

# Clase5\_GUI

April 10, 2023

## 1 Seminario de Lenguajes - Python

### 1.1 Cursada 2023

#### 1.1.1 Clase 5: GUI

## 2 ¿Sabén qué son las GUI?

### 2.1 GUI: Graphical User Interface

## 3 GUI en Python

- Vamos a usar [PySimpleGUI](#)
- Es un **framework** bastante simple para desarrollar interfaces gráficas en Python.
- Es software libre: <https://github.com/PySimpleGUI/PySimpleGUI>
- Se instala con pip: **pip install pysimplegui**
- Vamos a ir de a poco...
- Más info en: <https://pysimplegui.readthedocs.io/en/latest/cookbook/>
- También en <https://realpython.com/pysimplegui-python/>

## 4 ¿Con qué elementos podemos trabajar?

## 5 Popups: las ventanas más sencillas

```
[ ]: import PySimpleGUI as sg
sg.Popup('Mi primera ventanita')
```

- **import PySimpleGUI as sg**, permite acceder a los recursos por el nombre **sg**.

```
[ ]: import PySimpleGUI as sg

sg.Popup('Mi primera ventanita popup', button_color=('black', 'red'))

sg.PopupYesNo('Mi primera ventanita yes/no', button_color=('black', 'green'))
```

```
sg.PopupOKCancel('Mi primera ventanita ok/cancel', button_color=('black',
↪ 'grey'))

texto = sg.PopupGetText('Titulo', 'Ingresá un texto acá ...')
sg.Popup('Resultados', 'Ingresaste el siguiente texto: ', texto)
```

## 6 Creamos una ventana en PySimpleGUI

```
[ ]: import PySimpleGUI as sg

sg.Window(title="Hola Mundo!", layout=[[ ]], margins=(300, 350)).read()
```

- **margins**: tamaño de la ventana en pixeles.
- **read()**: devuelve una **tupla** con un **evento** y **los valores** de todos los elementos de entrada en la ventana.

## 7 Sobre los eventos

```
[ ]: import PySimpleGUI as sg

layout = [[sg.Text("Hola Mundo!"), [sg.Button("OK")]]

window = sg.Window("Manejo de eventos", layout, margins=(200, 150))

while True:
    event, values = window.read()

    if event == "OK" or event == sg.WIN_CLOSED:
        break

window.close()
```

- ¿Qué representa **sg.WIN\_CLOSED**?
- ¿Cuándo termina este programa?

## 8 ¿Cómo recuperamos los valores ingresados?

```
[ ]: import PySimpleGUI as sg

layout = [ [sg.Text('Ingresá primer valor'), sg.InputText()],
            [sg.Text('Ingresá segundo valor'), sg.InputText()],
            [sg.Button('Ok'), sg.Button('Cancel')] ]

window = sg.Window("Segunda Demo", layout, margins=(200, 150))
```

```

while True:
    event, values = window.read()

    if event == "Cancel" or event == sg.WIN_CLOSED:
        break
    print('Datos ingresados: ', values)

window.close()

```

- ¿De qué tipo es **values**? ¿Dónde se muestran los datos ingresados?

## 9 Layout: ¿cómo organizamos la UI?

Representa al esquema o diseño de nuestra UI: **cómo se distribuyen los elementos en la UI.**

```

[ ]: layout = [ [sg.Text('Ingresa primer valor'), sg.InputText()],
                [sg.Text('Ingresa segundo valor'), sg.InputText()],
                [sg.Button('Ok'), sg.Button('Cancel')] ]

```

- ¿De qué tipo es la variable layout?
- ¿Qué elementos estamos incluyendo?

**Tarea:** agregar un elemento nuevo donde se muestren los valores ingresados.

## 10 Elementos de la UI

- Acá van algunos disponibles en PySimpleGUI
  - Buttons: File Browse, Folder Browse, Color chooser, Date picker, etc.
  - Checkbox, Radio Button, Listbox
  - Slider, Progress Bar
  - Multi-line Text Input, Scroll-able Output
  - Image, Menu, Frame, Column, Graph, Table
- Referencias: <https://pysimplegui.readthedocs.io/en/latest/call%20reference/#element-and-function-call-reference>

## 11 Agreguemos elementos a nuestra ventana

```

[ ]: import PySimpleGUI as sg

layout = [[sg.Input('Ingresa algo')],
          [sg.Listbox(values=('Item 1', 'Item 2', 'Item 3'))],
          [sg.OK()]]

sg.Window("Elementos básicos", layout, margins=(200, 150)).read()

```

```
[ ]: sg.ChangeLookAndFeel('DarkAmber')

layout = [[sg.Listbox(values=('Item 1', 'Item 2', 'Item 3'),
    ↪background_color='yellow', size=(20,3)),
    [sg.Input('Last input')],
    [sg.ColorChooserButton(" Elegi color")],
    [sg.OK()]] ]
sg.Window("Más elementos ", layout, margins=(200, 150)).read()
```

## 12 PySimpleGUI en GitHub

- En [GitHub](#) hay mucha información y referencias útiles.
- Por ejemplo, hay [varios ejemplos](#) que pueden servir de base para nuestros desarrollos.
- Probemos el [visualizador de csv](#)

##

¡¡Las ventajas del software libre!!

## 13 Hablemos de la Actividad 1 ...

## 14 Seguimos la próxima ...