

TAREA 3

INTERFACES

GRÁFICAS

DE USUARIO

PARTICIPANTES

- Angie Carolina Chitiva Muñoz
- Diego Andrés Santoyo Panche
- Natalia Castaño Miranda
- John Jairo Leon Guerrero
- Leidy Jimena Garcia Bernal



● ¿Qué es Interfaz Gráfica del Usuario, en el contexto del desarrollo de aplicaciones?

Una Interfaz Gráfica de Usuario (GUI) es el medio visual e interactivo que permite a los usuarios comunicarse con una aplicación software o un sistema, a través de elementos gráficos como ventanas, botones, menús, iconos y otros componentes visuales. El propósito de una interfaz gráfica es representar visualmente el código del backend de la manera más entendible posible para el usuario para simplificarle las tareas diarias o comunes.



La GUI es clave porque facilita la interacción entre el usuario y el software de manera intuitiva, eliminando la necesidad de conocer comandos de texto o lenguajes técnicos de programación. Se enfoca en mejorar la experiencia del usuario al hacer las aplicaciones más accesibles y fáciles de usar, lo cual es crucial para el éxito de cualquier software.

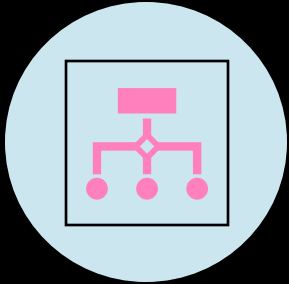


Que es Tkinter

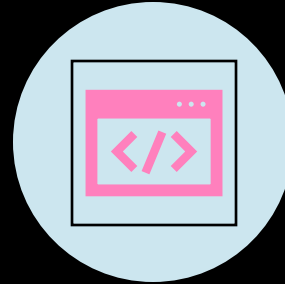
Es una de las muchas librerías para Python, y es utilizada para la creación de interfaces gráficas. Es parte de la biblioteca estándar de Python por lo que se incluye con su instalación. Esta librería permite a los desarrolladores crear aplicaciones de escritorio con elementos como botones, cuadros de texto, menús, etc.



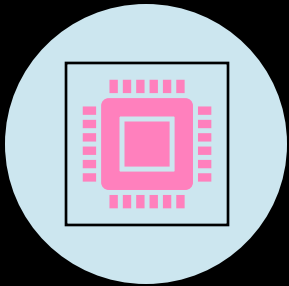
● ¿Qué son los Widgets?



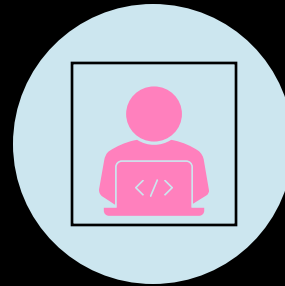
Los widgets son los componentes básicos de una GUI que permiten a los usuarios interactuar con la aplicación.



Tkinter ofrece una gran variedad de widgets, como botones, cuadros de texto, menús desplegables y mucho más.



Estos widgets se pueden personalizar para adaptarse a las necesidades específicas de cada aplicación.



Los widgets de Tkinter son fáciles de implementar y permiten a los desarrolladores crear interfaces intuitivas.

● Widgets de visualización



Label: Muestra texto, imágenes u otros contenidos en la interfaz.

Text: Permite a los usuarios editar y visualizar texto de manera más avanzada.

Canvas: Proporciona un área gráfica donde se pueden dibujar y manipular formas, imágenes y otros elementos visuales.

● Widgets de entrada



Entry: Permite a los usuarios ingresar y editar texto en un campo de entrada.

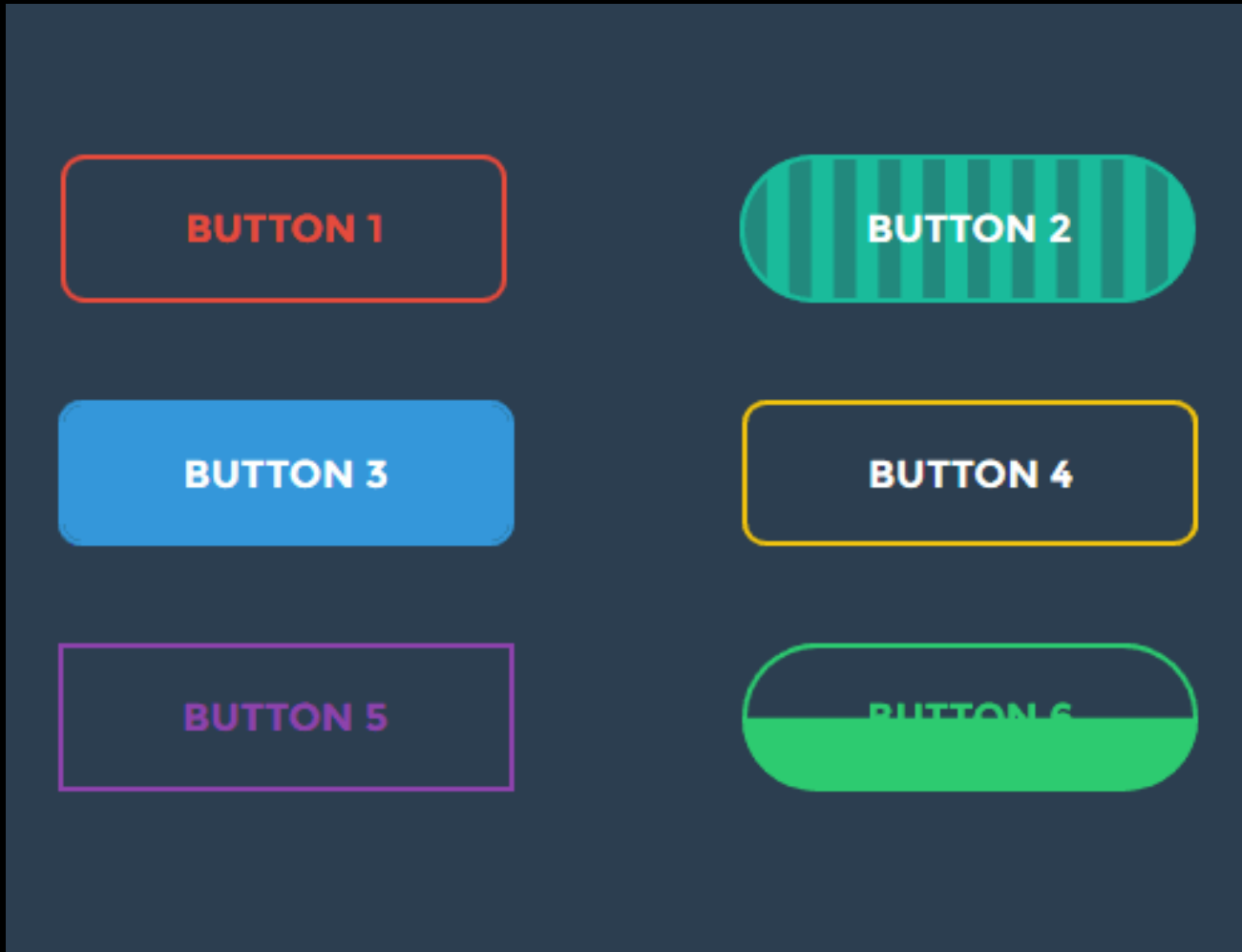


Spinbox: Ofrece un control de entrada numérica con botones para incrementar y decrementar el valor.



Listbox: muestra una lista de elementos seleccionables por el usuario

● Widgets de control



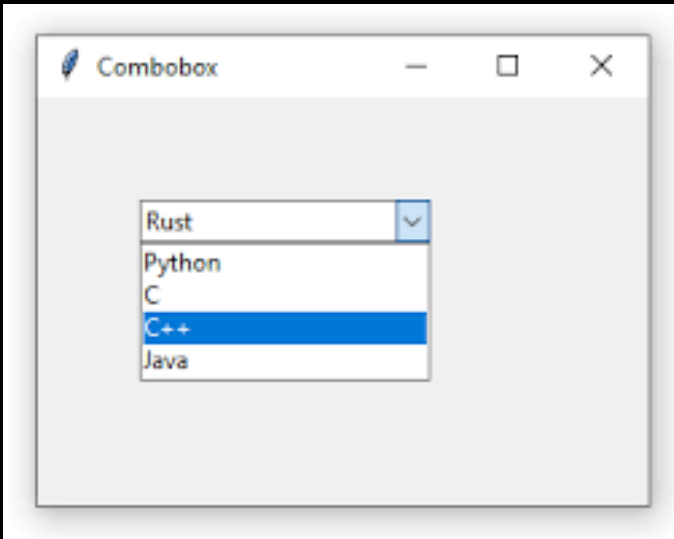
Button: representa un botón que se puede hacer clic para ejecutar una acción específica.

Checkbutton: permite a los usuarios seleccionar o deseleccionar una opción de una lista.

Radiobutton: ofrece un conjunto de opciones exclusivas, donde solo se puede seleccionar una a la vez.



4. Como se manejan los eventos de tkinter



En tkinter, el manejo de eventos es tan simple como agregar un comando, que convertiremos en una función. Si bien esta función que creamos es una función básica de una línea que simplemente llama a otra función, podemos ver cómo podemos crear más adelante funciones más complejas para nuestros eventos.

los eventos se manejan a través de un sistema de "bindings", que permite asociar funciones (o métodos) a ciertos eventos, como clics del ratón, pulsaciones de teclas, etc.





¿Qué es PyQt?

PyQt es un conjunto de bibliotecas de Python que proporciona una interfaz para el framework de desarrollo de aplicaciones Qt, que es muy popular para crear interfaces gráficas de usuario (GUI). Qt es un framework en C++ y es conocido por su robustez, flexibilidad y facilidad de uso.

Características de PyQt

Interfaz gráfica rica: PyQt permite crear aplicaciones con interfaces gráficas complejas y altamente personalizables, incluyendo botones, menús, cuadros de diálogo, y más.

Multiplataforma: Las aplicaciones desarrolladas con PyQt pueden ejecutarse en diferentes sistemas operativos, como Windows, macOS y Linux, sin cambios en el código.

Módulos extensivos: PyQt incluye varios módulos que abarcan diferentes funcionalidades, como gráficos, acceso a bases de datos, redes y más.



Diseñador gráfico: PyQt viene con una herramienta llamada Qt Designer, que permite diseñar visualmente las interfaces y luego generar código Python.

Soporte para programación orientada a objetos: Permite estructurar aplicaciones de manera modular y reutilizable, facilitando la organización del código.

Signal y Slot: Este patrón permite la comunicación entre objetos en la interfaz, lo que facilita la respuesta a eventos de manera eficiente.

Versiones de PyQt

Existen varias versiones de PyQt, siendo las más conocidas:

PyQt5: Basada en Qt 5, es muy popular y ampliamente utilizada.

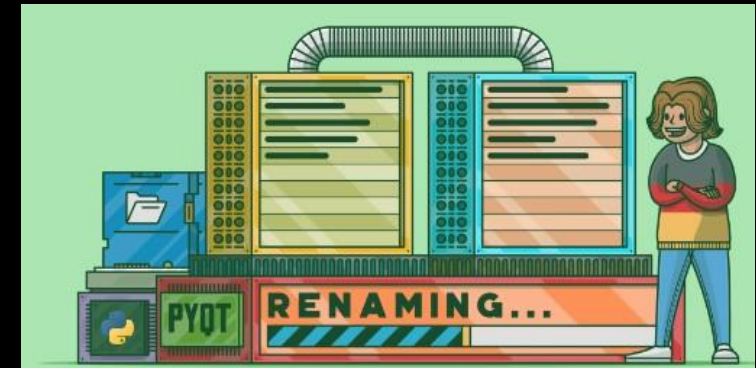
PyQt6: La versión más reciente, basada en Qt 6, con mejoras y nuevas características.

PyQt es una herramienta poderosa para desarrollar aplicaciones de escritorio en Python con interfaces gráficas sofisticadas. Es ideal para desarrolladores que buscan crear aplicaciones multiplataforma con una apariencia profesional.

Natalia Castaño Miranda

IMÁGENES PYQT

N A T A L I A
C A S T A Ñ O



● Referencias

- Grayson, J. (2000). *Tkinter for Python: A GUI programming guide for Python*. New York: Wiley.
- Brey, B. (2018). PyQt: The Complete Guide. *Python Software Foundation*. Retrieved from <https://www.python.org/doc/essays/pyqt/>
- Rivero, J. (2020). *PyQt Documentation*. Retrieved from <https://www.riverbankcomputing.com/static/Docs/PyQt5/>

