A red sign with white text

Description automatically generated

**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 |
| A screenshot of a computer  Description automatically generated |  |

1. Hola Mundo

|  |  |
| --- | --- |
| Proceso | Resultado |
|  |  |

1. Agregar dos textos

|  |  |
| --- | --- |
| Proceso | Resultado |
|  |  |

1. Modificaciones en la ventana de trabajo

|  |  |
| --- | --- |
| Proceso | Resultado |
|  |  |
| Con esta línea podemos definir:  Ancho x Altura x pixeles a la derecha  x pixeles a la izquierda |  |
| Hacemos una transparencia | A screenshot of a computer  Description automatically generated |

1. Widgets

|  |  |
| --- | --- |
| Proceso | Resultado |
| Diferentes labels |  |

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

|  |  |
| --- | --- |
| Proceso | Resultado |
| 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(),                  {"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.END, 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.END, "math.radians(")          elif valor\_boton == "Deg":  # Convertir a grados              entrada.insert(tk.END, "math.degrees(")          elif valor\_boton in ["sin", "cos", "tan", "log", "sqrt", "exp"]:  # Funciones matemáticas              entrada.insert(tk.END, 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.END, str(resultado))          elif valor\_boton == "Ans":  # Placeholder para último resultado              pass  # Puedes agregar la funcionalidad aquí          elif valor\_boton == "x²":  # Elevar al cuadrado              expresion = float(entrada.get())              resultado = expresion \*\* 2              entrada.delete(0, tk.END)              entrada.insert(tk.END, str(resultado))          elif valor\_boton == "π":  # Insertar el valor de pi              entrada.insert(tk.END, str(math.pi))          elif valor\_boton == "e":  # Insertar el valor de e              entrada.insert(tk.END, str(math.e))          else:  # Insertar cualquier otro valor              entrada.insert(tk.END, 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),      ('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="#FF69B4",          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(columna, 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 |
| 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)      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)  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(row=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(row=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, columnspan=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.