

Tarea 10_Ejercicio

Axel Daniel, Ramirez Urbina, 202010083

Escuela de Mecánica Eléctrica, Facultad de Ingeniería,
Universidad de San Carlos de Guatemala

A. Resumen

La calculadora en Python es una herramienta de conversión de unidades que ofrece una interfaz gráfica amigable. Permite a los usuarios convertir entre metros (m), centímetros (cm), kilogramos (kg), gramos (g) y libras (lb). La aplicación es intuitiva, con campos de entrada para ingresar valores numéricos y selección de unidades de entrada y salida a través de menús desplegables. Al presionar el botón "Convertir", la aplicación realiza la conversión y muestra el resultado en pantalla. Esto facilita el trabajo con diferentes unidades de medida de forma rápida y eficiente. Con su diseño simple y funcionalidad útil, la calculadora de conversiones es una herramienta valiosa para realizar tareas cotidianas y académicas que implican manipulación de unidades de medida.

B. Código

```
1 import tkinter as tk
2 from tkinter import ttk
3 from tkinter import messagebox
4
5 # Función para realizar la conversión de unidades
6 def convertir():
7     try:
8         valor = float(entry_valor.get())
9         unidad_entrada = combo_unidad_entrada.get()
10        unidad_salida = combo_unidad_salida.get()
11
12        # Lógica de conversión
13        resultado = valor # Por defecto, si las unidades son las mismas, el resultado es el mismo valor
14        if unidad_entrada == unidad_salida:
15            resultado = valor
16        elif unidad_entrada == "m" and unidad_salida == "cm":
17            resultado = valor * 100
18        elif unidad_entrada == "cm" and unidad_salida == "m":
19            resultado = valor / 100
20
21        elif unidad_entrada == "kg" and unidad_salida == "lb":
22            resultado = valor * 2.205
23        elif unidad_entrada == "lb" and unidad_salida == "kg":
24            resultado = valor / 2.205
25
26        elif unidad_entrada == "kg" and unidad_salida == "g":
27            resultado = valor * 1000
28        elif unidad_entrada == "g" and unidad_salida == "kg":
29            resultado = valor / 1000
30
31        else:
32            raise ValueError("No se puede realizar la conversion")
33
34        # Agregar más conversiones según sea necesario
35
36        # Mostrar el resultado
37        label_resultado.config(text=f"Resultado: {resultado} {unidad_salida}")
38    except ValueError:
39        messagebox.showerror("Error", "Por favor, ingrese un valor numérico válido.")
40
41 # Función para guardar el resultado en un archivo de texto
42 def guardar_resultado():
43     try:
44         resultado_texto = label_resultado.cget("text")
45         with open("resultado.txt", "w") as archivo:
46             archivo.write(resultado_texto)
47         messagebox.showinfo("Guardado", "El resultado se ha guardado correctamente en 'resultado.txt'.")
48     except Exception as e:
```

```
49         messagebox.showerror("Error", f"Error al guardar el resultado: {e}")
50
51 # Crear la ventana principal
52 root = tk.Tk()
53 root.title("Convertidor de Unidades")
54
55 # Crear y posicionar los widgets
56 label_valor = ttk.Label(root, text="Valor:")
57 label_valor.grid(row=0, column=0, padx=5, pady=5)
58
59 entry_valor = ttk.Entry(root)
60 entry_valor.grid(row=0, column=1, padx=5, pady=5)
61
62 label_unidad_entrada = ttk.Label(root, text="Unidad de entrada:")
63 label_unidad_entrada.grid(row=1, column=0, padx=5, pady=5)
64
65 combo_unidad_entrada = ttk.Combobox(root, values=["m", "cm", "kg", "g", "lb"])
66 combo_unidad_entrada.grid(row=1, column=1, padx=5, pady=5)
67 combo_unidad_entrada.current(0)
68
69 label_unidad_salida = ttk.Label(root, text="Unidad de salida:")
70 label_unidad_salida.grid(row=2, column=0, padx=5, pady=5)
71
72 combo_unidad_salida = ttk.Combobox(root, values=["m", "cm", "kg", "g", "lb"])
73 combo_unidad_salida.grid(row=2, column=1, padx=5, pady=5)
74 combo_unidad_salida.current(1)
75
76 btn_convertir = ttk.Button(root, text="Convertir", command=convertir)
77 btn_convertir.grid(row=3, column=0, columnspan=2, padx=5, pady=5, sticky="WE")
78
79 label_resultado = ttk.Label(root, text="Resultado:")
80 label_resultado.grid(row=4, column=0, columnspan=2, padx=5, pady=5)
81
82 btn_guardar = ttk.Button(root, text="Guardar resultado", command=guardar_resultado)
83 btn_guardar.grid(row=5, column=0, columnspan=2, padx=5, pady=5, sticky="WE")
84
85 # Ejecutar el bucle principal
86 root.mainloop()
```

C. Resultados

The screenshot shows the 'Convertidor de...' application window. It has a title bar with a feather icon and standard window controls. The interface includes a 'Valor:' label with an input field containing '1'. Below it are two dropdown menus: 'Unidad de entrada:' set to 'm' and 'Unidad de salida:' set to 'cm'. A 'Convertir' button is positioned below the dropdowns. The result is displayed as 'Resultado: 100.0 cm'. At the bottom is a 'Guardar resultado' button.

The screenshot shows the 'Convertidor de...' application window with the same layout as the previous one. In this instance, the 'Valor:' input field contains '100'. The 'Unidad de entrada:' dropdown is set to 'cm' and the 'Unidad de salida:' dropdown is set to 'm'. The 'Convertir' button is visible, and the result is displayed as 'Resultado: 1.0 m'. The 'Guardar resultado' button is at the bottom.

Convertidor de...

Valor:

Unidad de entrada:

kg

Unidad de salida:

g

Convertir

Resultado: 1000.0 g

Guardar resultado

Convertidor de...

Valor:

Unidad de entrada:

g

Unidad de salida:

kg

Convertir

Resultado: 1.0 kg

Guardar resultado

Convertidor de...

Valor:

Unidad de entrada:

lb

Unidad de salida:

kg

Convertir

Resultado: 4.535147392290249 kg

Guardar resultado

Convertidor de...

Valor:

Unidad de entrada:

kg

Unidad de salida:

lb

Convertir

Resultado: 2.205 lb

Guardar resultado