

# Desarrollo de un traductor de texto e imágenes (Julio de 2021)

Meyer T. Velasquez Autor

**Resumen** – Traductor de texto en idiomas como el inglés, alemán y español programado en Python versión 3.9.5, con interfaz gráfica para el ingreso de texto y subida de archivos, imágenes en este caso con tres formatos disponibles, y con la opción de audio de las traducciones en los tres idiomas que fueron mencionados anteriormente.

**Palabras clave** – Traductor, OCR, Imágenes, Idiomas

**Abstract** - Text translator in languages such as English, German and Spanish programmed in Python version 3.9.5, with a graphical interface for entering text and uploading files, images in this case with three available formats, and with the option of audio of the translations in the three languages that were mentioned above.

**Keywords** - Translator, OCR, Images, Languages

## I. INTRODUCCION

Los lenguajes son parte básica y fundamental de las culturas, donde en nuestra evolución avanza de manera vertiginosa, que el acceder a imágenes, textos audio es de vital para nuestra comunicación o acceso a información que en muchas ocasiones puede ser de utilidad sin importar la actividad, pero se encuentra en idiomas desconocidos, el acceder a imágenes que contengan texto en un idioma que no conocemos, al usar traductores online, se publican datos, descargar de software poco confiable, por eso al tener nuestro propia implementación de un traductor y poder mejorarlo usando librerías disponibles, tendremos una versión única y según nuestras necesidades, con pocas líneas de código, pero con un gran potencial desde traducir texto plano en inglés, alemán, español viceversa a los otros idiomas, con el audio de estas traducciones, se implementa un GUI sencilla para que el usuario lo use sin tener que usar una manual o que le genere dificultades, esta interfaz esta implementada en tkinter con labels y botones básicos de acceso.

Documento recibido el 30 de Julio de 2021. Este trabajo fue apoyado por Meyer Tovar Velasquez. Fundación universitaria del área andina.

## II. MANUAL DE USUARIO

### A. Arranque software

el archivo principal es main.py, donde al ejecutar nos abra la siguiente Gui, donde elegiremos una imagen, se recomienda imágenes con fondos blancos o grises, ya que no tiene implementado el reconocimiento de colores.

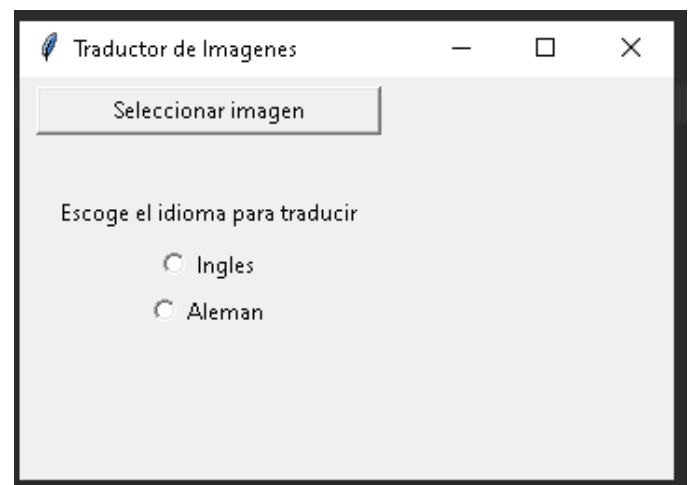


Figura 1. Interfaz inicial .

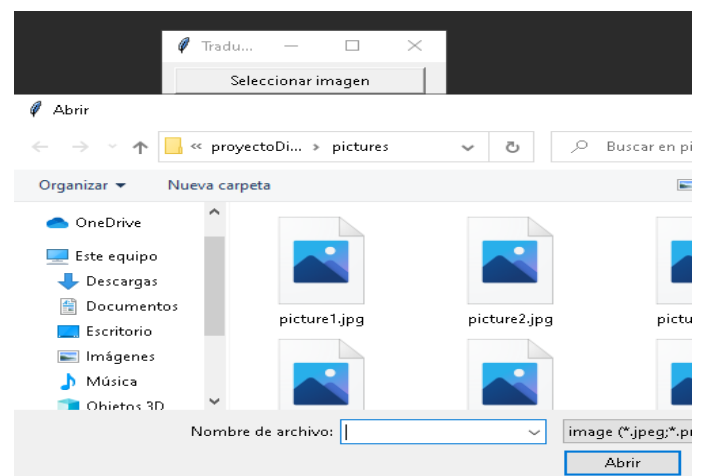


Figura 2. Cargue de imagen.

Imagen cargada que sera traducida, con la opcion de radio boton para seleccionar el idioma.



Figura 3. Imagen a traducir.

Después de elegir al idioma al que deseamos traducir nos dará una imagen de salida, con el texto traducido de la imagen de entrada.

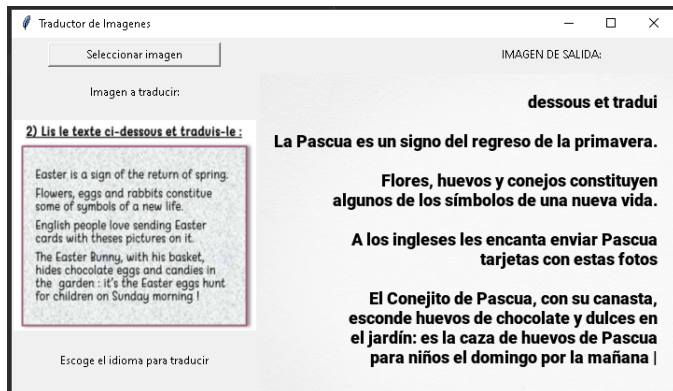


Figura 4. Imagen traducida.

Esta es la primera parte del módulo de traducción de imágenes, donde podemos implementar cualquier imagen con texto y el software extraerá el texto, que se enviará a otra imagen con el texto traducido, tener en cuenta no existe orden gramatical, se traduce línea por línea un caso similar como el traductor de Google.

4) Modulo traductor de texto en diferentes idiomas con la opción de activar el audio de texto traducido tienes los tres idiomas configurados, se encuentra tres botones, el de limpiar para ingresar de nuevo texto el limpia las dos cajas de texto y

el de traducir eligiendo un idioma inicial y luego una lista con un idioma de destino o el del texto traducido.

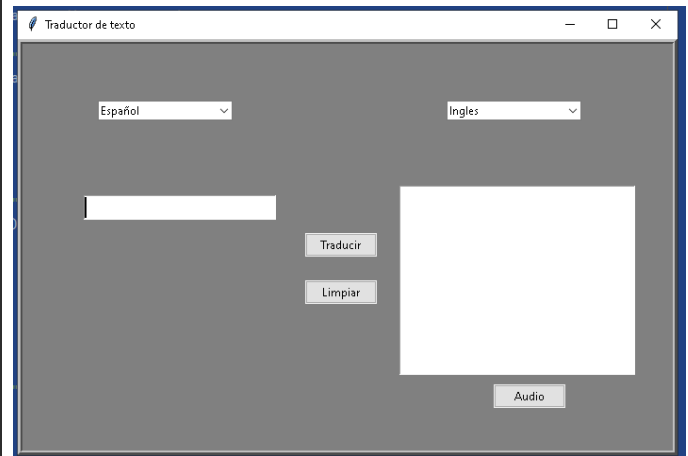


Figura 4. Modulo para traducir texto, donde tenemos dos listas para traducir desde un idioma o viceversa.



Figura 5. Ingresamos el texto que se desea traducir y en la siguiente caja de texto el texto traducido en el idioma seleccionado.

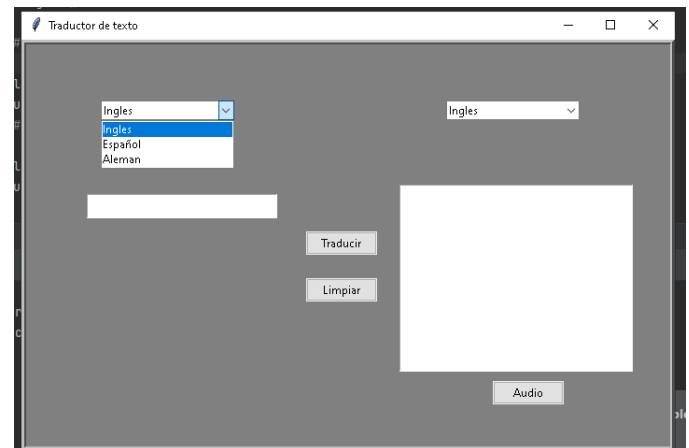


Figura 6. Listas desplegables de idioma de entrada y de salida.

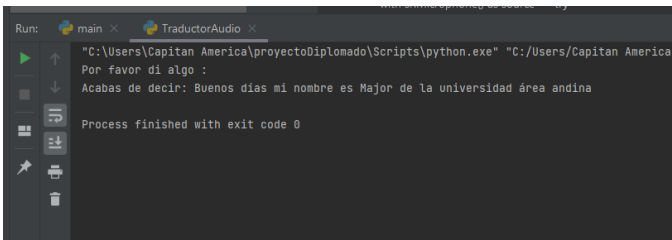


Figura 7. Traductor de voz a texto, hablamos es pañol, el software captura el audio y lo presenta en la consola.

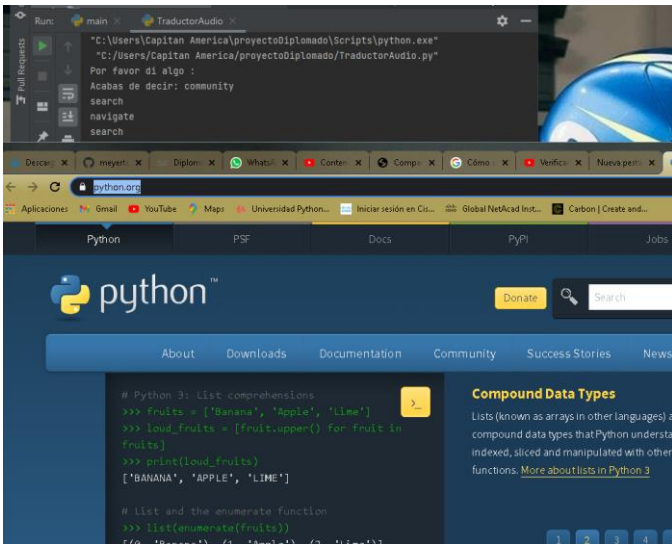


Figura 8. Traducto de voz a texto, si se cumplen los dos if, al pronunciar la palabra community y navigate, nos abre a pagina de python.

### III. MANUAL DEL DESARROLLADOR

Importación de paquetes necesarios para el traductor de imágenes.

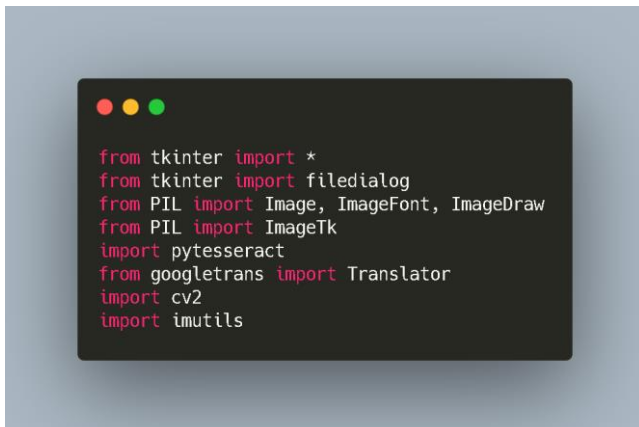


Figura 9. Importación de módulos y paquetes.

Función para selecciona el idioma



```
# function to choose the language to
translate the image
def english_idiom():
    global image, img_text

    if selected.get() == 1:
        # tesseract must be installed in
        windows and indicate the path so that it
        does not generate errors

pytesseract.pytesseract.tesseract_cmd =
r'C:\Program Files\Tesseract-
OCR\tesseract.exe'
img_text =
pytesseract.image_to_string(image)
# Image converted to text and
translated into Spanish
p = Translator()
p_translated =
p.translate(img_text, dest='spanish')
translated =
str(p_translated.text)
# translated text font type
settings, and on which image we will send
the text.
imas =
Image.open("pictures/plantillaTraductor.j
pg")
draw = ImageDraw.Draw(imas)
font =
ImageFont.truetype(r'C:\fonts\Roboto-
Black.ttf', 18)
text = translated
draw.text((15, 20), text,
fill="black", font=font, align="right")
# image with the translated text
that we will use late
imas.save("test.png")
src = cv2.imread("test.png")
# we will display the image in
the GUI in the lblOutputImage
imageToShowOutput =
cv2.cvtColor(src, cv2.COLOR_BGR2RGB)
im =
Image.fromarray(imageToShowOutput)
img =
ImageTk.PhotoImage(image=im)

lblOutputImage.configure(image=img)
lblOutputImage.image = img

# output image label
lblInfo3 = Label(root,
text="IMAGEN DE SALIDA:")
lblInfo3.grid(column=1, row=0,
padx=5, pady=5)

with open('traduccion.txt', mode='w')
as file:
```

```

        file.write(img_text + "\n\n\n" +
translated)
        print("hecho!")

# function to choose the language
def selected_image():
    # specify file type for images
    path_image =
filedialog.askopenfilename(filetypes=[
    ("image", ".jpeg"),
    ("image", ".png"),
    ("image", ".jpg")])
    # if there is a path we declare image
    globally
    if len(path_image) > 0:
        global image
        # we read the image and resize it
        image = cv2.imread(path_image)
        image = imutils.resize(image,
height=340)
        # display image and input in GUI
        imageToShow =
imutils.resize(image, width=320)
        im = Image.fromarray(imageToShow)
        img =
ImageTk.PhotoImage(image=im)

lblInputImage.configure(image=img)
        lblInputImage.image = img

        # Label image to translate- image
entrace
        lblInfo1 = Label(root,
text="Imagen a traducir:")
        lblInfo1.grid(column=0, row=1,
padx=5, pady=5)
        # we clean the radio button
selection
        lblOutputImage.image = ""
        selected.set(0)

image = None

# modal principal
root = Tk()
root.title("Traductor de Imagenes")
# Label where the input image will be
presented
lblInputImage = Label(root)
lblInputImage.grid(column=0, row=2)

# Label where the output image will be
presented
lblOutputImage = Label(root)
lblOutputImage.grid(column=1, row=1,
rowspan=6)

lblInfo2 = Label(root, text="Escoge el
idioma para traducir", width=25)

```

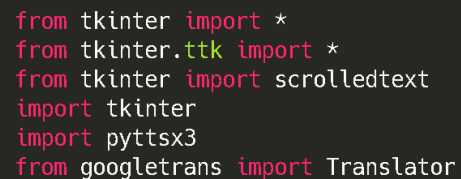
```

lblInfo2.grid(column=0, row=3, padx=5,
pady=5)
# selected allows us to handle integers
for radio button
selected = IntVar()
# radio button where we pass the modal
root, and call the function of
english_idiom
radio1 = Radiobutton(root, text="Ingles",
width=25, value=1, variable=selected,
command=english_idiom)
radio2 = Radiobutton(root, text="Aleman",
width=25, value=2, variable=selected,
command=english_idiom)
radio1.grid(column=0, row=4)
radio2.grid(column=0, row=5)
# button selected input image
btn = Button(root, text="Seleccionar
imagen", width=25,
command=selected_image)
btn.grid(column=0, row=0, padx=5, pady=5)
# an endless loop is created where we
have already declared root, until the
user closes the window
root.mainloop()

```

Se debe tener en cuenta para hacer uso de la librería pytesseract se debe establecer la ruta del ejecutable, ya que se presentan errores y se soluciona.

#### IV. CÓDIGO TRADUCTOR DE TEXTO



```

from tkinter import *
from tkinter.ttk import *
from tkinter import scrolledtext
import tkinter
import pyttsx3
from googletrans import Translator

```

Figura 10. Importación de paquetes y módulos.

```

# function to delete the text
def eraser_text():
    # delete characters from the text box
    boxText.delete("0", END)
    boxText2.delete("0", END)

# function to translate the entered text

```

```

def translate_text():
    inn = selected.get()
    if inn != "":
        if (selectOne.get() == "Español"
and (selectTwo.get()) == "Ingles":
            out =
Translator().translate(inn, dest="en")
            salida.set(out.text)
        elif (selectOne.get() == "Ingles"
and (selectTwo.get()) == "Español":
            out =
Translator().translate(inn, dest="es")
            salida.set(out.text)
        elif (selectOne.get() ==
"Español" and (selectTwo.get()) ==
"Aleman":
            out =
Translator().translate(inn, dest="de")
            salida.set(out.text)
        elif (selectOne.get() == "Aleman"
and (selectTwo.get()) == "Español":
            out =
Translator().translate(inn, dest="es")
            salida.set(out.text)
        elif (selectOne.get() == "Aleman"
and (selectTwo.get()) == "Ingles":
            out =
Translator().translate(inn, dest="en")
            salida.set(out.text)
        elif (selectOne.get() == "Ingles"
and (selectTwo.get()) == "Aleman":
            out =
Translator().translate(inn, dest="de")
            salida.set(out.text)

# function to give audio to the
translated text
def talk():
    # the button invokes a pytttsx3
instance
    voiceEngine = pytttsx3.init()
    # instance to the speech engine
    queues a command for the engine that is
rate the speech rate
    voiceEngine.setProperty("rate", 85)
    voiceEngine.say(boxText2.get())
    voiceEngine.runAndWait()

# modal or translator window
root = Tk()
root.title("Traductor de texto")
root.geometry("700x440")
root.config(relief="ridge", bd="5",
bg="gray")

# we declare variables of type String to
store the data of the boxes
selected = StringVar()
salida = StringVar()

```

```

# combobox # 1 we design the select and
the style
selectOne = Combobox(root,
values=["Ingles", "Español", "Aleman"])
selectOne.place(x=80, y=60)
selectOne.current(1)
# combobox # 2 we design the select and
the style
selectTwo = Combobox(root,
values=["Ingles", "Español", "Aleman"])
selectTwo.place(x=450, y=60)
selectTwo.current(0)
# text box where the information to be
translated will be entered
boxText =
tkinter.Entry(justify=tkinter.LEFT,
textvariable=selected, font=("Times New
Roman", 15))
boxText.grid(column=0, pady=160, padx=65)
# placing cursor in text area
boxText.focus()
# box tex # 1 text box where the
translated information will be sent
boxText2 =
tkinter.Entry(justify=tkinter.LEFT,
textvariable=salida, font=("Times New
Roman", 15))
boxText2.place(x=400, y=150, height=200,
width=250)

btn = Button(root, text="Limpiar",
command=eraser_text).place(x=300, y=250)
btnTrad = Button(root, text="Traducir",
command=translate_text).place(x=300,
y=200)

btnVoi = Button(root, text="Audio",
command=talk)
btnVoi.place(x=500, y=360)

# method in main window that repeats
infinitely until user exits or stops
root.mainloop()

```

## V. TRADUCTOR AUDIO

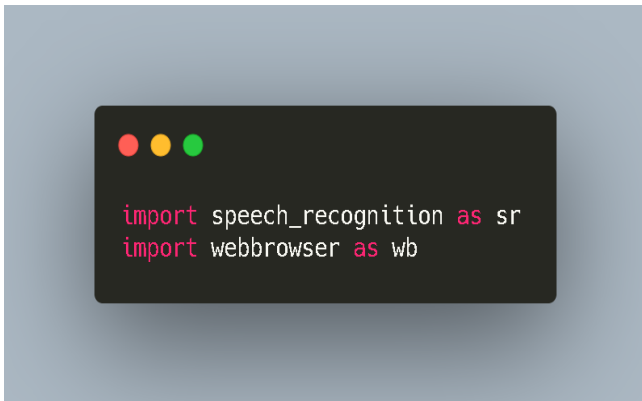


Figura 11. Módulos y paquetes importados

```
# creamos una instancia de la clase
Recognizer
r = sr.Recognizer()
r1 = sr.Recognizer()
r2 = sr.Recognizer()

# usamos la clase Microphone para
capturar el audio
with sr.Microphone() as source:
    print('Por favor di algo : ')
    # el audio sera capturado con el
    metodo listen
    audio = r2.listen(source)
    try:
        # usamos Recognizer de google
        para convertir el audio en texto
        text = r2.recognize_google(audio,
        language='es-ES')
        print('Acabas de decir:
        {}'.format(text))
    except sr.UnknownValueError:
        print('Disculpa no te escucho')
    except sr.RequestError as e:
        print("Fallo".format(e))

# creamos una condicion par abrir una url
en el navegador usando dos palabras
if "community" in
r1.recognize_google(audio):
    r1 = sr.Recognizer()
    url = 'https://www.python.org/'
    with sr.Microphone() as source:
        print('search')
        audio = r1.listen(source)

    try:
        get =
r1.recognize_google(audio)
        print(get)
        wb.get().open_new(url)
    except sr.UnknownValueError:
        print("error")
    except sr.RequestError as e:
        print("Fallo".format(e))
```

```
if "navigate" in
r.recognize_google(audio):
    r = sr.Recognizer()
    url = 'https://www.python.org/'
    with sr.Microphone() as source:
        print('search')
        audio = r.listen(source)

    try:
        get =
r.recognize_google(audio)
        print(get)
        wb.get().open_new(url)
    except sr.UnknownValueError:
        print("error")
    except sr.RequestError as e:
        print("Fallo".format(e))
```

## VI. CONCLUSIÓN

Se puso en practica las lecciones que nos fueron instruidas y el estudio de los módulos, se llega a implementar un traductor con diferentes opciones, en tiempo muy corto, código fácil de implementar, así como el uso de las librerías usadas , se pueden usar GUI de tkinter que no requieren mucho conocimiento para llegar a un resultado exitoso, como se puede llegar a observar el los módulos de imágenes y texto, en futuras versiones realizar una interfaz mas amigable para el usuario e integrada, así como el uso de dispositivos electrónicos , como ESP32 y pantalla tdt para enviar texto traducido.