Perfecto, Chema. Aquí tienes una versión mejorada, funcional y documentada del archivo **integracion/api\_whatsapp.py** que:

* Escucha mensajes entrantes por webhook.
* Detecta comandos válidos.
* Llama a funciones simuladas del backend.
* Responde al usuario por WhatsApp a través de la API de Twilio.
* Registra logs en consola.

**✅ integracion/api\_whatsapp.py**

import os

from flask import Flask, request

from twilio.rest import Client

import logging

# Configuración básica

app = Flask(\_\_name\_\_)

logging.basicConfig(level=logging.INFO)

# Credenciales Twilio desde variables de entorno

ACCOUNT\_SID = os.getenv("CLAVE\_TWILIO\_SID")

AUTH\_TOKEN = os.getenv("CLAVE\_TWILIO\_TOKEN")

FROM\_WA = os.getenv("NUMERO\_WHATSAPP\_REMISOR", "whatsapp:+1415XXXXXXX")

# Cliente Twilio

client = Client(ACCOUNT\_SID, AUTH\_TOKEN)

# Comandos disponibles

COMANDOS\_VALIDOS = {

"status": "✅ SentinelCore operativo.",

"ayuda": "🤖 Comandos: status | escanear | detener | informe",

"escanear": "🔍 Iniciando escaneo completo...",

"detener": "🛑 Cancelando tareas activas.",

"informe": "📄 Informe generado. Descárguelo desde el panel."

}

@app.route("/webhook", methods=["POST"])

def recibir\_mensaje():

mensaje = request.form.get("Body", "").strip().lower()

numero = request.form.get("From", "")

logging.info(f"Mensaje recibido de {numero}: {mensaje}")

respuesta = COMANDOS\_VALIDOS.get(mensaje, "⚠️ Comando no reconocido. Escriba 'ayuda'.")

# Enviar respuesta por WhatsApp

try:

client.messages.create(

body=respuesta,

from\_=FROM\_WA,

to=numero

)

logging.info(f"Respuesta enviada a {numero}: {respuesta}")

except Exception as e:

logging.error(f"Error al enviar mensaje: {e}")

return "Error interno", 500

return "OK", 200

# Para pruebas locales

if \_\_name\_\_ == "\_\_main\_\_":

app.run(port=5005, debug=True)

**🔐 Notas de seguridad y despliegue:**

* Protege este endpoint (/webhook) con verificación de firma de Twilio si lo expones públicamente.
* Requiere tener tu número aprobado en WhatsApp Business o Twilio Sandbox.
* Se recomienda usar ngrok para pruebas locales con webhook si aún no está en producción.

¿Te gustaría que también te genere una **plantilla JSON de Postman** para probar este webhook fácilmente, o continuo con la siguiente parte (autenticacion.py + JWT)?