**INFORME DEL PROYECTO DE INVESTIGACION**

**De voz a texto utilizando Python**

Integrantes:

Bianca Báez

Fernando Giménez

Fernanda Otero

Facultad de Ingeniería en Informática, Universidad Americana.

Introducción a la Programación.

Ing. Guillermo Ávila

24 de junio de 2023

**INTRODUCCION**

En la actualidad, existe una gran demanda por la automatización de tareas repetitivas y la transformación de voz a texto es una de ellas. Python, un lenguaje de programación con gran popularidad, brinda herramientas para hacerlo posible. En este proyecto, exploraremos cómo podemos transformar nuestro discurso en texto mediante el uso de librerías de Python y algunos pasos sencillos. Veremos la preparación de datos, la selección de la librería adecuada y cómo implementarla para obtener resultados precisos. Acompáñanos en este recorrido para aprender a transformar la voz en texto con Python de manera fácil y efectiva.

**JUSTIFICACIÓN**

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado por Guido van Rossum y lanzado por primera vez en 1991.. Elegimos utilizar Python en la creación del programa debido a que es frecuentemente utilizado en los últimos tiempos, por varias razones a continuación. Python se destaca por su diseño legible y su enfoque en la simplicidad y la facilidad de uso, teniendo también una amplia variedad de bibliotecas que facilitan el desarrollo de diferentes tipos de aplicaciones, siendo también productivo

**OBJETIVOS DE LA INVESTIGACIÓN**

Objetivo General:

* Desarrollar un programa que permita identificar textos y convertirlas en voces

Objetivos específicos:

Los objetivos específicos para realizar un programa de texto a voz con Python podrían ser:

* Utilizar técnicas de reconocimiento de voz para convertir el audio en texto.
* Implementar la funcionalidad para capturar audio utilizando un micrófono o cargar archivos de audio pregrabados en el programa.
* Mostrar o almacenar los audios obtenidos para una consecuente mejora

**MARCO TEÓRICO**

1. **PYTHON**

**1.1 Definición**

En términos técnicos, Python es un lenguaje de programación de alto nivel, orientado a objetos, con una semántica dinámica integrada, principalmente para el desarrollo web y de aplicaciones informáticas.

Es muy atractivo en el campo del Desarrollo Rápido de Aplicaciones (RAD) porque ofrece tipificación dinámica y opciones de encuadernación dinámicas.

Python es relativamente simple, por lo que es fácil de aprender, ya que requiere una sintaxis única que se centra en la legibilidad. Los desarrolladores pueden leer y traducir el código Python mucho más fácilmente que otros lenguajes.

Por tanto, esto reduce el costo de mantenimiento y de desarrollo del programa porque permite que los equipos trabajen en colaboración sin barreras significativas de lenguaje y experimentación. Además, soporta el uso de módulos y paquetes, lo que significa que los programas pueden ser diseñados en un estilo modular y el código puede ser reutilizado en varios proyectos. Una vez se ha desarrollado un módulo o paquete, se puede escalar para su uso en otros proyectos, y es fácil de importar o exportar.

Por otro lado, uno de los beneficios más importantes de Python es que tanto la librería estándar como el intérprete están disponibles gratuitamente, tanto en forma binaria como en forma de fuente.

Tampoco hay exclusividad, ya que Python y todas las herramientas necesarias están disponibles en todas las plataformas principales. Por lo tanto, es una opción multiplataforma, bastante tentadora para los desarrolladores que no quieren preocuparse por pagar altos costos de desarrollo.

* 1. **Historia**

Python fue creado por el informático Guido van Rossum, quien había estado trabajando con un lenguaje llamado ABC en su anterior trabajo en el Centrum Wiskunde & Informática (CWI) – Instituto Nacional de Investigación en Matemáticas e Informática en los Países Bajos-. Aunque le gustaban algunos aspectos de ABC, estaba frustrado por lo difícil que era difundir este lenguaje. Durante las vacaciones de Navidad de 1989, van Rossum decidió intentar crear su propio lenguaje. Poco más de un año después, en febrero de 1991, subió la primera versión de su creación a USENET. Por otro lado, le gustaba leer los textos de los episodios de «El circo volador de Monty Python», de la famosa compañía británica de comedia. Buscando un nombre que fuera «corto, único y ligeramente misterioso», eligió llamarlo Python.

* 1. **Usos de Python**

Python es un lenguaje de programación de propósito general, que es otra forma de decir que puede ser usado para casi todo. Lo más importante es que se trata de un lenguaje interpretado, lo que significa que el código escrito no se traduce realmente a un formato legible por el ordenador en tiempo de ejecución.

Este tipo de lenguaje también se conoce como «lenguaje de scripting» porque inicialmente fue pensado para ser usado en proyectos sencillos.

El concepto de «lenguaje de scripting» ha cambiado considerablemente desde su creación, porque ahora se utiliza Python para programar grandes aplicaciones de estilo comercial, en lugar de sólo las simples aplicaciones comunes.

Una encuesta realizada en 2019 entre los usuarios de Python indicó que los usos más populares eran para el desarrollo web y el análisis de datos. Sólo alrededor del 6 %  de los encuestados lo utilizaron para el desarrollo de juegos o el desarrollo de aplicaciones.

Esta dependencia de Python ha crecido aún más a medida que Internet se ha hecho más popular. Una gran mayoría de las aplicaciones y plataformas web dependen de su lenguaje, incluido el motor de búsqueda de Google, YouTube, y el sistema de transacciones orientado a la web de la Bolsa de Nueva York (NYSE).

En definitiva, sabes que el lenguaje debe ser realmente importante cuando se encarga de impulsar un sistema bursátil. De hecho, la NASA lo utiliza cuando programan sus equipos y maquinaria espacial.

Existen muchas aplicaciones comerciales para la programación en Python, pero el lenguaje también se ha afianzado en los círculos académicos, especialmente entre los que trabajan con grandes cantidades de datos.

También, puede ser usado para procesar texto, mostrar números o imágenes, resolver ecuaciones científicas y guardar datos. Nosotros en este caso, lo usaremos para desarrollar un programa de texto a voz

En resumen, se utiliza entre bastidores para procesar un montón de elementos que podrías necesitar o encontrar en tu(s) dispositivo(s), incluido el móvil.

* 1. **Funciones de Python**

El lenguaje de programación Python utiliza módulos de código que son intercambiables en lugar de una larga lista de instrucciones que era estándar para los lenguajes de programación funcional.

La implementación estándar de Python se llama «cpython«. En definitiva, no convierte su código en lenguaje de máquina o código máquina, algo que el hardware puede entender.

En realidad, lo convierte en algo llamado código de byte. Este código de bytes no puede ser entendido por la CPU. Así que necesitamos un intérprete llamado Máquina Virtual Python (PVM) que ejecuta los códigos de bytes.

El intérprete de Python realiza las siguientes tareas para ejecutar un programa:

* Paso 1: El intérprete lee un código o instrucción python. Luego verifica que la instrucción esté bien formateada, es decir, comprueba la sintaxis de cada línea. Si encuentra algún error, detiene inmediatamente la traducción y muestra un mensaje de error.
* Paso 2: Si no hay ningún error, es decir, si la instrucción o el código python está bien formateado, el intérprete lo traduce a su forma equivalente en un lenguaje intermedio llamado «código Byte». Así, después de la ejecución exitosa de la escritura o el código python, se traduce completamente en código Byte.
* Paso 3: El código del byte se envía a la Máquina Virtual Python, donde de nuevo se ejecuta el código del byte en PVM. Si se produce un error durante esta ejecución, ésta se detiene con un mensaje de error.

1. **VOZ HUMANA**
2. **Concepto**

La voz humana consiste en un [sonido rígido](https://es.wikipedia.org/w/index.php?title=Sonido_r%C3%ADgido&action=edit&redlink=1) emitido por un [ser humano](https://es.wikipedia.org/wiki/Ser_humano). Los [pulmones](https://es.wikipedia.org/wiki/Pulmones) deben producir un flujo de aire adecuado para que las dos [cuerdas vocales](https://es.wikipedia.org/wiki/Cuerdas_vocales) choquen entre sí. Las cuerdas vocales son una membrana vibradora, que realizan un ajuste fino de [tono](https://es.wikipedia.org/wiki/Tono_(m%C3%BAsica)) y [timbre](https://es.wikipedia.org/wiki/Timbre_(voz)). Los articuladores (tracto vocal) son, entre otros: la [lengua](https://es.wikipedia.org/wiki/Lengua_(anatom%C3%ADa)), el [paladar](https://es.wikipedia.org/wiki/Paladar), los [labios](https://es.wikipedia.org/wiki/Labios).

* 1. **Generalidades sobre voz humana**

La voz es un conjunto de ondas de presión acústica producidas por el tracto vocal y cuyo arreglo está gobernado por las reglas del lenguaje. El proceso de generación de voz es equivalente a un proceso de filtrado, en el cual el tracto vocal funciona como un filtro desde la apertura de las cuerdas vocales, hasta los labios. Cuando la señal de voz se propaga a través del tracto vocal, el espectro de esta señal queda determinado por la selectividad del tracto para dejar pasar determinadas frecuencias. La densidad espectral de potencia de una señal de voz, definida como la distribución de energía de la misma a través del espectro, tiende a cero en frecuencias bajas cercanas a cero y alcanza su máximo valor en algunos cientos de Hertz. La voz humana produce sonidos en un rango de 100 a 1 OOOOHz, pero prácticamente toda la información verbal está contenida en el rango de 300Hz a 6000Hz. El ancho de banda de la señal de voz para efectos prácticos puede considerarse entre 300Hz y 4000Hz. En este rango se encuentra la mayor parte del espectro de voz, por lo que puede considerarse adecuado para establecer comunicación a través de un medio electrónico; por ejemplo, el teléfono. Por su parte, el rango para entender frases y palabras (inteligibilidad) está contenido dentro de los 500Hz a 2500Hz. Para digitalizar la señal de voz, que es analógica por naturaleza, se emplea el principio de modulación por pulsos codificados (PCM). Esto se hace cumpliendo con el teorema de muestreo enunciado por Nyquist, que afirma que para recuperar con fidelidad una señal es necesario tomar muestras de ésta a una frecuencia de por lo menos el doble de su componente de mayor frecuencia. En este caso, los componentes de mayor frecuencia se encuentran alrededor de los 4000Hz, de modo que el muestreo se hace a 8KHz. Un parámetro fundamental utilizado en la caracterización de una señal de voz es su periodo de pitch, el cual se define como la duración de un periodo del segmento vocalizado. Espectralmente, está formado por la frecuencia fundamental y un conjunto de armónicos equiespaciados a la misma frecuencia. El periodo de pitch en hablantes masculinos es de 50 a 250Hz (periodo de 20 a 4ms), o de 160 a 32 muestras para una frecuencia de muestreo de 8KHz. En mujeres, el periodo de pitch se encuentra entre 120 y 500Hz (periodo de 8.3 a 2ms), ó 66 a 16 muestras a la misma frecuencia de muestreo.

* 1. **Procesamiento de voz**

El reconocimiento de voz es el proceso de convertir palabras habladas en texto. Python es compatible con muchos motores de reconocimiento de voz y API.

Una API (del inglés, application programming interface, en español, interfaz de programación de aplicaciones)​es una pieza de código que permite a diferentes aplicaciones comunicarse entre sí y compartir información y funcionalidades.

El reconocimiento automático del habla requiere como primer paso la representación de la señal de voz mediante una sucesión de vectores de parámetros acústicos, equiespaciados en el eje temporal, con la información suficiente para poder identificar los sonidos en las siguientes etapas del sistema de reconocimiento. Esta etapa suele conocerse con el nombre de parametrización. En la inmensa mayoría de sistemas de reconocimiento estos vectores de parámetros se obtienen a partir de un análisis espectral localizado de la señal de voz. Ello se debe a que tradicionalmente se ha supuesto que el oído humano es insensible a la fase de la transformada de Fourier localizada de la señal de voz [Sch75] y, por tanto, la información útil de la misma está contenida en su densidad espectral de potencia, que abreviadamente en esta memoria recibirá el nombre de espectro.

* 1. **Pasos para el procesamiento de la voz**

Grabación de voz: El primer paso es capturar la voz del usuario mediante un micrófono. Esta señal de voz se divide en pequeñas partes, llamadas "tramas" o "segmentos".

Preprocesamiento: Antes de que la señal de voz se pueda analizar, se realiza un preprocesamiento para eliminar el ruido de fondo y mejorar la calidad del audio. Esto puede incluir el filtrado de ruido y la normalización del volumen.

Extracción de características: En esta etapa, se extraen características relevantes de cada trama de voz. Estas características pueden incluir la energía de la señal, frecuencias dominantes y coeficientes cepstrales de frecuencia (MFCC, por sus siglas en inglés), entre otros.

Modelo acústico: El modelo acústico es una parte crucial del reconocimiento de voz. Se trata de un modelo estadístico que ha sido previamente entrenado utilizando grandes conjuntos de datos de voz. El modelo acústico se utiliza para determinar la probabilidad de que una secuencia de características de voz corresponda a cada posible unidad de habla (fonemas, palabras o frases).

Decodificación: En esta etapa, se utiliza un algoritmo de decodificación para buscar la secuencia de palabras o frases más probable dada la secuencia de características de voz y el modelo acústico. Esto implica evaluar diferentes combinaciones y seleccionar la más coherente en función de las probabilidades calculadas.

Post procesamiento: Después de la decodificación, se pueden aplicar técnicas de post procesamiento para mejorar aún más la precisión del reconocimiento de voz. Esto puede incluir el uso de modelos de lenguaje, corrección de errores ortográficos y mejoras en la puntuación.

Resultado final: Por último, se obtiene el resultado final del reconocimiento de voz, que suele ser un texto transcribiendo el habla del usuario.

* 1. **Librería Speech Recognition**

Las librerías o bibliotecas de programación son conjuntos de archivos de código que se utilizan para desarrollar software. Su objetivo es facilitar la programación, al proporcionar funcionalidades comunes, que ya han sido resueltas previamente por otros programadores.

Librería Speech Recognition es utilizada para realizar reconocimiento de voz, con soporte para varios motores y APIs, en línea y fuera de línea.

Este proyecto comenzó como una demostración técnica, pero ahora se ha consolidado como la herramienta más popular para la comunidad teniendo como autores iniciales;

* Uberi <me@anthonyz.ca> (Anthony Zhang)
* bobsayshilol
* arvindch <achembarpu@gmail.com> (Arvind Chembarpu)
* kevinismith <kevin\_i\_smith@yahoo.com> (Kevin Smith)
* haas85
* DelightRun <changxu.mail@gmail.com>
* maverickagm
* kamushadenes <kamushadenes@hyadesinc.com> (Kamus Hadenes)
* sbraden <braden.sarah@gmail.com> (Sarah Braden)
* tb0hdan (Bohdan Turkynewych)
* Thynix <steve@asksteved.com> (Steve Dougherty)
* beeedy <broderick.carlin@gmail.com> (Broderick Carlin)

**RESULTADOS DEL PROYECTO DE INVESTIGACIÓN**

**Creando un conversor de voz a texto - Python**

Crea un sintetizador de voz que sea capaz de realizar la transcripción de las palabras pronunciadas en el micrófono, convirtiéndolas en texto.

En Python existen muchas librerías que nos sirven para crear aplicaciones de conocimiento de voz. SpeechRecognition es la más popular;

import speech\_recognition as sr

A continuación, se crea una nueva instancia de la clase Recognizer;

r = sr.Recognizer()

Se necesita que el micrófono por defecto en nuestro sistema sea el da la fuente de datos que valla a capturar las palabras que vamos a decir;

with sr.Microphone() as source:

print("Say Something...")

audio = r.listen(source)

Imprimimos lo que captura del audio que se realiza mediante el método r.listen que esta presente en la clase Recognizer, que capturará el audio hasta que capte un silencio.

Una vez capturado el audio utilizamos el recognizer de Google para convertir el audio en texto, existen varios “recognize” que podemos usar; Google, Bing, IBM, Sphinx, etc.

try:

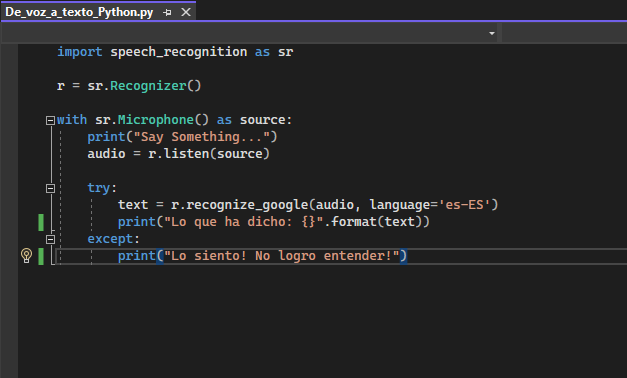
text = r.recognize\_google(audio, language='es-ES')

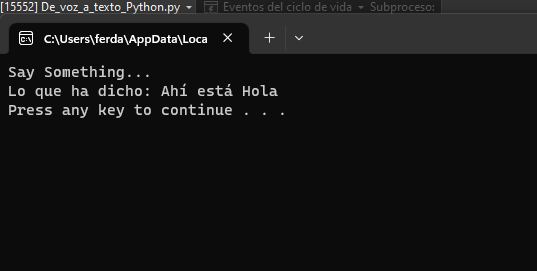
print("Lo que ha dicho: {}".format(text))

except:

print("Lo siento! No logro entender!")

Utilizamos un bloque “try” y “except” para controlar aquellos casos donde ocurra un error al entender lo que diga el usuario.

**Imágenes del código**

**Resultado**

**CONCLUSIONES**

Para concluir el proyecto, se puede comentar que el uso de la tecnología ha transformado significativamente la forma en que vivimos, trabajamos, nos comunicamos y nos relacionamos con los demás. Si bien hay preocupaciones sobre los efectos negativos, es importante reconocer los muchos beneficios que la tecnología nos ha brindado, como la eficiencia, la conectividad global y la mejora del acceso a la información. Es esencial encontrar un equilibrio entre el uso de la tecnología y la vida personal y social saludable para maximizar sus beneficios y minimizar sus riesgos potenciales. En definitiva, siendo así, Python, es un lenguaje de programación relativamente fácil de aprender, y las herramientas necesarias están disponibles para todos de forma gratuita. Esto hace que sea accesible para casi todo el mundo..

**REFERENCIAS BIBLIOGRAFICAS**

* Bill Venners January 13, 2003.

<https://www.artima.com/articles/the-making-of-python>

* Motor computacional de reconocimiento de voz.

Carlos Andrés Uribe Pérez

<https://www.artima.com/articles/the-making-of-python>

* Programación de algoritmos para análisis y reconocimiento de voz.

Paulina Rovira Sánchez /Fernando Mondragón Matus /Cuauhtémoc Gálvez Calvo/Germán Fabila García

https://repositorio.tec.mx/bitstream/handle/11285/629605/33068000968965.pdf?sequence=1&isAllowed=y

* CONVERTIR LA VOZ EN TEXTO CON PYTHON /TruzzBlogg

<https://github.com/tbcodes>

* SpeechRecognition 3.10.0

<https://pypi.org/project/SpeechRecognition/>