



Desarrollo de interfaces

Apuntes sobre PySide6

| | |
|---|----------|
| QRadioButton (Botón de radio). | 3 |
| Webgrafía: | 3 |
| QTabWidget (Pestañas con contenido). | 4 |
| Webgrafía: | 5 |
| QProgressBar (Progreso manual). | 6 |
| Webgrafía: | 6 |
| QDateTimeEdit (Fecha y hora) | 7 |
| Webgrafía: | 7 |
| QSlider (Barra deslizable) | 8 |
| Webgrafía: | 8 |
| QDial (Control giratorio) | 9 |
| Webgrafía: | 9 |

QRadioButton (Botón de radio).

- **Descripción:**

Con el widget QRadioButton podemos crear un botón de radio, cuya función es la de actuar como un botón de encendido o apagado, pero que usado en grupo sólo permite la activación de un botón de radio del grupo.

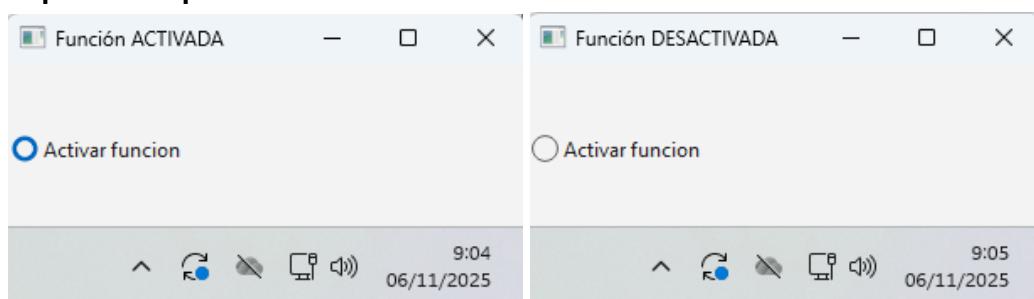
- **Principales métodos:**

- **.setChecked(bool)**. Permite cambiar el estado original del botón de radio.
- **.isChecked()**. Comprueba si está marcado o no.
- **.text()**. Devuelve el valor de texto del botón de radio.

- **Señales más frecuentes: (comunes a cualquier tipo de botón)**

- **clicked()**. Se activa al hacer clic en el botón de radio.
- **pressed()**. Se activa al presionar en el botón de radio.
- **released()**. Se activa al soltar después de pulsar en el botón de radio.
- **toggled()**. Se activa al cambiar el estado del botón de radio.

- **Capturas de pantalla del funcionamiento.**



Webgrafía:

Doc.qt.io. PySide6.QtWidgets.QAbstractButton. Recuperado de:
<https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QAbstractButton.html#PySide6.QtWidgets.QAbstractButton> [Consultado el 06/11/2025]

QTabWidget (Pestañas con contenido).

- **Descripción:**

Con el widget QTabWidget podemos crear un conjunto de pestañas, cada cual nos mostrará un widget diferente, según sea confeccionado.

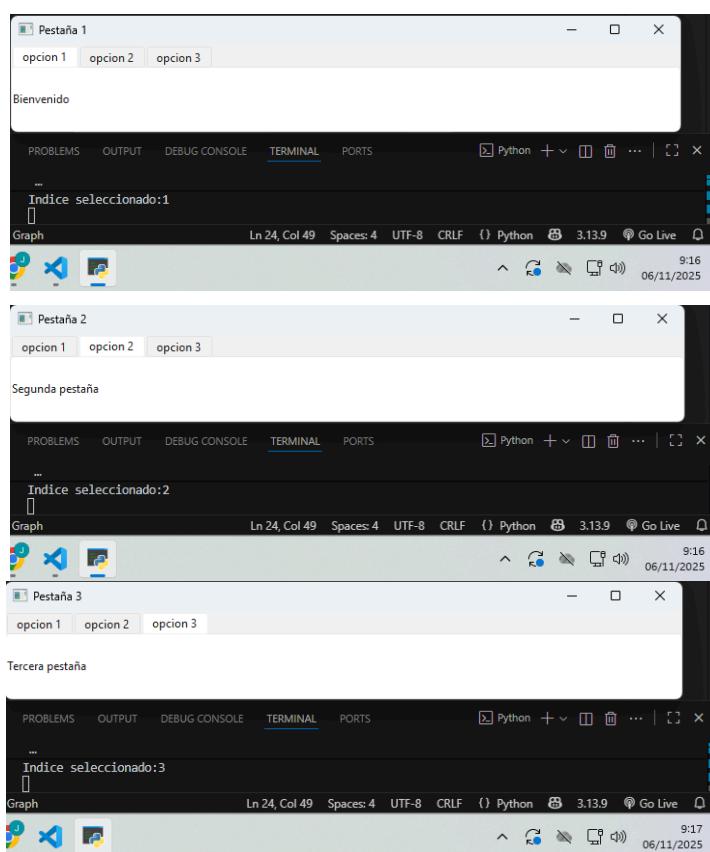
- **Principales métodos:**

- **.addTab(widget, str).** Sirve para añadir pestañas al widget.
- **.setCurrentIndex(int).** Permite cambiar la pestaña al índice proporcionado.
- **.count().** Devuelve la cuenta de la cantidad de pestañas que hay.
- **setTabText(index, text).** Cambia el título de la pestaña del índice proporcionado.

- **Señales más frecuentes:**

- **currentChanged().** Se activa al cambiar de pestaña
- **tabBarClicked().** Se activa al hacer clic en una pestaña
- **tabBarDoubleClicked().** Se activa al hacer doble clic sobre la pestaña
- **tabCloseRequested().** Se activa al intentar cerrar la pestaña(si está la opción)

- **Capturas de pantalla del funcionamiento.**



Webgrafía:

Doc.qt.io..PySide6.QtWidgets.QTabWidget. Recuperado de:
<https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QTabWidget.html#PySide6.QtWidgets.QTabWidget.tabBarClicked> [Consultado el 06/11/2025]

QProgressBar (Progreso manual).

- **Descripción:**

Con el widget QProgressBar podemos crear una barra de progreso con la que podemos mostrar el avance de un proceso.

- **Principales métodos:**

- **.setValue(int)**. Nos permite darle un valor a la barra de progreso.
- **.setRange(min, max)**. Con este método podemos delimitar los límites mínimo y máximo de la barra de progreso.
- **.reset()**. Con este método podemos reiniciar la barra de progreso a su valor inicial.

- **Señales más frecuentes:**

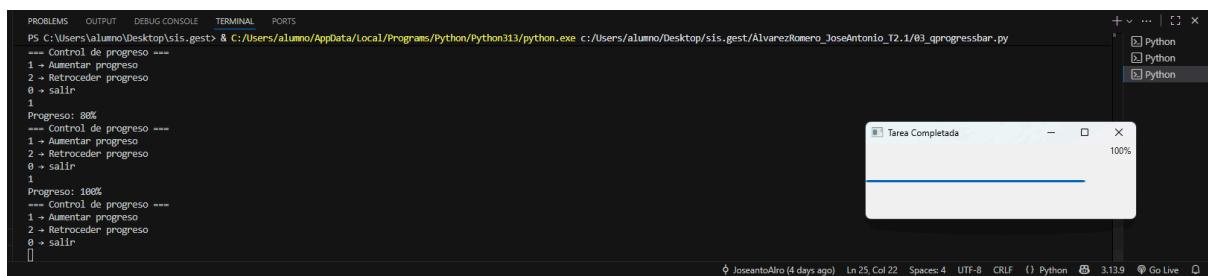
- No tiene señales propias, pero reacciona a las de otros widgets.

- **Capturas de pantalla del funcionamiento.**



A screenshot of a terminal window in a code editor. The terminal shows a Python script running. It displays a menu with options 1, 2, and 0, followed by two progress bar updates at 40% and 60%. A small modal dialog titled "Progreso actual: 60%" is visible in the foreground, showing a blue progress bar at the 60% mark.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\alumno\Desktop\sis.gest> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/sis.gest/AlvarezRomero_JoseAntonio_T2.1/03_qprogressbar.py
--- Control de progreso ---
1 -> Aumentar progreso
2 -> Retroceder progreso
0 -> salir
1
Progreso: 40%
--- Control de progreso ---
1 -> Aumentar progreso
2 -> Retroceder progreso
0 -> salir
1
Progreso: 60%
--- Control de progreso ---
1 -> Aumentar progreso
2 -> Retroceder progreso
0 -> salir
```



A screenshot of a terminal window in a code editor. The terminal shows a Python script running. It displays a menu with options 1, 2, and 0, followed by three progress bar updates at 80%, 100%, and 100%. A small modal dialog titled "Tarea Completada" is visible in the foreground, showing a blue progress bar at the 100% mark.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\alumno\Desktop\sis.gest> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/sis.gest/AlvarezRomero_JoseAntonio_T2.1/03_qprogressbar.py
--- Control de progreso ---
1 -> Aumentar progreso
2 -> Retroceder progreso
0 -> salir
1
Progreso: 80%
--- Control de progreso ---
1 -> Aumentar progreso
2 -> Retroceder progreso
0 -> salir
1
Progreso: 100%
--- Control de progreso ---
1 -> Aumentar progreso
2 -> Retroceder progreso
0 -> salir
```

Webgrafía:

Doc.qt.io. PySide6.QtWidgets.QProgressBar. Recuperado de:

<https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QProgressBar.html#id8>

[Consultado el 06/11/2025]

QDateTimeEdit (Fecha y hora)

- **Descripción:**

Con QDateTimeEdit podemos crear un campo modificable para poder introducir cualquier fecha que queramos.

- **Principales métodos:**

- **.setDateTime(QDateTime).** Con este método podemos determinar el valor inicial de nuestro campo QDateTimeEdit
- **.dateTime().** Devuelve el valor de la fecha actual introducida.
- **.setDisplayFormat(str).** Sirve para darle un formato específico a la fecha.

- **Señales más frecuentes:**

- **.dateChanged(QDate).** Señal que se activa cada vez que la fecha es cambiada.
- **.timeChanged(QTime).** Señal que se activa cada vez que la hora es cambiada.
- **.dateTimeChanged(QDateTime).** Señal que se activa cada vez que tanto la fecha como la hora es cambiada.

- **Capturas de pantalla del funcionamiento.**

```
22 print('Fecha elegida: ' + fecha.toString('dd/MM/yyyy hh:mm'))
23
24 app = QApplication()
25 window = Ventana()
26 window.show()
27 app.exec()
```

```
3
4 app = QApplication()
5 window = Ventana()
6 window.show()
7 app.exec()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\alumno\Desktop\sis.gest> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/sis.gest/ÁlvarezRomero_JoseAntonio_T2.1/04_qdatetimeedit.py
Fecha elegida: Wednesday, 12 de November de 2025 08:47
C:\Users\alumno\Desktop\sis.gest> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/sis.gest/ÁlvarezRomero_JoseAntonio_T2.1/04_qdatetimeedit.py
Fecha elegida: Friday, 12 de September de 2025 08:47

Webgrafía:

Doc.qt.io. PySide6.QtWidgets.QDateTimeEdit. Recuperado de:
<https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QDateTimeEdit.html#id4>
[Consultado el 11/11/2025]

QSlider (Barra deslizable)

- **Descripción:**

Con Qslider podemos crear una barra deslizable, como una barra para seleccionar el nivel de brillo o el nivel de volumen.

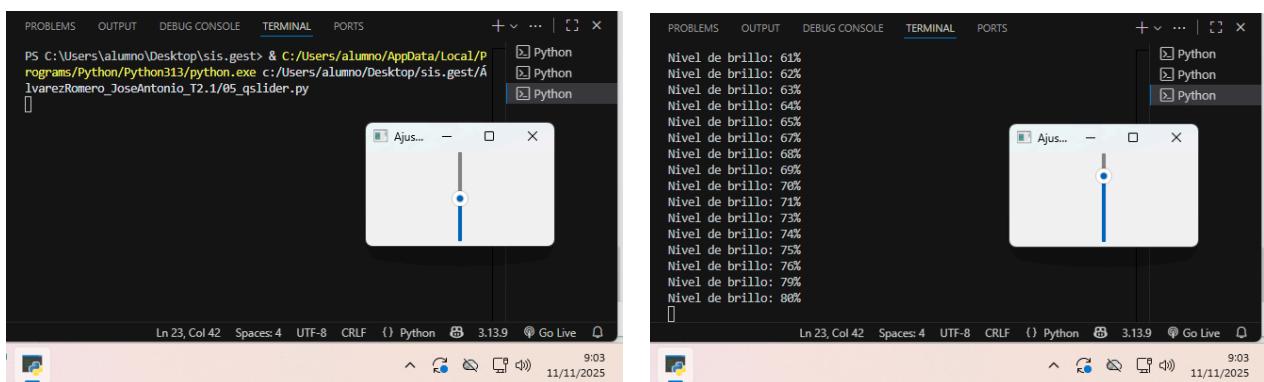
- **Principales métodos:**

- **.setMinimum()**. Delimita el valor mínimo que alcanza la barra.
- **.setMaximum()**. Delimita el valor máximo que alcanza la barra.
- **.setValue()**. Delimita el valor inicial de la barra.
- **.value()**. Devuelve el valor de la barra.

- **Señales más frecuentes: (comunes a cualquier tipo de botón)**

- **.valueChanged(int)**. Envía una señal cada vez que varía el valor de la barra, enviando el valor de dicha barra como int.

- **Capturas de pantalla del funcionamiento.**



Webgrafía:

Doc.qt.io. PySide6.QtWidgets.QSlider. Recuperado de:

<https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QSlider.html#PySide6.QtWidgets.QSlider> [Consultado el 11/11/2025]

QDial (Control giratorio)

- **Descripción:**

Con QSlider podemos crear un control giratorio, como un dial para seleccionar el nivel de volumen de un amplificador.

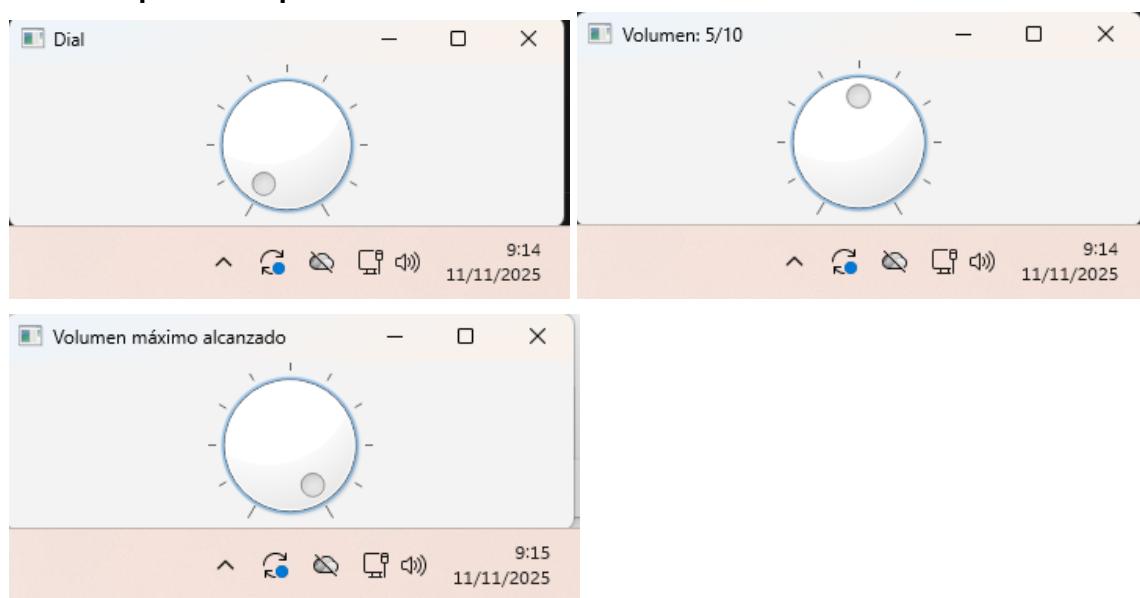
- **Principales métodos:**

- **.setNotchesVisible(True).** Hace visibles las marcas del dial.
- **.setRange(min, max).** Delimita los valores mínimos y máximos que puede alcanzar el dial.
- **.setValue().** Delimita el valor inicial del dial.
- **.value().** Devuelve el valor del dial.

- **Señales más frecuentes: (comunes a cualquier tipo de botón)**

- **.valueChanged(int).** Envía una señal cada vez que varía el valor del dial, enviando el valor de dicho dial como int.

- **Capturas de pantalla del funcionamiento.**



Webgrafía:

[Doc.qt.io. PySide6.QtWidgets.QSlider. Recuperado de:](https://doc.qt.io/qtforpython-6/PySide6/QtWidgets.QSlider.html#PySide6.QtWidgets.QSlider)

<https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/QDial.html#PySide6.QtWidgets.QDial> [Consultado el 11/11/2025]