

INSTRUMENTACION VIRTUAL

SCROLLBAR Y SLIDER

Los ScrollBar son muy usados para desplegar información oculta en documentos muy grandes, donde lo que estamos leyendo u observando supera el campo de visibilidad, en este caso los procesadores de texto o visores de texto tienen a los costados ScrollBars para poder avanzar o retroceder en el documento que estemos trabajando. El ScrollBar está compuesto por un Slider y aparte agrega otras funcionalidades como flechas en los extremos y un área detrás del Slider que al dar click en esa área podremos movernos más rápidamente en el documento.

Los sliders son más usados para desplegar valor enteros que van desde un rango mínimo a un rango máximo.

Los ScrollBar como ya mencionamos se usan mucho para moverse dentro de un documento o ventana, dentro de las librerías de PySide6 podemos encontrar dos tipos de widgets de este tipo son el HorizontalScrollBar y el VerticalScrollBar, estos dos widgets son instancias de la clase QScrollBar. El ScrollBar tiene los siguientes controles.

- **Slider handle.** Este control es la parte movable del ScrollBar nos permite desplazarnos a cualquier parte del documento o imagen rápidamente.
- **Scroll arrows.** Estas son las flechas que están a los costados del ScrollBar y su función también es ayudarnos a desplazarnos en un documento o imagen. Al usar los scroll arrows podremos observar como el slider del ScrollBar también va cambiando ya sea incrementándose o decrementándose.
- **Page control.** El page control es la parte del fondo del ScrollBar y es el área donde arrastramos el slider de arriba a abajo o de izquierda a derecha. La cantidad de movimientos del slider puede cambiarse en la propiedad pageStep. El pageStep indica cuanto se va a mover el slider del ScrollBar cuando el usuario presiona las teclas de avanzar página o retroceder página y puede cambiarse con el método setPageStep().

METODOS

El método que nos sirve para establecer o extraer el valor del ScrollBar es el método value(), aparte del método value tenemos más métodos disponibles para el ScrollBar los cuales son.

- **setValue().** Este método asigna el valor al ScrollBar.
- **minimum().** Este método retorna el valor mínimo del ScrollBar.
- **maximum().** Este método retorna el valor máximo del ScrollBar.
- **setMinimum().** Este método asigna el valor mínimo al ScrollBar.
- **setMaximum().** Este método asigna el valor máximo al ScrollBar.

SEÑALES

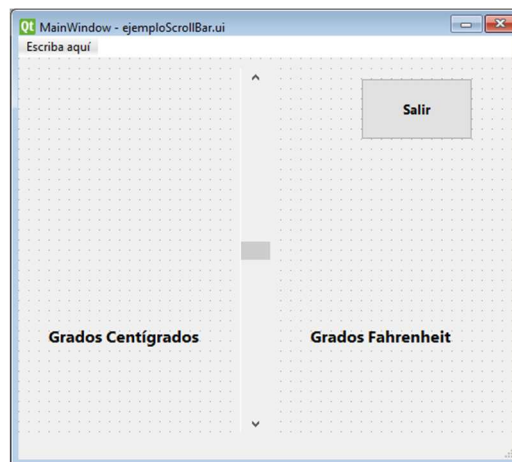
Las señales emitidas por el ScrollBar son las siguientes.

- **valueChanged()**. Esta señal es emitida cuando el valor del ScrollBar es cambiando, esto es cuando el slider es movido.
- **sliderPressed()**. Esta señal es emitida cuando el usuario empieza a arrastrar el slider del ScrollBar.
- **sliderMoved()**. Esta señal es emitida cuando el usuario arrastra el slider del ScrollBar.
- **sliderReleased()**. Esta señal es emitida cuando el usuario suelta el slider del ScrollBar.
- **actionTriggered()**. Esta señal es emitida cuando el ScrollBar es cambiando por interacción del usuario.

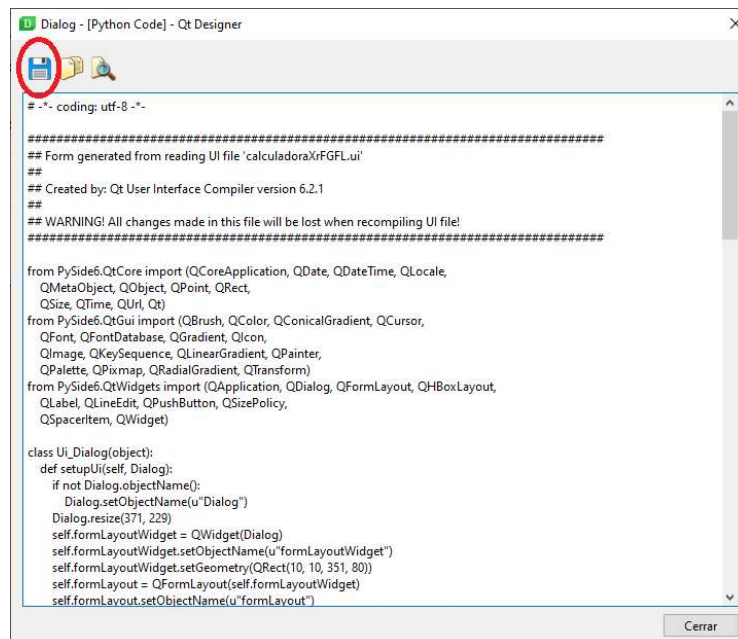
Los Sliders heredan de la clase QSlider, existen dos tipos de widgets de este tipo son el HorizontalSlider y el VerticalSlider. Los Sliders tiene métodos similares a los del ScrollBar y también generan señales como valueChanged(), sliderPressed(), sliderMoved(), sliderReleased().

Para este tema se van a crear dos ejercicios uno con ScrollBar y otro con Slider. Primero realizaremos el ejercicio con ScrollBar el cual es una aplicación para convertir de grados Centígrados a grados Fahrenheit, los pasos son los siguientes.

1. Abrimos el Designer y creamos un dialogo sin botones y agregamos cuatro Labels, un ScrollBar y un PushButton.
2. Empezaremos por cambiar las propiedades text de los Labels, dos tendrán la leyenda “Grados Centígrados”, la otra “Grados Fahrenheit” y las últimas dos no tendrán nada en su propiedad text. Las dos Labels sin texto cambiamos su nombre en la propiedad *objectName* por lblCenti y lblFahr.
3. Lo siguiente es modificar las propiedades del ScrollBar la primera es la propiedad minimum y le asignamos el valor de -100 y la segunda es maximum y le asignamos el valor 100. Lo siguiente es cambiar su nombre en *objectName* por sbTemp.
4. Por último, al botón de salir modificamos su propiedad *objectName* para cambiar su nombre por pbSalir. Después vamos a programar la acción del botón entrando en el modo de edición de señales y slots. Colocaremos el puntero del mouse sobre el botón este cambiará a color rojo y damos click en el botón y sin soltar arrastramos y nos aparecerá una línea roja con un símbolo de tierra y en cualquier zona cercana de la ventana soltamos el raton y nos aparecerá la ventana de configuración de conexión. En la parte izquierda seleccionamos la señal clicked() y del lado derecho seleccionamos el metodo close(). En caso de no aparecer este metodo hay que seleccionar el checkbox de la parte inferior de la ventana que dice “*show signals and slots inherited from QWidget*” damos click en el botón Ok para aceptar los cambios, volvemos al modo *edit widgets* y guardamos la interfaz con el nombre scrollBar.ui. La interfaz se vera de la siguiente manera.



- Después de crear la interfaz convertimos el archivo “ui” en código Python. Primer vamos al menú Formulario/View Python Code...



6. Después creamos un nuevo proyecto en PyCharm o PyScripter y escribimos el siguiente código.

```
import sys
from PySide6.QtWidgets import QApplication, QMainWindow
from scrollBar import Ui_MainWindow

class MiFormulario(QMainWindow, Ui_MainWindow):
    def __init__(self):
        super().__init__()
        self.setupUi(self)

    self.sbTemp.valueChanged.connect(self.calcularTemp)
    self.pbSalir.clicked.connect(self.salir)

    fahr = 32 + 1.8 * self.sbTemp.value()
    self.lblCenti.setText(str(self.sbTemp.value()) + "
°C")
    self.lblFahr.setText(f"{fahr:.2f} °F")

    def calcularTemp(self):
        fahr = 32 + 1.8 * self.sbTemp.value()
        self.lblCenti.setText(str(self.sbTemp.value()) + "
°C")
        self.lblFahr.setText(f"{fahr:.2f} °F")

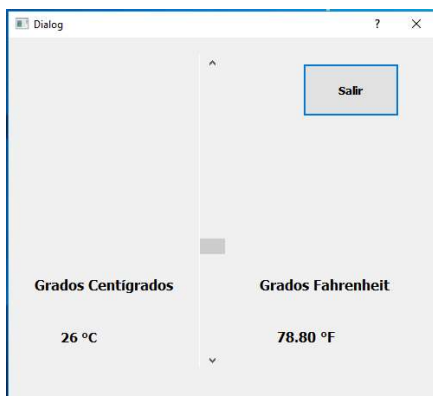
    def salir(self):
        self.close()

if __name__ == "__main__":
    app = QApplication(sys.argv)
    w = MiFormulario()
    w.show()
    sys.exit(app.exec())
```

La salida del programa es la siguiente.



Al mover el ScrollBar veremos que cambian los valores de los grados centígrados y los grados Fahrenheit como se ve a continuación.



SLIDER

El siguiente ejemplo es usar un el widget Slider esta interfaz es más complicada que las que se han visto hasta este momento, vamos a combinar varios widgets y la idea es modificar el color de fondo de un Label variando las componentes R, G y B de la propiedad *styleSheet*, la interfaz contará con dos RadioButtons con los que podremos seleccionar si cambiar el color de fondo o el color del texto que tendrá el Label. La interfaz quedaría de la siguiente manera.



Esta interfaz contiene más Widgets, realizaremos los siguientes pasos para elaborarla.

1. Abrir el Designer y seleccionar un dialogo sin borones y arrastrar dentro de la ventana cuatro Labels, dos RadioButtons, tres Sliders y un PushButton.
2. Seleccionamos el Label y en su propiedad *text* escribimos “Instrumentación Virtual”, modificamos la propiedad *Font* y modificamos el tamaño de la letra para que se extienda a lo largo del Label, y seleccionamos el texto para que aparezca en negrita. Por último, modificamos su propiedad *objectName* y le asignamos el nombre *lblColor*.
3. Lo siguiente es agrupar los RadioButtons dentro de un *HorizontalLayout*, después a un RadioButton hay que cambiar la propiedad *text* por “Fondo” y el siguiente RadioButton su propiedad *text* también la cambiamos y ponemos “Texto”, ya por último cambiamos sus propiedades *objectName* por *rbFondo* y *rbTexto* respectivamente.
4. Lo siguiente es modificar las propiedades de los tres Slider, primero modificamos la propiedad *maximum* de cada Slider y ponemos 255 la propiedad *minimum* la dejamos en cero, después a cada slider modificamos su propiedad *objectName* y ponemos *hsRojo*, *hsVerde*, *hsAzul* respectivamente.
5. Después hay que modificar los tres Labels que desplegaran el valor de los Sliders. Primero modificamos su propiedad *text* y borramos su contenido, después modificaremos la propiedad *objectName* de cada uno por *lblRojo*, *lblVerde*, *lblAzul*.
6. Por último, modificamos el PushButton cambiamos su propiedad *objectName* por *pbSalir*, después programamos la función de salir de la misma forma que lo programamos con el *ScrollBar* en el modo editar señales y slots.

7. Guardamos la interfaz con el nombre color.ui. La interfaz se vera de la siguiente forma.



8. Abrimos el PyCharm o PyScripter y creamos un proyecto nuevo y escribimos el siguiente código.

```
import sys
from PySide6.QtWidgets import QApplication, QMainWindow
from color import Ui_MainWindow

class MiFormulario(QMainWindow, Ui_MainWindow):
    bColor = [None]*3
    tColor = [None]*3

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        self.hsRojo.valueChanged.connect(self.cambiarColor)
        self.hsVerde.valueChanged.connect(self.cambiarColor)
        self.hsAzul.valueChanged.connect(self.cambiarColor)
        self.pbSalir.clicked.connect(self.salir)
        self.hsRojo.setValue(127)
        self.hsVerde.setValue(127)
        self.hsAzul.setValue(127)
        for i in range(3):
            self.bColor[i] = 127
```



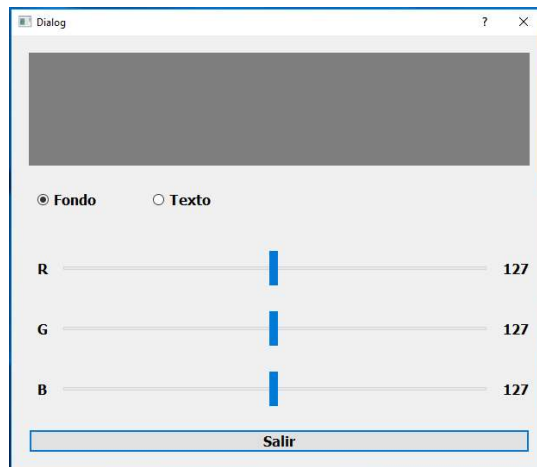
```
        self.tColor[i] = 127
    self.rbFondo.setChecked(True)
    self.cambiarColor()

def cambiarColor(self):
    if self.rbFondo.isChecked():
        self.bColor[0] = self.hsRojo.value()
        self.bColor[1] = self.hsVerde.value()
        self.bColor[2] = self.hsAzul.value()
        self.lblRojo.setText(f"{self.bColor[0]}")
        self.lblVerde.setText(f"{self.bColor[1]}")
        self.lblAzul.setText(f"{self.bColor[2]}")
        str1 = f"background-color: rgb({self.bColor[0]},
{self.bColor[1]}, {self.bColor[2]}); " \
            f"color: rgb({self.tColor[0]},
{self.tColor[1]}, {self.tColor[2]})"
        self.lblColor.setStyleSheet(str1)
    elif self.rbTexto.isChecked():
        self.tColor[0] = self.hsRojo.value()
        self.tColor[1] = self.hsVerde.value()
        self.tColor[2] = self.hsAzul.value()
        self.lblRojo.setText(f"{self.tColor[0]}")
        self.lblVerde.setText(f"{self.tColor[1]}")
        self.lblAzul.setText(f"{self.tColor[2]}")
        str2 = f"background-color: rgb({self.bColor[0]},
{self.bColor[1]}, {self.bColor[2]}); " \
            f"color: rgb({self.tColor[0]},
{self.tColor[1]}, {self.tColor[2]})"
        self.lblColor.setStyleSheet(str2)

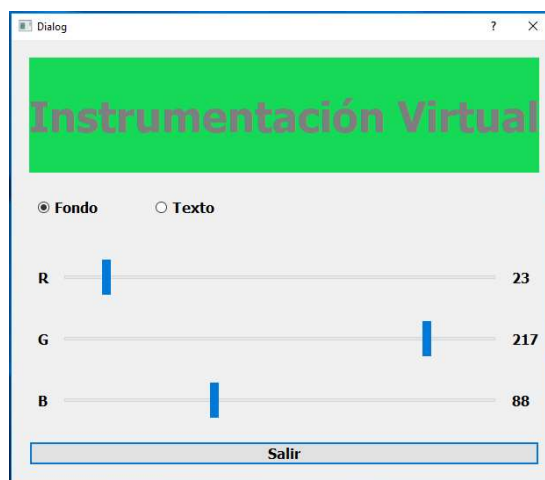
def salir(self):
    self.close()

if __name__ == "__main__":
    app = QApplication(sys.argv)
    w = MiFormulario()
    w.show()
    sys.exit(app.exec())
```

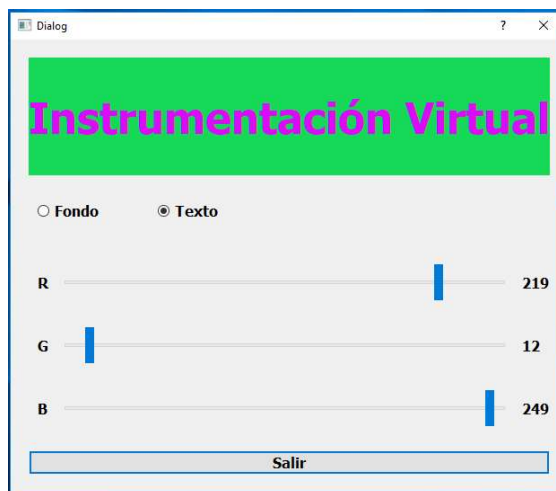

La salida de este programa es la siguiente.



Al modificar los Sliders con el RadioButton Fondo seleccionado veremos como el fondo va cambiando como se ve en la siguiente imagen.



Si seleccionamos el RadioButton que dice Texto y variamos la posición de los Sliders veremos cómo el texto cambia de color como se ve en la siguiente imagen.



La forma de cambiar el color en PySide6 es a través de código CSS en la propiedad styleSheet podemos introducir lo siguiente.

```
background-color: red
```

Veremos que el fondo se pondrá de color rojo y si debajo de esa instrucción ponemos la siguiente

```
color: white
```

Veremos que el texto cambiara de color. Hay otra forma de cambiar el color sin usar las constantes usadas por CSS y es con la instrucción `rgb(R, G, B)` con esta instrucción podemos cambiar el color cambiando sus componentes Rojo, Verde, Azul. Podremos hacerlo de la siguiente forma.

```
Background-color: rgb(100, 75, 25)
```

```
color: rgb(30, 200, 170)
```

Esta última forma es la que utilizamos para cambiar el color tanto del texto como el de fondo. Cuando la señal `valueChanged` es emitida se tomó el valor de cada Slider y se crea una cadena de caracteres con las dos instrucciones de arriba y cambiamos su componente Rojo, Verde y Azul que tomamos de los Sliders y aparte se desplegará el valor en los Label que están a cada lado de los Sliders.