



## Ciclo 1

# Fundamentos de programación con Python

## Sesión 22: Introducción a Interfaz grafica con tkinter 2

Programa Ciencias de la Computación e Inteligencia Artificial  
Escuela de Ciencias Exactas e Ingeniería  
Universidad Sergio Arboleda  
Bogotá





# Agenda

1. Interfaces graficas de usuario
2. Las ideas principales
3. El programa GUI más sencillo
4. Programación basada en eventos
5. Ampliación de ejemplo
6. Ejemplo Radiobutton





# 1. Interfaces gráficas de usuario

- Las ideas clave de las interfaces gráficas de usuario (ventanas en pantalla, iconos, menús, botones, etc., y un dispositivo señalador (ratón)) fueron desarrollaron en Xerox PARC a finales de los años 70.
- Estas ideas fueron adoptadas por Apple, primero en el Lisa (1983) y luego en el popular Macintosh (1984). En 1985, Microsoft introdujo Windows, primero como aplicación y después como sistema operativo.
- Python hace que sea muy fácil implementar interfaces gráficas de usuario simples.





## 2. Las ideas principales

- La estructura del programa es un bucle, que pide al usuario un comando y luego llama a una función en un comando y luego llama a una función para llevarlo a cabo.
- Los programas GUI tienen una estructura similar, salvo que el bucle principal es proporcionado por un módulo, y las funciones que hacen el trabajo son llamadas en respuesta a los clics del ratón en los botones, etc.





### 3. El programa GUI más sencillo en Python

```
# Use el modulo Tkinter
```

```
import Tkinter
```

```
# Crear la ventana de nivel superior (o raíz)
```

```
top = Tkinter.Tk()
```

```
# Crear un boton ...
```

```
quitButton = Tkinter.Button(top, text="Quit",  
                             command=top.destroy)
```

```
# ... Y mostrarlo en la ventana
```

```
quitButton.grid()
```

```
# Inicia el bucle principal: responde al ratón, etc
```

```
Tkinter.mainloop()
```







### 3. El programa GUI más sencillo en Python

```
top = Tkinter.Tk()
```

Esto debe estar presente para crear una ventana. Siempre se necesita al menos una ventana para poner botones, etc





### 3. El programa GUI más sencillo en Python

```
quitButton = Tkinter.Button(top, text="Quit",  
                             command=top.destroy)
```

debe estar presente, para  
asociar el botón con la ventana  
raíz

opcional

especifica una llamada de retorno que se utilizará cuando se pulse el botón





### 3. El programa GUI más sencillo en Python

```
quitButton.grid()
```

Esto utiliza el gestor de diseño llamado grid para colocar el botón en la ventana raíz.  
Sin esta línea, el botón no se mostrará.







### 3. El programa GUI más sencillo en Python

**Tkinter.mainloop()**

- Esto inicia el bucle principal, que rastrea el ratón y trabaja cuando y donde ha sido presionado
- Al pulsar el ratón sobre el botón Quit se hace la llama de retorno: es decir, que se llame al método `top.destroy`, que termina la ventana raíz.





## 4. Programación basada en eventos

- Las aplicaciones GUI utilizan un estilo de programación llamado impulsada por eventos. Los eventos son movimientos del ratón, clics del ratón, pulsaciones de teclas, y muchos eventos de nivel superior contruidos a partir de estos.
- Por ejemplo, hacer clic con el ratón mientras el puntero está sobre genera un evento de pulsación de botón.
- Algunos eventos se manejan completamente dentro del bucle principal proporcionado por Tkinter.
- Por ejemplo, los movimientos del ratón se utilizan para actualizar la posición del puntero en la pantalla; hacer clic en el botón de minimizar de la ventana tiene el efecto habitual.





## 4. Programación basada en eventos

- Otros eventos, por lo general de mayor nivel, como la pulsación de botones, la selección de menús o la escritura en un campo de texto, se manejan de una manera que involucra el código del usuario.
- Esto se controla mediante la definición de *callbacks*.
- Ejemplo: si tenemos un botón, el evento que nos interesa es pulsarlo. Cuando se crea el botón, el parámetro de comando se utiliza para especificar qué función llamar cuando se presiona el botón.

```
quitButton = Tkinter.Button(top, text="Quit",  
                             command=top.destroy)
```





## 5. Ampliación del ejemplo

- Primero vamos a añadir algo que nos permita mostrar un mensaje al usuario. Tkinter proporciona Label para este propósito.

```
import Tkinter
```

```
top = Tkinter.Tk()
```

```
messageLabel = Tkinter.Label(top, text="Hello World!")  
messageLabel.grid()
```

```
quitButton = Tkinter.Button(top, text="Quit",  
                             command=top.destroy)
```

```
quitButton.grid()
```

```
Tkinter.mainloop()
```





## 5. Ampliación del ejemplo

- En lugar de mostrar el mensaje inmediatamente, vamos a añadir otro botón con una llamada de retorno que mostrará el mensaje







## 5. Ampliación del ejemplo

```
import Tkinter
```

```
def display():
```

```
    messageLabel.configure(text="Hello World!")
```

```
top = Tkinter.Tk()
```

```
messageLabel = Tkinter.Label(top, text="")
```

```
messageLabel.grid()
```

```
showButton =
```

```
Tkinter.Button(top, text="Show", command=display)
```

```
showButton.grid()
```

```
quitButton =
```

```
Tkinter.Button(top, text="Quit", command=top.destroy)
```

```
quitButton.grid()
```

```
Tkinter.mainloop()
```





## 5. Puntos a tener en cuenta

```
import Tkinter

def display():
    messageLabel.configure(text="Hello World!")

top = Tkinter.Tk()

messageLabel = Tkinter.Label(top, text="")
messageLabel.grid()

showButton = Tkinter.Button(top, text="Show", command=display)
showButton.grid()

quitButton = Tkinter.Button(top, text="Quit", command=top.destroy)
quitButton.grid()

Tkinter.mainloop()
```

definición antes del uso

sin paréntesis





## 5. Terminología

El término genérico para un elemento GUI (botón, menú, etiqueta, ...) es un widget.





## 5. Cambiar el diseño

Podemos utilizar argumentos opcionales con el método grid para controlar cómo se colocan los widgets.

```
import Tkinter

def display():
    messageLabel.configure(text="Hello World!")

top = Tkinter.Tk()

messageLabel = Tkinter.Label(top, text="", width=12)
messageLabel.grid(row=0, column=0)

showButton = Tkinter.Button(top, text="Show", command=display)
showButton.grid(row=0, column=1)

quitButton = Tkinter.Button(top, text="Quit", command=top.destroy)
quitButton.grid(row=0, column=2)

Tkinter.mainloop()
```





## 5. Obtención de información del usuario

```
import Tkinter

def display():
    name = textVar.get()
    messageLabel.configure(text="Hello "+name)

top = Tkinter.Tk()

textVar = Tkinter.StringVar("")
textEntry = Tkinter.Entry(top,textvariable=textVar,width=12)
textEntry.grid(row=0,column=0)

messageLabel = Tkinter.Label(top,text="",width=12)
messageLabel.grid(row=1,column=0)

showButton = Tkinter.Button(top,text="Show",command=display)
showButton.grid(row=1,column=1)

quitButton = Tkinter.Button(top,text="Quit",command=top.destroy)
quitButton.grid(row=1,column=2)

Tkinter.mainloop()
```







## 5. Idea importante

- El widget **Entry** permite al usuario introducir texto, pero no tiene almacenamiento para el texto incorporado.
- Tenemos que crear un objeto **Tkinter.StringVar** y dárselo al objeto Entry.
- Luego podemos utilizar el método **get** del StringVar para obtener el texto.
- Este estilo de programación también es necesario con varios otros widgets de Tkinter.
- *Debe ser una StringVar, no una variable de cadena ordinaria.*





## 6. Ejemplo de Radiobutton

```
def display():
    name = textVar.get()
    ch = choice.get()
    if ch == 1:
        message = "Hello "+name
    elif ch == 2:
        message = "Goodbye "+name
    else:
        message = ""
    messageLabel.configure(text=message)

top = Tkinter.Tk()
textVar = Tkinter.StringVar("")
textEntry = Tkinter.Entry(top,textvariable=textVar,width=12)
textEntry.grid(row=0,column=0)
messageLabel = Tkinter.Label(top,text="",width=12)
messageLabel.grid(row=1,column=0)
choice = Tkinter.IntVar(0)
helloButton = Tkinter.Radiobutton(top,text="Hello",
                                   variable=choice,value=1,command=display)
helloButton.grid(row=1,column=1)
goodbyeButton = Tkinter.Radiobutton(top,text="Goodbye",
                                     variable=choice,value=2,command=display)
goodbyeButton.grid(row=1,column=2)
quitButton = Tkinter.Button(top,text="Quit",command=top.destroy)
quitButton.grid(row=1,column=3)
Tkinter.mainloop()
```





## 6. Ejemplo de Radiobutton

```
def display():
    name = textVar.get()
    ch = choice.get()
    if ch == 1:
        message = "Hello "+name
    elif ch == 2:
        message = "Goodbye "+name
    else:
        message = ""
    messageLabel.configure(text=message)

top = Tkinter.Tk()
textVar = Tkinter.StringVar("")
textEntry = Tkinter.Entry(top,textvariable=textVar,width=12)
textEntry.grid(row=0,column=0)
messageLabel = Tkinter.Label(top,text="",width=12)
messageLabel.grid(row=1,column=0)
choice = Tkinter.IntVar(0)
helloButton = Tkinter.Radiobutton(top,text="Hello",
                                   variable=choice,value=1,command=display)
helloButton.grid(row=1,column=1)
goodbyeButton = Tkinter.Radiobutton(top,text="Goodbye",
                                     variable=choice,value=2,command=display)
goodbyeButton.grid(row=1,column=2)
quitButton = Tkinter.Button(top,text="Quit",command=top.destroy)
quitButton.grid(row=1,column=3)
Tkinter.mainloop()
```





## 6. Ejemplo de Radiobutton

```
def display():
    name = textVar.get()
    ch = choice.get()
    if ch == 1:
        message = "Hello "+name
    elif ch == 2:
        message = "Goodbye "+name
    else:
        message = ""
    messageLabel.configure(text=message)

top = Tkinter.Tk()
textVar = Tkinter.StringVar("")
textEntry = Tkinter.Entry(top,textvariable=textVar,width=12)
textEntry.grid(row=0,column=0)
messageLabel = Tkinter.Label(top,text="",width=12)
messageLabel.grid(row=1,column=0)
choice = Tkinter.IntVar(0)
helloButton = Tkinter.Radiobutton(top,text="Hello",
                                   variable=choice,value=1,command=display)
helloButton.grid(row=1,column=1)
goodbyeButton = Tkinter.Radiobutton(top,text="Goodbye",
                                     variable=choice,value=2,command=display)
goodbyeButton.grid(row=1,column=2)
quitButton = Tkinter.Button(top,text="Quit",command=top.destroy)
quitButton.grid(row=1,column=3)
Tkinter.mainloop()
```





El futuro digital  
es de todos

MinTIC

# Preguntas



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC 2022