
GESTIÓN DE EVENTOS Y FUNCIONES PERSONALIZADAS CON TKINTER

Aprender a conectar los widgets de la interfaz con funciones que ejecutan lógica personalizada, como cálculos, validaciones, cambios en tiempo real, y toma de decisiones.

1. ASOCIAR FUNCIONES A BOTONES (COMMAND=)

- Las funciones deben definirse antes de ser usadas. No se colocan paréntesis al pasar el nombre de la función (ej: `command=saludar`, no `saludar()`).

```
def saludar():  
    print("Hola, mundo!")
```

```
boton = tk.Button(ventana, text="Saludar", command=saludar)
```

EXPLICACIÓN LÍNEA POR LÍNEA:

DEF SALUDAR():

- Esta línea define una función en Python llamada saludar.
- Toda la lógica que estará indentada debajo de esta línea será ejecutada cuando se llame a la función.
- Las funciones en Tkinter se utilizan para responder a eventos, como un clic de botón.

PRINT("HOLA, MUNDO!")

- . Esta es la acción que ejecuta la función saludar. Cuando se invoque saludar(), Python imprimirá "Hola, mundo!" en la consola. En una aplicación Tkinter con ventana gráfica, esta impresión no se verá en la interfaz, sino en la terminal donde se ejecutó el script.

```
BOTON = TK.BUTTON(VENTANA, TEXT="SALUDAR",  
COMMAND=SALUDAR)
```

- Aquí se está creando un botón usando la biblioteca `tkinter.tk.Button(...)`: llama al constructor del widget tipo botón.
- `ventana`: es el contenedor donde se ubicará el botón (debe haberse definido previamente con `ventana = tk.Tk()`).
- `text="Saludar"`: es el texto que aparecerá en el botón.
- `command=saludar`: este es el vínculo entre el botón y la función `saludar`.
- Cuando el usuario haga clic en el botón, se ejecutará la función `saludar`. Notar que NO lleva paréntesis (`saludar`, no `saludar()`), porque queremos pasar la referencia a la función, no ejecutarla inmediatamente.

2. CAPTURAR Y PROCESAR DATOS DE ENTRADA

- Obtener valores desde Entry, Text, Checkbutton, Radiobutton, etc. Convertir tipos (int(), float()) y validar rangos o formatos.python

```
def calcular_doble():  
    valor = entrada.get()  
    if valor.isdigit():  
        resultado = int(valor) * 2  
        etiqueta_resultado.config(text=f"El doble es: {resultado}")  
    else:  
        etiqueta_resultado.config(text="Ingrese un número válido")
```

EXPLICACIÓN LÍNEA POR LÍNEA:

DEF CALCULAR_DOBLE():

- Se define una función llamada `calcular_doble`, que será ejecutada al presionar un botón.
- Todo lo que está indentado debajo de esta línea pertenece a esa función.

`VALOR = ENTRADA.GET()`

- `entrada.get()` lee el contenido que el usuario escribió en el widget Entry llamado `entrada`. Ese valor se guarda como texto en la variable `valor`. Ejemplo: si el usuario escribió 7, `valor` será "7" (una cadena de texto).

EXPLICACIÓN LÍNEA POR LÍNEA:

IF VALOR.ISDIGIT():

- Esta línea verifica si el valor ingresado está compuesto solo por dígitos, es decir, si es un número entero positivo.
- Si el valor es válido (por ejemplo "25"), entra al bloque if.
- Si el valor no es numérico (por ejemplo "abc" o está vacío), ejecuta el bloque else.do debajo de esta línea pertenece a esa función.

RESULTADO = INT(VALOR) * 2

- Convierte el texto valor a un número entero con int(valor).
- Calcula el doble del número (* 2) y guarda el resultado en la variable resultado.

```
ETIQUETA_RESULTADO.CONFIG(TEXT=F"EL DOBLE ES:  
{RESULTADO}")
```

- Modifica el texto del widget Label llamado etiqueta_resultado.
- Usa un f-string para mostrar el resultado como texto, por ejemplo: "El doble es: 14".

ELSE:

- Si el valor ingresado no era un número, se ejecuta este bloque alternativo.

```
ETIQUETA_RESULTADO.CONFIG(TEXT="INGRESE UN NÚMERO VÁLIDO")
```

- Muestra un mensaje de error al usuario en la etiqueta etiqueta_resultado, indicando que el dato ingresado no es correcto.

3. VALIDACIONES CON CONDICIONALES (IF/ELSE)

- Verificar condiciones: números positivos, campos vacíos, valores en rangos.

```
def verificar_edad():  
    edad = entrada.get()  
    if edad.isdigit():  
        edad = int(edad)  
        if edad >= 18:  
            resultado.config(text="Es mayor de edad")  
        else:  
            resultado.config(text="Es menor de edad")  
    else:  
        resultado.config(text="Ingrese una edad válida")
```

4. MANIPULAR EL ESTADO DE WIDGETS

- Usar `.config()` para actualizar texto o propiedades.
- Cambiar color, tamaño, habilitar/deshabilitar.

```
def cambiar_color():  
    etiqueta.config(fg="red", font=("Arial", 14, "bold"))
```

5. EVENTOS MÁS ALLÁ DEL BOTÓN: BIND

- Capturar teclas, clicks, etc.

```
def al_presionar_tecla(event):  
    print("Presionó:", event.char)  
  
ventana.bind("<Key>", al_presionar_tecla)
```

4. MANIPULAR EL ESTADO DE WIDGETS

- Usar `.config()` para actualizar texto o propiedades.
- Cambiar color, tamaño, habilitar/deshabilitar.

```
def cambiar_color():  
    etiqueta.config(fg="red", font=("Arial", 14, "bold"))
```