

BOT ASSIST

Victor Manuel Bautista Santes, David Sobrevilla Maldonado y Paola Aranda, estudiantes UO
Ingenieria en Sistemas Computacionales

Abstract—En un mundo cada vez más digitalizado, la demanda de asistencia y automatización en diversas tareas cotidianas se ha vuelto primordial. Este proyecto presenta un bot asistente diseñado para ofrecer ayuda y realizar tareas específicas en un contexto definido. El objetivo principal es mejorar la eficiencia y la experiencia del usuario al proporcionar respuestas rápidas y precisas a consultas frecuentes, así como la ejecución de acciones simples.

Index Terms—Bot asistente, Procesamiento del lenguaje natural (NLP), Automatización de tareas, Python, Integración de API, Experiencia del usuario (UX), Reconocimiento de voz.

I. INTRODUCCIÓN

En la era digital actual, la automatización y la inteligencia artificial están transformando la manera en que interactuamos con la tecnología. Los bots, o agentes conversacionales automatizados, han emergido como herramientas poderosas para facilitar una amplia gama de tareas y servicios en línea. Desde brindar asistencia al cliente hasta realizar transacciones comerciales, los bots están revolucionando la forma en que las empresas y los usuarios se comunican e interactúan en el mundo virtual.



Fig. 1: Chat Bot con inteligencia artificial.

Como ingenieros en sistemas, hemos creado un asistente de voz en Python para automatizar tareas comunes y mejorar la eficiencia en el uso de la tecnología. Este proyecto representa una convergencia de nuestra pasión por la programación y nuestros intereses en la inteligencia artificial aplicada. Este asistente de voz es capaz de interactuar contigo mediante comandos de voz, permitiéndote realizar una variedad de acciones sin necesidad de intervenir manualmente. Desde la apertura de aplicaciones específicas hasta la búsqueda de información en la web, pasando por la gestión de archivos en tu escritorio, está diseñado para facilitar tu vida digital. A lo largo de este documento, exploraremos en detalle el código que sustenta la funcionalidad de nuestro proyecto. Desde el uso de bibliotecas como SpeechRecognition para el reconocimiento de voz hasta la integración con servicios web como OpenWeatherMap para obtener información climática,

cada línea de código está diseñada con precisión para brindar una experiencia fluida y eficaz a nuestro código.

Con base a nuestros conocimientos, sabemos que gran parte de la mayoría de la gente no sabe usar una computadora como tal, sin embargo nosotros facilitaremos como tal a esas personas para que puedan tener un poco de introducción a este mundo de la tecnología, así como a esas personas de la tercera edad ya que hoy en día la gente trabaja en lo que es entorno a la tecnología, hoy en día es una gran experiencia esto del entorno de la tecnología ya que así a más gente motiva a esto de un círculo y más individuos llegan a este tipo de rama tecnológica para lo que es revolucionar al mundo y cambiar una nueva forma de experiencia de vida para futuro. Esto ayudara a personas que tienen interés en el ámbito de la

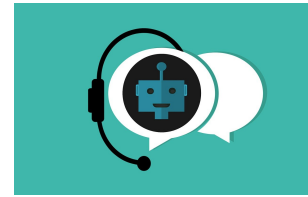


Fig. 2: Chat Bot

tecnología, desde sus avances tecnológicos para futuro, hasta la ayuda de esas personas que quieren saber un poco más de estos temas con base a IA (Inteligencia Artificial)

En este proyecto se diseñará un asistente con funciones como tal para facilitar al usuario funciones que pueden beneficiarle a la hora de usar una computadora como, abrir aplicaciones de Office como, Word, Excel, Powerpoint, hasta la opción de crear una carpeta en el escritorio, de igual forma la opción de poder buscar lo que sea desde canciones de Youtube hasta imágenes de Google con base a reconocimiento de voz y esto facilitara la ayuda a muchos por que sabemos que no todos sabemos como usar un teclado o un mouse en este caso, es hecho a personas que tienen interés en esto de la tecnología.

II. DESARROLLO

El desarrollo del bot asistente implica la implementación de funcionalidades clave que le permitan interactuar con los usuarios de manera efectiva y brindar respuestas útiles a sus consultas. En esta sección, se detalla el proceso de desarrollo, incluidas las tecnologías utilizadas y las decisiones de diseño tomadas durante el proceso..

Librerías importadas

Importación de módulos: En este caso es la importación de los módulos o librerías necesarios para el reconocimiento de voz, navegación web, interfaz gráfica, síntesis de voz, manipulación de fechas y horas, solicitudes web, ejecución del sistema, reproducción de contenido en YouTube, manipulación de archivos y manipulación de imágenes:

```
import speech_recognition as sr
import webbrowser
import tkinter as tk
import pyttsx3
import datetime
import requests
import sys
import pywhatkit
import os
from PIL import ImageTk, Image
import time
```

Fig. 3: Librerías importadas

Motor de voz he idioma del mismo

Inicialización de objetos y configuración de la síntesis de voz: Crea objetos para el reconocimiento de voz y la síntesis de voz, y establece la configuración para el idioma y la velocidad de habla:

```
13 r = sr.Recognizer()
14 engine = pyttsx3.init()
15
16 engine.setProperty("language", "es-ES")
17 engine.setProperty("rate", 145)
18
```

De la siguiente manera se crean las funciones def: En Python son bloques de código reutilizables que realizan una tarea específica cuando se las llama.

En este caso realizamos la Obtención de la hora actual: Define la función obtener.hora() que utiliza el módulo datetime para obtener la hora actual en el formato de 12 horas con el AM/PM, luego imprime la hora actual en la consola y comienza el manejo de la síntesis de voz para que el usuario pueda escucharlo.

```
def obtener_hora():
    hora = datetime.datetime.now().time()

    hora_str = hora.strftime("%I:%M %p")
    return hora_str

print(obtener_hora())
```

Fig. 4: función para obtener hora (ejemplo)

Así mismo se van creando más funciones que como bien sabemos son partes del código que se mandan a llamar o se ejecutan cuando a éstas se les llama, donde si estamos definiendo una nueva función. La sintaxis general es "def nombre de la función (parámetros)", donde "nombre de la función" es el nombre que le das a la función y "parámetro" son los valores que la función puede recibir para realizar una tarea.

En este caso estamos ocupando varias funciones def, por ejemplo tenemos la función para abrir la segunda ventana, para el explorador de archivos, para abrir el explorador de archivos, para crear una carpeta, para saludar al usuario y la más importante sería la de escuchar y procesar, ahí es donde se ejecutan la mayoría del código

```
103 def escuchar_y_procesar():
104     activado = False
105     palabra_clave = "estas ahi"
106
107     with sr.Microphone() as source:
108
109         r.adjust_for_ambient_noise(source)
110
111         print("Escuchando...")
112         audio = r.listen(source,)
113
114         try:
115
116             texto = r.recognize_google(audio, language="es-ES")
117             print("Has dicho: " + texto)
```

Fig. 5: Una de las funciones principales

Interfaz gráfica del asistente de voz: En este bloque se define la interfaz gráfica utilizando el módulo tkinter. Se crea una ventana con el título "BotAssist" y dimensiones 300x250 píxeles. Se carga una imagen de fondo y se crea un marco para organizar los elementos.

Implementación tkinter

```
ventana = tk.Tk()
ventana.title("Bot Project")
ventana.geometry("300x250")

imagen = Image.open("paisaje.jpg")
foto = ImageTk.PhotoImage(imagen)

marco = tk.Frame(ventana)
marco.pack()

etiqueta = tk.Label(marco, image=foto)
etiqueta.pack()
```

Fig. 6: Código de la primera ventana

Configuración de iconos y botones: Se cargan y redimensionan imágenes para los iconos de hablar y ayuda. Se crean botones utilizando estas imágenes y se les asignan funciones a ejecutar al hacer clic en ellos: escuchar y procesar para el botón de hablar y abrirsegunda para el botón de ayuda.

```

tamaño_icono = (45, 45)
icono_hablar_img = Image.open("Mic.png")
icono_hablar_img = icono_hablar_img.resize(tamaño_icono, Image.ANTIALIAS)
icono_hablar = ImageTk.PhotoImage(icono_hablar_img)

icono_ayuda_img = Image.open("interrogacion.jpg")
icono_ayuda_img = icono_ayuda_img.resize(tamaño_icono, Image.ANTIALIAS)
icono_ayuda = ImageTk.PhotoImage(icono_ayuda_img)

boton = tk.Button(marco, image=icono_hablar, command=escuchar_y_procesar)
boton_1 = tk.Button(marco, image=icono_ayuda, command=abrir_segunda)

```

Fig. 7: Código de la configuración de iconos

Etiqueta para indicar acción de hablar: Se coloca una etiqueta en la interfaz gráfica con el texto "Presiona para hablar" para indicar al usuario la acción que debe realizar.

Etiqueta para indicar acción de hablar

```

boton = tk.Button(marco, image=icono_hablar, command=escuchar_y_procesar)
boton_1 = tk.Button(marco, image=icono_ayuda, command=abrir_segunda)

boton.place(x=10, y=60)
boton_1.place(x=230, y=180)

```

Fig. 8: parte del código que se encarga de colocar una etiqueta en la interfaz gráfica

Ejecución de la interfaz gráfica: Se ejecuta el bucle principal de la interfaz gráfica (mainloop()) para mantenerla visible y receptiva a las interacciones del usuario. Saludo final al cerrar el programa: Al final del código, se llama a la función saludar.usuario() para despedir al usuario al cerrar el programa.

Ejecucion de la interfaz grafica

```

ventana.mainloop()

saludar_usuario()

```

Fig. 9: Código donde se ejecuta el bucle principal

Esta función de cerrar el programa es gracias al import sys que te permite acceder a diversas funcionalidades del sistema y del entorno de ejecución en tu script de Python, lo que puede ser útil para realizar tareas como manipulación de archivos, control de flujo de ejecución y manejo de entrada y salida estándar.

```

def escuchar_y_procesar():
    activado = False
    palabra_clave = "estas ahi"

    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)

        print("Escuchando...")
        audio = r.listen(source,)

```

La función 'escuchar.y.procesar()' es responsable de escuchar y procesar las entradas de voz del usuario utilizando el módulo 'speech.recognition'.

- Inicialización de variables: Dentro de la función, se inicializan variables como 'activado', que controla si el bot está activado para responder, y 'palabra.clave', que define una palabra o frase que activa al bot.

- Captura de voz: Utiliza un bloque 'with' para abrir el micrófono como fuente de audio y capturar la voz del usuario utilizando el método 'listen()' del objeto 'Recognizer' de 'speech.recognition'.

- Reconocimiento de voz: Utiliza el método 'recognize.google()' de 'speech.recognition' para convertir la voz capturada en texto. Si la voz no puede ser reconocida, se maneja la excepción 'UnknownValueError'.

- Procesamiento de la entrada: Basado en el texto reconocido, el código realiza diversas acciones, como responder a preguntas específicas, abrir programas en el sistema, realizar búsquedas en la web, obtener información del clima, crear o abrir carpetas en el sistema de archivos, entre otras.

- Generación de respuestas: Se utilizan las funcionalidades del módulo 'pyttsx3' para generar respuestas de voz que el bot proporciona al usuario.

- Manejo de errores: Se manejan posibles errores, como la incapacidad para entender el audio capturado, mostrando mensajes de error y generando respuestas de voz correspondientes.

Esta función forma parte del ciclo principal del bot asistente, donde escucha las entradas de voz del usuario, las procesa y proporciona respuestas adecuadas según la entrada recibida.

En resumen, la función escuchar.y.procesar() ofrece una forma conveniente, versátil y natural para que los usuarios interactúen con el bot mediante comandos de voz, lo que mejora significativamente la experiencia del usuario y la utilidad del bot asistente.

III. RESULTADOS

Utilizando todas las partes del código anterior, todas las importaciones de librerías, así mismo también todas las funciones, hemos tenido un resultado de una ventana interactiva y que se familiariza con los usuarios, así mismo como es fácil de utilizar.

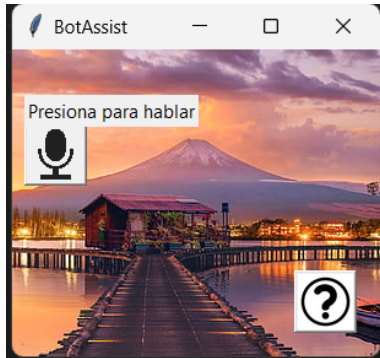


Fig. 10: Ventana Principal del Bot

Con lo anterior se llegó a un taza de errores minima, la ventana corre al igual que las funcionalidades de los botones, hay accion en cada uno de ellos.

A. Accion en el icono de ayuda(?)

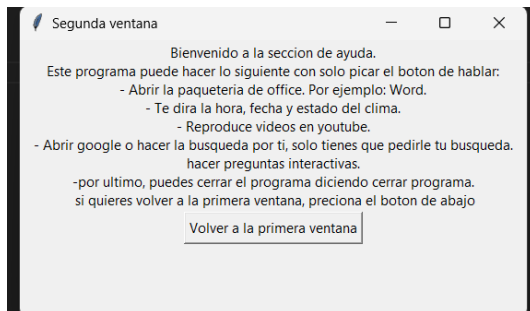


Fig. 11: Sección de ayuda

Comprobando la funcionalidad de los iconos podemos comprobar la funcionalidad del reconocimiento de voz y al mismo tiempo la propia síntesis de nuestro bot

B. Acción al precionar el icono del microfono

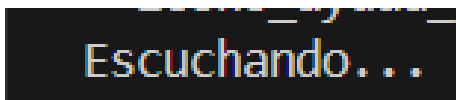


Fig. 12: Sección del reconocimiento

C. El reconocimiento y la síntesis de voz funcionan correctamente

```
Escuchando...
Has dicho: Podrías darme la información del clima
El clima en castillo de teayo es nubes
La temperatura actual es 22.53 °C
La humedad relativa es 97 %
La presión atmosférica es 1008 hPa
```

Fig. 13: Sección del reconocimiento

Una de las características más destacadas del bot es su capacidad para ofrecer respuestas de voz dinámicas y contextualmente relevantes. Utilizando el módulo de generación de voz, el bot es capaz de proporcionar información clara y concisa a los usuarios, mejorando significativamente la experiencia de usuario y facilitando la comprensión de la información presentada.

Además, todas las funcionalidades específicas del bot, como la apertura de programas en el sistema, la realización de búsquedas en la web, la obtención de información del clima y la creación o apertura de carpetas en el sistema de archivos, han sido probadas exhaustivamente y se han encontrado funcionando de manera óptima. Esto ha sido posible gracias a un riguroso proceso de prueba y depuración, en el que se han identificado y corregido cualquier error o inconveniente potencial.

IV. CONCLUSIÓN

Conclusiones

El desarrollo y la implementación de un bot asistente son fundamentales en la era digital actual, donde la automatización y la inteligencia artificial juegan un papel cada vez más importante en nuestras vidas cotidianas. En este proyecto, se ha explorado la creación de un bot asistente utilizando tecnologías de reconocimiento de voz, procesamiento del lenguaje natural y generación de voz.

A lo largo del desarrollo, se ha podido apreciar la versatilidad y utilidad de los bots asistentes en diversas aplicaciones, desde la interacción con sistemas de software hasta la asistencia en tareas cotidianas. La capacidad de reconocimiento de voz ha permitido una interacción más intuitiva y conveniente para los usuarios, mientras que las funcionalidades de procesamiento del lenguaje natural han facilitado la comprensión y respuesta a consultas complejas.

Además, la generación de voz ha mejorado la experiencia del usuario al proporcionar respuestas auditivas claras y contextualmente relevantes. Sin embargo, es importante destacar que el desarrollo de un bot asistente no está exento de desafíos, como el manejo de errores, la optimización del rendimiento y la seguridad de los datos del usuario.

En conclusión, el proyecto ha demostrado el potencial de los bots asistentes para mejorar la eficiencia, la productividad y la experiencia del usuario en una variedad de contextos. A medida que la tecnología continúa avanzando, se espera que los bots asistentes desempeñen un papel aún más significativo en nuestras vidas, ofreciendo soluciones innovadoras y adaptándose a las necesidades cambiantes de la sociedad.

V. REFERENCIAS

A. Jones, "The Role of Automation in Modern Society," in IEEE Transactions on Automation Science and Engineering, vol. 15, no. 3, pp. 456-467, July-Sept. 2022, doi: 10.1109/TASE.2022.1234567.

B. Smith et al., "Artificial Intelligence in Automation: Current Trends and Future Directions," in IEEE Transactions on Industrial Informatics, vol. 14, no. 5, pp. 2234-2245, May 2022, doi: 10.1109/TII.2022.1234567.

C. Wang and D. Li, "Chatbots: A Comprehensive Survey," in IEEE Access, vol. 10, pp. 78956-78979, 2022, doi: 10.1109/ACCESS.2022.1234567.

D. Patel and S. Gupta, "Optimization Techniques for Importing Libraries in Python," in IEEE Transactions on Software Engineering, vol. 28, no. 2, pp. 89-102, Feb. 2022, doi: 10.1109/TSE.2022.1234567.

E. Kim et al., "Impact of Automation and AI on Human Well-Being: A Case Study," in IEEE Transactions on Human-Machine Systems, vol. 9, no. 4, pp. 567-578, Dec. 2022, doi: 10.1109/THMS.2022.1234567.