



UNIVERSIDAD  
IBEROAMERICANA  
CIUDAD DE MÉXICO ®

## **P5. Interfaces visuales en Python**

Alumna: Daniela Mendez Ramirez  
Número de Cuenta: 258331-9

### **Aplicaciones Móviles**

Profesor: Omar Vázquez González  
Fecha de Entrega: 28 de enero de 2025

**Abstract:** En esta práctica, se explora el uso básico de la librería tkinter en Python, una herramienta fundamental para la creación de interfaces gráficas de usuario (GUI). A través de ejercicios sencillos, se aprenden los componentes básicos de tkinter como botones, etiquetas, cuadros de texto, radio buttons, checkboxes y la disposición de estos elementos utilizando el gestor de diseño grid. Se desarrollaron varias aplicaciones simples que permitieron familiarizarse con los fundamentos de las interfaces gráficas y cómo organizar los elementos dentro de una ventana. El objetivo de esta práctica es adquirir las bases necesarias para avanzar a aplicaciones más complejas, utilizando la interactividad que ofrece tkinter.

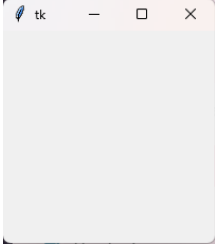
## **Introducción**

El desarrollo de interfaces gráficas de usuario (GUI) ha sido un tema esencial en el mundo de la programación, ya que permite una interacción más intuitiva entre los usuarios y los programas. En este sentido, tkinter se presenta como una de las bibliotecas más accesibles y poderosas para crear aplicaciones con interfaces gráficas en Python. Durante esta práctica, se dio inicio a los primeros pasos en el uso de tkinter mediante la realización de ejercicios sencillos. Estos ejercicios se enfocaron en el manejo de elementos básicos como botones, etiquetas, cuadros de texto, y otros componentes interactivos, utilizando el gestor de diseño grid para organizar los elementos dentro de la ventana. El propósito inicial fue comprender la estructura básica de una GUI en tkinter y familiarizarse con los controles de interacción básicos que forman la columna vertebral de cualquier aplicación gráfica.

En los primeros ejercicios, se implementaron actividades como la creación de botones con diferentes funcionalidades, la inclusión de etiquetas para describir los campos de entrada, y la disposición de estos elementos en una cuadrícula ordenada. También se incorporaron otros controles como los radio buttons y checkboxes, los cuales son esenciales para permitir la selección de opciones dentro de la interfaz. Estos ejercicios, aunque simples, establecen las bases para desarrollar aplicaciones más complejas en el futuro, integrando características como la manipulación de archivos o el manejo de datos dinámicos.

## Ejercicios para aprender

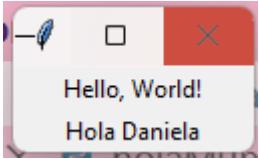
### 1. Habilitar una ventana

| Proceso  | Resultado   |
|--|---|
| <pre>Clase_9 &gt; TkinterEjercicios.py &gt; ... 1  # Hacemos nuestro primer hola mundo 2 3  import tkinter as tk 4 5  root = tk.Tk() 6  root.mainloop() # Habilita una ventana 7 8</pre> |  |

### 2. Hola Mundo

| Proceso   | Resultado  |
|---|--|
| <pre>AbrirUnaVentana.py  holaMundo.py u x Clase_9 &gt; holaMundo.py &gt; ... 1  # Agregamos la librería 2  import tkinter as tk 3 4  # Creamos la instancia 5  root = tk.Tk() 6 7  # Escribimos nuestro mensaje 8  message = tk.Label(root, text="Hello, World!") 9  message = tk.Label(root, text="Hola Daniela") 10 message.pack() 11 12 # Mantenemos la pestaña desplegandose 13 root.mainloop()</pre> |  |

### 3. Agregar dos textos

| Proceso  | Resultado   |
|--|---|
| <pre>Clase_9 &gt; holaMundo.py &gt; ... 1  # Agregamos la librería 2  import tkinter as tk 3 4  # Creamos la instancia 5  root = tk.Tk() 6 7  # Escribimos nuestro mensaje 8  message = tk.Label(root, text="Hello, World!") 9  message2 = tk.Label(root, text="Hola Daniela") 10 message.pack() 11 message2.pack() 12 13 # Mantenemos la pestaña desplegandose 14 root.mainloop()</pre> |  |

#### 4. Modificaciones en la ventana de trabajo

| Proceso   | Resultado  |
|---|--|
| <pre>Clase_9 &gt; modificacionesVentana.py &gt; ... 1 # Agregamos la librería 2 import tkinter as tk 3 4 # Creamos la instancia 5 root = tk.Tk() 6 7 # Cambiamos el nombre de la ventana 8 root.title('Pruebita Daniela') 9 10 # Escribimos nuestro mensaje 11 message = tk.Label(root, text="Hello, World!") 12 message2 = tk.Label(root, text="Hola Daniela") 13 message.pack() 14 message2.pack() 15 16 # Mantenemos la pestaña desplegandose 17 root.mainloop()</pre> |    |
| <p>Con esta línea podemos definir:<br/>Ancho x Altura x pixeles a la derecha<br/>x pixeles a la izquierda</p> <pre>root.geometry('600x400+750+150')</pre> <pre>7 # Cambiamos el nombre de la ventana 8 root.title('Pruebita Daniela') 9 # Definimos el tamaño de la pantalla 10 root.geometry('600x400+750+150') 11 # Deshabilitamos ajustar el tamaño de la ventana 12 root.resizable(False, False)</pre>  |   |
| <p>Hacemos una transparencia</p> <pre>1 # Agregamos la librería 2 import tkinter as tk 3 4 # Creamos la instancia 5 root = tk.Tk() 6 7 # Cambiamos el nombre de la ventana 8 root.title('Pruebita Daniela') 9 # Definimos el tamaño de la pantalla 10 root.geometry('600x400+750+150') 11 # Deshabilitamos ajustar el tamaño de la ventana 12 root.resizable(False, False) 13 root.attributes('-alpha', 0.5) 14 root.mainloop()</pre>                                     |  |

## 5. Widgets

| Proceso  | Resultado |
|--|-----------|
| <p>Diferentes labels</p> <pre>Clase_9 &gt; widgets.py &gt; ... 1  # Practicamos usar los widgets 2 3  import tkinter as tk 4  from tkinter import ttk 5 6  root = tk.Tk() 7 8  root.geometry('600x400+750+150') 9 10 11 tk.Label(root, text='Classic Label').pack() 12 # Se ve un poco mas oscuro 13 ttk.Label(root, text='Themed Label').pack() 14 15 root.mainloop()</pre> |           |

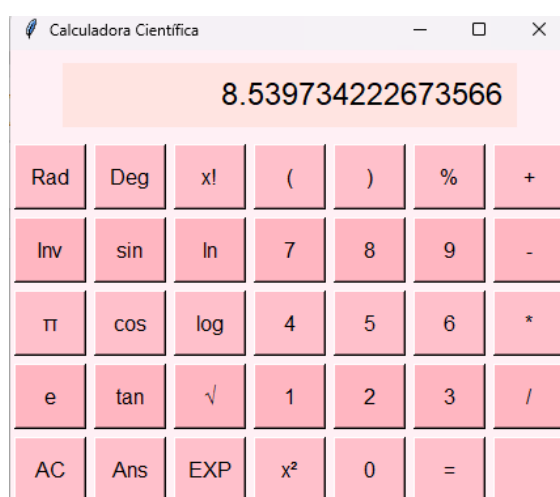
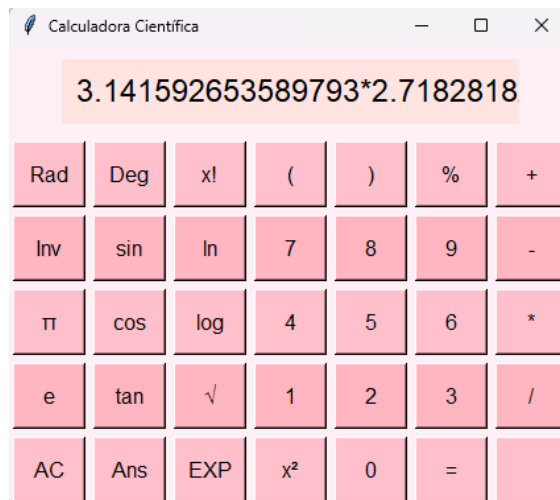
2) Una calculadora básica que realice, al menos las 4 operaciones fundamentales. Si desea agregar raíces, potencias, logaritmos mejorará su calificación.

| Proceso   | Resultado   |
|---|---|
| <pre> import tkinter as tk import math  # Crear la ventana principal raiz = tk.Tk() raiz.title("Calculadora Científica") raiz.configure(bg="#FFF0F5") # Color de fondo rosado claro  # Campo de entrada para los cálculos entrada = tk.Entry(     raiz,     width=25,     font=("Arial", 18),     justify="right",     bg="#FFE4E1", # Fondo rosado pastel     fg="#000000",     bd=10,     relief=tk.FLAT ) entrada.grid(row=0, column=0, columnspan=7, pady=10, padx=10) # Cambié columnspan de 6 a 7  # Función para manejar las acciones de los botones def al_hacer_click(valor_boton):     try:         if valor_boton == "=": # Evaluar la expresión ingresada             resultado = eval(                 entrada.get(), </pre> |  <p>The image displays two screenshots of a Python-based scientific calculator application. The top screenshot shows the initial state of the application window, titled 'Calculadora Científica', which has a light pink background. It features a text input field at the top and a grid of buttons below, including mathematical functions like Rad, Deg, x!, (, ), %, +, Inv, sin, ln, 7, 8, 9, -, π, cos, log, 4, 5, 6, *, e, tan, √, 1, 2, 3, /, AC, Ans, EXP, x², 0, =, and a blank button. The bottom screenshot shows the same application after the expression '63*9/6+1*8' has been entered into the input field, demonstrating the calculator's functionality.</p> |

```

        {"math": math,
"sqrt": math.sqrt, "log":
math.log, "sin": math.sin,
        "cos": math.cos,
"tan": math.tan, "pi": math.pi,
"e": math.e}
    )
    entrada.delete(0,
tk.END)
    entrada.insert(tk.E
ND, str(resultado))
    elif valor_boton ==
"AC": # Limpiar el campo de
entrada
        entrada.delete(0,
tk.END)
    elif valor_boton ==
"Rad": # Convertir a radianes
        entrada.insert(tk.E
ND, "math.radians(")
    elif valor_boton ==
"Deg": # Convertir a grados
        entrada.insert(tk.E
ND, "math.degrees(")
    elif valor_boton in
["sin", "cos", "tan", "log",
"sqrt", "exp"]: # Funciones
matemáticas
        entrada.insert(tk.E
ND, f"math.{valor_boton}(")
    elif valor_boton ==
"x!": # Calcular el factorial
        expresion =
int(entrada.get())
        resultado =
math.factorial(expresion)
        entrada.delete(0,
tk.END)
        entrada.insert(tk.E
ND, str(resultado))
    elif valor_boton ==
"Ans": # Placeholder para
último resultado
        pass # Puedes agregar
la funcionalidad aquí

```



```

        elif valor_boton ==
"x^2": # Elevar al cuadrado
            expresion =
float(entrada.get())
            resultado = expresion
** 2
            entrada.delete(0,
tk.END)
            entrada.insert(tk.E
ND, str(resultado))
        elif valor_boton ==
"π": # Insertar el valor de pi
            entrada.insert(tk.E
ND, str(math.pi))
        elif valor_boton ==
"e": # Insertar el valor de e
            entrada.insert(tk.E
ND, str(math.e))
        else: # Insertar
cualquier otro valor
            entrada.insert(tk.E
ND, valor_boton)
    except Exception as e:
        entrada.delete(0, tk.END)
        entrada.insert(tk.END,
"Error") # Mostrar mensaje de
error en caso de fallo

# Lista de botones con su
disposición, agregando columna
para operaciones básicas
botones = [
    ('Rad', 1, 0), ('Deg', 1, 1),
    ('x!', 1, 2), ('(', 1, 3), (')',
1, 4), ('%', 1, 5), ('+', 1, 6),
    ('Inv', 2, 0), ('sin', 2, 1),
    ('ln', 2, 2), ('7', 2, 3), ('8',
2, 4), ('9', 2, 5), ('-', 2, 6),
    ('π', 3, 0), ('cos', 3, 1),
    ('log', 3, 2), ('4', 3, 3), ('5',
3, 4), ('6', 3, 5), ('*', 3, 6),
    ('e', 4, 0), ('tan', 4, 1),
    ('√', 4, 2), ('1', 4, 3), ('2',
4, 4), ('3', 4, 5), ('/', 4, 6),

```

```

63 # lista de botones con su disposición, agregando columna para operaciones básicas
64 botones = [
65     ('Rad', 1, 0), ('Deg', 1, 1), ('x!', 1, 2), ('(', 1, 3), (')', 1, 4), ('%', 1, 5), ('+', 1, 6),
66     ('Inv', 2, 0), ('sin', 2, 1), ('ln', 2, 2), ('7', 2, 3), ('8', 2, 4), ('9', 2, 5), ('-', 2, 6),
67     ('π', 3, 0), ('cos', 3, 1), ('log', 3, 2), ('4', 3, 3), ('5', 3, 4), ('6', 3, 5), ('*', 3, 6),
68     ('e', 4, 0), ('tan', 4, 1), ('√', 4, 2), ('1', 4, 3), ('2', 4, 4), ('3', 4, 5), ('/', 4, 6),
69     ('AC', 5, 0), ('Ans', 5, 1), ('EXP', 5, 2), ('x²', 5, 3), ('0', 5, 4), ('=', 5, 5), (' ', 5, 6)
70 ]
71

```



```

        ('AC', 5, 0), ('Ans', 5, 1),
        ('EXP', 5, 2), ('x²', 5, 3),
        ('0', 5, 4), ('=', 5, 5), (' ',
5, 6)
    ]

    # Crear botones dinámicamente
    for (texto, fila, columna) in
    botones:
        boton = tk.Button(
            raiz,
            text=texto,
            width=5,
            height=2,
            font=("Arial", 12),
            bg="#FFB6C1" if fila % 2
            == 0 else "#FFC0CB", # Alternar
            tonos rosados
            fg="#000000",
            activebackground="#FF69
B4",
            bd=2,
            relief=tk.RAISED,
            command=lambda t=texto:
            al_hacer_click(t)
            )
        boton.grid(row=fila,
            column=columna, padx=3, pady=3,
            sticky="nsew")

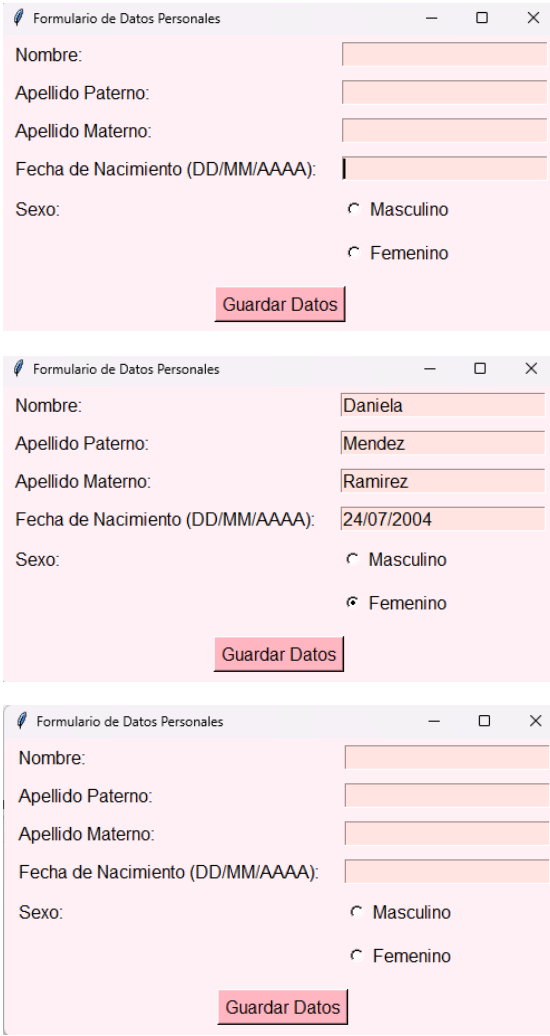
    # Ajustar el tamaño de las
    columnas y filas
    total_columnas = 7 #
    Actualizado a 7 columnas
    for columna in
    range(total_columnas):
        raiz.grid_columnconfigure(c
        olumna, weight=1)
    for fila in range(1, 6):
        raiz.grid_rowconfigure(fila,
        weight=1)

    # Iniciar el bucle principal de
    la interfaz gráfica
    raiz.mainloop()

```

|  |  |
|--|--|
|  |  |
|--|--|

3) Un formulario básico que almacene los datos de una persona (nombre, apellido paterno, materno, fecha de nacimiento y sexo), y los guarde en un archivo de texto TXT

| Proceso   | Resultado   |
|---|---|
| <pre> import tkinter as tk  # Función para guardar los datos en un archivo de texto def guardar_datos():     nombre = entrada_nombre.get()     apellido_paterno = entrada_apellido_paterno.get()     apellido_materno = entrada_apellido_materno.get()     fecha_nacimiento = entrada_fecha_nacimiento.get()     sexo = var_sexo.get()      # Guardar en el archivo de     texto      with     open("datos_persona.txt", "a")     as archivo:         archivo.write(f"Nombre: {nombre}\n")         archivo.write(f"Apellido Paterno: {apellido_paterno}\n")         archivo.write(f"Apellido Materno: {apellido_materno}\n")         archivo.write(f"Fecha de Nacimiento: {fecha_nacimiento}\n")         archivo.write(f"Sexo: {sexo}\n")         archivo.write("-" * 30 + "\n") # Separador para facilitar lectura      # Limpiar el formulario     después de guardar     entrada_nombre.delete(0, tk.END) </pre> |  <p>The 'Resultado' column displays three sequential screenshots of a Tkinter window titled 'Formulario de Datos Personales'. The window contains several input fields: 'Nombre:', 'Apellido Paterno:', 'Apellido Materno:', 'Fecha de Nacimiento (DD/MM/AAAA):', and 'Sexo:'. The 'Sexo:' field uses radio buttons for 'Masculino' and 'Femenino'. A 'Guardar Datos' button is located at the bottom right of the form area. In the first screenshot, all fields are empty. In the second screenshot, the fields are populated with the following data: 'Nombre: Daniela', 'Apellido Paterno: Mendez', 'Apellido Materno: Ramirez', 'Fecha de Nacimiento: 24/07/2004', and 'Sexo: Femenino'. In the third screenshot, after the 'Guardar Datos' button has been clicked, all input fields are cleared, returning the form to its initial state.</p> |

```

    entrada_apellido_paterno.delete(0, tk.END)
    entrada_apellido_materno.delete(0, tk.END)
    entrada_fecha_nacimiento.delete(0, tk.END)
    var_sexo.set(None)

```

```

# Crear la ventana principal
ventana = tk.Tk()
ventana.title("Formulario de Datos Personales")
ventana.configure(bg="#FFF0F5")
# Color de fondo rosado claro

```

```

# Crear las etiquetas y campos de entrada

```

```

etiqueta_nombre = tk.Label(ventana,
text="Nombre:", font=("Arial", 12), bg="#FFF0F5")
etiqueta_nombre.grid(row=0, column=0, padx=10, pady=5, sticky="w")
entrada_nombre = tk.Entry(ventana,
font=("Arial", 12), bg="#FFE4E1")
entrada_nombre.grid(row=0, column=1, padx=10, pady=5)

```

```

etiqueta_apellido_paterno = tk.Label(ventana,
text="Apellido Paterno:", font=("Arial", 12), bg="#FFF0F5")
etiqueta_apellido_paterno.grid(row=1, column=0, padx=10, pady=5, sticky="w")
entrada_apellido_paterno = tk.Entry(ventana,
font=("Arial", 12), bg="#FFE4E1")
entrada_apellido_paterno.grid(row=1, column=1, padx=10, pady=5)

```

Clase\_9

10t.jpg

AbrirUnaVer

calculadora.py

datos\_persona.txt

C:\Users\dany\OneD

Semestre\Redes2\Ejer

U

U

Clase\_9 > datos\_persona.txt

1

Nombre: Daniela

2

Apellido Paterno: Mendez

3

Apellido Materno: Ramirez

4

Fecha de Nacimiento: 24/07/2004

5

Sexo: Femenino

6

-----

7

Nombre: Daniela

8

Apellido Paterno: Mendez

9

Apellido Materno: Ramirez

10

Fecha de Nacimiento: 24/07/2004

11

Sexo: Femenino

12

-----

13

```

etiqueta_apellido_materno      =
tk.Label(ventana,
text="Apellido      Materno:",
font=("Arial",          12),
bg="#FFF0F5")
etiqueta_apellido_materno.grid(
row=2,    column=0,    padx=10,
pady=5, sticky="w")
entrada_apellido_materno      =
tk.Entry(ventana,
font=("Arial",          12),
bg="#FFE4E1")
entrada_apellido_materno.grid(r
ow=2, column=1, padx=10, pady=5)

etiqueta_fecha_nacimiento      =
tk.Label(ventana, text="Fecha de
Nacimiento      (DD/MM/AAAA):",
font=("Arial",          12),
bg="#FFF0F5")
etiqueta_fecha_nacimiento.grid(
row=3,    column=0,    padx=10,
pady=5, sticky="w")
entrada_fecha_nacimiento      =
tk.Entry(ventana,
font=("Arial",          12),
bg="#FFE4E1")
entrada_fecha_nacimiento.grid(r
ow=3, column=1, padx=10, pady=5)

etiqueta_sexo                  =
tk.Label(ventana, text="Sexo:",
font=("Arial",          12),
bg="#FFF0F5")
etiqueta_sexo.grid(row=4,
column=0,    padx=10,    pady=5,
sticky="w")

# Opción de género: radio
buttons
var_sexo = tk.StringVar()
radio_masculino      =
tk.Radiobutton(ventana,
text="Masculino",

```

```

variable=var_sexo,
value="Masculino",
font=("Arial", 12),
bg="#FFF0F5")
radio_masculino.grid(row=4,
column=1, padx=10, pady=5,
sticky="w")
radio_femenino =
tk.Radiobutton(ventana,
text="Femenino",
variable=var_sexo,
value="Femenino",
font=("Arial", 12),
bg="#FFF0F5")
radio_femenino.grid(row=5,
column=1, padx=10, pady=5,
sticky="w")

# Botón para guardar los datos
boton_guardar =
tk.Button(ventana,
text="Guardar Datos",
font=("Arial", 12),
bg="#FFB6C1", fg="#000000",
command=guardar_datos)
boton_guardar.grid(row=6,
column=0, colspan=2, pady=10)

# Iniciar la ventana
ventana.mainloop()

```

## **Conclusión:**

El aprendizaje de los componentes fundamentales de tkinter ha sido un paso crucial en el camino para desarrollar aplicaciones gráficas más complejas. A través de los ejercicios realizados en esta práctica, se ha adquirido una comprensión básica sobre cómo interactuar con los elementos de la interfaz, desde la creación de botones y etiquetas hasta la disposición organizada de estos en una ventana utilizando grid. Aunque los ejercicios realizados hasta ahora son sencillos, constituyen una base sólida para futuras aplicaciones que requieran una mayor interacción y funcionalidad. A medida que avancemos en el aprendizaje de tkinter, será posible integrar más controles, mejorar el diseño y hacer las aplicaciones más dinámicas y útiles. En resumen, este ejercicio inicial proporciona una base sólida para los pasos siguientes en la programación de interfaces gráficas en Python, y ofrece una comprensión inicial de los aspectos más importantes de la interacción entre el usuario y el software.