

Manual Técnico del Asistente Virtual

1. Comandos necesarios:

- ``pip install -r requirements.txt``: Este comando instalará todas las dependencias necesarias para ejecutar el proyecto del asistente virtual.

2. Explicación del Proyecto:

El Asistente Virtual es un programa desarrollado en Python que permite interactuar con una computadora mediante comandos de voz. El objetivo principal del proyecto es proporcionar una forma conveniente y rápida de realizar tareas utilizando la voz como entrada.

3. Explicación Técnica del Desarrollo:

El asistente se ha desarrollado utilizando el lenguaje de programación Python y se han utilizado diversas bibliotecas y módulos para agregar funcionalidades específicas

- ``speech_recognition``: Esta biblioteca se utiliza para reconocer y capturar comandos de voz del usuario. Se utiliza el servicio de reconocimiento de voz de Google para convertir el audio en texto.

- ``gtts`` (Google Text-to-Speech): Esta biblioteca se utiliza para convertir texto en voz.

Permite al asistente responder a través de la síntesis de voz.

- ``webbrowser``: Este módulo se utiliza para abrir páginas web en un navegador. Permite al asistente abrir Google y reproducir videos de YouTube.
- ``datetime``: Este módulo se utiliza para obtener la hora actual del sistema. Permite al asistente proporcionar la hora actual cuando se le solicita.
- ``wikipedia``: Esta biblioteca se utiliza para buscar y obtener información de Wikipedia. Permite al asistente buscar en Wikipedia y proporcionar un resumen de los resultados.
- ``smtplib``: Este módulo se utiliza para enviar correos electrónicos. Permite al asistente enviar correos electrónicos a través de un servidor SMTP.
- ``cv2`` (OpenCV): Esta biblioteca se utiliza para acceder y manipular la cámara de la computadora. Permite al asistente tomar fotos utilizando la cámara.
- ``googletrans``: Esta biblioteca se utiliza para traducir texto. Permite al asistente traducir los resultados de Wikipedia a otros idiomas.
- ``pywhatkit``: Esta biblioteca se utiliza para buscar y reproducir videos de YouTube. Permite al asistente buscar videos y reproducirlos en el navegador web.

4. Explicación Técnica de las Funciones (Comandos):

- ``escuchar(nombre)``: Esta función utiliza el reconocimiento de voz para capturar comandos de voz del usuario. Se utiliza la biblioteca ``speech_recognition`` para realizar la captura de audio y convertirlo en texto. Los pasos para esta función son:

1. Se crea una instancia del reconocedor de voz utilizando ``reconocedor = sr.Recognizer()``.

2. Se abre el micrófono como fuente de audio utilizando ``with sr.Microphone() as source:``.

3. Se escucha el audio del micrófono y se captura utilizando ``audio = reconecedor.listen(source, phrase_time_limit=5)``.

4. Se utiliza el servicio de reconocimiento de voz de Google para convertir el audio en texto mediante ``reconecedor.recognize_google(audio, language='es')``.

5. Se comprueba si el nombre del asistente está presente en el comando reconocido. Si es así, se extrae el comando del nombre utilizando ``comando.replace(nombre, "").strip()`` y se devuelve el comando resultante. En caso contrario, se devuelve una cadena vacía.

- ``hablar(texto)``: Esta función utiliza la síntesis de voz para convertir texto en voz. Se utiliza la biblioteca ``gtts`` para generar el archivo de audio correspondiente al texto proporcionado como parámetro. Los pasos para esta función son:

1. Se crea una instancia de la clase ``gTTS`` con el texto y el idioma deseado, por ejemplo, ``tts = gTTS(text=texto, lang='es')``.

2. Se guarda el archivo de audio generado utilizando ``tts.save("audio.mp3")``.

- ``reproducir_video(url)``: Esta función utiliza el módulo ``webbrowser`` para abrir una URL en el navegador web predeterminado. Se utiliza para reproducir videos de YouTube proporcionando la URL del video como parámetro. El único paso para esta función es ``webbrowser.open(url)``.

- ``obtener_hora_actual()``: Esta función utiliza el módulo ``datetime`` para obtener la hora actual del sistema. El único paso para esta función es ``datetime.datetime.now()`` para

obtener el objeto ``datetime`` que representa la hora actual, y luego se utiliza el método ``strftime`` para formatear la hora como una cadena.

- ``buscar_wikipedia(consulta)``: Esta función utiliza la biblioteca ``wikipedia`` para buscar información en Wikipedia basada en una consulta proporcionada como parámetro. Los pasos para esta función son:

1. Se utiliza ``wikipedia.summary(consulta, sentences=2)`` para obtener un resumen de los resultados de búsqueda.

2. Se utiliza la biblioteca ``googletrans`` para realizar la traducción del resumen obtenido a español. Se crea una instancia de ``Translator`` con ``translator = Translator(to_lang="es", from_lang="en")`` y luego se utiliza ``translator.translate(respuesta)`` para obtener la traducción del resumen.

- ``abrir_google()``: Esta función utiliza el módulo ``webbrowser`` para abrir la página de inicio de Google en el navegador web predeterminado. El único paso para esta función es ``webbrowser.open("https://www.google.com")``.

- ``enviar_email(destinatario, asunto, mensaje)``: Esta función utiliza el módulo ``smtplib`` para enviar un correo electrónico. Los pasos para esta función son:

1. Se define el remitente del correo electrónico y se lee la contraseña desde un archivo utilizando la función ``leer_contraseña_desde_archivo(pwd)``.

2. Se crea una instancia de ``smtplib.SMTP`` con el servidor SMTP y el puerto correspondiente, por ejemplo, ``server = smtplib.SMTP(servidor_smtp, puerto_smtp)``.

3. Se inicia la conexión segura con el servidor utilizando ``server.starttls()``.

4. Se autentica en el servidor utilizando ``server.login(remitente, password)``.

5. Se construye el correo electrónico utilizando ``email = f"Subject: {asunto}\n\n{mensaje}"``.

6. Se envía el correo electrónico utilizando ``server.sendmail(remitente, destinatario, email)``.

- ``buscar_y_reproducir_video(busqueda)``: Esta función utiliza la biblioteca ``pywhatkit`` para buscar y reproducir videos de YouTube basados en una búsqueda proporcionada como parámetro. El único paso para esta función es ``rep.playonyt(busqueda)``.

- ``tomar_foto()``: Esta función utiliza el módulo ``cv2`` para acceder a la cámara de la computadora y tomar una foto. Los pasos para esta función son:

1. Se crea una instancia de ``cv2.VideoCapture`` con el índice ``0`` para acceder a la cámara predeterminada.

2. Se utiliza ``cam.read()`` para leer el cuadro actual de la cámara.

3. Si se obtiene un cuadro válido, se guarda como una imagen utilizando ``cv2.imwrite("foto.png", frame)``.

5. Conclusiones:

El Asistente Virtual es un proyecto que combina el reconocimiento de voz, la síntesis de voz y diversas bibliotecas y módulos de Python para ofrecer una forma intuitiva y conveniente de interactuar con una computadora mediante comandos de voz. El proyecto se puede personalizar y ampliar según las necesidades específicas de cada usuario. Al seguir

las instrucciones de instalación y utilizar los comandos mencionados, se puede disfrutar de todas las funcionalidades proporcionadas por el asistente virtual.

Recuerda tener en cuenta los requisitos del sistema y las dependencias mencionadas en el archivo `requirements.txt` para asegurarte de que el proyecto funcione correctamente.