DESARROLLO DE INTERFACES

Adrián Solís León 2ºDAM

11/11/2025

Adrián Solís León 2ºDAM

QRadioButton:

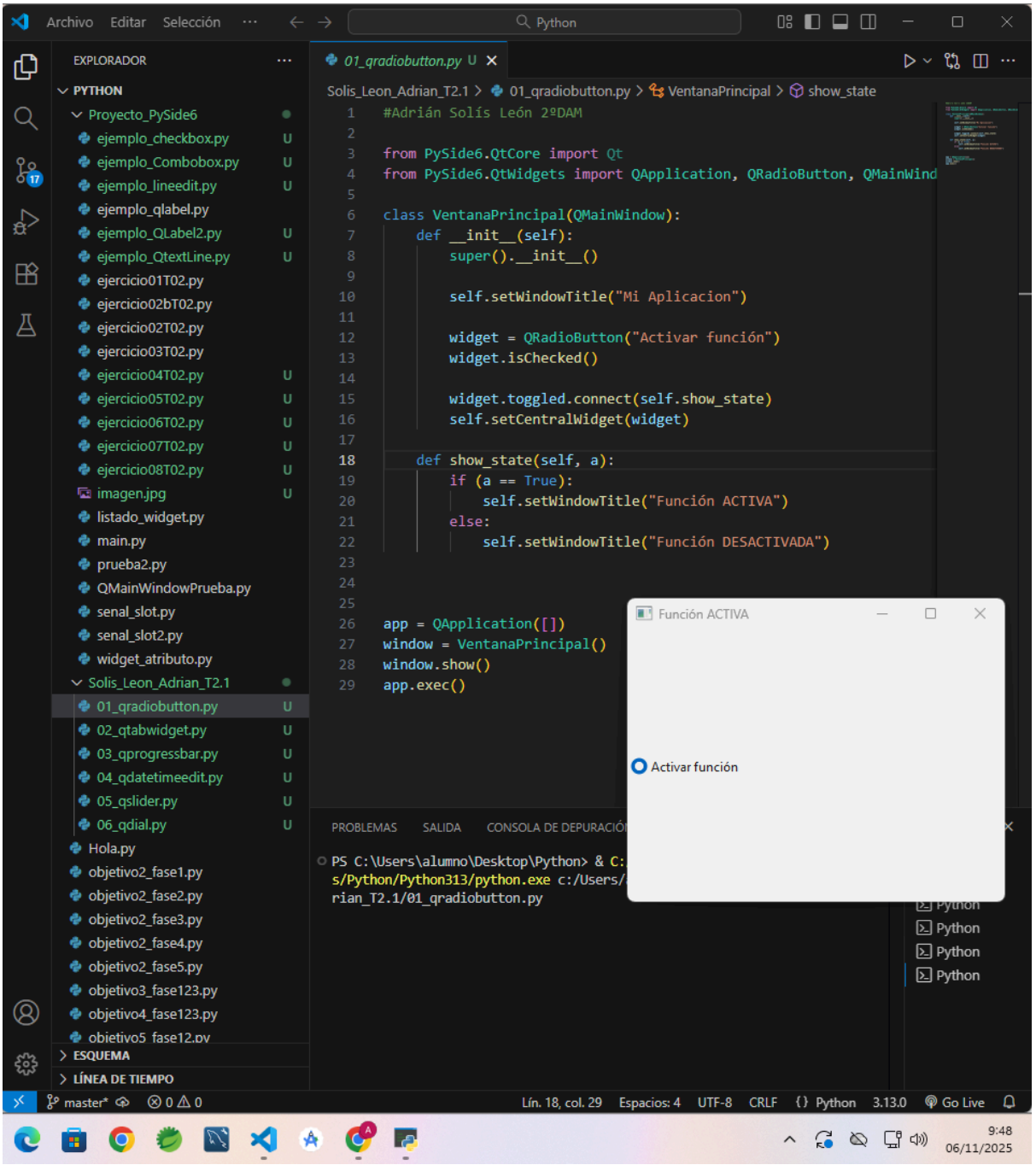
Se usa principalmente para seleccionar una única opción entre varias.

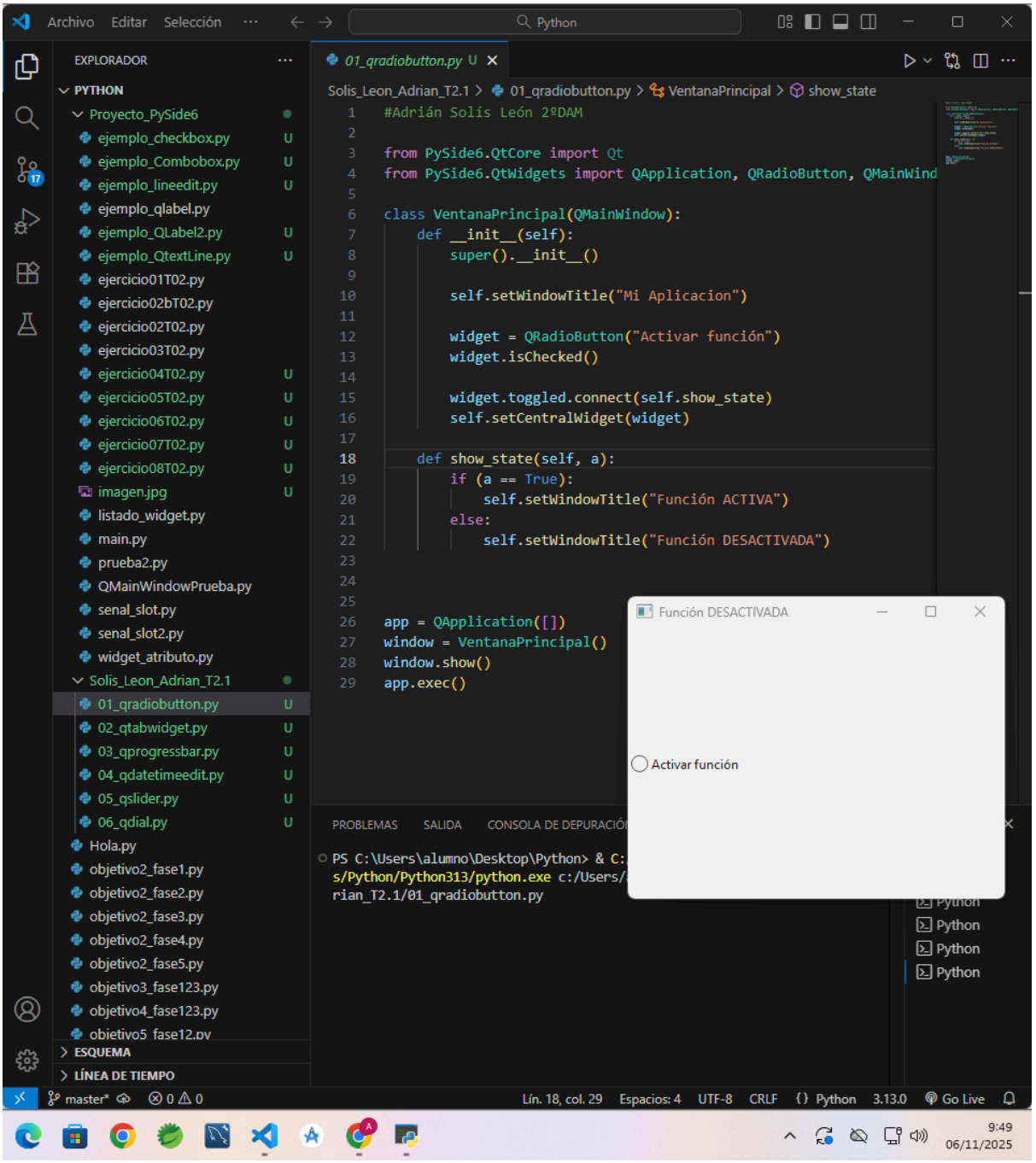
Algunos de sus métodos son:

.setChecked(bool): Para que al arrancar la aplicación el botón esté activado o no.
.isChecked(): Devuelve un booleano, dependiendo si el botón está activado o no.
.text(): El texto del botón de radio.

Señal más frecuente:

toggled(bool): Se emite cuando el botón cambia de estado, true si está marcado o false si lo desmarcan.


Archivo Editar Selección
Python
EXPLORADOR
PYTHON
Proyecto_PySide6
ejemplo_checkbox.py
ejemplo_Combobox.py
ejemplo_lineedit.py
ejemplo_qlabel.py
ejemplo_QLabel2.py
ejemplo_QtextLine.py
ejercicio01T02.py
ejercicio02bT02.py
ejercicio02T02.py
ejercicio03T02.py
ejercicio04T02.py
ejercicio05T02.py
ejercicio06T02.py
ejercicio07T02.py
ejercicio08T02.py
imagen.jpg
listado_widget.py
main.py
prueba2.py
QMainWindowPrueba.py
senal_slot.py
senal_slot2.py
widget_atributo.py
Solis_Leon_Adrian_T2.1
01_qradiobutton.py
02_qtabwidget.py
03_qprogressbar.py
04_qdatetimeedit.py
05_qslider.py
06_qdial.py
Hola.py
objetivo2_fase1.py
objetivo2_fase2.py
objetivo2_fase3.py
objetivo2_fase4.py
objetivo2_fase5.py
objetivo3_fase123.py
objetivo4_fase123.py
objetivo5_fase12.py
ESQUEMA
LÍNEA DE TIEMPO
01_qradiobutton.py U
Solis_Leon_Adrian_T2.1 > 01_qradiobutton.py > VentanaPrincipal > show_state
#Adrián Solís León 2ºDAM
from PySide6.QtCore import Qt
from PySide6.QtWidgets import QApplication, QRadioButton, QMainWind
class VentanaPrincipal(QMainWindow):
def __init__(self):
super().__init__()
self.setWindowTitle("Mi Aplicacion")
widget = QRadioButton("Activar función")
widget.isChecked()
widget.toggled.connect(self.show_state)
self.setCentralWidget(widget)
def show_state(self, a):
if (a == True):
self.setWindowTitle("Función ACTIVA")
else:
self.setWindowTitle("Función DESACTIVADA")
app = QApplication([])
window = VentanaPrincipal()
window.show()
app.exec()
Función ACTIVA
Activar función
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN
PS C:\Users\alumno\Desktop\Python> & C:
s/Python/Python313/python.exe c:/Users/
rian_T2.1/01_qradiobutton.py
Python
Python
Python
Python
master*
Lín. 18, col. 29 Espacios: 4 UTF-8 CRLF Python 3.13.0 Go Live
9:48
06/11/2025



QTabWidget:

Se usa principalmente para agrupar el contenido en pestañas.

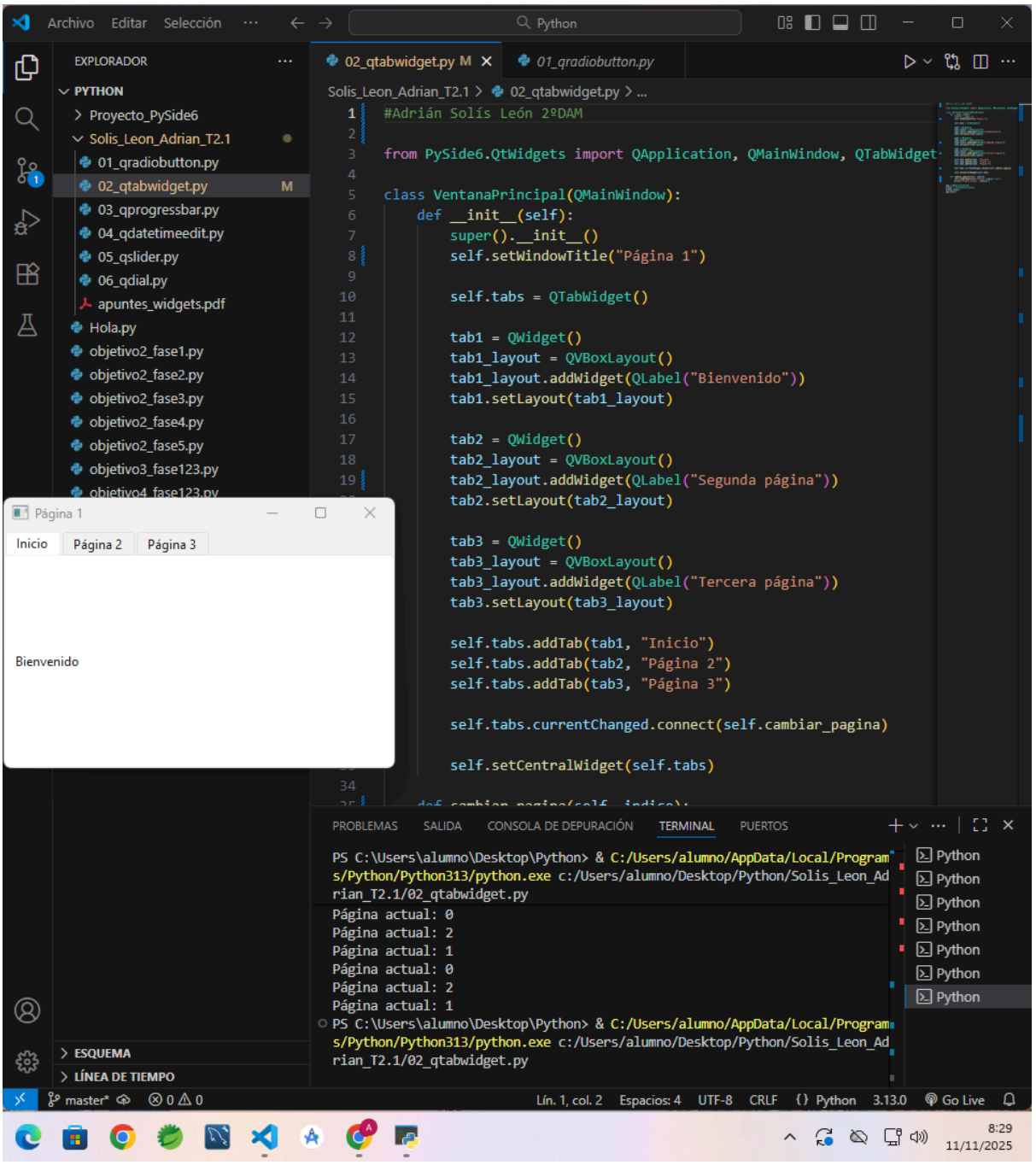
Algunos de sus métodos son:

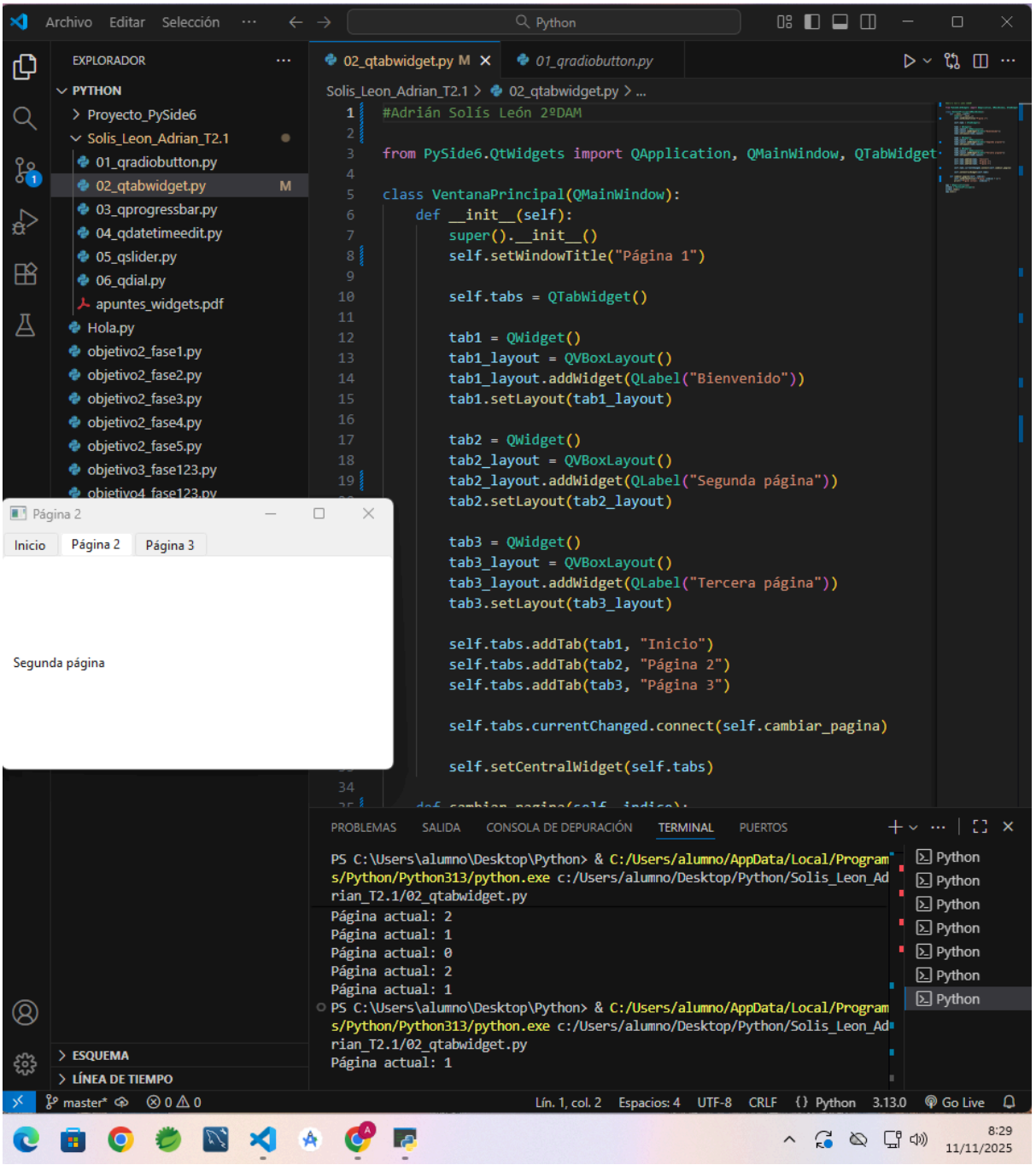
.addTab(widget, “Título”): Agrega una pestaña con la página y las etiquetas dadas al widget de la pestaña y devuelve el índice de la pestaña en la barra de pestañas.

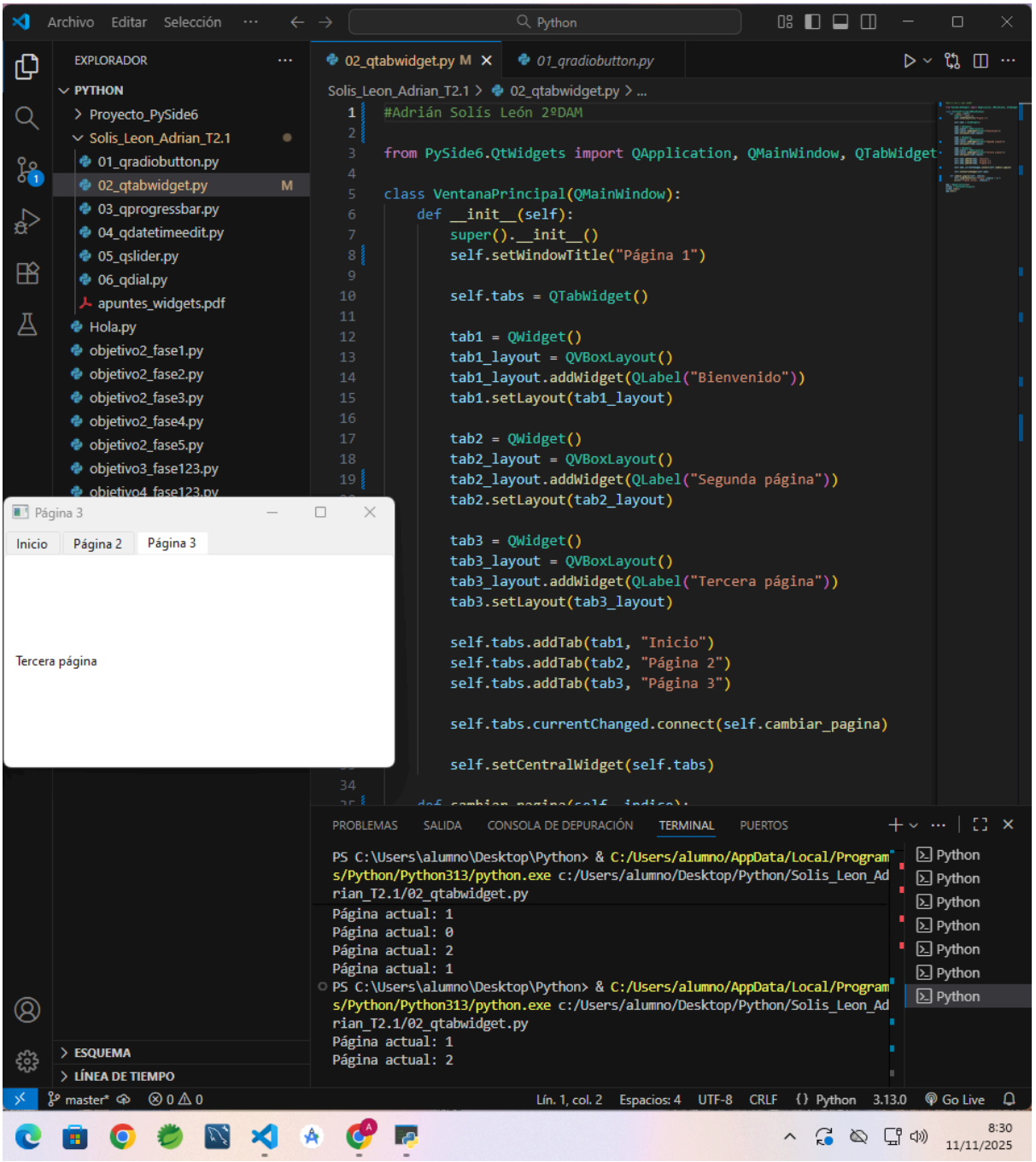
.setCurrentIndex(int): Convierte el widget en el widget actual. El widget utilizado debe ser una página en este widget de pestaña.
.count(): Devuelve los widgets que se muestran en la pestaña o ninguno.

Señal más frecuente:

currentChanged(int): Se emite cada vez que cambia el índice de la página actual, devuelve la nueva posición del índice o -1 si no hay nueva.




Archivo Editar Selección
Python
EXPLORADOR
PYTHON
Proyecto_PySide6
Solis_Leon_Adrian_T2.1
01_qradiobutton.py
02_qtabwidget.py
03_qprogressbar.py
04_qdatetimeedit.py
05_qslider.py
06_qdial.py
apuntes_widgets.pdf
Hola.py
objetivo2_fase1.py
objetivo2_fase2.py
objetivo2_fase3.py
objetivo2_fase4.py
objetivo2_fase5.py
objetivo3_fase123.py
02_qtabwidget.py M
01_qradiobutton.py
Solis_Leon_Adrian_T2.1 > 02_qtabwidget.py > ...
#Adrián Solís León 2ºDAM
from PySide6.QtWidgets import QApplication, QMainWindow, QTabWidget
class VentanaPrincipal(QMainWindow):
def __init__(self):
super().__init__()
self.setWindowTitle("Página 1")
self.tabs = QTabWidget()
tab1 = QWidget()
tab1_layout = QVBoxLayout()
tab1_layout.addWidget(QLabel("Bienvenido"))
tab1.setLayout(tab1_layout)
tab2 = QWidget()
tab2_layout = QVBoxLayout()
tab2_layout.addWidget(QLabel("Segunda página"))
tab2.setLayout(tab2_layout)
tab3 = QWidget()
tab3_layout = QVBoxLayout()
tab3_layout.addWidget(QLabel("Tercera página"))
tab3.setLayout(tab3_layout)
self.tabs.addTab(tab1, "Inicio")
self.tabs.addTab(tab2, "Página 2")
self.tabs.addTab(tab3, "Página 3")
self.tabs.currentChanged.connect(self.cambiar_pagina)
self.setCentralWidget(self.tabs)
Página 2
Inicio
Página 2
Página 3
Segunda página
PROBLEMAS
SALIDA
CONSOLA DE DEPURACIÓN
TERMINAL
PUERTOS
PS C:\Users\alumno\Desktop\Python> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/Python/Solis_Leon_Adrian_T2.1/02_qtabwidget.py
Página actual: 2
Página actual: 1
Página actual: 0
Página actual: 2
Página actual: 1
PS C:\Users\alumno\Desktop\Python> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/Python/Solis_Leon_Adrian_T2.1/02_qtabwidget.py
Página actual: 1
Python
ESQUEMA
LÍNEA DE TIEMPO
master*
Lín. 1, col. 2
Espacios: 4
UTF-8
CRLF
Python
3.13.0
Go Live
8:29
11/11/2025



QProgressBar:

Se usa para mostrar el progreso de una tarea o valor.

Algunos de sus métodos son:

.setValue(int): Establece el valor de la propiedad
.setRange(min,max): Establece los valores mínimo y máximo de la barra de progreso en mínimo y máximo respectivamente.

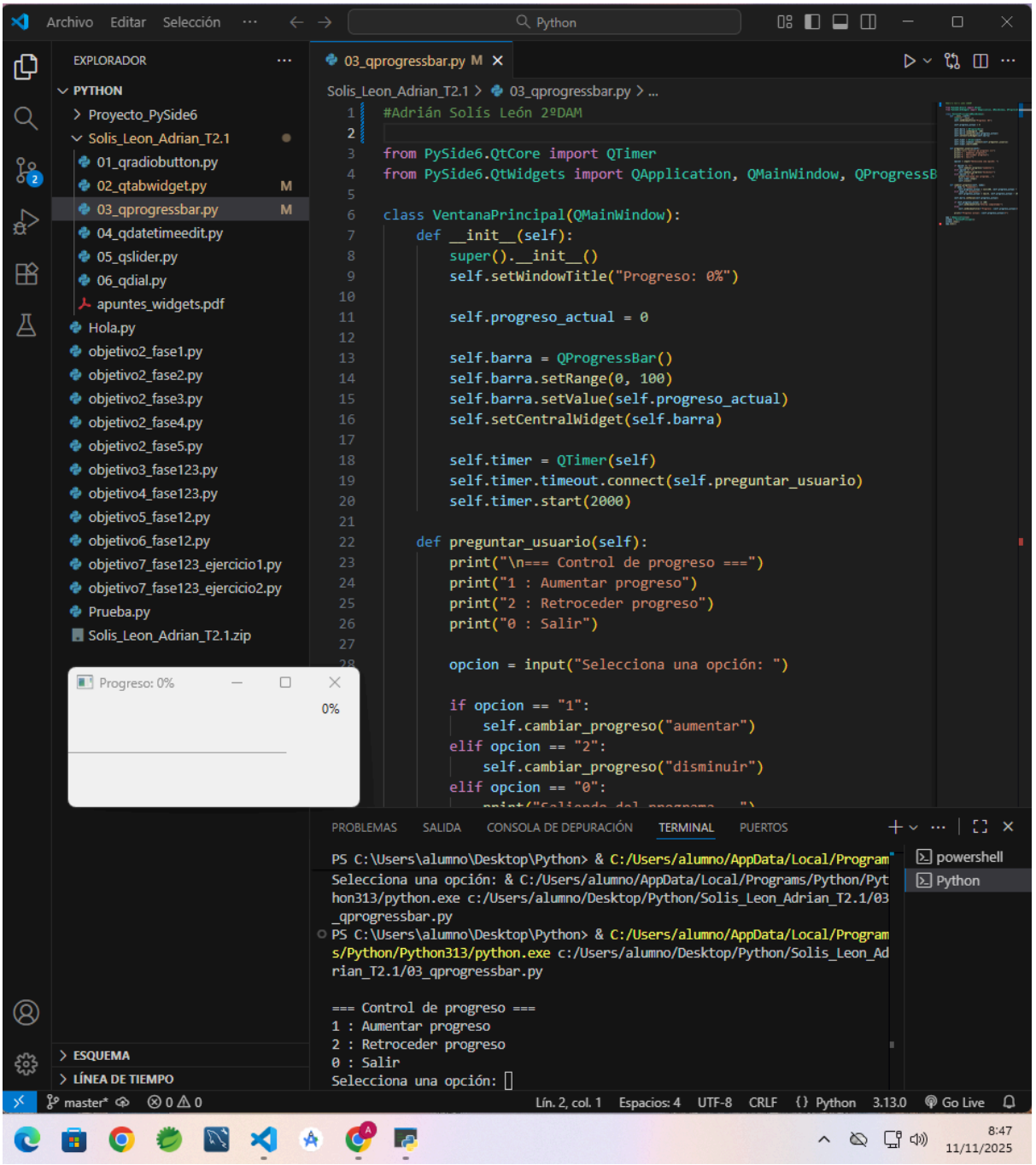
Si el máximo es menor que el mínimo, el mínimo se convierte en el único valor legal.

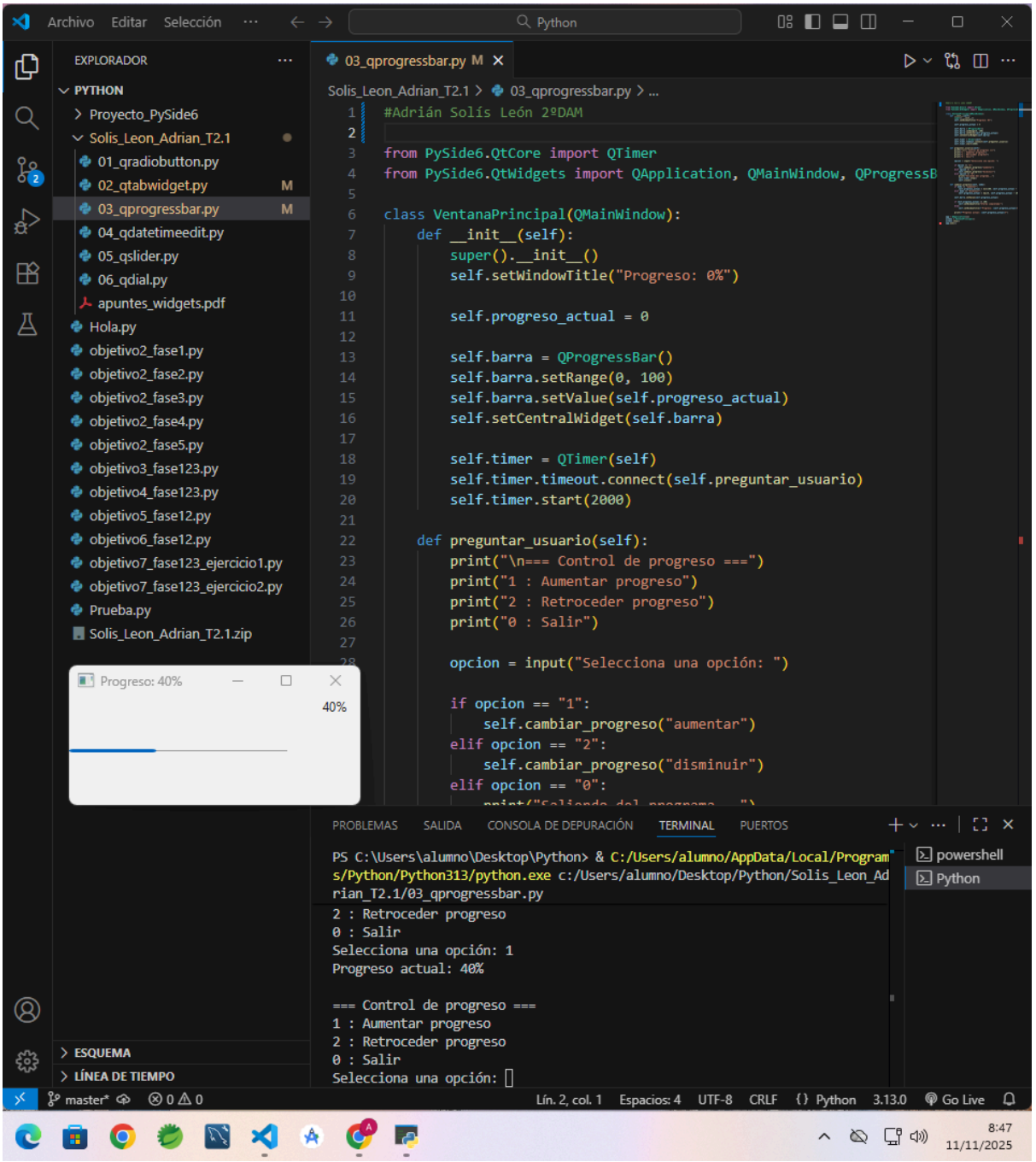
Si el valor actual queda fuera del nuevo rango, la barra de progreso se restablece con reset() .

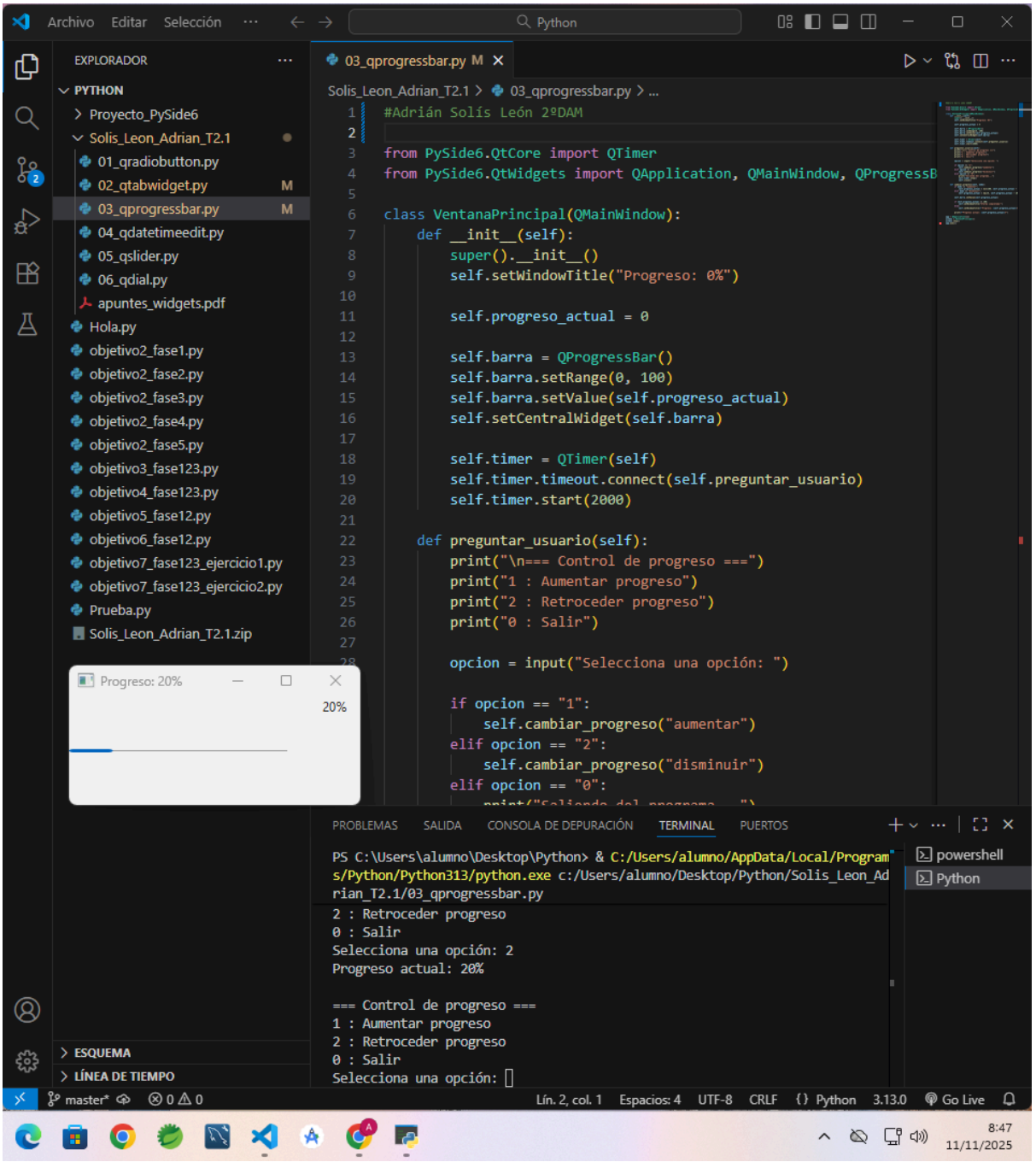
QProgressBar se puede establecer en estado indeterminado usando setRange(0, 0).

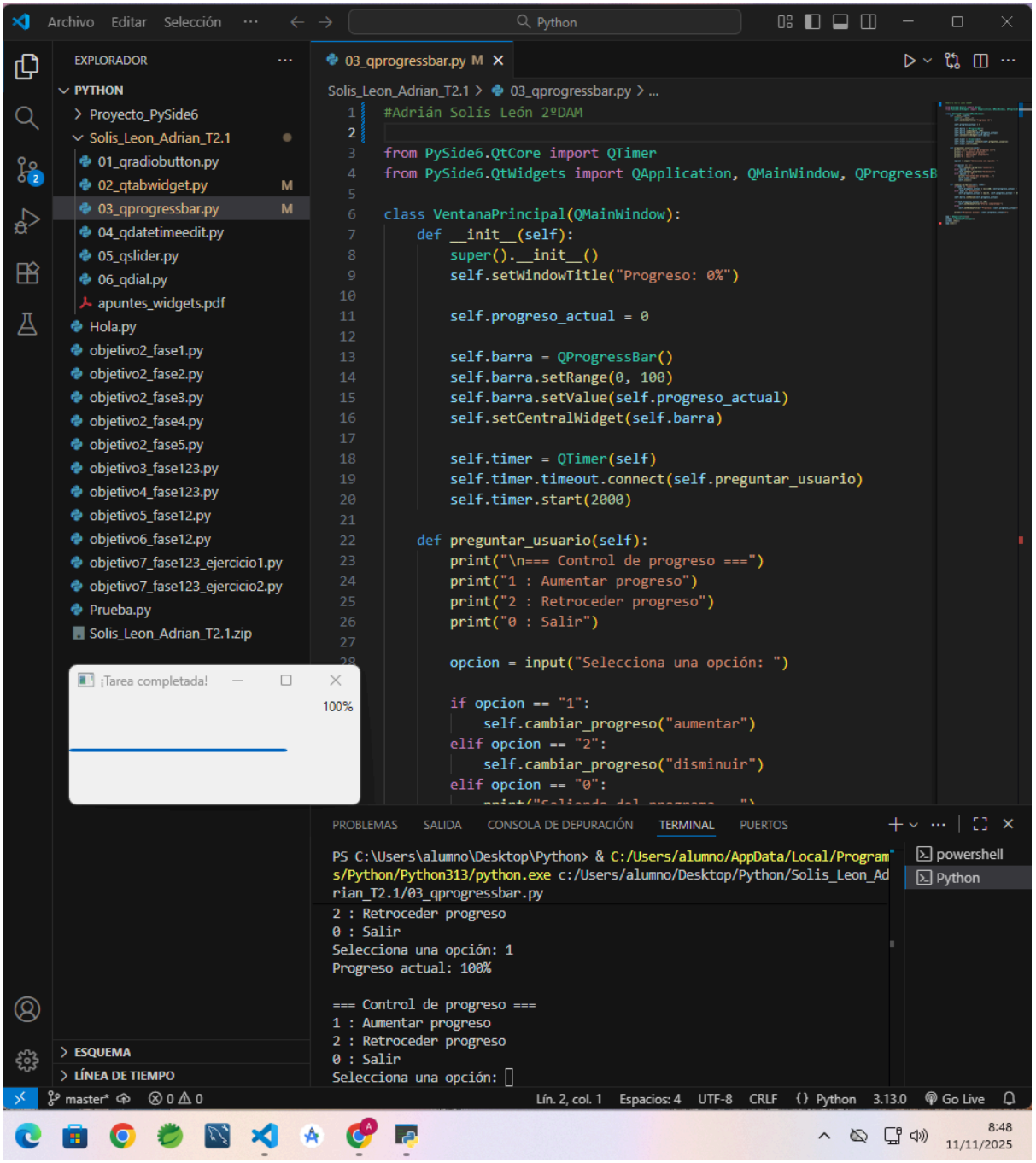
.reset(): Restablece la barra de progreso

No tiene señales propias pero sí reacciona a la de otros Widgets.


Archivo Editar Selección
Python
EXPLORADOR
PYTHON
Proyecto_PySide6
Solis_Leon_Adrian_T2.1
01_qradiobutton.py
02_qtabwidget.py M
03_qprogressbar.py M
04_qdatetimeedit.py
05_qslider.py
06_qdial.py
apuntes_widgets.pdf
Hola.py
objetivo2_fase1.py
objetivo2_fase2.py
objetivo2_fase3.py
objetivo2_fase4.py
objetivo2_fase5.py
objetivo3_fase123.py
objetivo4_fase123.py
objetivo5_fase12.py
objetivo6_fase12.py
objetivo7_fase123_ejercicio1.py
objetivo7_fase123_ejercicio2.py
Prueba.py
Solis_Leon_Adrian_T2.1.zip
03_qprogressbar.py M
Solis_Leon_Adrian_T2.1 > 03_qprogressbar.py > ...
#Adrián Solís León 2ºDAM
from PySide6.QtCore import QTimer
from PySide6.QtWidgets import QApplication, QMainWindow, QProgressB
class VentanaPrincipal(QMainWindow):
def __init__(self):
super().__init__()
self.setWindowTitle("Progreso: 0%")
self.progreso_actual = 0
self.barra = QProgressBar()
self.barra.setRange(0, 100)
self.barra.setValue(self.progreso_actual)
self.setCentralWidget(self.barra)
self.timer = QTimer(self)
self.timer.timeout.connect(self.preguntar_usuario)
self.timer.start(2000)
def preguntar_usuario(self):
print("\n=== Control de progreso ===")
print("1 : Aumentar progreso")
print("2 : Retroceder progreso")
print("0 : Salir")
opcion = input("Selecciona una opción: ")
if opcion == "1":
self.cambiar_progreso("aumentar")
elif opcion == "2":
self.cambiar_progreso("disminuir")
elif opcion == "0":
Progreso: 0%
0%
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\alumno\Desktop\Python> & C:/Users/alumno/AppData/Local/Program
Selecciona una opción: & C:/Users/alumno/AppData/Local/Programs/Python/Pyt
hon313/python.exe c:/Users/alumno/Desktop/Python/Solis_Leon_Adrian_T2.1/03
_qprogressbar.py
PS C:\Users\alumno\Desktop\Python> & C:/Users/alumno/AppData/Local/Program
s/Python/Python313/python.exe c:/Users/alumno/Desktop/Python/Solis_Leon_Ad
rian_T2.1/03_qprogressbar.py
=== Control de progreso ===
1 : Aumentar progreso
2 : Retroceder progreso
0 : Salir
Selecciona una opción:
powershell
Python
ESQUEMA
LÍNEA DE TIEMPO
master* 0 0
Lín. 2, col. 1 Espacios: 4 UTF-8 CRLF Python 3.13.0 Go Live
8:47
11/11/2025


Archivo Editar Selección
Python
EXPLORADOR
PYTHON
Proyecto_PySide6
Solis_Leon_Adrian_T2.1
01_qradiobutton.py
02_qtabwidget.py M
03_qprogressbar.py M
04_qdatetimeedit.py
05_qslider.py
06_qdial.py
apuntes_widgets.pdf
Hola.py
objetivo2_fase1.py
objetivo2_fase2.py
objetivo2_fase3.py
objetivo2_fase4.py
objetivo2_fase5.py
objetivo3_fase123.py
objetivo4_fase123.py
objetivo5_fase12.py
objetivo6_fase12.py
objetivo7_fase123_ejercicio1.py
objetivo7_fase123_ejercicio2.py
Prueba.py
Solis_Leon_Adrian_T2.1.zip
03_qprogressbar.py M
Solis_Leon_Adrian_T2.1 > 03_qprogressbar.py > ...
#Adrián Solís León 2ºDAM
from PySide6.QtCore import QTimer
from PySide6.QtWidgets import QApplication, QMainWindow, QProgressB
class VentanaPrincipal(QMainWindow):
def __init__(self):
super().__init__()
self.setWindowTitle("Progreso: 0%")
self.progreso_actual = 0
self.barra = QProgressBar()
self.barra.setRange(0, 100)
self.barra.setValue(self.progreso_actual)
self.setCentralWidget(self.barra)
self.timer = QTimer(self)
self.timer.timeout.connect(self.preguntar_usuario)
self.timer.start(2000)
def preguntar_usuario(self):
print("\n=== Control de progreso ===")
print("1 : Aumentar progreso")
print("2 : Retroceder progreso")
print("0 : Salir")
opcion = input("Selecciona una opción: ")
if opcion == "1":
self.cambiar_progreso("aumentar")
elif opcion == "2":
self.cambiar_progreso("disminuir")
elif opcion == "0":
Progreso: 40%
40%
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\alumno\Desktop\Python> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/Python/Solis_Leon_Adrian_T2.1/03_qprogressbar.py
2 : Retroceder progreso
0 : Salir
Selecciona una opción: 1
Progreso actual: 40%
=== Control de progreso ===
1 : Aumentar progreso
2 : Retroceder progreso
0 : Salir
Selecciona una opción:
powershell
Python
ESQUEMA
LÍNEA DE TIEMPO
master*
Lín. 2, col. 1 Espacios: 4 UTF-8 CRLF Python 3.13.0 Go Live
8:47
11/11/2025


Archivo Editar Selección
Python
EXPLORADOR
PYTHON
Proyecto_PySide6
Solis_Leon_Adrian_T2.1
01_qradiobutton.py
02_qtabwidget.py M
03_qprogressbar.py M
04_qdatetimeedit.py
05_qslider.py
06_qdial.py
apuntes_widgets.pdf
Hola.py
objetivo2_fase1.py
objetivo2_fase2.py
objetivo2_fase3.py
objetivo2_fase4.py
objetivo2_fase5.py
objetivo3_fase123.py
objetivo4_fase123.py
objetivo5_fase12.py
objetivo6_fase12.py
objetivo7_fase123_ejercicio1.py
objetivo7_fase123_ejercicio2.py
Prueba.py
Solis_Leon_Adrian_T2.1.zip
03_qprogressbar.py M
Solis_Leon_Adrian_T2.1 > 03_qprogressbar.py > ...
#Adrián Solís León 2ºDAM
from PySide6.QtCore import QTimer
from PySide6.QtWidgets import QApplication, QMainWindow, QProgressB
class VentanaPrincipal(QMainWindow):
def __init__(self):
super().__init__()
self.setWindowTitle("Progreso: 0%")
self.progreso_actual = 0
self.barra = QProgressBar()
self.barra.setRange(0, 100)
self.barra.setValue(self.progreso_actual)
self.setCentralWidget(self.barra)
self.timer = QTimer(self)
self.timer.timeout.connect(self.preguntar_usuario)
self.timer.start(2000)
def preguntar_usuario(self):
print("\n=== Control de progreso ===")
print("1 : Aumentar progreso")
print("2 : Retroceder progreso")
print("0 : Salir")
opcion = input("Selecciona una opción: ")
if opcion == "1":
self.cambiar_progreso("aumentar")
elif opcion == "2":
self.cambiar_progreso("disminuir")
elif opcion == "0":
Progreso: 20%
20%
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\alumno\Desktop\Python> & C:/Users/alumno/AppData/Local/Programs/Python/Python313/python.exe c:/Users/alumno/Desktop/Python/Solis_Leon_Adrian_T2.1/03_qprogressbar.py
2 : Retroceder progreso
0 : Salir
Selecciona una opción: 2
Progreso actual: 20%
=== Control de progreso ===
1 : Aumentar progreso
2 : Retroceder progreso
0 : Salir
Selecciona una opción:
powershell
Python
ESQUEMA
LÍNEA DE TIEMPO
master*
Lín. 2, col. 1 Espacios: 4 UTF-8 CRLF Python 3.13.0 Go Live
8:47
11/11/2025



QDateTimeEdit:

Se usa para seleccionar fecha y hora.

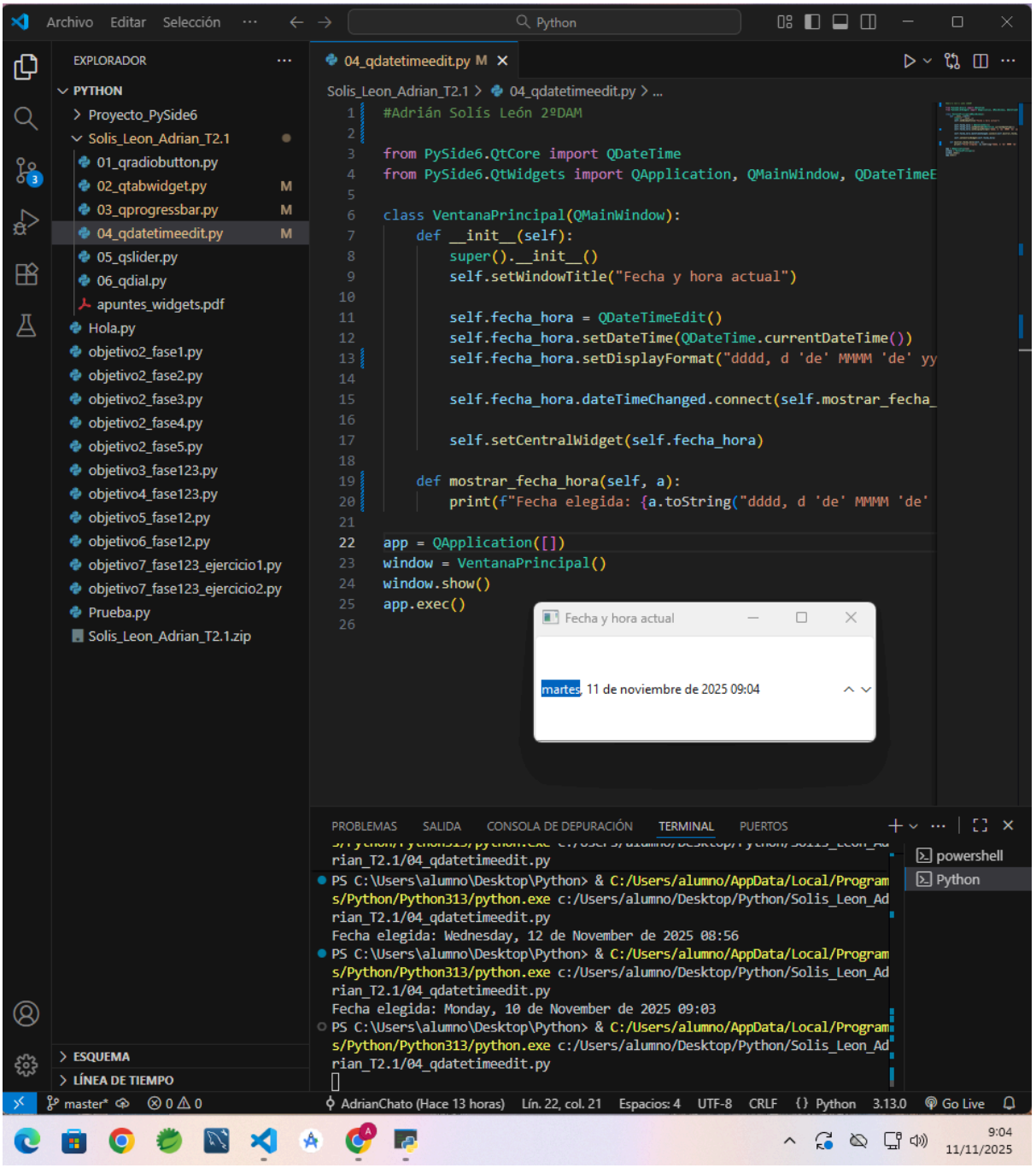
Algunos de sus métodos son:

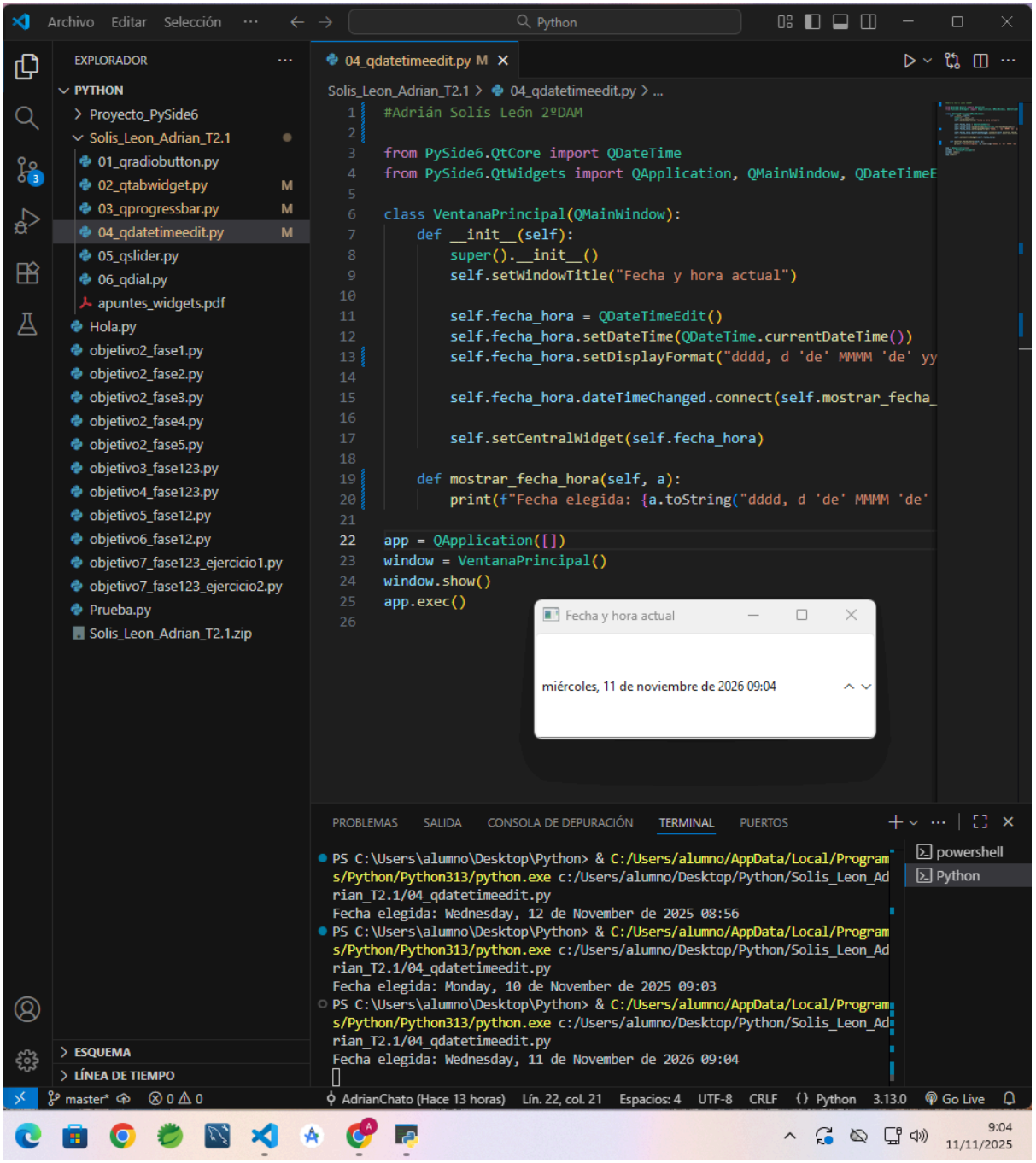
.setDateTime(QDateTime): Establece la propiedad dateTime().
.dateTime(): Obtiene la propiedad dateTime

.setDisplayFormat(str): Configura la propiedad displayFormat

Señales más frecuentes:

dateChanged(QDate): Se emite cada vez que se cambia de fecha, la nueva fecha se pasa por la entrada.
timeChanged(QTime): Se emite cada vez que se cambia de hora, la nueva hora se pasa por la entrada.
dateTimeChanged(QDateTime): Se emite cada vez que cambie fecha o hora, se pasa por la entrada los nuevos datos.





QSlider:

Se usa como control deslizante para valores numéricos.

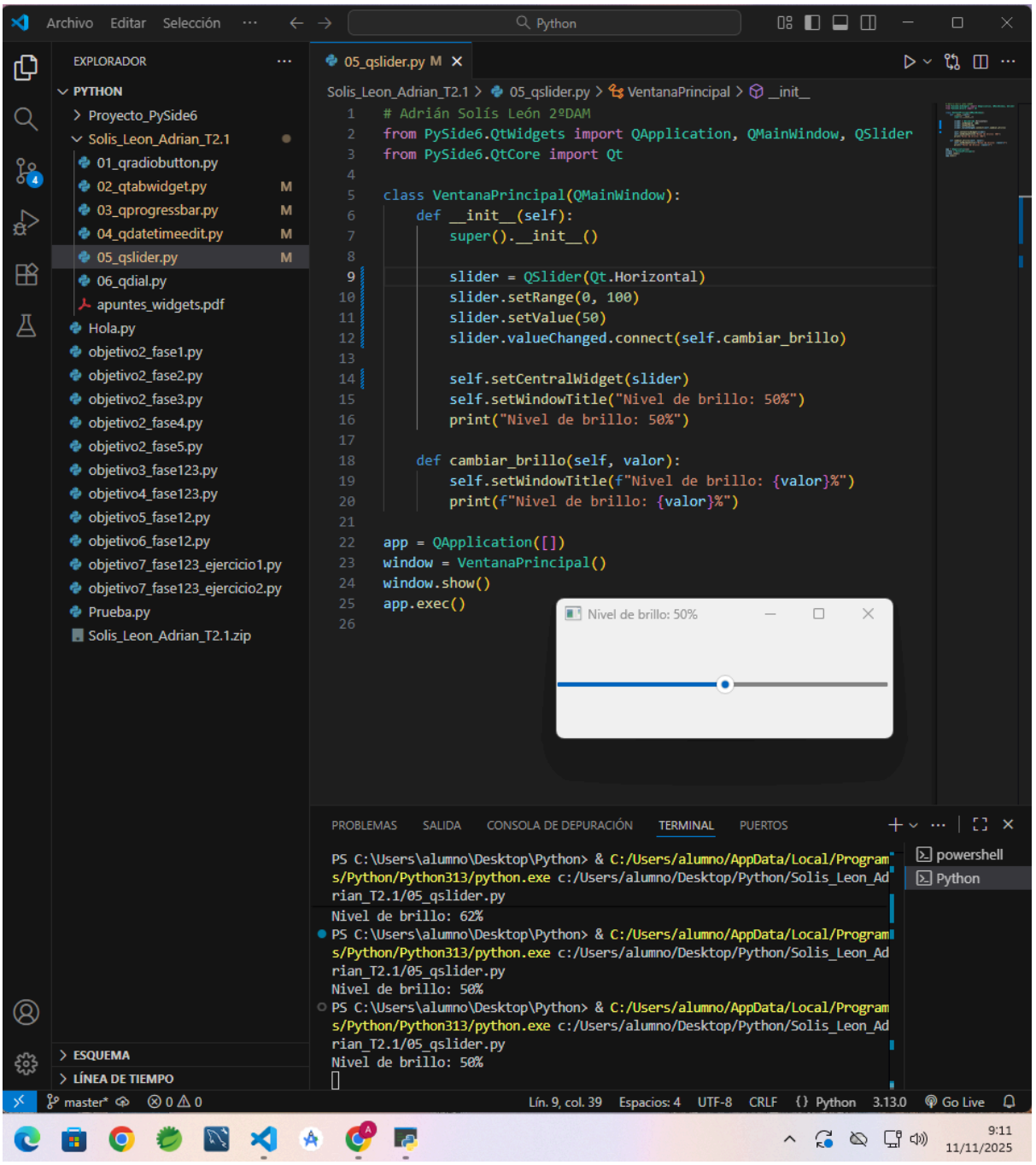
Algunos de sus métodos son:

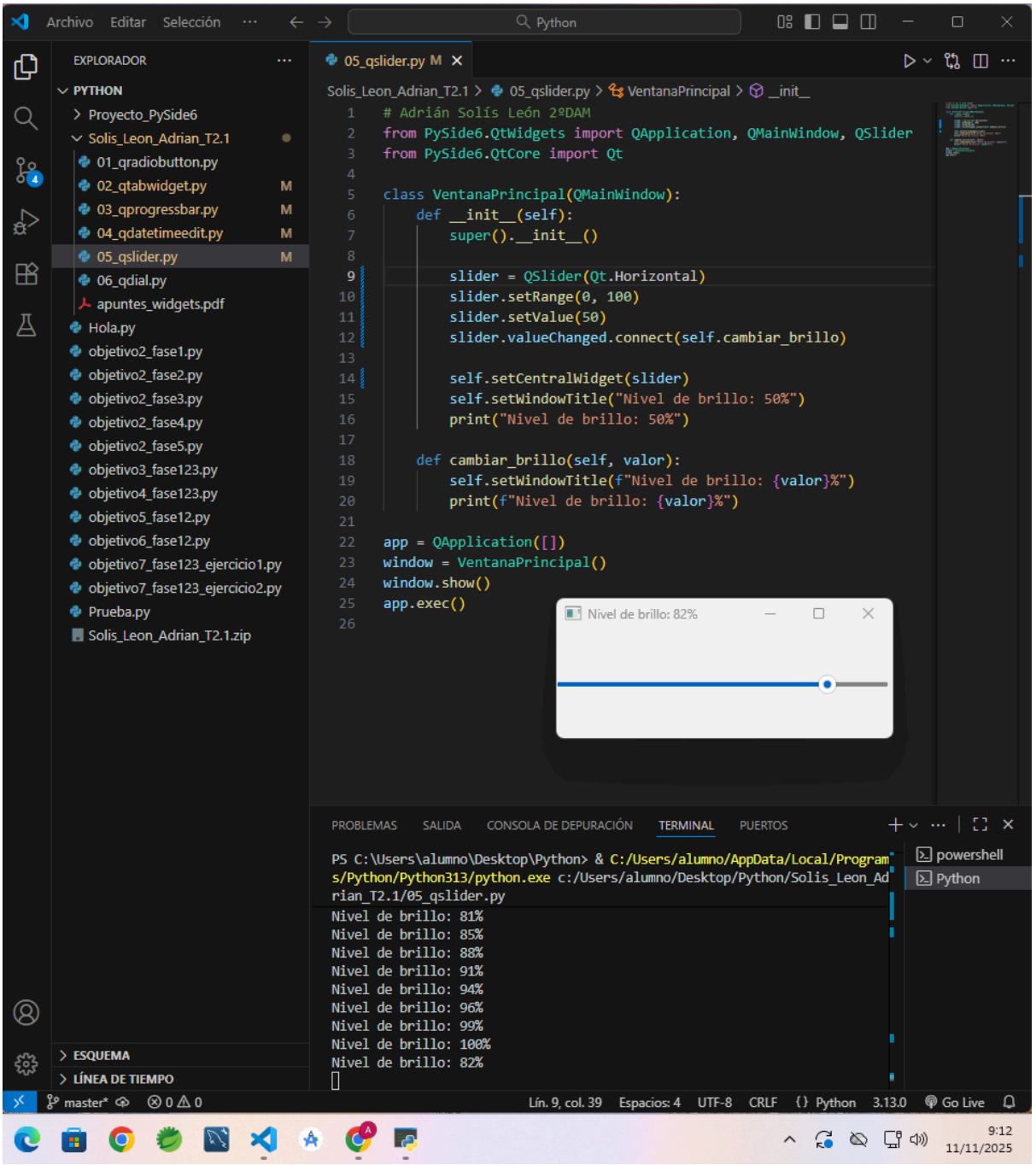
.setMinimum(): Establece el mínimo de la propiedad.

.setMaximum(): Establece el máximo de la propiedad.
.setValue(): Establece el valor de la propiedad.
.value(): Devuelve el valor de la propiedad, un entero.

Señal más frecuente:

valueChanged(int): Se emite cuando cambia el valor del control deslizante, el valor que se pasa es el nuevo y notifica el nuevo valor.





QDial:

Es parecido a QSlider pero es un control giratorio en vez de deslizante.

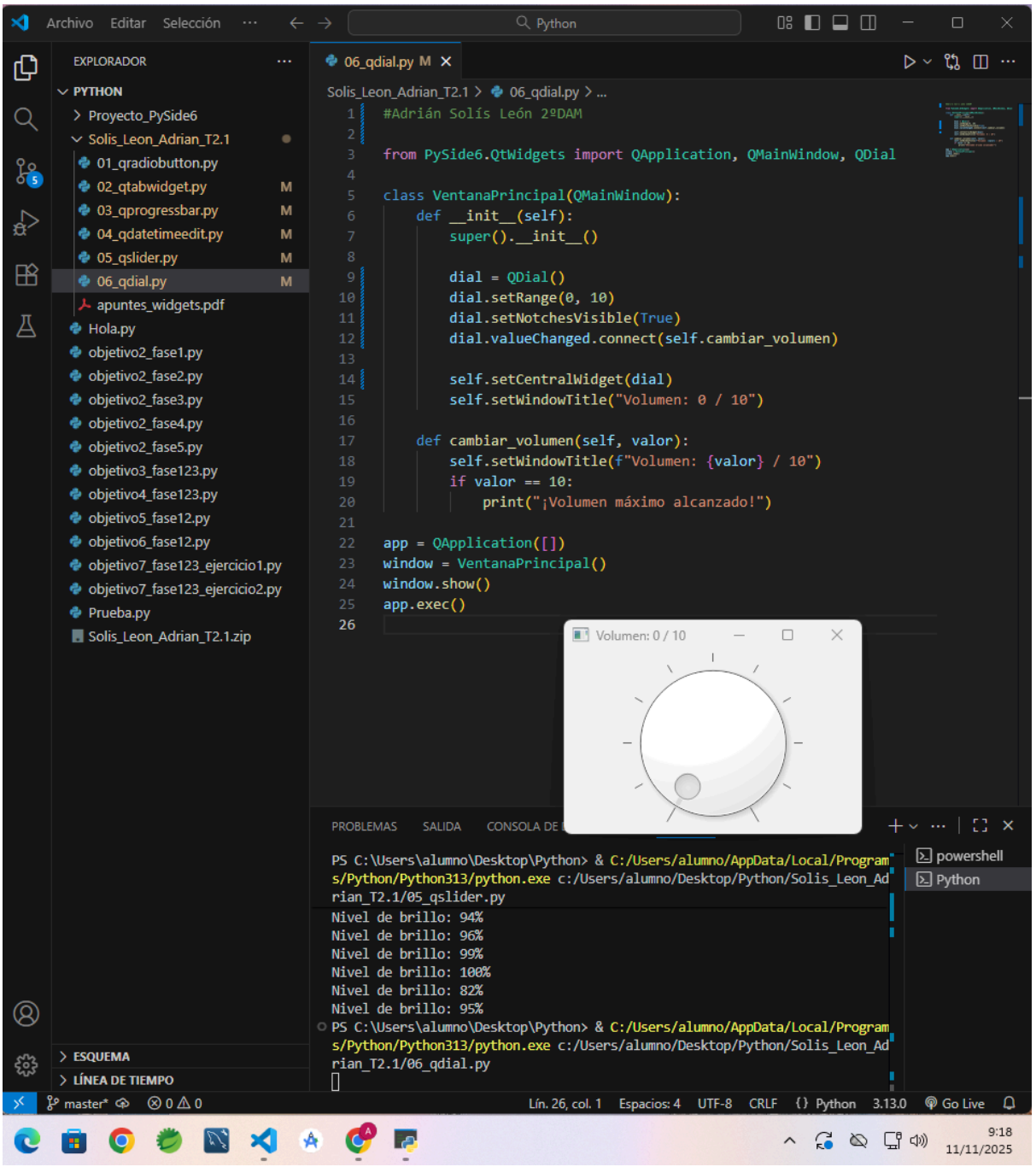
Algunos de sus métodos son:

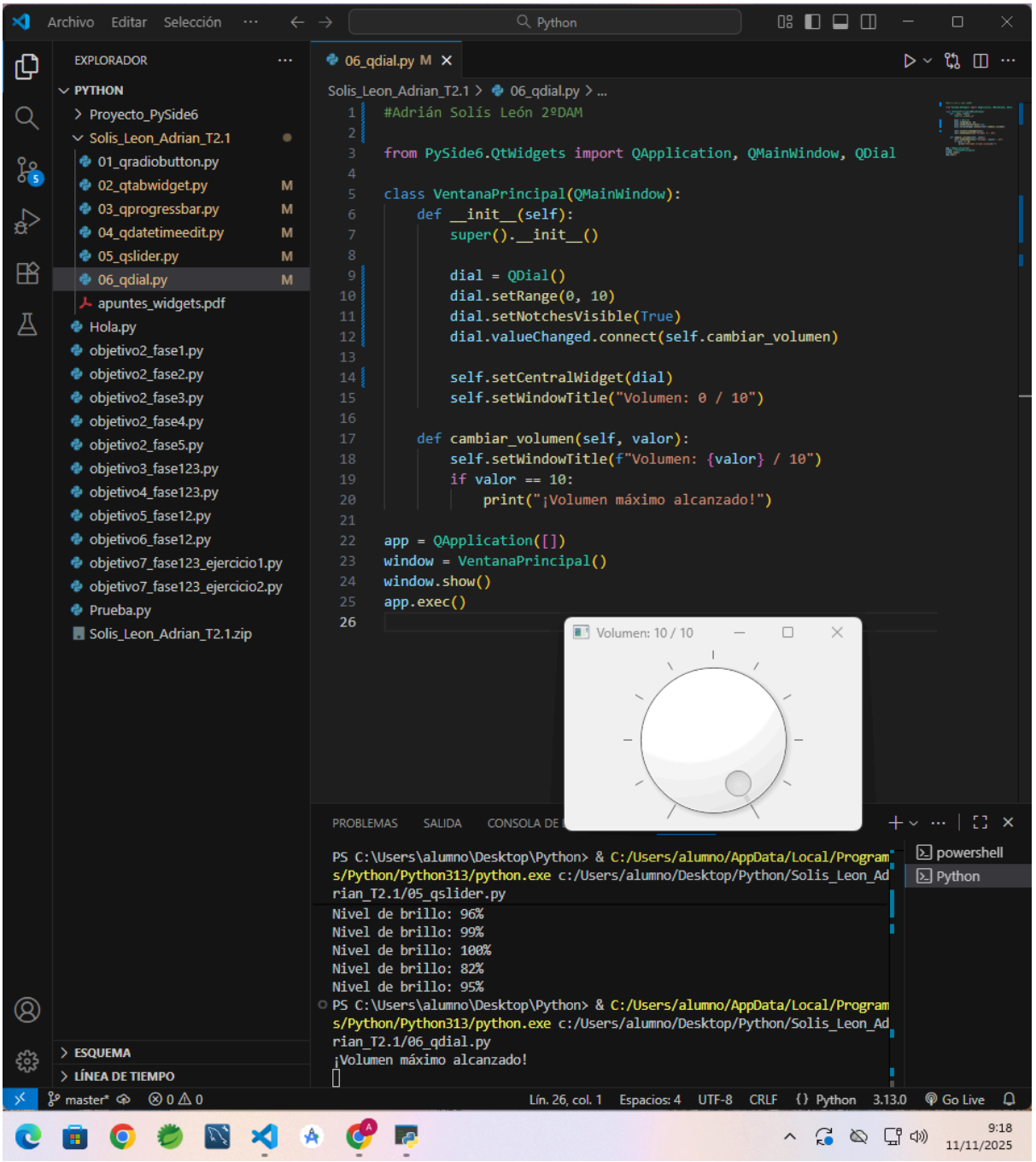
.setNotchesVisible(True): Establece el booleano de la propiedad.
.setRange(min,max): Establece el valor mínimo y máximo del control.

.setValue(): Establece el valor de la propiedad.

Su señal más frecuente:

valueChanged(int): Se emite cuando cambia el valor del control giratorio, el valor que se pasa es el nuevo y notifica el nuevo valor.





Webgrafía:

https://doc.qt.io/qtforpython/
https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html
https://doc.qt.io/qt-6/qtdesigner-manual.html

Apuntes QTimer del ejercicio.