UNIVERSIDAD PRIVADA “FRANZ TAMAYO”

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA EN SISTEMAS

“Encriptación y Desencriptación de archivos”

ESTUDIANTE:

JOSE YHILMAR VILLCA MAYTA

DOCENTE:

ING. HENRY RAUL VARGAS GRILLO

MATERIA:

SEGURIDAD INFORMATICA

ENLACE:

LINK DE GITHUB

El Alto – Bolivia
2022

INDICE

CAPITULO I

1. INTRODUCCION

1.1. Presentación del problema

La criptografía se creó hace mucho tiempo como una respuesta a una necesidad, esta necesidad era de la poder enviar y recibir mensajes sin que su contenido pudiera ser leído sin su clave correspondiente.

Es un pilar fundamental para garantizar la seguridad de cualquier comunicación que se realice a través de medios informáticos. En pocas palabras la criptografía sirve para mantener segura la información que se transmite en el mensaje.

Mientras mas usuarios tienen acceso a la tecnología muchos ignoran o no tienen conciencia sobre el hecho de que puedan vulnerar con facilidad sus documentos ya sea: imágenes, documentos, datos personales, datos familiares, etc.

1.2. Argumentación sobre la importancia de objeto de estudio

El objeto de estudio del proyecto es poder lograr que los usuarios sean conscientes sobre la seguridad de información y protección en sus dispositivos inteligentes por medio de la protección de sus datos y la importancia que tiene por mas insignificante que parezca.

Con este proyecto, podremos educar a las personas a concientizar un poco la información confidencial de sus archivos, gracias a la encriptación, la cual les ayudará a mantener la información segura, debido a que el cifrado que se utilizará no podrá ser desencriptado sin la respectiva llave.

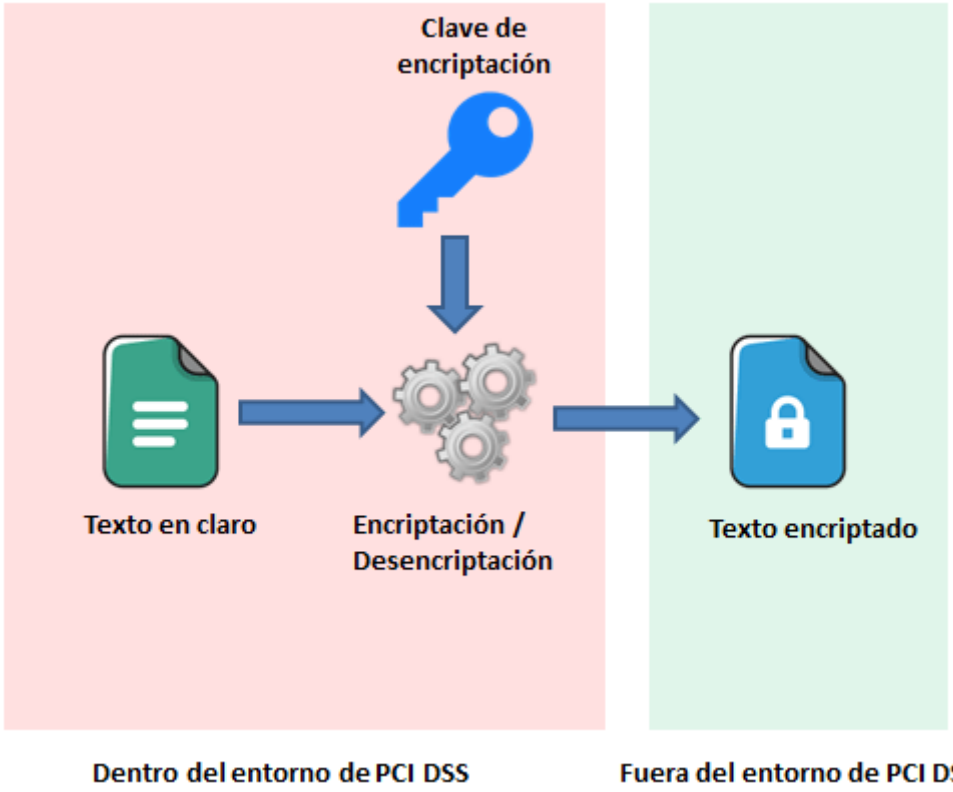
CAPITULO II

2. Marco Teórico

2.1. Criptografía

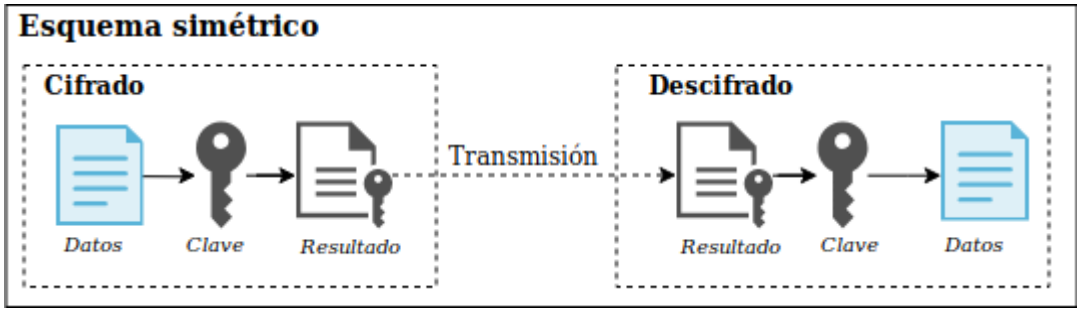
La criptografía se ha definido como el ámbito de la criptología que se ocupa de las técnicas de cifrado y descifrado destinadas a alterar las representaciones

lingüísticas de ciertos mensajes con el fin de hacerlos ininteligibles a receptores no autorizados.



2.2. Criptografía simétrica

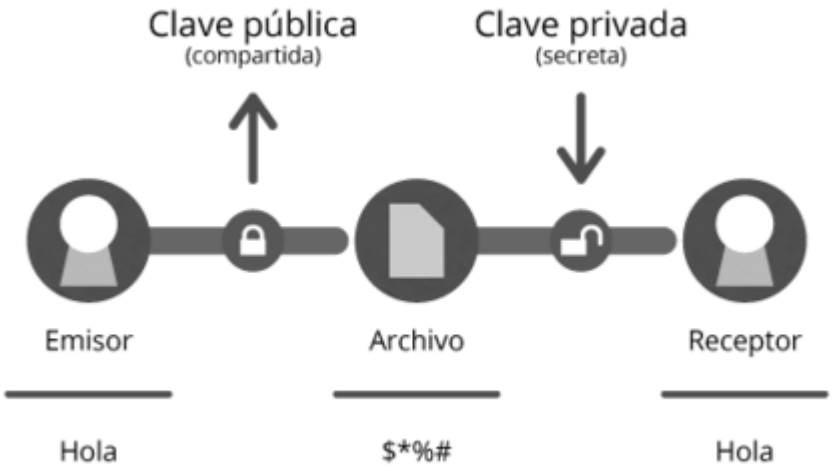
La criptografía simétrica utiliza la misma llave para cifrar y descifrar el mensaje de datos, es decir se que comparte el mismo secreto. Es por esta razón que la seguridad de este proceso depende de la posibilidad de que una persona no autorizada consiga la llave secreta.



Lo cual nos quiere decir que este tipo de criptografía es una de las más seguras, debido a que no se puede descifrar la información de este encriptado, sin la llave generada.

2.3. Criptografía asimétrica

El cifrado asimétrico también es conocido como cifrado de clave publica o cifrado de dos claves. En este tipo de cifrado se utiliza una clave para el cifrado y otra para el descifrado. Ambas claves están relacionadas, pero de una forma demasiado compleja como para encontrar una a partir de la otra si no se conoce concretamente la relación. El cifrado se realiza con lo que se llama clave publica y el descifrado con la clave privada.



Gracias a la anterior definición, podemos deducir que el cifrado simétrico y asimétrico son dos formas seguras de poder realizar el descifrado de archivos, claves, palabras y contraseñas, a diferencia del cifrado simétrico, el cifrado asimétrico utiliza dos claves distintas, una para poder cifrar los datos y otra para poder descifrarlos.

CAPITULO III

3. Justificación

3.1. Justificación Técnica

Para la elaboración del proyecto, se utilizará el lenguaje de programación Python, el cual nos ayudará a crear el algoritmo para poder encriptar y desencriptar los diferentes archivos.

3.2. Justificación Social

El proyecto, ayudará a las personas y empresarios a poder tener seguros sus archivos, gracias a que sus archivos se encontrarán cifrados con una llave única que solo el tendrá acceso, el cual no puede ser descifrado si no se tiene la llave generada para poder descifrar dicho archivo, gracias a esto, se podrá mantener la confidencialidad de los archivos.

3.3. Justificación Económica

Para el presente proyecto los recursos para utilizarse no son elevados, debido a que solo necesitamos tener instalado Python, un editor de código como Visual Studio Code y una computadora para poder realizar la programación del algoritmo, en este caso, debido a que las herramientas de software pueden ser encontradas de forma gratuita en la web, el costo es mínimo, además de ayudar a las personas a ocultar información secreta de sus archivos, como ser estados de cuentas, socios y proveedores.

CAPITULO IV

4.1. Planteamiento del problema

La seguridad es vital para las personas de todas partes del mundo, además de la seguridad, es necesario concientizar a las personas sobre los permisos que otorgan sobre el acceso a sus dispositivos, dado este caso, gracias al gran incremento de cibercriminales, es necesario mantener niveles de seguridad dentro de los equipos,

debido a que hoy en día existen herramientas que logran acceder al dispositivo, logrando robar información.

Hoy en día es necesario el tener un método para poder asegurar la información dentro de los archivos, debido a que muchas veces existen archivos que deben mantenerse de forma privada, como ser: Estados de cuenta, socios y principales proveedores dentro de la empresa, ya que la competencia hoy en día, puede buscar el robo de la información para perjudicar a las empresas gracias al fácil acceso y poca seguridad de este tipo de archivos

4.1.1. Formulación del problema

¿Como demostrar y concientizar sobre la necesidad de mantener los archivos encriptado de forma segura?

4.2. Objetivos

4.2.1. Objetivo General

Desarrollar un programa que permita encriptar y desencriptar archivos como documentos con una llave única, utilizando un cifrado simétrico.

4.2.2. Objetivos Específicos

- Aprender a utilizar Python para la encriptación y desencriptación de archivos con la librería Fernet.
- Demostrar a los usuarios lo sencillo que es tener la información de sus documentos seguros gracias a la encriptación.
- Realizar el proyecto con fines educativos con la intención de mostrar el proceso de encriptación automatizado

4.3. Alcances y Limites

4.3.1. Alcances

El presente proyecto realizara una encriptación y desencriptación de un solo archivo a la vez utilizando el cifrado simétrico con la librería Fernet.

4.3.2. Limites

El presente proyecto se ejecutará únicamente desde la consola de comandos, además se encriptarán únicamente archivos con el formato ".txt".

MARCO PRACTICO

5. Librerías

5.1. Fernet

Fernet es un sistema de cifrado/descifrado simétrico que utiliza las mejores prácticas actuales. También autentica el mensaje, lo que significa que el destinatario puede saber si el mensaje ha sido alterado de alguna manera con respecto a lo que se envió originalmente. (*Fernet System for Symmetric Encryption*, 2020)

5.2. Modulo Pathlib

El módulo pathlib le puede resultar útil si desea a crear o mover archivos en el sistema de archivos de su programa de Python, enumerar los archivos del sistema de archivos que coincidan con una extensión o un patrón determinado o crear rutas de archivos apropiadas para el sistema operativo basadas en colecciones de cadenas sin procesar. (DavidMuller, 2020)

6. *Código fuente*

6.1. Código encriptacion.py

```
from cryptography.fernet import Fernet
from pathlib import Path
from os import listdir

#ruta = r'/home/jose/Escritorio/prueba1'
#from matplotlib.pyplot import cla
def generarClave():
    archivo = r'llave.key'
    objeto = Path(archivo)
    if not objeto.is_file():
        clave = Fernet.generate_key()

        with open("llave.key", "wb") as key_file:
            key_file.write(clave)

def cargarClave():
    return open("llave.key","rb").read()

# ENCRIPTACION Y DESENCRIPTACION

def encriptar(archivo,clave):
    f = Fernet(clave)
    with open(archivo,"rb") as file:
        mensaje = file.read()

    datos_encriptados = f.encrypt(mensaje)
    with open(archivo, "wb") as file:
        file.write(datos_encriptados)

def desencriptar(archivo,clave):
    f = Fernet(clave)
    with open(archivo,"rb") as file:
        mensaje_encriptado = file.read()

    datos = f.decrypt(mensaje_encriptado)

    with open(archivo,"wb") as file:
        file.write(datos)

# MANEJO DE ARCHIVOS
def abrir(archivo):
```

```
    a = open(archivo,"rt", encoding="utf-8")
    print()
    print(a.read())
    print()

def crear(linea, archivos):
    with open(archivos, 'a') as file: # a = append crea o edita en el caso
de que exista
        file.write("\n"+linea)
```

6.2. Código menú.py

```
import encriptacion
from os import listdir
from os.path import isfile,join


encriptacion.generarClave()
clave=encriptacion.cargarClave()

ruta = r'/home/jose/Escritorio/prueba1'

def listar(ruta):
    archivo = [a for a in listdir(ruta) if isfile(join(ruta,a)) if
a.endswith('txt')]
    return archivo
listado_archivos = listar(ruta)

menuprincipal = int(input("Menu principal\n1.\t Ver Archivos\n2.\t Crear -
Editar archivos\n3.\t Encriptar Archivos\n4.\t Desencriptar archivos\n0. \t
Salir del Programa\n "))

while menuprincipal != 0:
    if menuprincipal == 1:

        print(listado_archivos)
        a = input("Seleccione archivo a leer: ")
        encriptacion.abrir(a)
    elif menuprincipal == 2:
        opcion = input ("Ingresa un nuevo archivo ")
        linea = input("Texto a agregar:")
        encriptacion.crear(linea, opcion)
    elif menuprincipal == 3:

        print(listado_archivos)
        opcion = input('Elija el archivo que desee encriptar: ')
        if opcion == 'archivo1.txt':
            encriptacion.encriptar(opcion,clave)
            print("Archivo encriptado, el texto es: ")
            encriptacion.abrir(opcion)
```

```
        if opcion == listado_archivos:
            encriptacion.encriptar(opcion,clave)
            print("Archivo encriptado, el texto es: ")
            encriptacion.abrir(opcion)
        if opcion == 'archivo3.txt':
            encriptacion.encriptar(opcion,clave)
            print("Archivo encriptado, el texto es: ")
            encriptacion.abrir(opcion)


    elif menuprincipal == 4:
        print(listado_archivos)
        archivo = input('Elija el archivo que desee desencriptar: ')
        if archivo == 'archivo1.txt':
            encriptacion.desencriptar(archivo,clave)
            print("Archivo desencriptado, el texto es: ")
            encriptacion.abrir(archivo)
        if archivo == 'archivo2.txt':
            encriptacion.desencriptar(archivo,clave)
            print("Archivo desencriptado, el texto es: ")
            encriptacion.abrir(archivo)
        if archivo == 'archivo3.txt':
            encriptacion.desencriptar(archivo,clave)
            print("Archivo desencriptado, el texto es: ")
            encriptacion.abrir(archivo)
    else:
        print('Digite una opcion correcta')

    menuprincipal = int(input("Menu principal\n1.\t Ver Archivos\n2.\t Crear
- Editar archivos\n3.\t Encriptar Archivos\n4.\t Desencriptar archivos\n0.
\t Salir del Programa\n "))
else:
    print('Salimos del programa')
```

6.3. Ejecución

6.3.1. Ver archivos

```
┌──(jose㉿kali)-[~/Escritorio/prueba1]
└─$ /bin/python3 /home/jose/Escritorio/prueba1/menu.py
Menu principal
1.       Ver Archivos
2.       Crear - Editar archivos
3.       Encriptar Archivos
4.       Desencriptar archivos
0.       Salir del Programa
 1
['archivo1.txt', 'archivo2.txt', 'archivo3.txt']
Seleccione archivo a leer: archivo1.txt

hola aqui practicando la criptografia
nueva linea?
otra linea mas
```

6.3.2. Crear – Editar archivos

```
Menu principal
1.       Ver Archivos
2.       Crear - Editar archivos
3.       Encriptar Archivos
4.       Desencriptar archivos
0.       Salir del Programa
 2
Ingresa un nuevo archivo archivo2.txt
Texto a agregar:la encriptacion es de hecho algo importante e interesante
Menu principal
1.       Ver Archivos
2.       Crear - Editar archivos
3.       Encriptar Archivos
4.       Desencriptar archivos
0.       Salir del Programa
 1
['archivo1.txt', 'archivo2.txt', 'archivo3.txt']
Seleccione archivo a leer: archivo2.txt

banco de datos 2
la encriptacion es de hecho algo importante e interesante
```

6.3.3. Encriptar archivos

```
Menu principal
1.       Ver Archivos
2.       Crear - Editar archivos
3.       Encriptar Archivos
4.       Desencriptar archivos
0.       Salir del Programa
 3
['archivo1.txt', 'archivo2.txt', 'archivo3.txt']
Elija el archivo que desee encriptar: archivo1.txt
Archivo encriptado, el texto es:

gAAAAABiuPUOb0KvjdQEFSiG2vwtPqVQ7XbRII28-5ZdECdmJSORsLvaFog8ddSjbuPq2cf-uqehD_62pDUoacT6mPXhxq8--
rKx6E_4JIPISJbFLLtfJXGqpdTenGQbMvFz1h88=
```

6.3.4. Desencriptar archivos

```
Menu principal
1.       Ver Archivos
2.       Crear - Editar archivos
3.       Encriptar Archivos
4.       Desencriptar archivos
0.       Salir del Programa
 4
['archivo1.txt', 'archivo2.txt', 'archivo3.txt']
Elija el archivo que desee desencriptar: archivo1.txt
Archivo desencriptado, el texto es:

hola aqui practicando la criptografia
nueva linea?
otra linea mas
```

6.3.5. Salir del programa

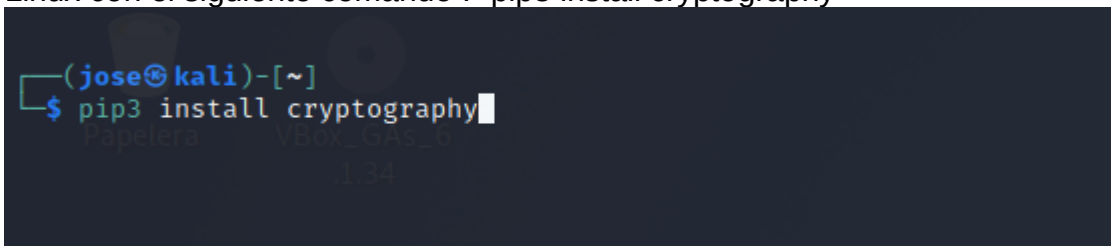
```
Menu principal
1.       Ver Archivos
2.       Crear - Editar archivos
3.       Encriptar Archivos
4.       Desencriptar archivos
0.       Salir del Programa
 0
Salimos del programa

┌──(jose㉿kali)-[~/Escritorio/prueba1]
└─$
```

MANUAL DE USUARIO

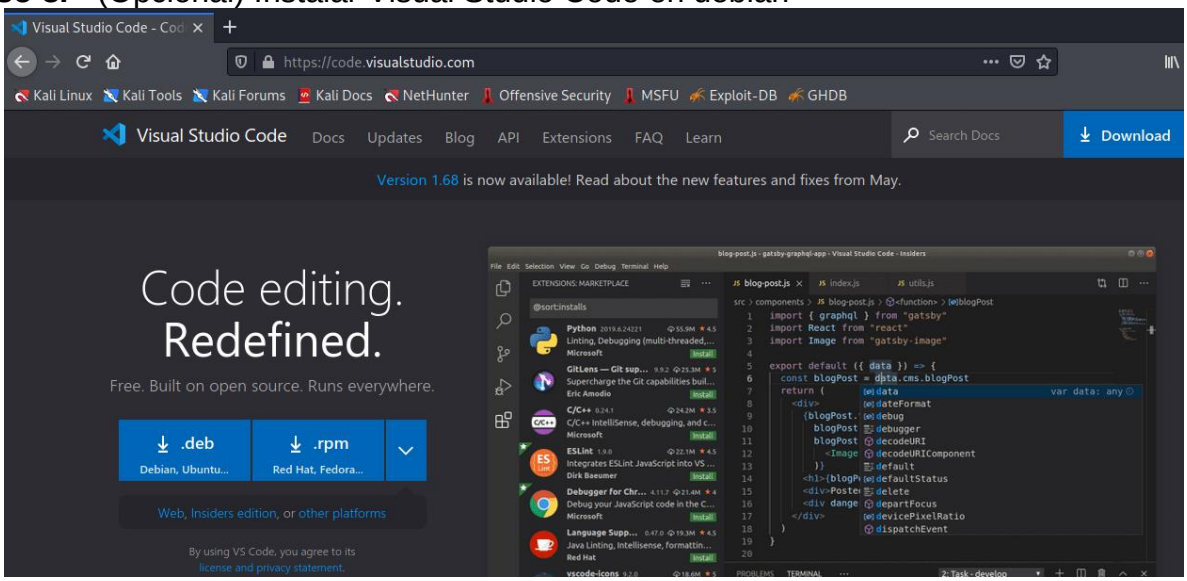
INSTALACION

Paso 1. Descargar e instalar Python adjunto el link de descarga

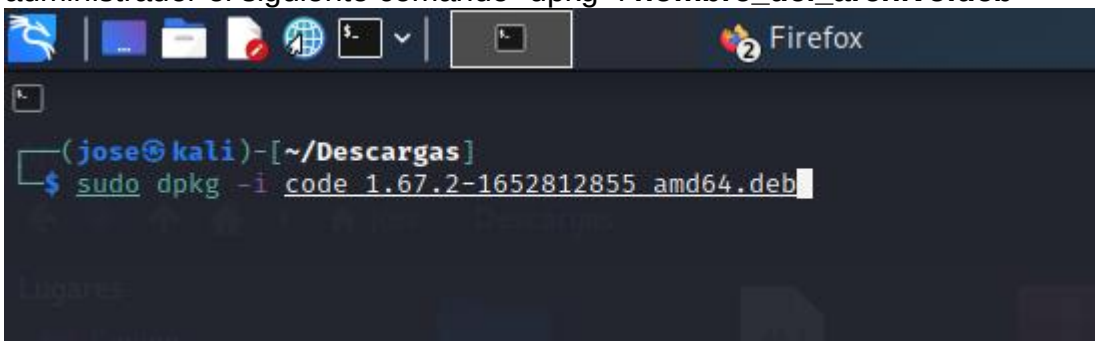
Paso 2. Instalar el modulo cryptography desde cmd en Windows o terminal en Linux con el siguiente comando : “pip3 install cryptography”



Paso 3. (Opcional) Instalar Visual Studio Code en debian

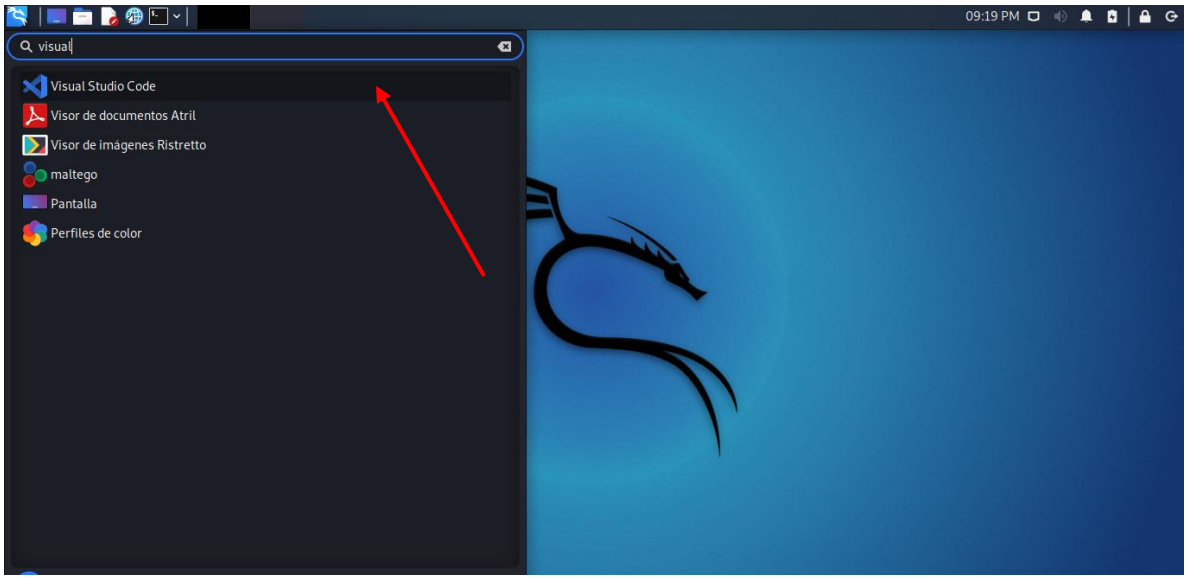


Paso 4. Una vez descargado instalar mediante la terminal con permisos de administrador el siguiente comando “dpkg -i **nombre_del_archivo.deb**”

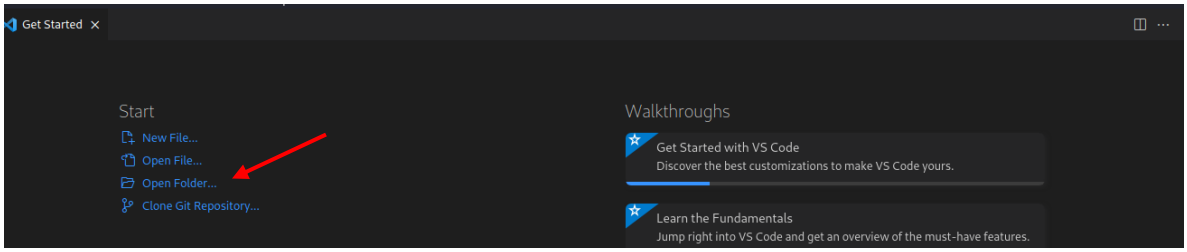


Ejecución del programa

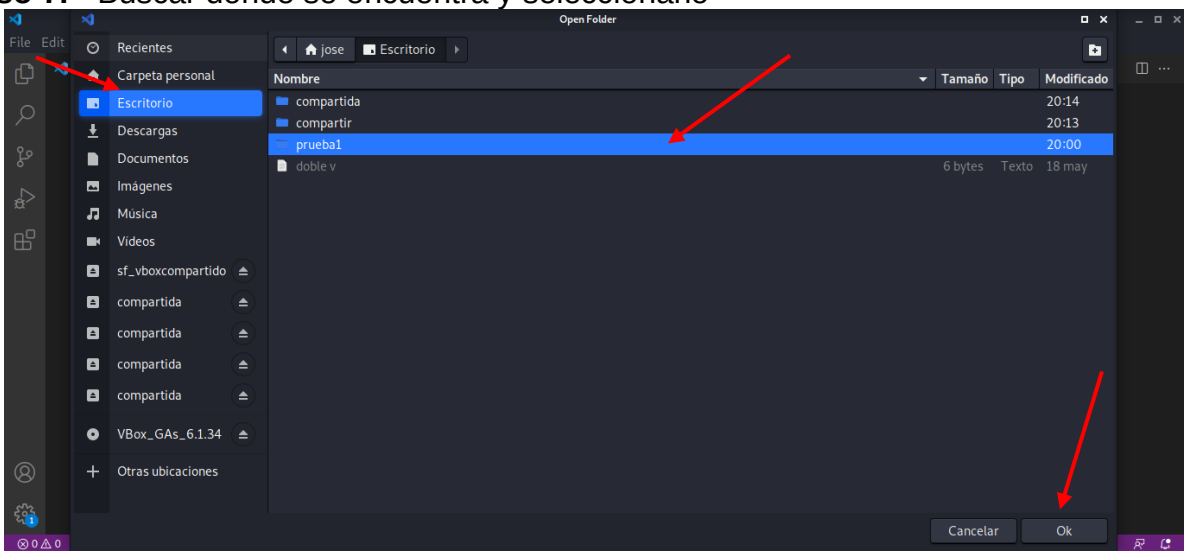
Paso 5. Abrir un editor de código de tu preferencia en mi caso Visual Studio Code



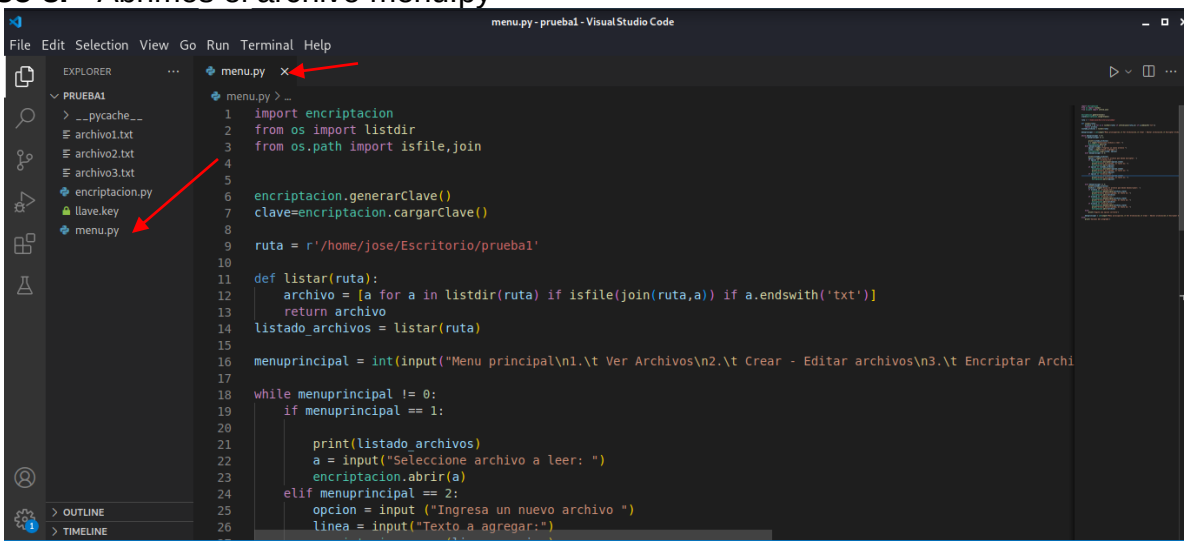
Paso 6. Abrir el directorio



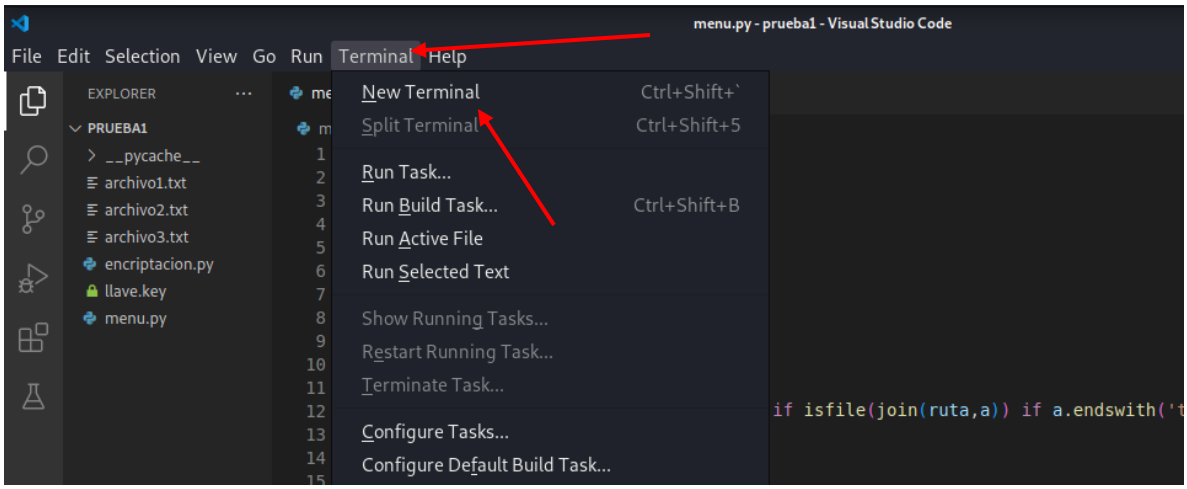
Paso 7. Buscar donde se encuentra y seleccionarlo



Paso 8. Abrimos el archivo menú.py



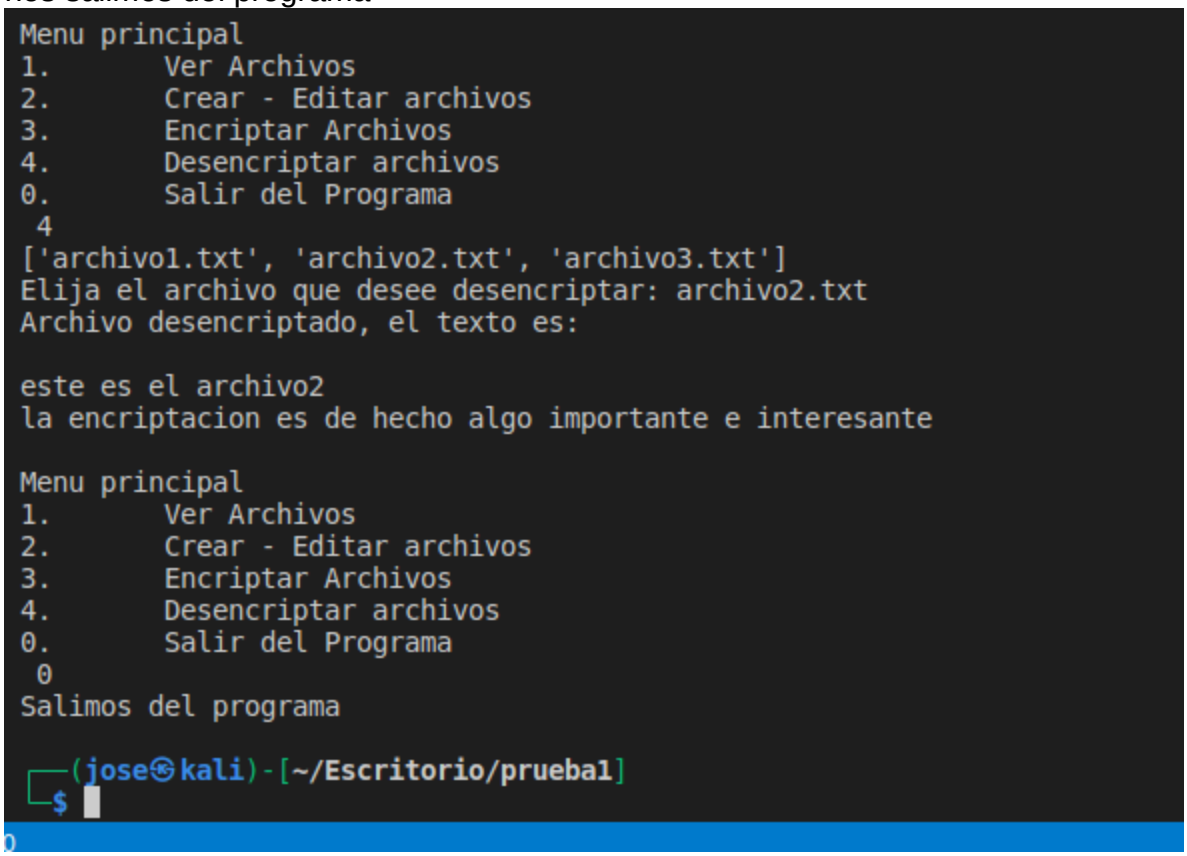
Paso 9. Click en terminal > new terminal



Paso 10. Una vez se ejecute la terminal ejecutar el siguiente comando :
“python3 menu.py”

Paso 11. Seleccione cualquier opción que desee. Por ejemplo: abrimos un archivo para observar su contenido y posteriormente encriptarlo.

Paso 12. Posteriormente desencriptamos, mostramos el contenido y por ultimo nos salimos del programa



REFERENCIAS

Fernet system for symmetric encryption. (2020). Pythoninformer.com. https://www.pythoninformer.com/python-libraries/cryptography/fernet/

DavidMuller. (2020, August 19). *Cómo usar el módulo pathlib para manipular las rutas de sistemas de archivos en Python 3*. Digitalocean.com; DigitalOcean. https://www.digitalocean.com/community/tutorials/how-to-use-the-pathlib-module-to-manipulate-filesystem-paths-in-python-3-es

Fernet (symmetric encryption) — Cryptography 38.0.0.dev1 documentation. (2022). Cryptography.io. https://cryptography.io/en/latest/fernet/