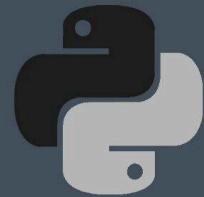


DESARROLLO DE INTERFACES



José Carlos Mora

11/11/2025

2ºDAM

Tarea 1.2

Índice

QRadioButton.....	3
QTabWidget.....	4
QProgresBar.....	6
QDateTimeEdit.....	8
QSlider.....	9
QDial.....	11

QRadioButton

- **Descripción general:** es un tipo de widget de interfaz gráfica de usuario que permite al usuario elegir una única opción entre varias opciones ya predefinidas
- **Principales métodos:**
 - **.setChecked(bool):** Es un boolean que establece en qué estado va a estar el botón, es decir, si es True, el botón estaría seleccionado
 - **.isChecked():** Devuelve en qué estado está el botón mediante True o False
 - **.text():** Muestra el texto visible de las opciones
 - **.setText():** Cambia el texto de las opciones
- **Señales (slots) más frecuentes y su uso**
 - **toggled(bool):** Devuelve cuando el estado del botón cambia
 - **pressed():** Devuelve cuando el usuario presiona el botón sin soltarlo
 - **released():** Devuelve cuando el usuario suelta el botón presionado

- **Captura de pantalla de la ejecución del ejercicio propuesto**

The screenshot shows a Python development environment with the following details:

- IDE Interface:** The top bar includes "Go", "Run", "Terminal", and "Help". The title bar says "Proyectos Python".
- Project Explorer:** Shows files like "ejercicio01.py", "ejercicio02.py", etc., under "Desarrollo de Interfaces > Tema 2 > Tarea 2.1".
- Code Editor:** Displays Python code for a Qt application. The code initializes a QMainWindow and sets its window title based on a checked QRadioButton state.
- Terminal:** Shows the command "PS C:\Users\alumno\Desktop\Python313\Proyectos Python> & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DAM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio01.py"".
- Output Window:** Shows the application's window titled "Función ACTIVADA" with a radio button labeled "Activar función".
- Bottom Bar:** Includes tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "PORTS". It also shows file navigation icons and a search bar.
- Status Bar:** Displays "Ln 15, Col 56", "Spaces: 4", "UTF-8", "CRLF", "Python 3.13.9", and the date "11/11/2025".

- **Webgrafía:**

- <https://www.tutorialspoint.com/>
- <https://doc.qt.io/qt-6/qtdesigner-manual.html>
- <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html>
- Apuntes del tema

QTabWidget

- **Descripción general:** Es un widget que permite organizar la interfaz de un usuario en pestañas, con la que cada pestaña contiene un widget independiente

- **Principales métodos:**

- **addTab(widget, “Título”):** Añade una nueva pestaña indicando el título que tendrá y el widget

- **.setCurrentIndex(int):** Cambia la pestaña que está visible actualmente según el número que le indiques
 - **.count():** Devuelve el número total de pestañas que hay
 - **.removeTab():** Elimina la pestaña en la posición que indiques
 - **.currentIndex():** Devuelve la posición de la pestaña en la que se encuentra actualmente
- **Señales (slots) más frecuentes y su uso**
 - **currentChanged(int):** Devuelve al cambiar de pestaña
 - **tabCloseRequested(int):** Devuelve cuando el usuario pide cerrar una pestaña
 - **tabBarClicked(int):** Devuelve cuando se hace click en una pestaña
 - **tabBarDoubleClicked(int):** Devuelve cuando se hace doble click en una pestaña

● Captura de pantalla de la ejecución del ejercicio propuesto

The screenshot shows the PySide6 IDE interface. On the left, the code editor displays Python code for a QTabWidget application. On the right, a terminal window shows the command to run the script and its output. A small preview window in the center shows the application's window with three tabs labeled "Pestaña 1", "Pestaña 2", and "Pestaña 3".

```

Go Run Terminal Help ← → Proyectos Python
Ejercicio02.py ✘ ejercicio03.py ✘ ejercicio04.py ✘ ejercicio05.py ✘ ejercicio06.py ✘ ⌂ ⌂ ⌂ ... Python + - ⊖ ⊖ ⊖ ⊖ ⊖ ...
Desarrollo de Interfaces > Tema 2 > Tarea 2.1 > Ejercicio02.py > ...
1 # José Carlos Mora 2º DAM
2
3 from PySide6.QtWidgets import QApplication, QMainWindow, QTabWidget, QLabel
4
5 class MainWindow(QMainWindow):
6     def __init__(self):
7         super().__init__()
8         self.setWindowTitle("Pestaña 1")
9
10        self.tabs = QTabWidget()
11
12        self.tabs.addTab(QLabel("Bienvenido"), "Pestaña 1")
13        self.tabs.addTab(QLabel("Segunda pestaña"), "Pestaña 2")
14        self.tabs.addTab(QLabel("Tercera pestaña"), "Pestaña 3")
15
16        self.titulos = ["Pestaña 1", "Pestaña 2", "Pestaña 3"]
17
18        self.tabs.currentChanged.connect(self.cambiar_titulo)
19
20        self.setCentralWidget(self.tabs)
21
22    def cambiar_titulo(self, indice):
23        print(indice)
24        self.setWindowTitle(self.titulos[indice])
25
26
27 app = QApplication([])
28 window = MainWindow()
29 window.show()
30 app.exec()
31
32
33 PS C:\Users\alumno\Desktop\DM\Proyectos Python> & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio01.py"
34 PS C:\Users\alumno\Desktop\DM\Proyectos Python> & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio06.py"
35 ● PS C:\Users\alumno\Desktop\DM\Proyectos Python> & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio02.py"
36
37

```

- **Webgrafía:**

- <https://www.tutorialspoint.com/>
- <https://doc.qt.io/qt-6/qtdesigner-manual.html>
- <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html>
- Apuntes del tema

QProgressBar

- **Descripción general:** Permite mostrar el progreso de una tarea en forma de barra

- **Principales métodos:**

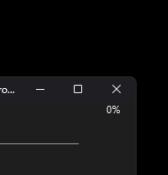
- **.setValue(int):** Define el valor actual de la barra
- **.setRange(min, max):** Define los límites de la barra
- **.reset():** Reinicia la barra
- **.setMinimum(int):** Fija el valor mínimo de la barra
- **.setTextVisible(bool):** Muestra u oculta el texto que tenga que la barra
- **.setStyleSheet(str):** Sirve para personalizar el diseño de la barra

- **Señales (slots) más frecuentes y su uso:**

No tiene señales propias, pero reacciona a la de otros widgets como por ejemplo:

- **QSlider**
 - **QDial**

- Captura de pantalla de la ejecución del ejercicio propuesto



```
ejercicio03.py U ejercicio04.py U ejercicio05.py U ejercicio06.py U
```

Desarrollo de Interfaces > Tema 2 > Tarea 2.1 > ejercicio03.py > MainWindow > __init__

```
6 class MainWindow(QMainWindow):
7     def __init__(self):
8         super().__init__()
9
10        self.progreso_actual = 0
11
12        self.barra = QProgressBar()
13        self.barra.setRange(0, 100)
14        self.barra.setValue(self.progreso_actual)
15
16
17        self.setCentralWidget(self.barra)
18        self.setWindowTitle("Progreso: 0%")
19
20
21
22        self.timer = QTimer()
23        self.timer.timeout.connect(self.preguntar_usuario)
24        self.timer.start(2000)
25
26
27        self.titulos = ["Progreso: 0%", "Progreso: 20%", "Progreso: 40%", "Progreso: 60%", "Progreso: 80%", "Completado"]
28
29    def preguntar_usuario(self):
30
31        print("\n== Control de progreso ==")
32        print("1 → Aumentar progreso")
33        print("2 → Retroceder progreso")
34        print("0 → Salir")
35        opcion = input("Elige una opción: ")
36
37        if opcion == "1":
38            self.cambiar_progreso("aumentar")
39        elif opcion == "2":
40            self.cambiar_progreso("disminuir")
41        elif opcion == "0":
42            print("Adios")
43            self.timer.stop()
44            self.close()
```

PS C:\Users\alumno\Desktop\DM\Proyectos Python & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio01.py"

PS C:\Users\alumno\Desktop\DM\Proyectos Python & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio06.py"

PS C:\Users\alumno\Desktop\DM\Proyectos Python & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/Ejercicio02.py"

1

PS C:\Users\alumno\Desktop\DM\Proyectos Python & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio06.py"

PS C:\Users\alumno\Desktop\DM\Proyectos Python & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio03.py"

== Control de progreso ==
1 → Aumentar progreso
2 → Retroceder progreso
0 → Salir
Elige una opción:

• **Webgrafía:**

- <https://www.tutorialspoint.com/>
 - <https://doc.qt.io/qt-6/qtdesigner-manual.html>
 - <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html>
 - Apuntes del tema

QDateTimeEdit

- **Descripción general:** Sirve para seleccionar la fecha y hora, donde se puede editar
- **Principales métodos:**
 - **.setDateTime(QDateTime):** Establece la fecha y hora
 - **.dateTime():** Devuelve la fecha y hora actual
 - **.setDisplayFormat(str):** Cambia el formato de la fecha
 - **.setMinimumDate(QDate):** Establece la fecha mínima
 - **.clear():** Borra todo el contenido de la fecha y hora
 - **.setCalendarPopup(bool):** Sirve para mostrar un calendario emergente
- **Señales (slots) más frecuentes y su uso:**
 - **dateChanged(QDate):** Cambia la fecha
 - **timeChanged(QTime):** Cambia la hora
 - **dateTimeChanged(QDateTime):** Cambia tanto fecha como hora
 - **editingFinished():** Devuelve cuando termine la edición manual

- Captura de pantalla de la ejecución del ejercicio propuesto

The screenshot shows the PyCharm IDE interface. On the left, there are three tabs: 'ejercicio04.py U' (selected), 'ejercicio05.py U', and 'ejercicio06.py U'. The main editor area contains Python code for a PyQt application. The terminal on the right shows the command line and the output of the application's execution.

```

1 # Jose Carlos Mora 2º DAM
2
3 from PySide6.QtWidgets import QApplication, QMainWindow, QDateTimeEdit
4 from PySide6.QtCore import QDateTime
5
6 class MainWindow(QMainWindow):
7     def __init__(self):
8         super().__init__()
9
9         self.dt_edit = QDateTimeEdit()
10        self.dt_edit.setDateTime(QDateTime.currentDateTime())
11
12        self.dt_edit.setDisplayFormat("ddd, d 'de' MMMM 'de' yyyy hh:mm")
13
14        self.dt_edit.dateTimeChanged.connect(self.mostrar_fecha)
15
16        self.setCentralWidget(self.dt_edit)
17
18        self.mostrar_fecha(self.dt_edit.dateTime())
19
20    def mostrar_fecha(self, fecha_hora):
21
22        self.setWindowTitle(fecha_hora.toString("ddd, d 'de' MMMM 'de' yyyy hh:mm"))
23        print("Fecha elegida:", fecha_hora.toString("ddd, d 'de' MMMM 'de' yyyy hh:mm"))
24
25
26
27
28
29
30
31
32 app = QApplication([])
33 ventana = MainWindow()
34 ventana.show()
35 app.exec()
36

```

Terminal output:

```

PS C:\Users\alumno\Desktop\DM\Proyectos Python> & "C:/Program Files/Python313/python.exe" "C:/Users/alumno/Desktop/DM/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio04.py"
Wednesday, 12...
miercoles 12 de noviembre de 2025 09:30 ~
Fecha elegida: Tuesday, 11 de November de 2025 09:30
Fecha elegida: Wednesday, 12 de November de 2025 09:30

```

- Webgrafía:

- <https://www.tutorialspoint.com/>
- <https://doc.qt.io/qt-6/qtdesigner-manual.html>
- <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html>
- Apuntes del tema

QSlider

- **Descripción general:** Sirve para crear un control deslizante o vertical que permite seleccionar un valor numérico dentro de un rango
- **Principales métodos:**
 - **.setMinimum():** Define el valor mínimo

- **.setMaximum():** Define el valor máximo
 - **.setValue():** Fija el valor actual
 - **.Value:** Devuelve el valor actual
 - **.setOrientation(Qt.Orientation):** Define la orientación
 - **.setTracking(bool):** Devuelve señales constantemente mientras se mueve, siempre y cuando sea true
 - **.setTickPosition(QSlider.TicksBelow):** Define la posición de las marcas
 - **Señales (slots) más frecuentes y su uso:**
 - **valueChanged(int):** Se emite cuando cambia el valor
 - **sliderPressed():** Devuelve cuando se presiona el control
 - **sliderReleased():** Devuelve cuando se suelta el control
 - **sliderMoved(int):** Devuelve cuando se arrastra el control
 - **Captura de pantalla de la ejecución del ejercicio propuesto**



```
ejercicio05.py U ejercicio05.py U
Desarrollo de Interfaces > Tema 2 > Tarea 2.1 > ejercicio05.py > ...
1 # José Carlos Mora 2º DAM
2
3 from PySide6.QtWidgets import QApplication, QMainWindow, QSlider
4 from PySide6.QtCore import Qt
5
6 class MainWindow(QMainWindow):
7     def __init__(self):
8         super().__init__()
9
10        self.slider = QSlider(Qt.Horizontal)
11
12        self.setCentralWidget(self.slider)
```

PS C:\Users\alumno\Desktop\DAW\Proyectos Python> & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DAW/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio05.py"

Nivel de brillo: 50 %

- **Webgrafía:**
 - <https://www.tutorialspoint.com/>
 - <https://doc.qt.io/qt-6/qtdesigner-manual.html>
 - <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html>
 - Apuntes del tema

QDial

- **Descripción general:** Sirve para hacer un control en forma de rueda donde el usuario puede elegir un número dentro de un rango
- **Principales métodos:**
 - **.setNotchesVisible(True):** Muestra las muescas o marcas
 - **.setRange(min, max):** Define los valores mínimo y máximo
 - **.setValue():** Devuelve el valor actual

- **.setEnabled(bool):** Habilita o deshabilita el control
- **.setInvertedAppearance(bool):** Invierte la dirección visual
- **Señales (slots) más frecuentes y su uso:**
 - **valueChanged(int):** Se emite cuando cambia el valor
 - **sliderPressed():** Devuelve cuando se presiona el control
 - **sliderReleased():** Devuelve cuando se suelta el control
 - **sliderMoved(int):** Devuelve cuando se arrastra el control

● Captura de pantalla de la ejecución del ejercicio propuesto



The screenshot shows a Windows desktop environment with a Python application running in a window titled "Vol..". The window contains a large dial control. Below the window, a status bar displays the text "Vol.. Volumen: 0 / 10". The desktop background is dark, and there are other icons and windows visible in the background.

```

Go Run Terminal Help ← → 🔍 Proyectos Python
ejercicio06.py ✘ Desarrollo de Interfaces > Tema 2 > Tarea 2.1 > ejercicios06.py > ...
1 # José Carlos Mora 2º DAM
2
3 from PySide6.QtWidgets import QApplication, QMainWindow, QDial
4
5 class MainWindow(QMainWindow):
6     def __init__(self):
7         super().__init__()
8
9
10        self.dial = QDial()
11        self.dial.setRange(0, 10)
12        self.dial.setNotchesVisible(True)
13        self.dial.setValue(0)
14
15
16        self.dial.valueChanged.connect(self.cambiar_volumen)
17
18
19        self.setCentralWidget(self.dial)
20
21
22        self.cambiar_volumen(self.dial.value())
23
24    def cambiar_volumen(self, valor):
25
26        self.setWindowTitle("Volumen: " + str(valor) + " / 10")
27
28
29        if valor == 10:
30            print("¡Volumen máximo alcanzado!")
31
32
33
34    app = QApplication([])
35    ventana = MainWindow()
36    ventana.show()
37    app.exec()
38

```

TERMINAL

```

PS C:\Users\alumno\Desktop\DA\Proyectos Python> & "C:/Program Files/Python313/python.exe" "c:/Users/alumno/Desktop/DAV/Proyectos Python/Desarrollo de Interfaces/Tema 2/Tarea 2.1/ejercicio06.py"

```

- <https://www.tutorialspoint.com/>

- <https://doc.qt.io/qt-6/qtdesigner-manual.html>
- <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html>
- Apuntes del tema