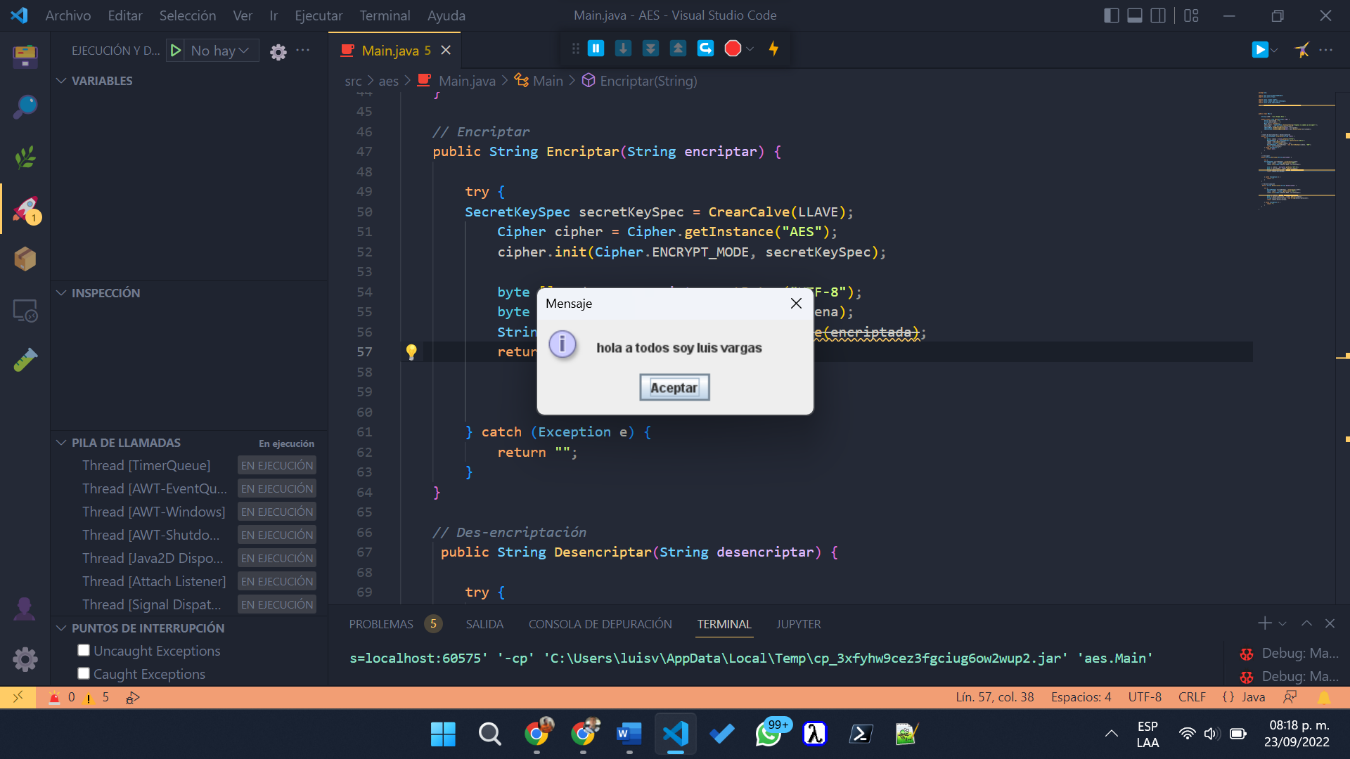
**}**

1. **Corrida o pruebas.**









1. **Conclusiones.**

Como se pudo constatar , el programa pide al usuario una cadena de texto por pantalla, después le muestra la cadena de texto ya encriptada con la codificación UTF-8, y al último la verificación del texto que el usuario introdujo que es el compendio de mensajes.

Este algoritmo podría ser muy útil cuando un usuario requiera que le envíen por algún medio tecnológico alguna información ya sean videos, música, texto que no cualquier persona deba o pueda ver, ejemplo: alguna escritura de propiedades inmobiliarias, códigos de acceso a sistemas de vigilancia o bancarios de alguna empresa, grabaciones de seguridad con temas delicados, o videos de seguridad.

1. **Referencias.**
2. *Cifrado AES y RSA*. (s. f.). Recuperado 23 de septiembre de 2022, de https://www.boxcryptor.com/es/encryption/
3. *¿Qué es el cifrado AES y cómo funciona?* (2022, 9 marzo). Ciberseguridad. Recuperado 23 de septiembre de 2022, de https://ciberseguridad.com/guias/prevencion-proteccion/criptografia/cifrado-aes/