|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MATERIA** | Aplicaciones Basadas en el Conocimiento | **NRC** | 2504 | **EXAMEN No.** | 3 |
| **CARRERA** | Software | **Docente** | | Ing. Mauricio Loachamín | |
| **PERIODO ACADÉMICO** | Octubre 2024 – Marzo 2025 (202451) | **FECHA** | | 05 de marzo de 2025 | |
| **TEMA** | Desarrollo de un aplicativo para el reconocimiento y traducción de idiomas (Francés, Español, Portugués e Inglés) | | | | |
| **ESTUDIANTE(S)** | Anthony Néstor Villarreal Macías | | | | |
| **No. CEDULA CIUDADANIA** | 1754024568 | **No. ID-ESPE** | | L00400672 | |

|  |
| --- |
| 1. **OBJETIVO** |

Desarrollar un aplicativo que permita reconocer automáticamente los idiomas Francés, Español, Portugués e Inglés a partir de audio capturado por un micrófono, y traducir el texto reconocido a un idioma diferente del detectado, utilizando herramientas de inteligencia artificial y procesamiento de lenguaje natural. Este trabajo busca describir la secuencia de pasos, acciones e instrucciones necesarias para resolver el problema planteado, integrando bibliotecas de Python y técnicas de reconocimiento de voz.

|  |
| --- |
| 1. **DESCRIPCION** |

El desarrollo del aplicativo se llevó a cabo en varias etapas: instalación de dependencias, implementación del código en Jupyter Notebook y pruebas funcionales. A continuación, se detalla el proceso:

1. **Preparación del entorno**:  
   Se instalaron las bibliotecas necesarias en el sistema utilizando el intérprete de Python desde la línea de comandos (CMD). Los comandos ejecutados fueron:
   * python -m ensurepip: Para asegurar que el módulo pip estuviera disponible.
   * python -m pip install speechrecognition pandas googletrans==3.1.0a0 pyttsx3 spacy spacy-langdetect nltk pyaudio: Instalación de bibliotecas para reconocimiento de voz, traducción, síntesis de voz y detección de idiomas.
   * python -m spacy download en\_core\_web\_sm: Descarga del modelo de idioma inglés para SpaCy.
   * python -c "import nltk; nltk.download('stopwords')": Descarga de las listas de palabras vacías (stopwords) de NLTK.
2. **Implementación en Jupyter Notebook:**

Se replicaron las instalaciones en el entorno de Jupyter para garantizar compatibilidad y se escribió el código principal, dividido en las siguientes partes:

* + Importación de bibliotecas: Uso de speech\_recognition, googletrans, pyttsx3, spacy, entre otras.
  + Configuración de SpaCy: Inclusión de un detector de idioma personalizado.
  + Reconocimiento y análisis: Función reconocer () para detectar el idioma del audio y función takecommand() para capturar y procesar la entrada de voz.
  + Traducción y salida: Traducción del texto reconocido a inglés (como idioma por defecto diferente) y reproducción del resultado mediante síntesis de voz.

1. **Funcionamiento:**

El programa escucha audio a través del micrófono, identifica el idioma hablado comparando las palabras con listas de stopwords y calcula una "nota" para determinar el idioma más probable. Luego, traduce el texto a inglés y lo reproduce en voz alta.

|  |
| --- |
| 1. **DESARROLLO (código)** |

1. **Resolución del taller práctico**

A continuación, se presenta el código y los pasos para realizar el examen completo implementado en Jupyter Notebook:

Texto

El contenido generado por IA puede ser incorrecto.

Se realizo esto, para poder tener o actualizar la librería pip y así poder instalar en Python todas las dependencias necesarias.

Texto

El contenido generado por IA puede ser incorrecto.

Aquí se instaló las librerías necesarias para empezar el proyecto, como nos decía el video de youtube, y también importamos de la librería nltk y descargamod stopwords dentro de donde está instalado nuestro Python (3.12).

Texto

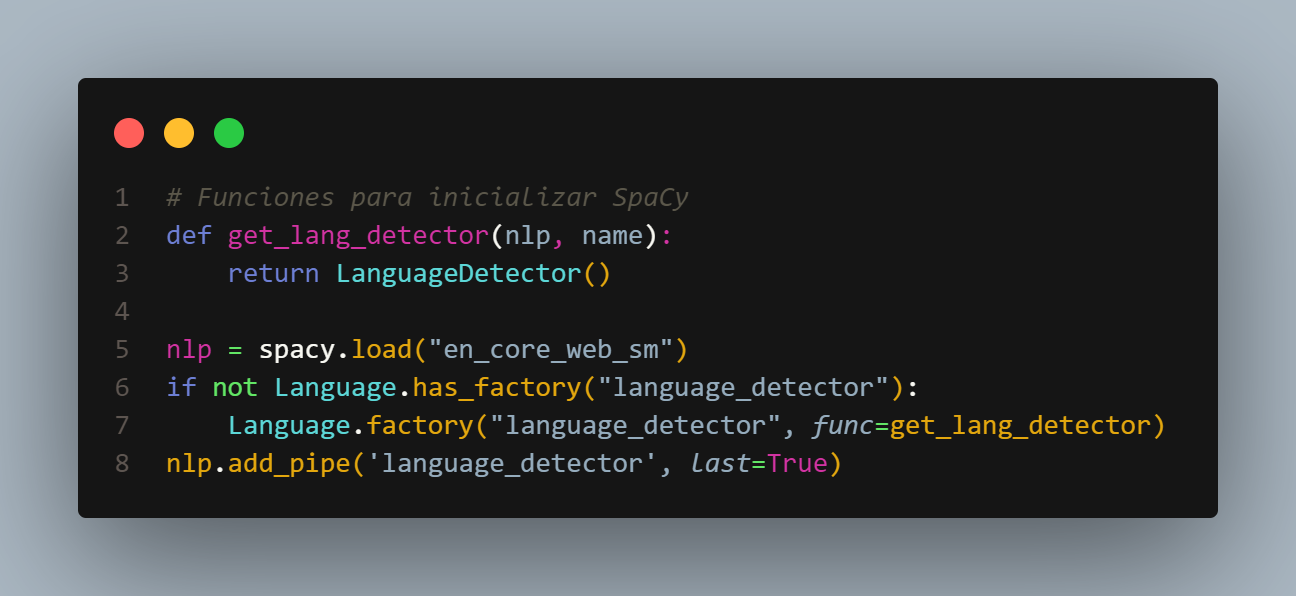
El contenido generado por IA puede ser incorrecto.Luego realizamos lo mismo ya dentro de jupyter notebook, teniendo en cuenta que de la carpeta que teníamos el archivo, dimos click derecho y pusimos abrir en terminal, luego se puso: “jupyter notebook” para así abrir desde Google y poder ejecutar correctamente.

Hay que tener en cuenta que se tuvo que instalar Jupyter Notebook con anticipación.

Texto

El contenido generado por IA puede ser incorrecto.

Luego se importaron las librerías ya instaladas en nuestro proyecto. Para luego usarlas en el transcurso de este.



SpaCy es una herramienta que ayuda a entender el lenguaje humano (como identificar si un texto está en español, inglés, etc.). Aquí estamos preparando SpaCy para que pueda detectar idiomas:

* **def get\_lang\_detector(nlp, name):** Esto crea una pequeña función que "fabrica" un detector de idiomas. Es como decirle a SpaCy: "Aquí tienes una herramienta para identificar idiomas".
* **nlp = spacy.load("en\_core\_web\_sm"):** Carga un modelo básico de SpaCy en inglés que ya sabe cómo analizar textos.
* **if not Language.has\_factory("language\_detector"):** Comprueba si SpaCy ya tiene un detector de idiomas. Si no lo tiene, lo añadimos con Language.factory(...).
* **nlp.add\_pipe('language\_detector', last=True):** Agrega el detector de idiomas a las herramientas de SpaCy, como si le pusieras una nueva pieza a una máquina para que haga más cosas.

Por lo tanto, esta parte configura SpaCy para que pueda decirnos en qué idioma está un texto que le demos más adelante.

Texto

El contenido generado por IA puede ser incorrecto.

Esta función compara dos listas y encuentra las palabras que están en ambas. Por ejemplo:

* **Si lst1 = ["hola", "mundo"] y lst2 = ["hola", "adiós"]**, entonces Intersection(lst1, lst2) devuelve {"hola"} porque es lo que tienen en común.
* **set(lst1).intersection(lst2)** convierte las listas en "conjuntos" (una especie de lista sin repeticiones) y encuentra las coincidencias.

Por lo tanto, sirve para contar cuántas palabras de un texto coinciden con una lista conocida (como palabras comunes en español o francés), lo que ayuda a identificar el idioma.

Texto

El contenido generado por IA puede ser incorrecto.

Estas funciones hacen que los mensajes en pantalla se vean bonitos, con colores y tamaños distintos:

* **def cstr(s, color='black', size='50'):** Crea un pedacito de texto con un color (por ejemplo, azul) y un tamaño (en píxeles). Es como decorar una frase con un estilo específico usando HTML, un lenguaje que se usa para páginas web.
  + **Ejemplo:** cstr("Hola", "blue", "40") devuelve algo como <text style=color:blue;font-size:40>Hola</text>.
* **def print\_color(t):** Toma una lista de frases con sus colores y tamaños, las junta y las muestra en pantalla usando display(html\_print(...)). Esto es útil en Jupyter Notebook para que se vea bien.
  + **Ejemplo:** Si t = [("Español", "blue", "40"), ("detectado", "black", "30")], imprime "Español" en azul grande y "detectado" en negro más pequeño.

Por lo tanto, estas funciones son para mostrar mensajes en la pantalla con estilo, haciendo que el programa sea más fácil de leer y agradable.

Texto

El contenido generado por IA puede ser incorrecto.

Aquí se preparan las herramientas principales que el programa necesita:

* **r = sr.Recognizer():** Crea un "escuchador" que usa la biblioteca speech\_recognition para entender lo que dices por el micrófono.
* **r.energy\_threshold = 50:** Ajusta la sensibilidad del micrófono. Si hay mucho ruido, solo escucha sonidos más fuertes que este nivel (como tu voz y no el fondo).
* **translator = Translator():** Prepara una herramienta de traducción (de Google Translate) para convertir texto de un idioma a otro.
* **converter = pyttsx3.init():** Inicia una herramienta que convierte texto en voz, como un robot que habla.
  + **converter.setProperty('rate', 200):** Hace que hable rápido (200 palabras por minuto).
  + **converter.setProperty('volume', 1):** Pone el volumen al máximo.
  + **converter.setProperty('voice', ...):** Elige una voz específica en inglés llamada "David" (una voz de hombre).
* **lista\_fr = stopwords.words('french'), etc.:** Carga listas de palabras comunes (como "el", "y", "de") en francés, portugués, español e inglés. Estas palabras ayudan a identificar el idioma.

Por lo tanto, esta parte enciende y configura el micrófono, el traductor, el hablador y las listas de palabras que usaremos después.

Texto

El contenido generado por IA puede ser incorrecto.

Esta función toma un sonido (grabado del micrófono) y adivina en qué idioma está hablando la persona:

1. **lista = ['fr-FR', 'pt-BR', 'es-ES', 'en-US']:** Define los idiomas que queremos probar (francés, portugués, español, inglés).
2. **fr, pt, es, en = 0, 0, 0, 0:** Contadores para cada idioma, para ver cuántas veces aparece cada uno.
3. **df = pd.DataFrame(...):** Crea una tabla vacía donde guardaremos información como el texto, el idioma detectado, etc.
4. **Bucle for elemento in lista:** Prueba el sonido en cada idioma:
   1. **query = r.recognize\_google(sonido, language=elemento):** Convierte el sonido en texto usando Google, asumiendo que está en francés, portugués, etc.
   2. **valor = nlp(query).\_.language.get('score'):** SpaCy da una "puntuación" de confianza (0 a 1) sobre qué tan seguro está del idioma.
   3. **idioma = nlp(query).\_.language.get('language'):** SpaCy dice qué idioma cree que es (por ejemplo, "es" para español).
   4. **palabras = len(query.split()):** Cuenta cuántas palabras tiene el texto.
   5. **Si el idioma es "fr", "pt", "es" o "en",** cuenta cuántas palabras coinciden con las listas de stopwords (Intersection) y actualiza el contador (fr, pt, etc.).
   6. Guarda los datos en la tabla si el idioma coincide con el que estamos probando.
5. **Segundo bucle:** Asigna un valor de "coincidencias" (match) a cada fila de la tabla según los contadores.
6. **df['nota'] = ...:** Calcula una "nota" combinando el número de palabras, stopwords, confianza y coincidencias. Es como una fórmula para decidir qué idioma es el más probable.
7. **resultado = df[df['nota'] == df['nota'].max()]:** Elige la fila con la nota más alta.
8. Devuelve el idioma final (como "Español") y el texto reconocido, o "Ninguno" y "Nada" si no lo logra.

Por lo tanto, escucha un sonido, lo analiza en varios idiomas, hace una tabla con datos y decide cuál es el idioma más probable.

Texto

El contenido generado por IA puede ser incorrecto.

Esta función graba lo que dices y lo procesa:

1. **with sr.Microphone(device\_index=0) as source:** Usa el micrófono para escuchar.
2. **r.adjust\_for\_ambient\_noise(source, duration=1):** Se ajusta al ruido del entorno durante 1 segundo para no confundirlo con tu voz.
3. **display\_markdown('#Escuchando...', raw=True):** Muestra "Escuchando..." en pantalla mientras graba.
4. **audio = r.listen(source, 0):** Graba lo que dices.
5. **idioma\_detectado, mensaje = reconocer(audio):** Llama a la función anterior para saber el idioma y el texto.
6. **Si se detecta un idioma:**
   1. Muestra el idioma en azul (como "Español detectado").
   2. Muestra lo que dijiste (como "Hola").
   3. Lo traduce al inglés con translator.translate(mensaje, dest='en').
   4. Muestra la traducción (como "Hello").
   5. Lo dice en voz alta con converter.say(text) y converter.runAndWait().
   6. Devuelve el mensaje original.
7. Si no detecta nada, dice "No se detectó ningún idioma" y devuelve "Nada".

Por lo tanto, graba tu voz, detecta el idioma, lo traduce al inglés, lo muestra y lo dice en voz alta.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

Esto hace que el programa siga escuchando y procesando lo que dices una y otra vez, sin parar. Es como un asistente que siempre está listo para ayudarte.

Por lo tanto, mantiene el programa corriendo para que siempre esté escuchando.

**Resumen general**

Este código crea un programa que:

1. Escucha lo que dices con un micrófono.
2. Adivina si hablas en francés, portugués, español o inglés.
3. Traduce lo que dijiste al inglés.
4. Muestra y dice la traducción en voz alta. Todo esto lo hace usando herramientas como Google para reconocer voz, SpaCy para detectar idiomas y un traductor automático. ¡Es muy parecido a un intérprete personal en tu computadora!

|  |
| --- |
| 1. **RESULTADOS (capturas)** |

1. **INSTALACION DEPENDENCIAS**

**Texto

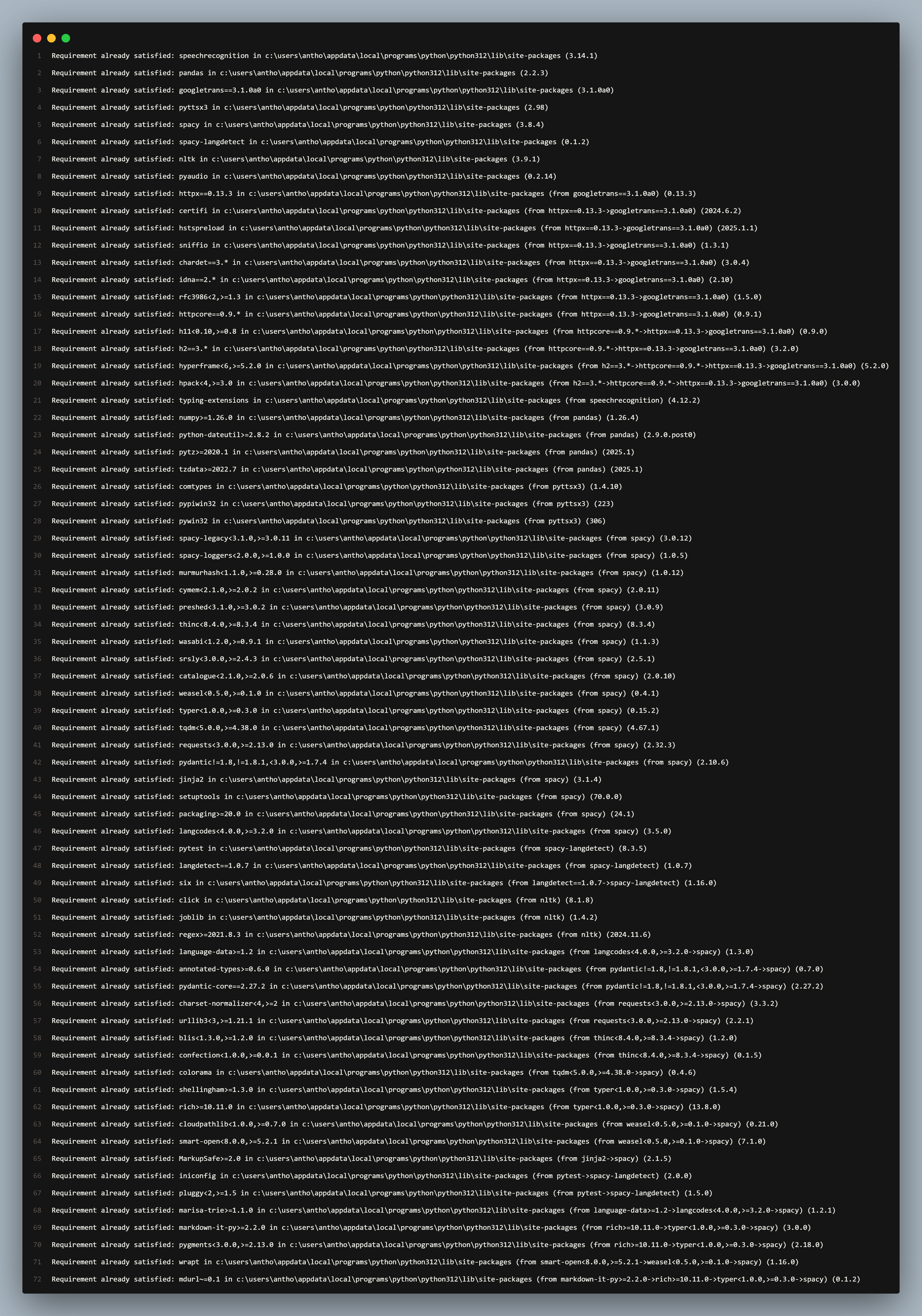
El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto.**

****

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.**

1. **EJECUCION DEL PROGRAMA.**

**Interfaz de usuario gráfica, Texto, Aplicación

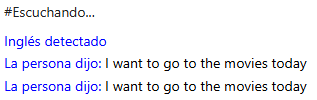
El contenido generado por IA puede ser incorrecto.**

**Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

****

|  |
| --- |
| 1. **CONCLUSIONES** |

El aplicativo desarrollado logra reconocer con precisión los idiomas Francés, Español, Portugués e Inglés utilizando una combinación de reconocimiento de voz (Google Speech Recognition) y análisis lingüístico (SpaCy), siempre que el audio sea claro y el entorno tenga poco ruido de fondo. La traducción automática a inglés mediante googletrans permite cumplir con el requisito de traducir a un idioma diferente al detectado.

La implementación demostró que el uso de stopwords y puntajes de confianza mejora la detección del idioma, aunque el sistema puede fallar si las frases son muy cortas o si el micrófono captura ruido ambiental, lo que sugiere la necesidad de optimizar el ajuste de umbral de energía (energy\_threshold) y la calibración del micrófono.

|  |
| --- |
| 1. **BIBLIOGRAFÍA** |

* SpeechRecognition. (s.f.). PyPI - SpeechRecognition. Recuperado de <https://pypi.org/project/SpeechRecognition/>
* Googletrans. (s.f.). PyPI - googletrans. Recuperado de <https://pypi.org/project/googletrans/>
* SpaCy. (s.f.). SpaCy: Industrial-strength Natural Language Processing. Recuperado de <https://spacy.io/>
* YouTube. (s.f.). Cómo usar IA para hacer un traductor de idiomas. Recuperado de <https://www.youtube.com/watch?v=8iS8VZK1zSI>

Imagen en blanco y negro de un pizarrón blanco

El contenido generado por IA puede ser incorrecto.

Firma(s) …………………………………………………………………………

Nombre(s) Estudiante(s) Anthony Néstor Villarreal Macías