



ACTIVIDAD DE APRENDIZAJE TRADUCTOR TKINTER.

La presente actividad tiene como objetivo realizar las siguientes acciones para el código denominado Traductor con Tkinter y Python. A continuación, se listan los puntos que se deben realizar y que deben ser entregados totalmente funcional en una carpeta llamada Traductor_Tkinter con los el cual debe ser montados en la plataforma Territorium.

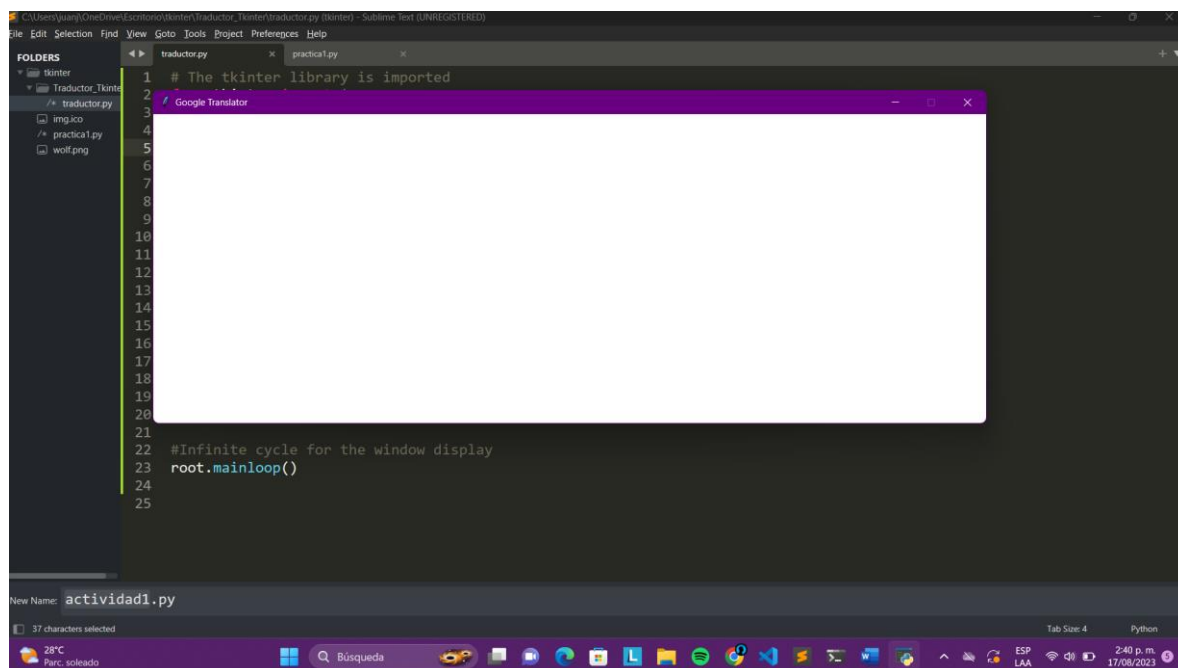
1. Crear una carpeta con el nombre Traductor_Tkinter.
2. Se debe instalar la librería googletrans (API de Google translate), para ello abra la consola y escriba el siguiente comando: `pip install googletrans`.
3. Abrir el editor de Código de preferencia y crear un archivo con el nombre traductor.py y guardarlo dentro de la carpeta que fue creada en el punto 1.
4. En el archivo creado en el punto 2 escribir el siguiente código:

```
1 from tkinter import *
2 from tkinter import ttk, messagebox
3 import googletrans
4 from googletrans import Translator
5
6 root = Tk()
7 root.title("Google Translator")
8 root.geometry("1080x400")
9 root.resizable(False, False)
10
11 root.configure(bg="white")
12 root.mainloop()
```

Nota: En este código estamos importando el googletrans para el manejo de la Api. También tkinter para el manejo de la ventana y el messagebox para el manejo de ventanas emergentes.



5. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:



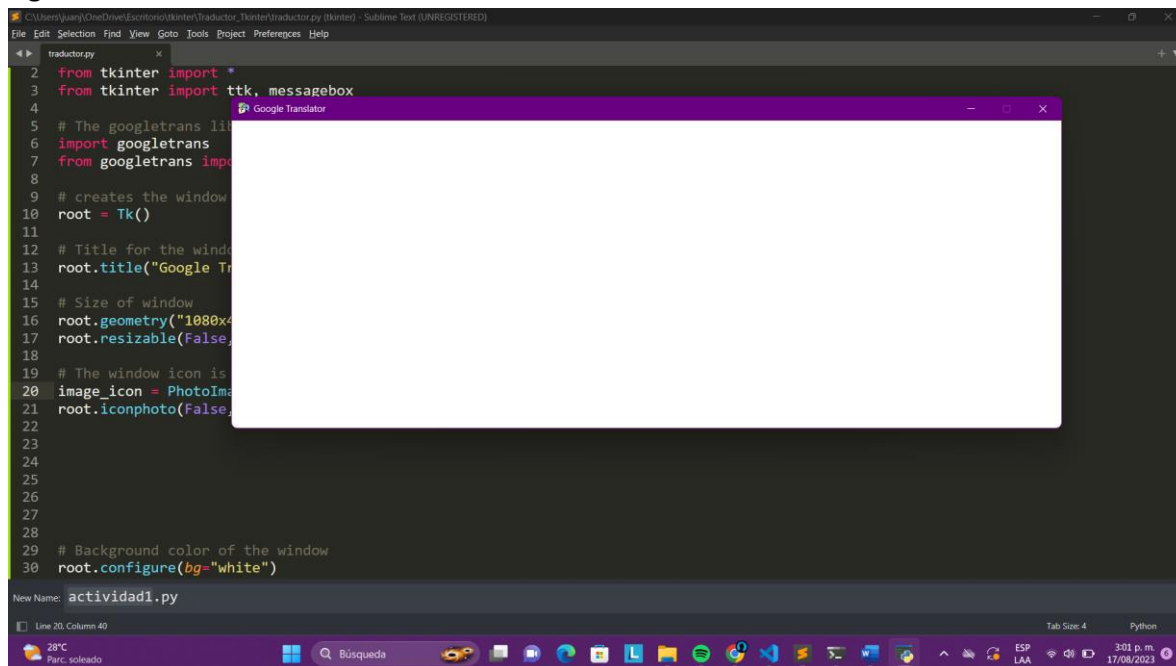
6. Ahora, después de la línea donde tenemos el `root.resizable`, vamos a agregar el icono de traductor a la ventana (se debe descargar un icono de la página de preferencia y lo debe guardar en la misma carpeta donde tiene el archivo `.py`. Este icono se va a ubicar en la parte superior de la venta al lado del título de la ventana, seleccione un icono que represente la funcionalidad de traducción de la aplicación.) y debe escribir el siguiente código:

```
9 #agregamos un icono
10 image_icon = PhotoImage(file="traductor.png")
11 root.iconphoto(False, image_icon)
```

Nota: traductor.png es el nombre del icono que fue descargado, si tiene otro nombre debe realizar el ajuste.



7. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:

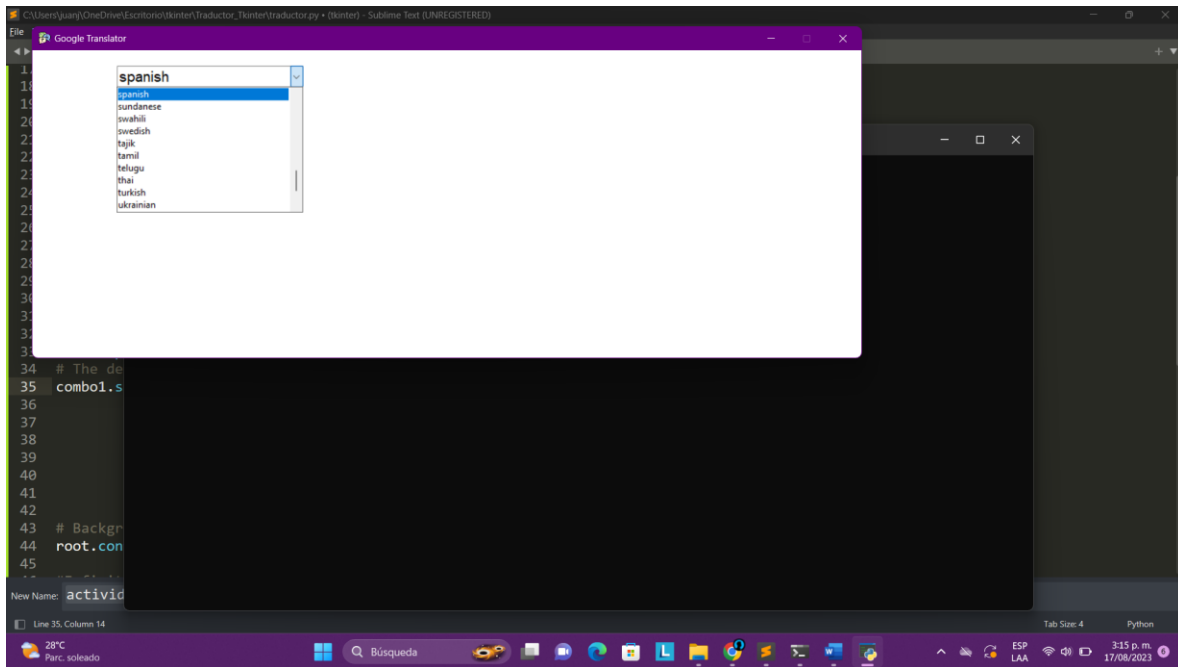


8. Ahora vamos a crear el primer combobox para cargar los idiomas de origen, escribimos el siguiente código (nos ubicamos después de `root.iconphoto`):

```
12
13 #se instancia language con los lenguajes que tiene la API de googletrans
14 language = googletrans.LANGUAGES
15 #languageV tiene el diccionario de elementos de language
16 languageV = list(language.values())
17 #se accede a los keys del diccionario de los idiomas
18 lang1 = language.keys()
19
20 #primer combo de la aplicacion
21 combo1 = ttk.Combobox(root, values=languageV, font="Roboto 14", state="r")
22 #ubicacion del combo
23 combo1.place(x=110, y=20)
24 #en el combo se muestra spanish
25 combo1.set("Spanish")
```



9. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer lasiguiente ventana:

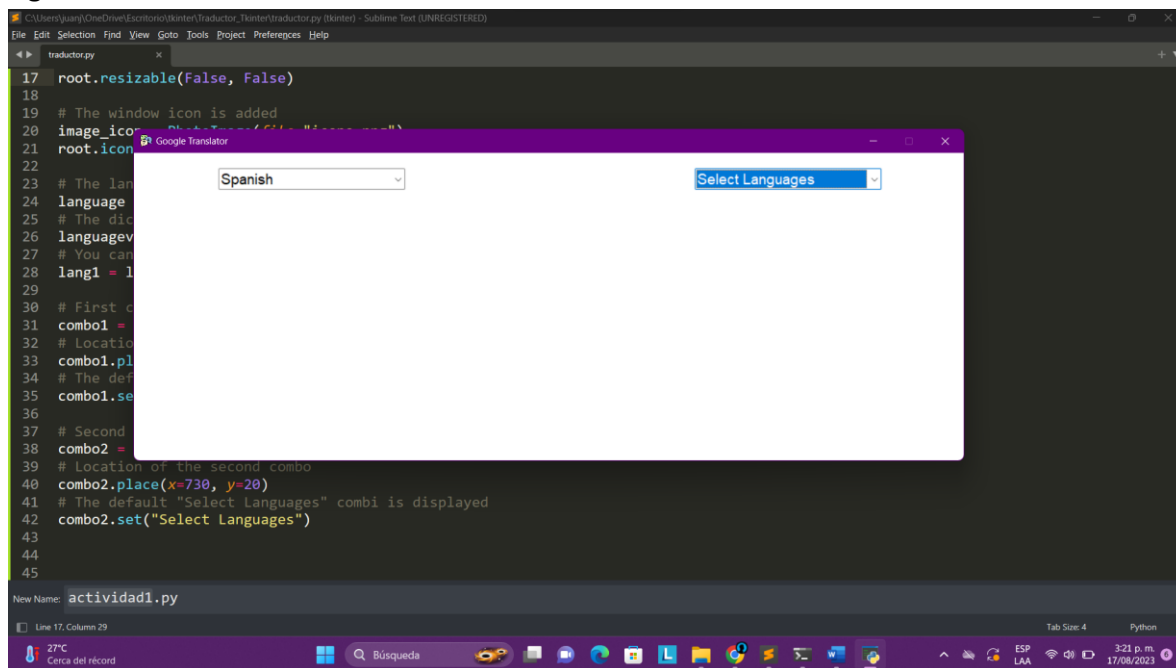


10. Ahora vamos a crear el segundo combobox para cargar los idiomas de destino, escribimos el siguiente código (nos ubicamos después de `combo1.set("Spanish")`):

```
26
27 #segundo combo de la aplicacion
28 combo2 = ttk.Combobox(root, values=languageV, font="Roboto 14", state="r")
29 #ubicacion del combo
30 combo2.place(x=730, y=20)
31 #en el combo se muestra spanish
32 combo2.set("Select Language")
```



11. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:



12. Vamos a crear el primer label para mostrar el texto Spanish, escribimos el siguiente código (nos ubicamos después de `combo1.set("Spanish")`):

```
27 #creamos el label para el idioma español
28 Label1 = Label(root, text="Spanish", font="segoe 30 bold", bg="white", width=18, bd=5, relief="groove")
29 Label1.place(x=10, y=50)
```

Nota: place significa la ubicación del label en el eje x y y.

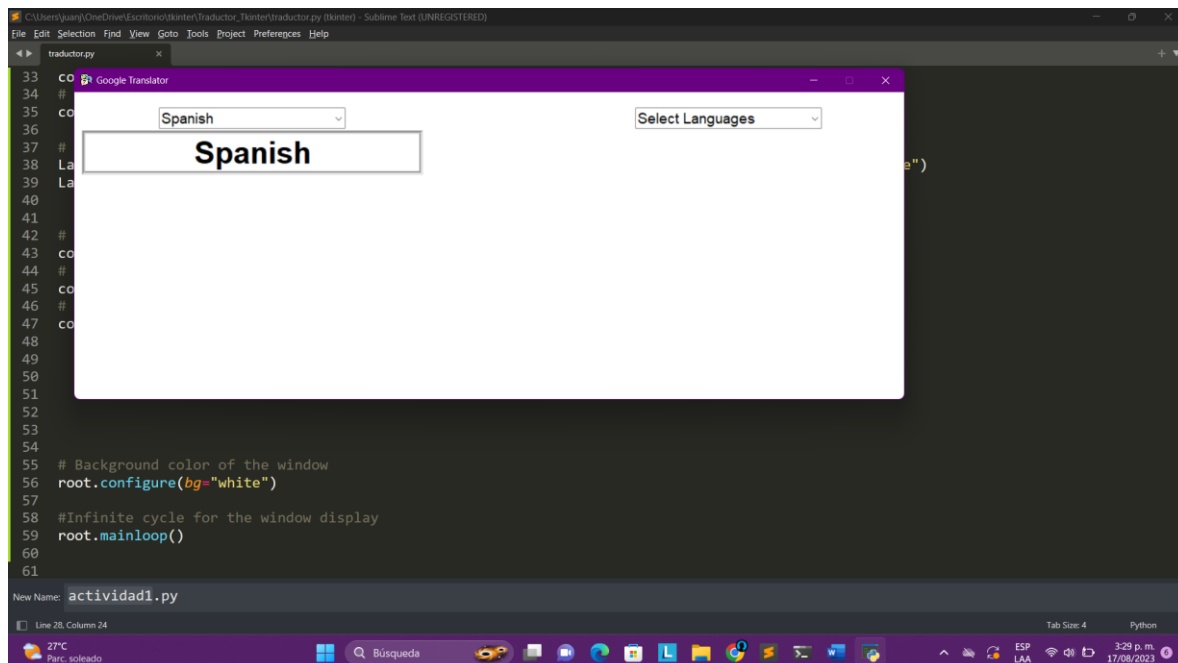
Nota: relief es un efecto del label.

Nota: root es donde se agrega el label.

Nota: bg es el color de fondo.



13. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:



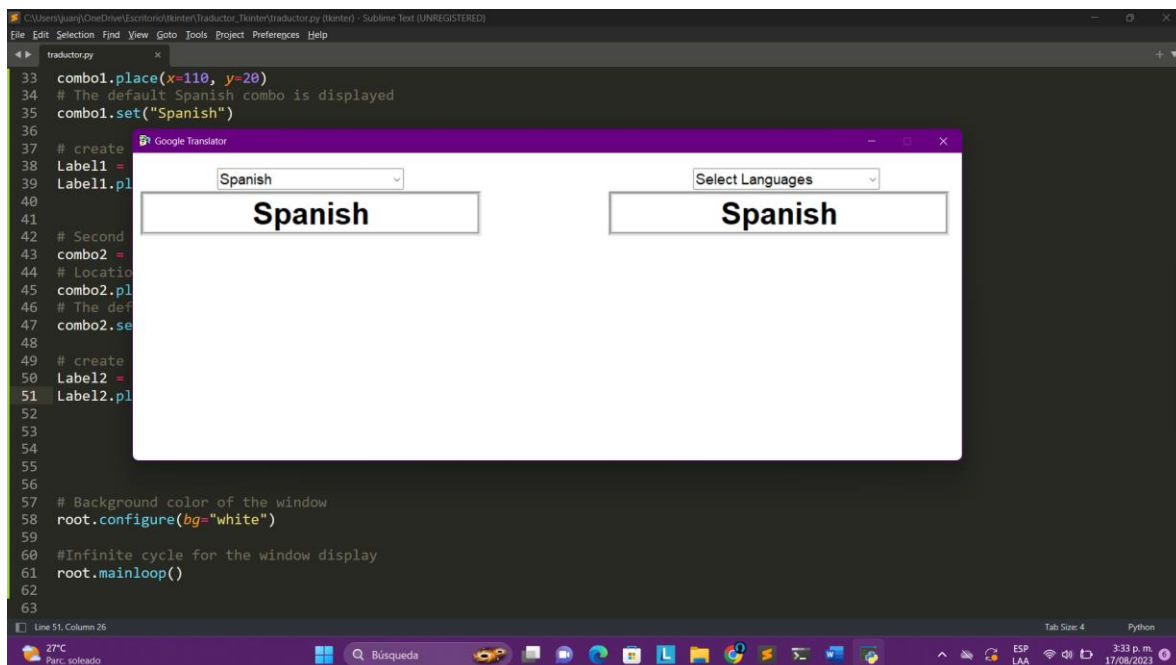
14. Ahora Vamos a crear el segundo label para mostrar el texto Spanish, escribimos el siguiente código (nos ubicamos después de `combo2.set("Select Language")`):

```
37
38 #creamos el label para el idioma español
39 Label2 = Label(root, text="Spanish", font="segoe 30 bold", bg="white", width=18, bd=5, relief="groove")
40 Label2.place(x=620, y=50)
```

3



15. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer lasiguiente ventana:

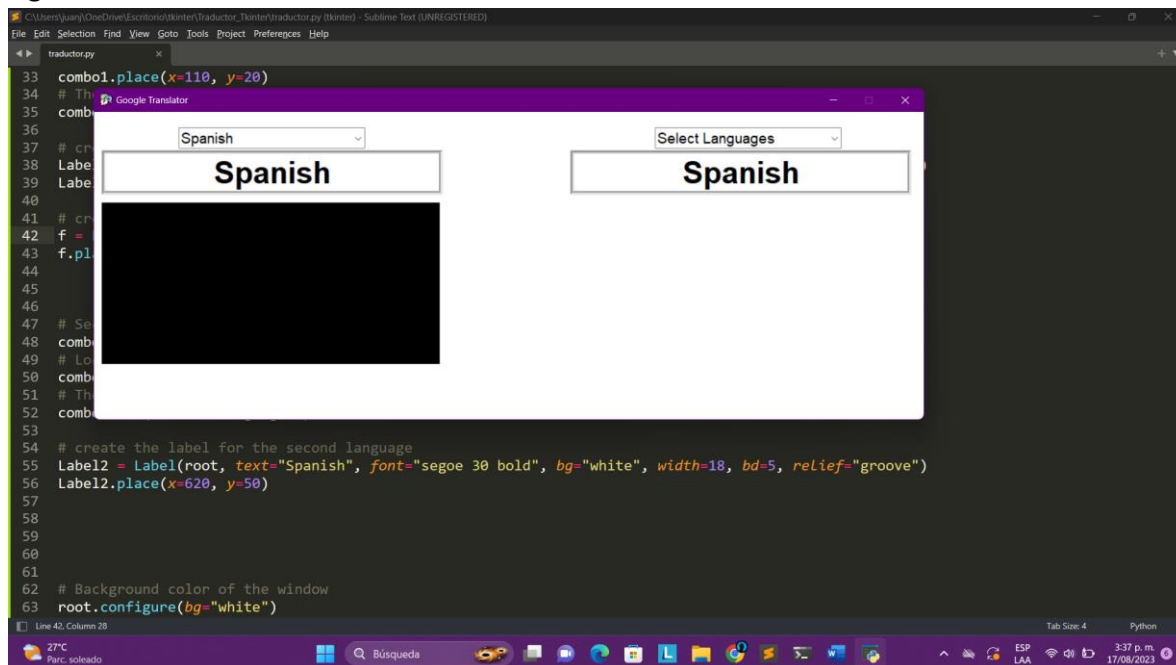


16. Vamos a crear el primer frame para mostrar el cajón donde se escribir el texto a traducir, escribimos el siguiente código (nos ubicamos después de Label1.place(x=10, y=50)):

```
30
31 #creamos el cajon para el contenido del texto a ingresar
32 f = Frame(root, bg="Black", bd=5)
33 f.place(x=10, y=118, width=440, height=210)
```



17. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:



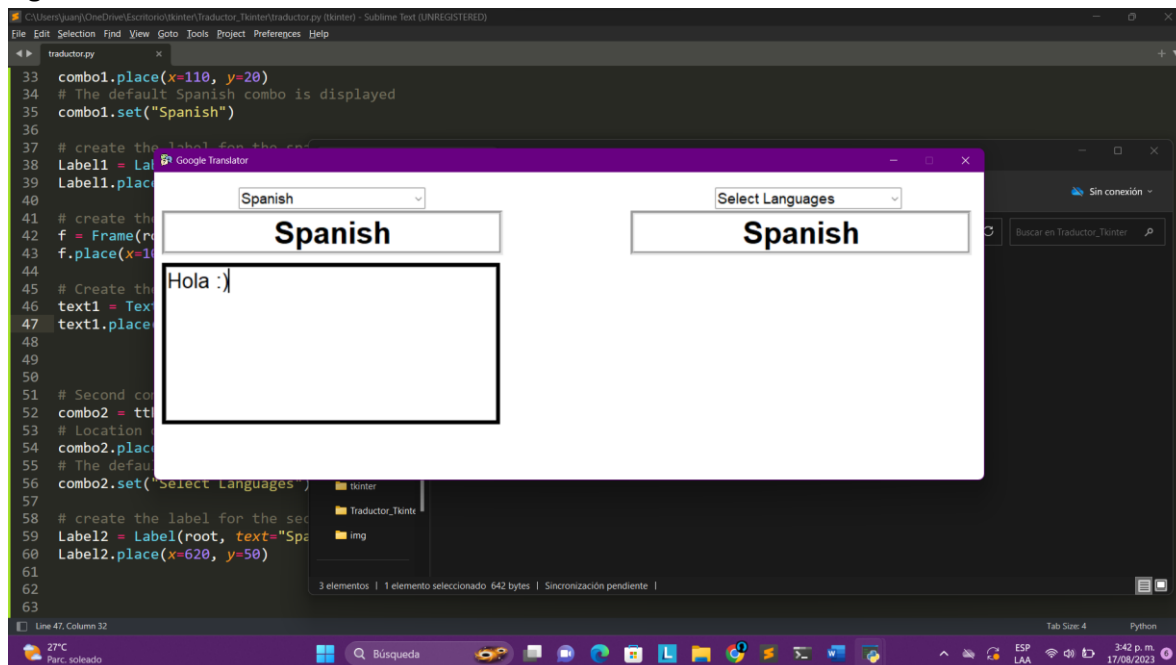
18. Vamos a crear el primer text para permitir la escritura del texto a traducir, escribimos el siguiente código (nos ubicamos después de `f.place(x=10, y=118, width=440, height=210)`):

```
34
35 #creamos la opcion para permitir el ingreso de texto en el frame
36 text1 = Text(f, font="Robote 20", bg="white", relief="groove", wrap=WORD)
37 text1.place(x=0, y=0, width=430, height=200)
```

Nota: wrap permite el manejo de la cadena de texto que se ingresa



19. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:



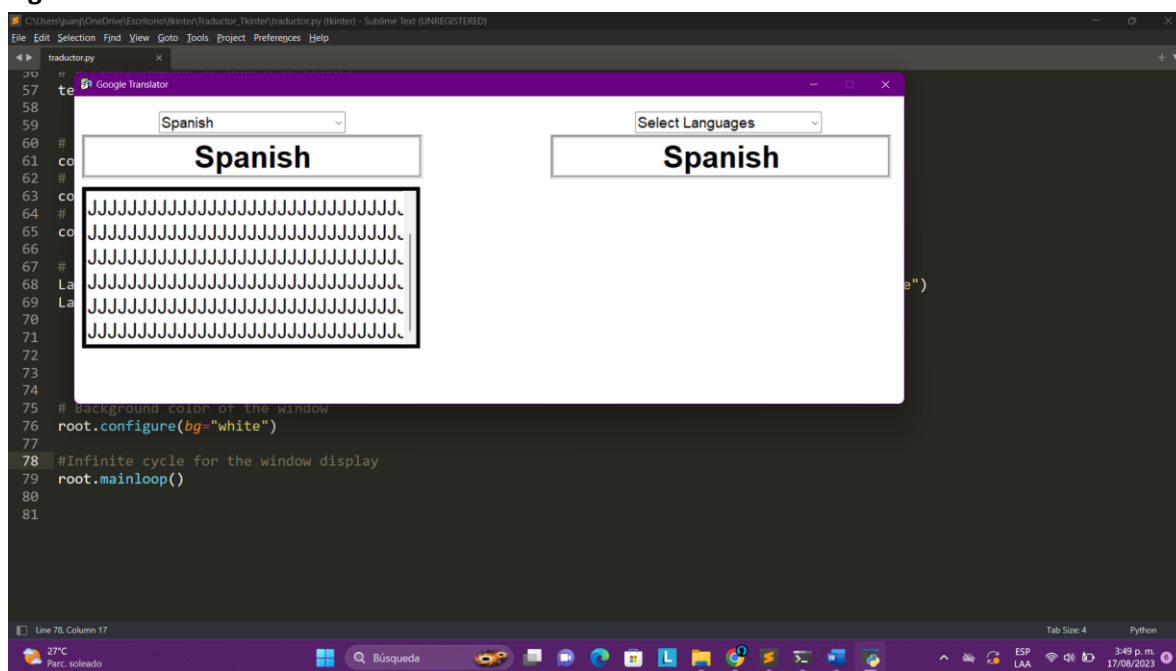
Nota: Verificar que se pueda escribir en el cajón de texto.



20. Vamos a crear un scroll para el primer text para permitir la escritura dentro de todo el cajón, escribimos el siguiente código (nos ubicamos después de `text1.place(x=0, y=0, width=430, height=200)`):

```
38
39 scrollbar1 = Scrollbar(f)
40 #empaquetado vertical
41 scrollbar1.pack(side="right", fill="y")
42 #yview es orientacion vertical
43 scrollbar1.configure(command=text1.yview)
44 #yscrollcommand es el scroll vertical
45 text1.configure(yscrollcommand=scrollbar1.set)
```

21. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:

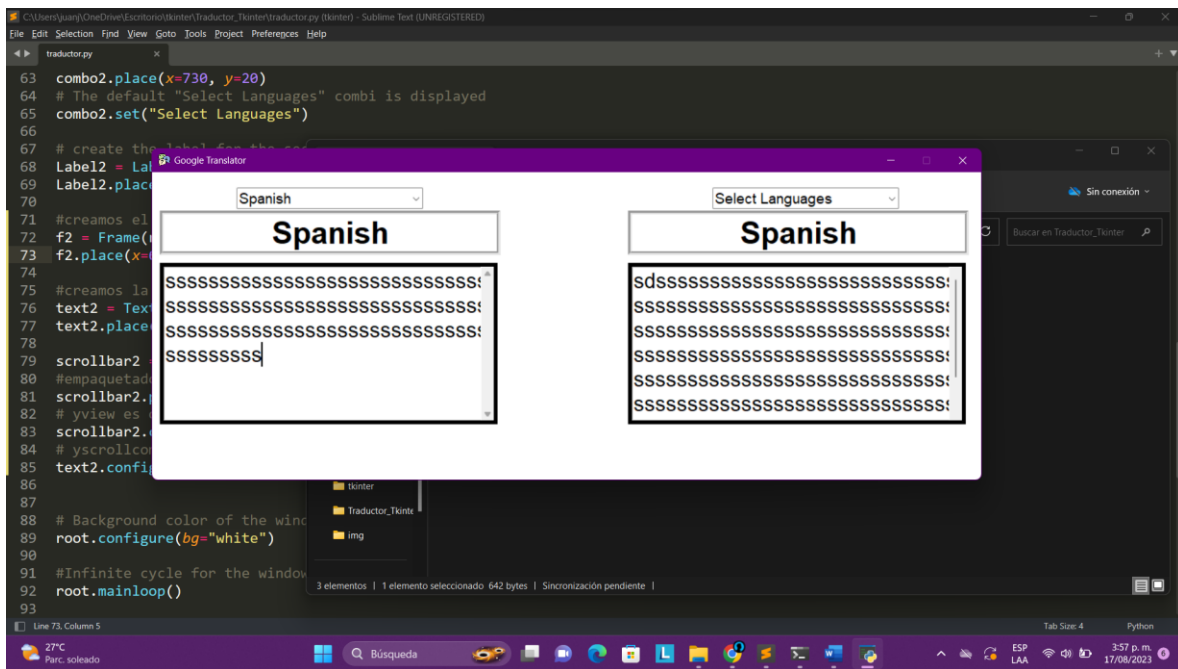


22. Vamos a crear el frame, el text y el scroll para el segundo text para permitir la escritura dentro de todo el cajón, escribimos el siguiente código (nos ubicamos después de `Label2.place(x=620, y=50)`):



```
57
58 #creamos el cajon para el contenido del texto a ingresar
59 f2 = Frame(root, bg="Black", bd=5)
60 f2.place(x=620, y=118, width=440, height=210)
61
62 #creamos la opcion para permitir el ingreso de texto en el frame
63 text2 = Text(f2, font="Roboto 20", bg="white", relief="groove", wrap=WORD)
64 text2.place(x=0, y=0, width=430, height=200)
65
66 scrollbar2 = Scrollbar(f2)
67 #empaquetado vertical
68 scrollbar2.pack(side="right", fill="y")
69 #yview es orientacion vertical
70 scrollbar2.configure(command=text2.yview)
71 #yscrollcommand es el scroll vertical
72 text2.configure(yscrollcommand=scrollbar2.set)
```

23. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:



24. Vamos a crear el label con la imagen del icono de intercambiar (se debe descargar un icono de la página de preferencia y lo debe guardar en la misma carpeta donde tiene el archivo .py. Este icono se va a ubicar en medio de los dos cajones donde se escribe el texto, seleccione un icono que represente la traducción del idioma), escribimos el siguiente código (nos ubicamos después de root.iconphoto(False, image_icon)):

```
12
13 arrow_image = PhotoImage(file="intercambiar.png")
14 image_label = Label(root, image=arrow_image, width=150)
15 image_label.place(x=460, y=50)
```



25. Vamos a crear un método para que al momento de seleccionar el idioma se carguen en los label el idioma seleccionado, escribimos el siguiente código (nos ubicamos después de `root.geometry("1080x400")`):

```
9
10 #metodo que permite cambiar el label de acuerdo a la seleccion del combobox
11 def label_change():
12     #se obtienen los datos del combobox
13     c1 = combo1.get()
14     c2 = combo2.get()
15     #se carga en el label el elemento seleccionado del combobox
16     Label1.configure(text=c1)
17     Label2.configure(text=c2)
18     #ejecuta el metodo label_change pasados 1 segundo (carga en el label el dato luego de 1 segundo)
19     root.after(1000, label_change)
```

26. Vamos a invocar el método construido en el paso anterior, escribimos el siguiente código (nos ubicamos después de `text2.configure(yscrollcommand=scrollbar2.set)`):

```
89
90 label_change()
```

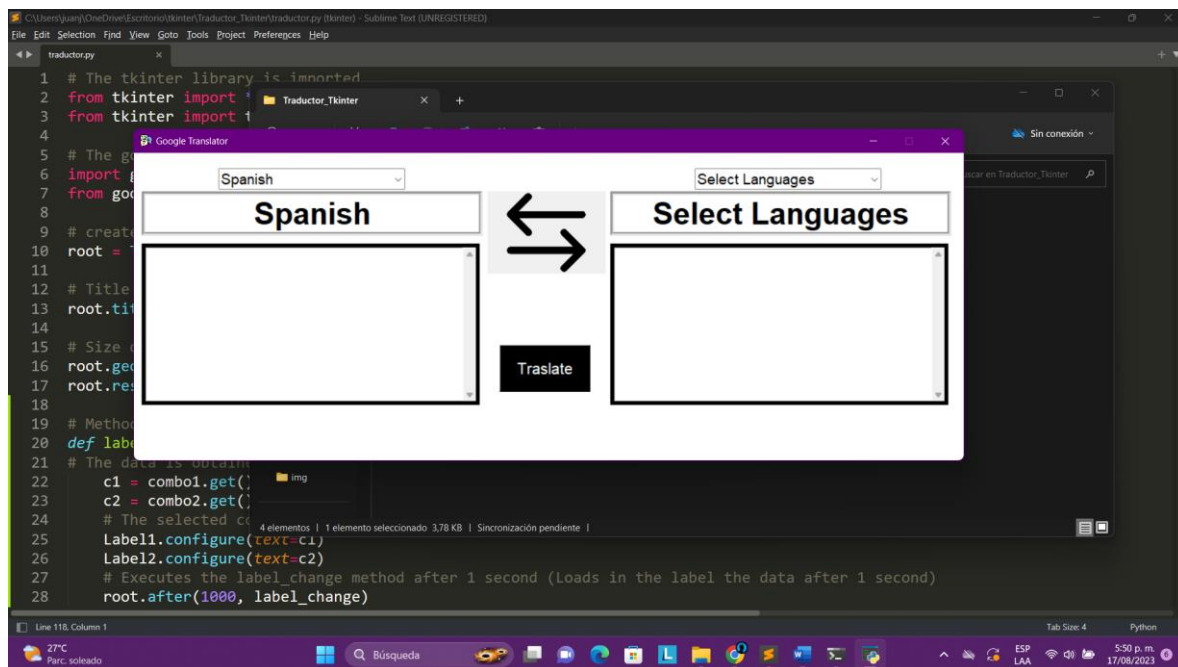
27. Vamos a crear el botón y que al dar clic sobre el cambie el tono e invoque el método `traslate_now`, escribimos el siguiente código (nos ubicamos después de `text2.configure(yscrollcommand=scrollbar2.set)`):

```
99 translate = Button(root, text="Traslate", font=("Roboto", 15), activebackground="white", cursor="hand2", bd=1, width=10, height=2, bg="black", fg="white", command=traslate_now)
100 translate.place(x=476, y=250)
```

Nota: En la imagen se visualizan dos líneas de código, pero ud. debe hacerla en una sola línea de código, la captura quedó de esta forma por temas de espacio.



28. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:

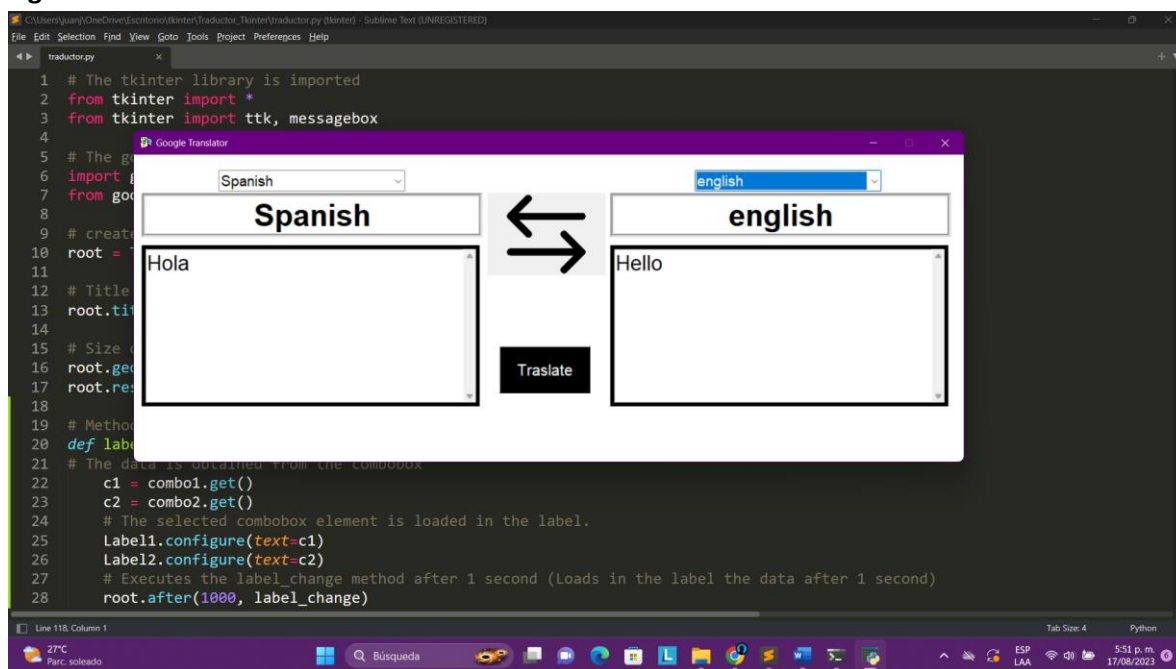


29. Vamos a crear el método para que traduzca el texto, escribimos el siguiente código (nos ubicamos después de `root.after(1000, label_change)`):

```
22 def traslate_now():
23     text = text1.get(1.0, END)
24     t1 = Translator()
25     #src es el origen del idioma donde debe tomar el texto
26     #dest es la opcion del idioma al cual se debe traducir
27     trans_text = t1.translate(text, src=combo1.get(), dest=combo2.get())
28     #se instancia el texto
29     trans_text = trans_text.text
30     #se borra el contenido que pueda tener el cajon de texto
31     text2.delete(1.0, END)
32     #se inserta la palabra traducida
33     text2.insert(END, trans_text)
```



30. Ejecutar el archivo y realizar una captura de pantalla completa (se debe visualizar fecha y hora del sistema) al resultado (lo debe anexar a continuación) y nos debe aparecer la siguiente ventana:



Notas Finales:

Si se presenta error en la línea `trans_text = t1.translate(text_src=combo1.get(), dest=combo2.get())`

- Se debe abrir el cmd y ejecutar el siguiente comando: `pip install googletrans==4.0.0-rc1`
- con el comando: `pip install googletrans==3.1.0a0`

Actividades:

- Debe generar un documento de Word con las capturas de pantalla solicitadas.
- El código debe documentarlo en inglés. Si en el código se encuentra información en español debe realizar la traducción respectiva.
- La evidencia para entregar consta de la carpeta con el archivo .py y las imágenes descargadas que fueron incluidas en el código. También debe incluir el documento con las capturas de pantalla de la aplicación.
- La actividad se debe montar en la plataforma Territorium en el espacio denominado Evidencia Actividad de Aprendizaje Traductor Tkinter.

7



- El espacio de la evidencia estará habilitado el día 17/08/2023 hasta las 19:00 horas.