

Universidad Autónoma de Tamaulipas

Facultad de Ingeniería Tampico



ASIGNATURA

PROGRAMACION DE INTERFACES Y PUERTOS

6. Semestre – Grupo “I”
2025 -1

TRABAJO

Programas en Clase y Ejercicios

UNIDAD

1 - MODELOS DE INTERACCIÓN COMPUTACIONAL

Docente: Dr. García Ruiz Alejandro H.

Integrante del Equipo	Nivel de Participación
Izaguirre Cortes Emanuel	33%
Turrubiates Mejia Gilberto	33%
García Salas Yahir Misael	33%
Total:	100%



INDICE

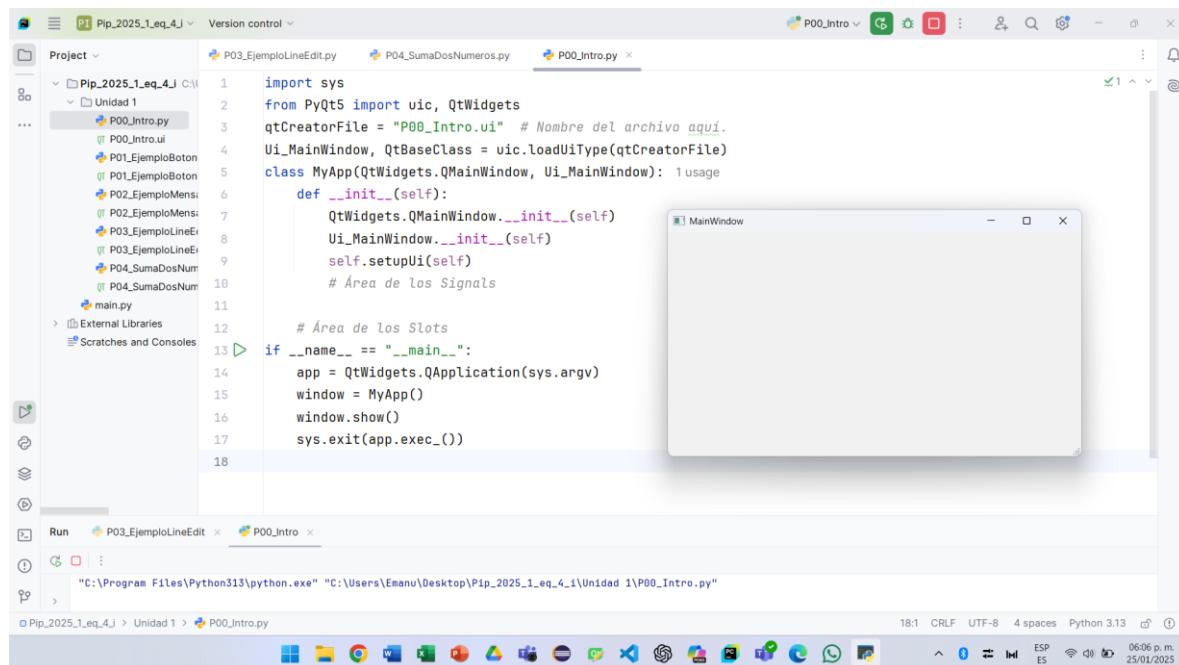
REPOSITORIO(S)	2
C1. P00_INTRO	2
C2. P01_EJEMPLOBOTON.....	3
C3. P02_EJEMPLOMENSAJEEMERGENTE	4
C4. P03_EJEMPLOLINEEDIT	5
C5. P04_SUMADOSNUMEROS	6
EJERCICIOS	8
DISTANCIA DE UN PUNTO A OTRO PUNTO	8
PAR O IMPAR.....	9
ÁREA DE UN CUADRADO	10
ÁREAS DE UN PENTÁGONO	11
COMPROBAR SI ES MAYOR DE EDAD UNA PERSONA.....	12
IMPRIMIR EN CONSOLA/MESSAGEBOX UNA TABLA DE MULTIPLICAR	14
CANTIDAD DE CARACTERES DE UNA CADENA	15
CUANTAS HORAS LE RESTAN AL DÍA PARA TERMINARSE	16
CALCULAR LA FACTORIAL DE UN NÚMERO	17



Repositorio(s)

Actividad	Repository
Todas	Emanuel-Izaguirre-Cortes-03/Programacion_int_prt

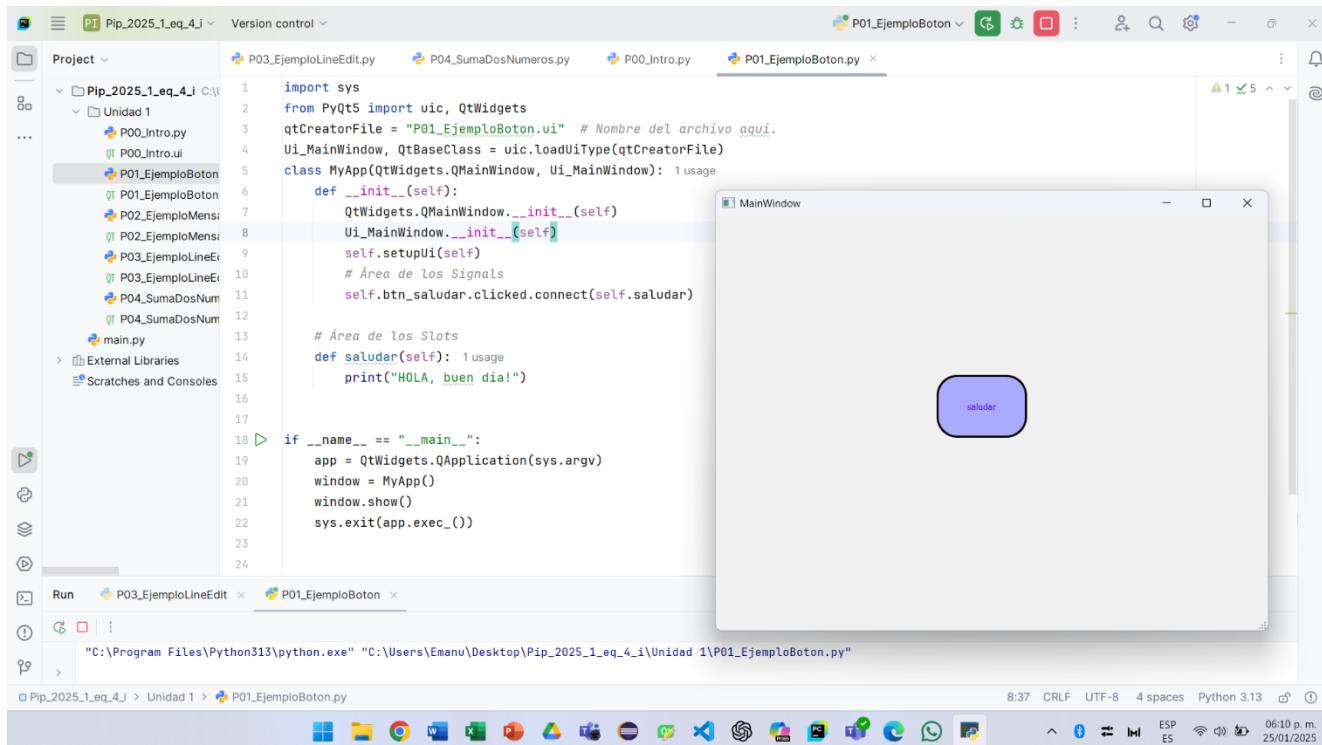
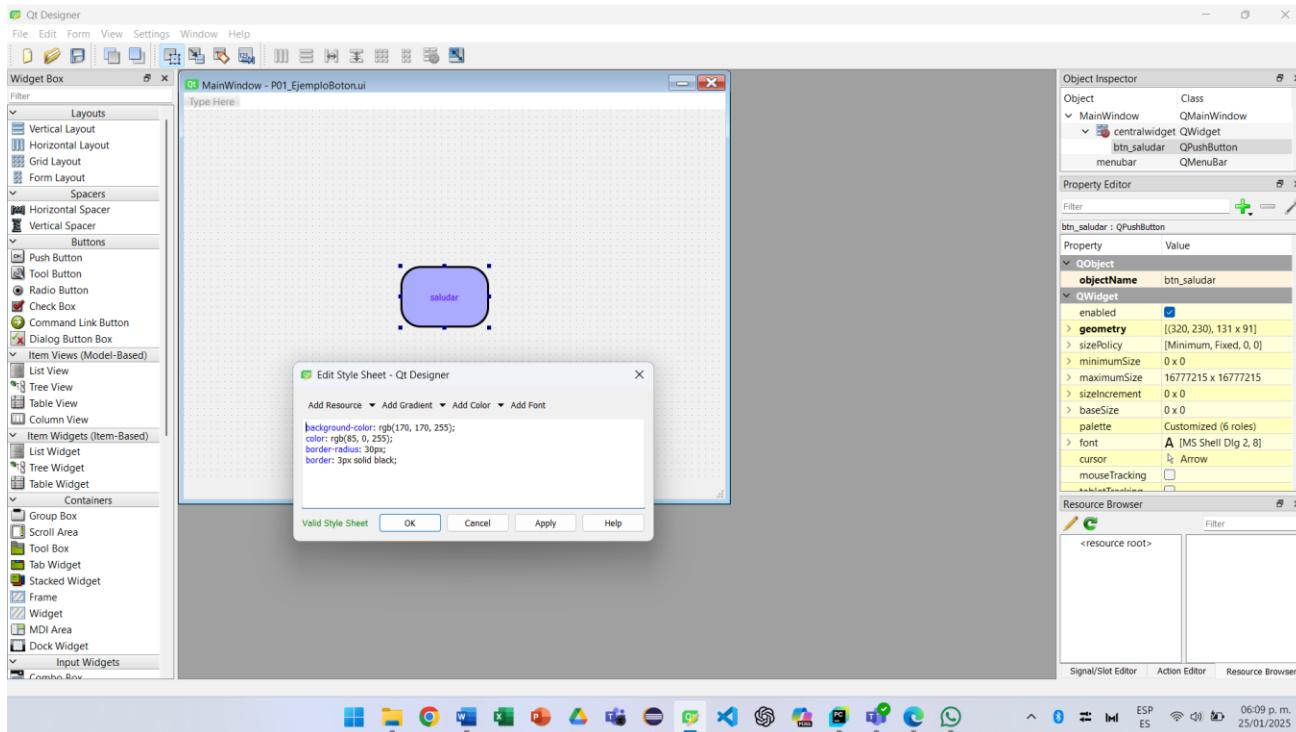
C1. P00_Intro



```
import sys
from PyQt5 import QtWidgets
qtCreatorFile = "P00_Intro.ui" # Nombre del archivo aquí.
Ui_MainWindow, QtBaseClass = QtWidgets.QUiLoader().load(qtCreatorFile)
class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
    def __init__(self):
        QtWidgets.QMainWindow.__init__(self)
        Ui_MainWindow.__init__(self)
        self.setupUi(self)
        # Área de los Signals
    # Área de los Slots
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MyApp()
    window.show()
    sys.exit(app.exec_())
```

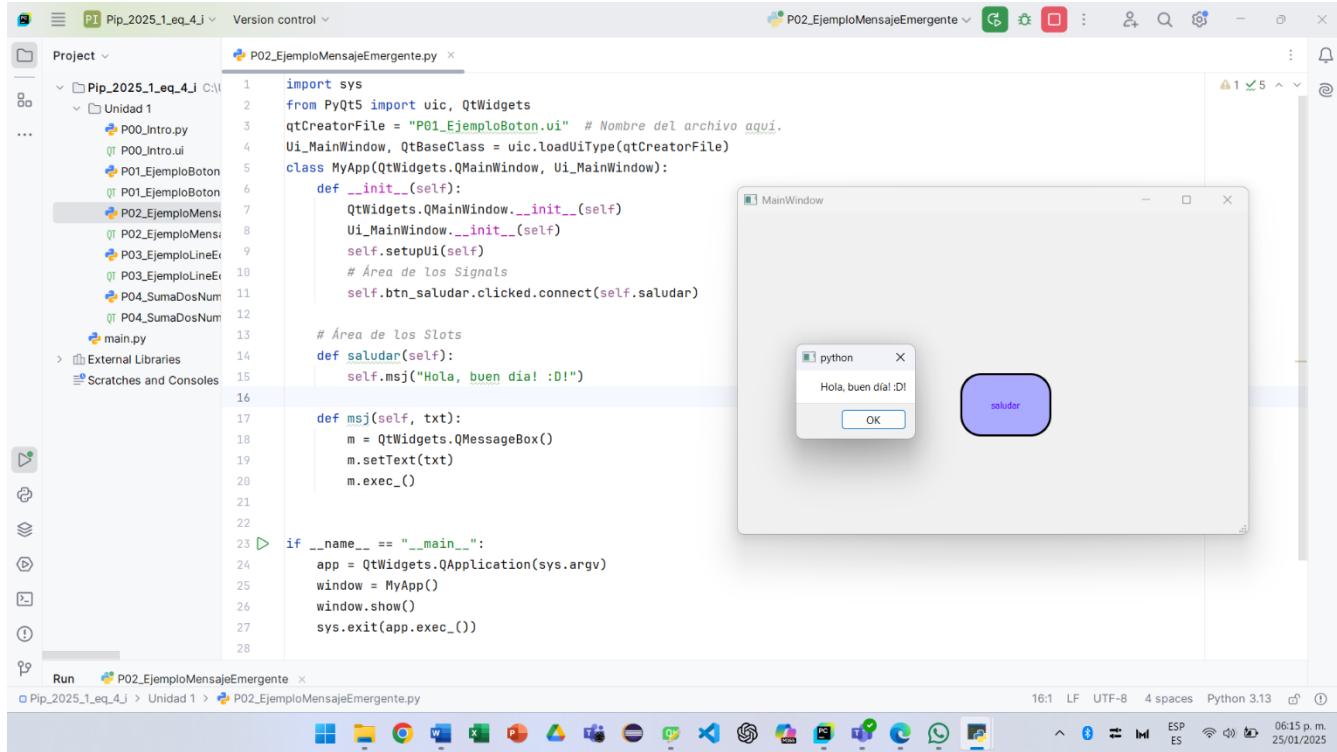


C2. P01_EjemploBoton





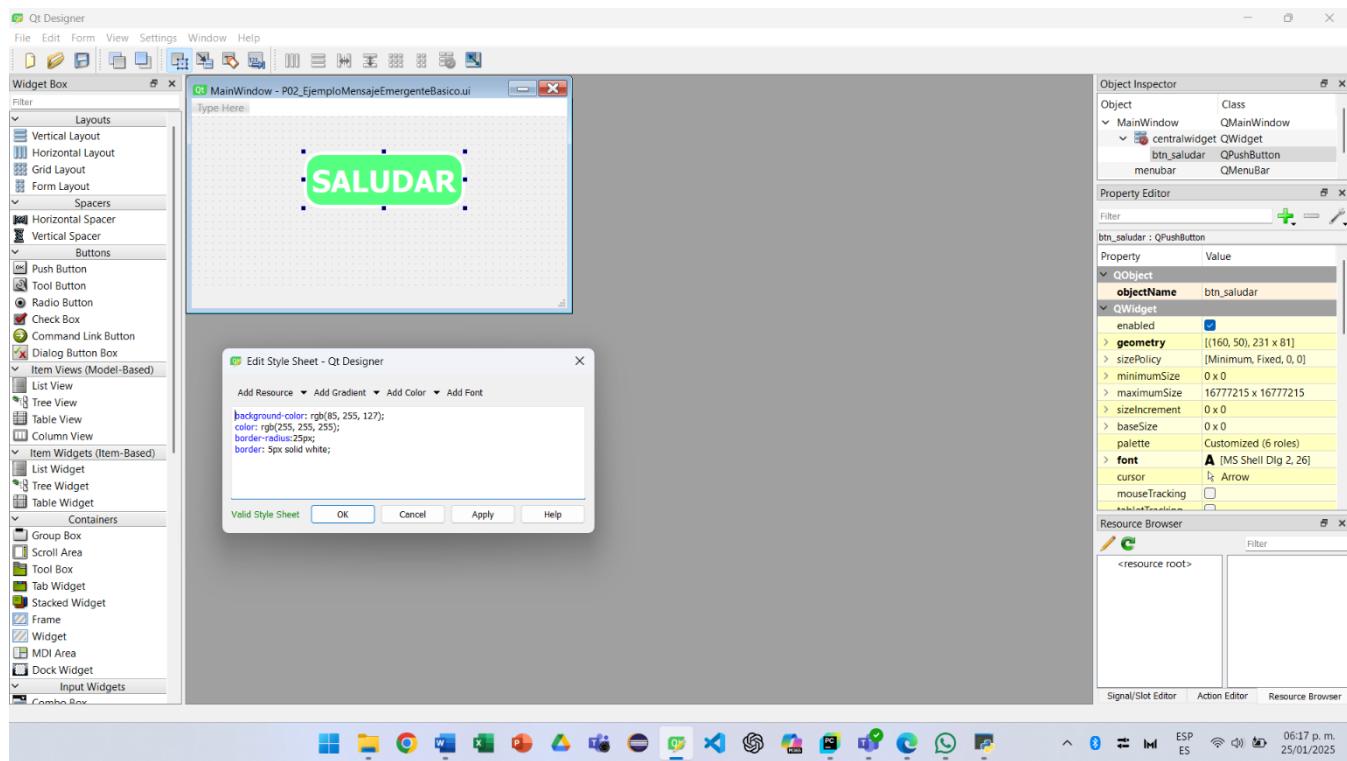
C3. P02_EjemploMensajeEmergente



The screenshot shows the PyCharm IDE interface. On the left is the project tree with files like P00_Intro.py, P01_EjemploBoton.py, P02_EjemploMensajeEmergente.py, P03_EjemploLineEdit.py, P04_SumaDosNum.py, and main.py. The right side shows the code editor for P02_EjemploMensajeEmergente.py:

```
1 import sys
2 from PyQt5 import uic, QtWidgets
3 qtCreatorFile = "P01_EjemploBoton.ui" # Nombre del archivo aquí.
4 Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
5 class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
6     def __init__(self):
7         QtWidgets.QMainWindow.__init__(self)
8         Ui_MainWindow.__init__(self)
9         self.setupUi(self)
10        # Área de los Signals
11        self.btn_saludar.clicked.connect(self.saludar)
12
13        # Área de los Slots
14    def saludar(self):
15        self.msj("Hola, buen día :D!")
16
17    def msj(self, txt):
18        m = QtWidgets.QMessageBox()
19        m.setText(txt)
20        m.exec_()
21
22
23 if __name__ == "__main__":
24     app = QtWidgets.QApplication(sys.argv)
25     window = MyApp()
26     window.show()
27     sys.exit(app.exec_())
```

A floating window titled "python" displays the message "Hola, buen día :D!". A purple button labeled "saludar" is highlighted in the application window.





C4. P03_EjemploLineEdit

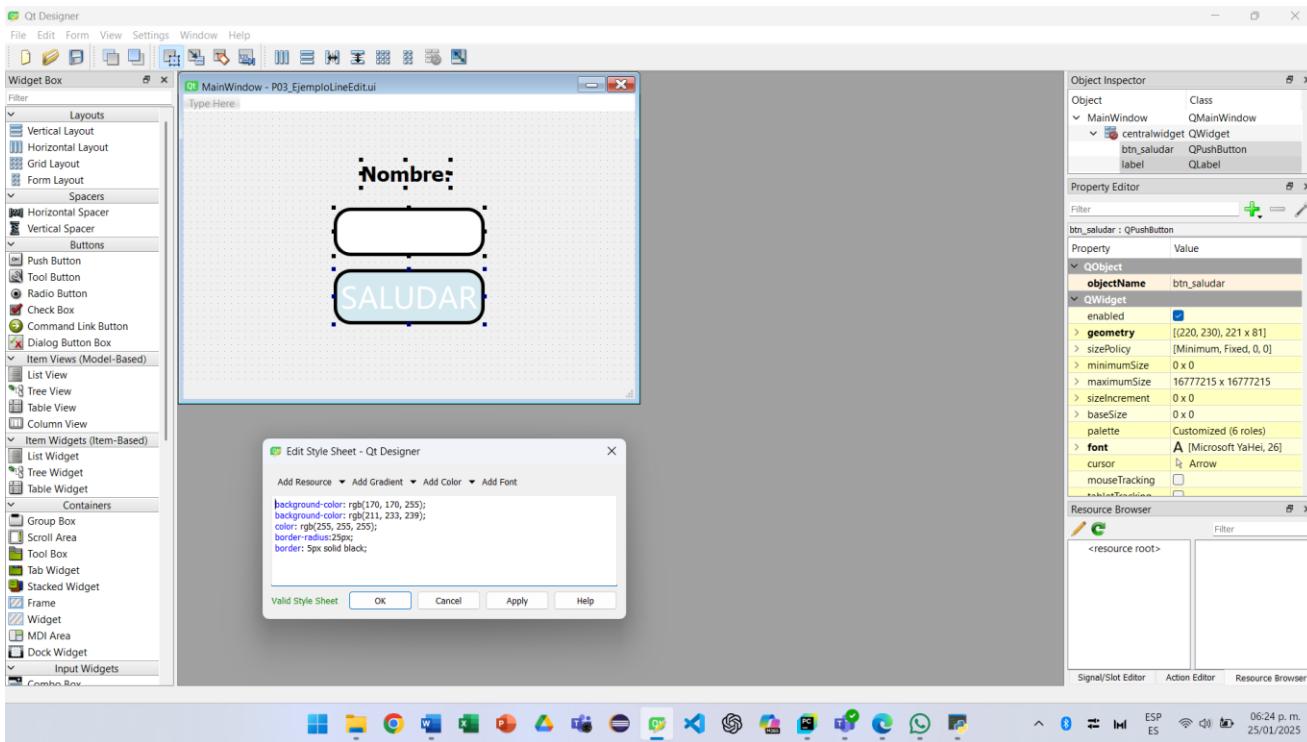
The screenshot shows the PyCharm IDE interface. On the left, the project structure for 'Pip_2025_1_eq_4_j' is visible, including files like P00_Intro.py, P01_EjemploBoton.py, P02_EjemploMensaje.py, P03_EjemploLineEdit.py (which is selected), and P04_SumaDosNum.py. The main editor window displays the Python code for P03_EjemploLineEdit.py. On the right, a screenshot of a running application window titled 'Nombre:' shows a text input field containing 'emanuel' and a button labeled 'SALUDAR'. A message box at the bottom left says 'Hola emanuel, buen dia!:D!'. The status bar at the bottom indicates the time is 13:24 and the system is running on Python 3.13.

```
import sys
from PyQt5 import uic, QtWidgets
qtCreatorFile = "P03_EjemploLineEdit.ui" # Nombre del archivo aquí.
Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
    def __init__(self):
        QtWidgets.QMainWindow.__init__(self)
        Ui_MainWindow.__init__(self)
        self.setupUi(self)
        # Área de los Signals
        self.btn_saludar.clicked.connect(self.saludar)

    # Área de los Slots
    def saludar(self):
        usage
        cadena = self.txt_nombre.text() # str
        if cadena== "":
            self.msj("Hola " + cadena + ", buen dia!:D!")
        else:
            self.msj("Debes ingresar tu nombre para continuar")

    def msj(self, txt): 2 usages
        m = QtWidgets.QMessageBox()
        m.setText(txt)
        m.exec_()

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MyApp()
    window.show()
    sys.exit(app.exec_())
```





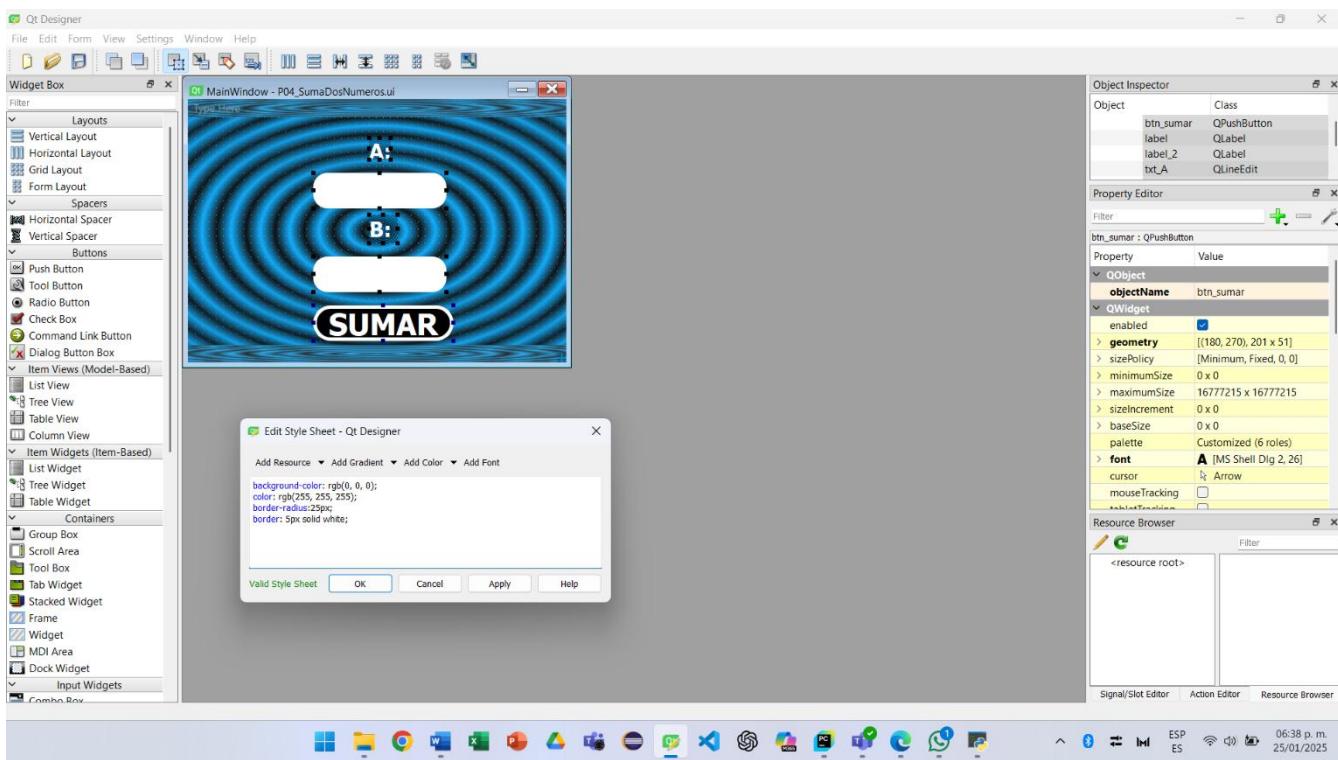
C5. P04_SumaDosNumeros

The screenshot shows the PyCharm IDE interface. On the left is the project tree with files like Pip_2025_1_eq_4_j, Unidad 1, and main.py. The main editor window displays the Python code for P04_SumaDosNumeros.py. A message box in the center says "La suma es: 11.0". To the right is a screenshot of a PyQt5 application window titled "MainWindow". It has two text input fields labeled "A:" and "B:", each containing the number "5" and "6" respectively. Below them is a large button labeled "SUMAR".

```
1 import sys
2 from PyQt5 import uic, QtWidgets
3 qtCreatorFile = "P04_SumaDosNumeros.ui" # Nombre del archivo aquí.
4 Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
5 class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
6     def __init__(self):
7         QtWidgets.QMainWindow.__init__(self)
8         Ui_MainWindow.__init__(self)
9         self.setupUi(self)
10        # Área de los Signals
11        self.btn_sumar.clicked.connect(self.sumar)
12
13    # Área de los Slots
14    def sumar(self): 1usage
15        try:
16            a = float(self.txt_A.text())
17            b = float(self.txt_B.text())
18            r = a+b
19            self.msj("La suma es: " + str(r))
20        except Exception as error:
21            print(error)
22
23    def msj(self, txt): 1usage
24        m = QtWidgets.QMessageBox()
25        m.setText(txt)
26        m.exec_()
27
28
29
30 if __name__ == "__main__":
31     app = QtWidgets.QApplication(sys.argv)
32     window = MyApp()
33     window.show()
34     sys.exit(app.exec_())
```

This screenshot is similar to the one above, showing the PyCharm IDE with the same project structure and code. However, the line of code `self.txt_A.text()` is highlighted in red, indicating a syntax error or a warning.

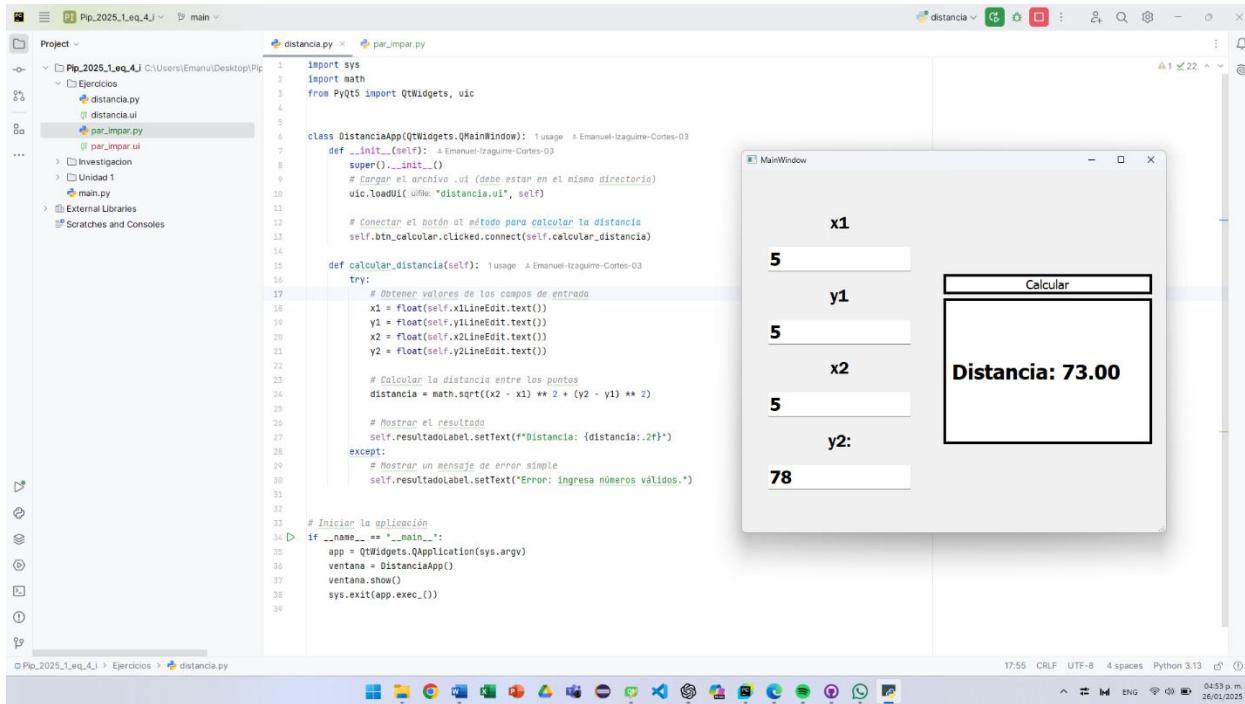
```
1 import sys
2 from PyQt5 import uic, QtWidgets
3 qtCreatorFile = "P04_SumaDosNumeros.ui" # Nombre del archivo aquí.
4 Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
5 class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
6     def __init__(self):
7         QtWidgets.QMainWindow.__init__(self)
8         Ui_MainWindow.__init__(self)
9         self.setupUi(self)
10        # Área de los Signals
11        self.btn_sumar.clicked.connect(self.sumar)
12
13    # Área de los Slots
14    def sumar(self): 1usage
15        try:
16            a = float(self.txt_A.text())
17            b = float(self.txt_B.text())
18            r = a+b
19            self.msj("La suma es: " + str(r))
20        except Exception as error:
21            print(error)
22
23    def msj(self, txt): 1usage
24        m = QtWidgets.QMessageBox()
25        m.setText(txt)
26        m.exec_()
27
28
29
30 if __name__ == "__main__":
31     app = QtWidgets.QApplication(sys.argv)
32     window = MyApp()
33     window.show()
34     sys.exit(app.exec_())
```





Ejercicios

Distancia de un punto a otro punto



The screenshot shows the PyCharm IDE interface. On the left is the project tree for 'Pip.2025_1_eq_4_i' containing files like 'distancia.py', 'par_impar.py', and 'main.py'. The main editor window displays Python code for a distance calculator application. The code imports sys, math, and QtWidgets. It defines a class 'DistanciaApp' that initializes a window with four input fields ('x1', 'y1', 'x2', 'y2') and a 'Calcular' button. It connects the button's click event to a method that calculates the distance using the formula $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$. The result is displayed in a label. The code also handles errors for invalid input. The bottom right shows the application running with inputs 5, 5, 5, 78 and output 'Distancia: 73.00'.

```
import sys
import math
from PyQt5 import QtWidgets, uic

class DistanciaApp(QtWidgets.QMainWindow):
    def __init__(self):
        super().__init__()
        # Cargar el archivo .ui (debe estar en el mismo directorio)
        uic.loadUi('distancia.ui', self)

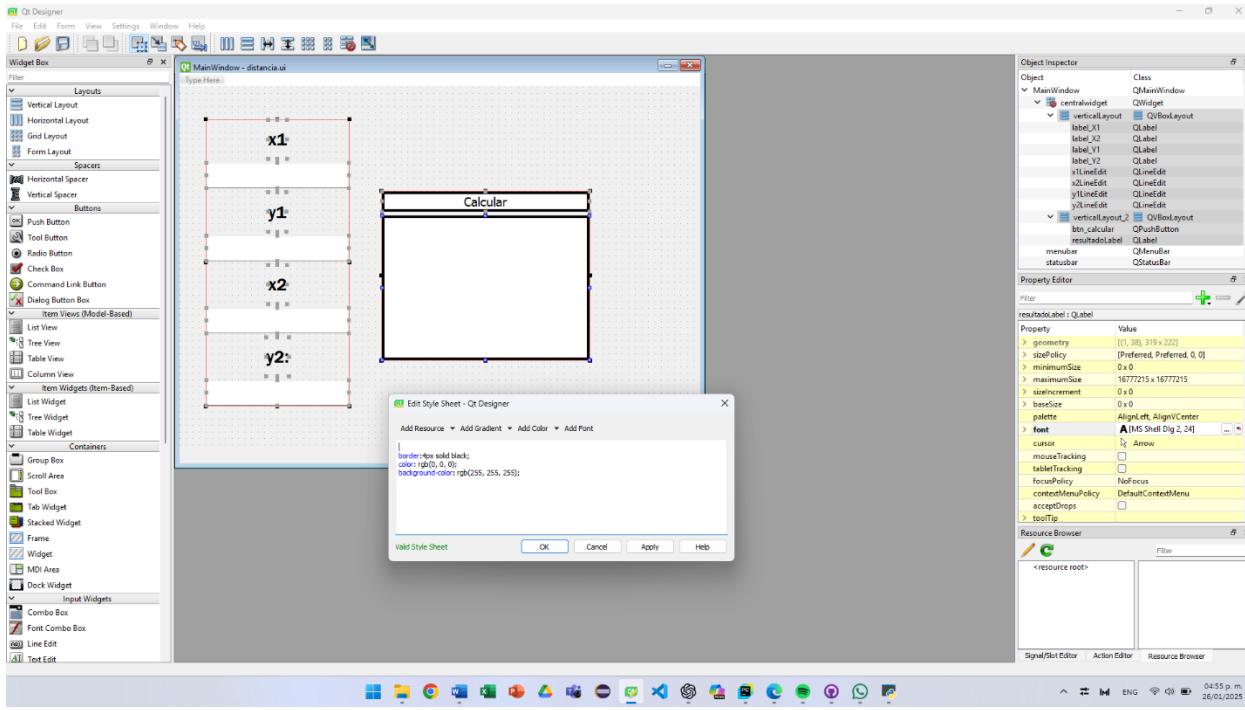
        # Conectar el botón al método para calcular la distancia
        self.btn_calcular.clicked.connect(self.calcular_distancia)

    def calcular_distancia(self):
        try:
            # Obtener valores de los campos de entrada
            x1 = float(self.xlineEdit.text())
            y1 = float(self.ylineEdit.text())
            x2 = float(self.x2lineEdit.text())
            y2 = float(self.y2lineEdit.text())

            # Calcular la distancia entre los puntos
            distancia = math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)

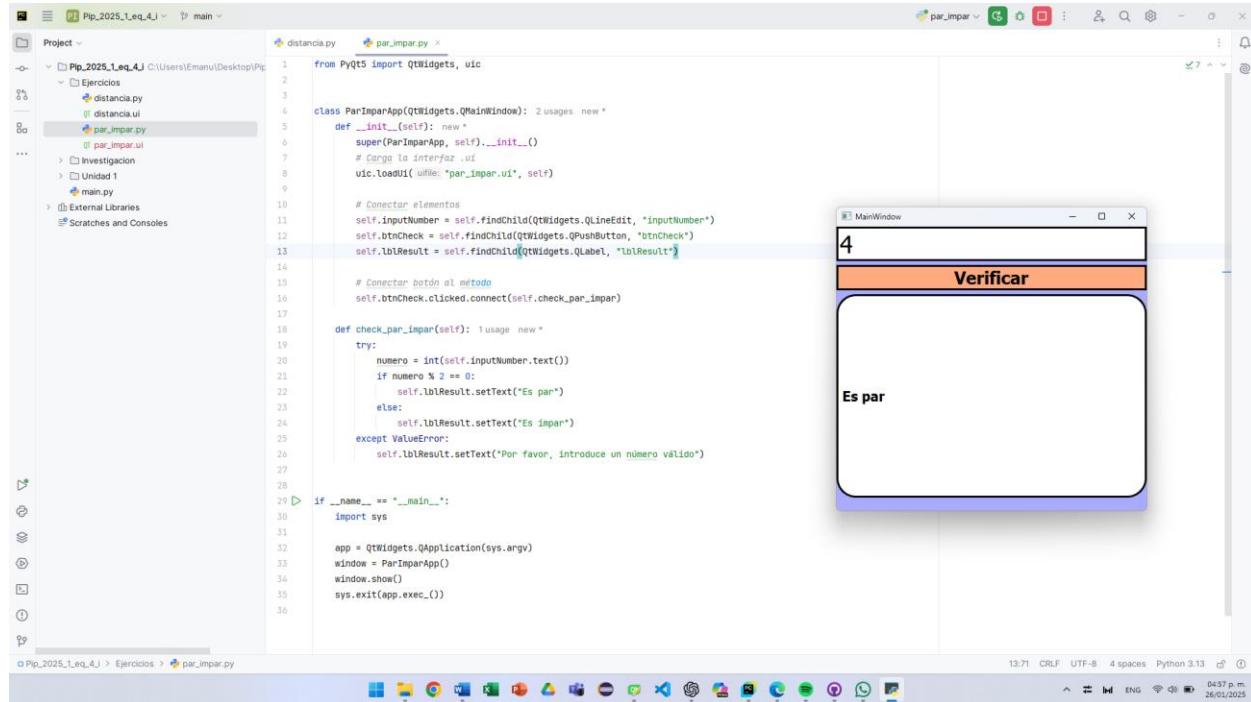
            # Mostrar el resultado
            self.resultadoLabel.setText(f'Distancia: {distancia:.2f}')
        except:
            # Mostrar un mensaje de error simple
            self.resultadoLabel.setText('Error: ingresa números válidos.')

    # Iniciar la aplicación
if __name__ == '__main__':
    app = QtWidgets.QApplication(sys.argv)
    ventana = DistanciaApp()
    ventana.show()
    sys.exit(app.exec_())
```





Par o impar



```
Project Pip_2025_1_eq_4 J C:\Users\Emanu\Desktop\Pip_2025_1_eq_4 J
- Ejercicios
  - distancia.py
  - par_impar.ui
  - par_impar.py
  - main.py
  - man.py
  - External Libraries
  - Scratches and Consoles

distancia.py
par_impar.py

class ParImparApp(QtWidgets.QMainWindow):
    def __init__(self):
        super(ParImparApp, self).__init__()
        # Carga la interfaz .ui
        uic.loadUi('par_impar.ui', self)

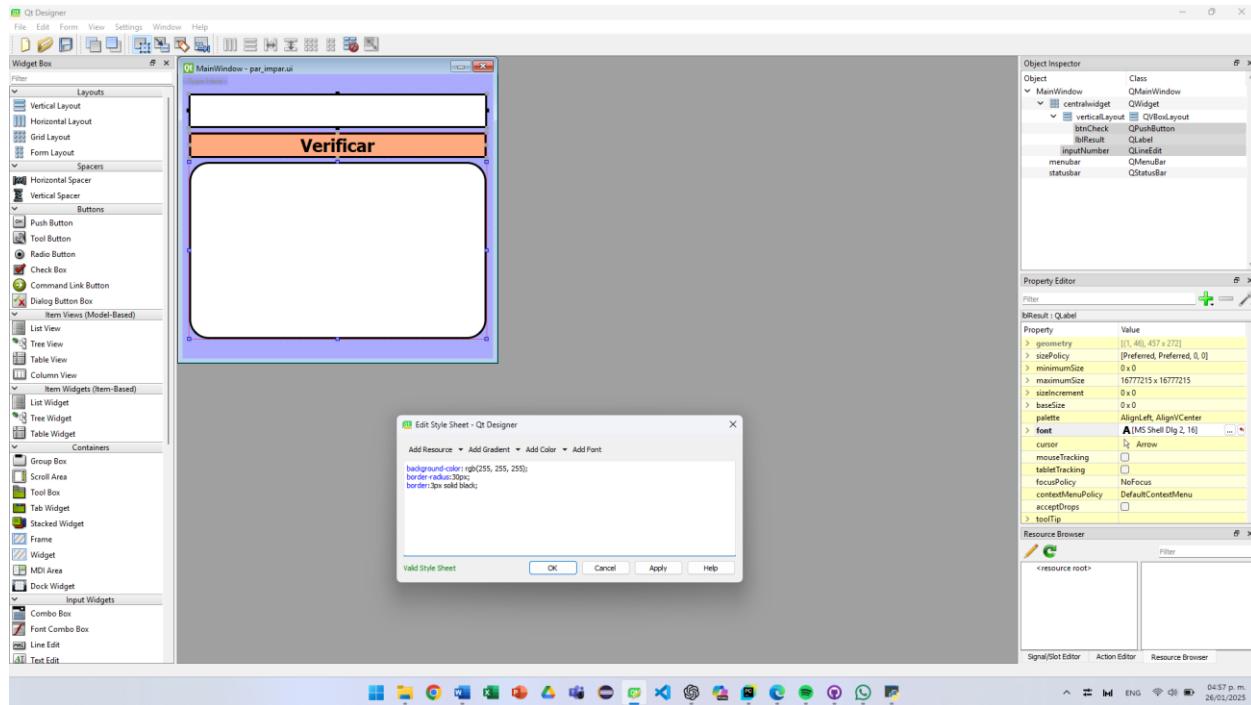
        # Conectar elementos
        self.inputNumber = self.findChild(QtWidgets.QLineEdit, "inputNumber")
        self.btnCheck = self.findChild(QtWidgets.QPushButton, "btnCheck")
        self.lblResult = self.findChild(QtWidgets.QLabel, "lblResult")

        # Conectar botón al método
        self.btnCheck.clicked.connect(self.check_par_impar)

    def check_par_impar(self):
        try:
            numero = int(self.inputNumber.text())
            if numero % 2 == 0:
                self.lblResult.setText("Es par")
            else:
                self.lblResult.setText("Es impar")
        except ValueError:
            self.lblResult.setText("Por favor, introduce un número válido")

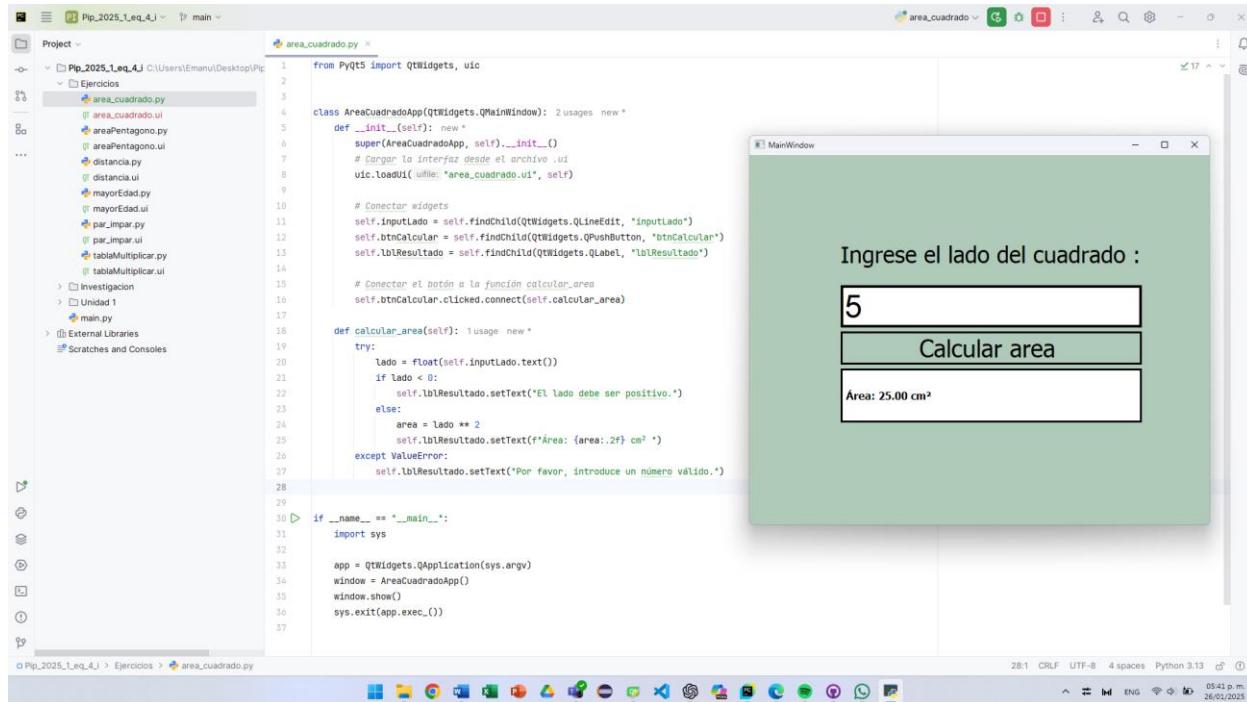
if __name__ == "__main__":
    import sys

    app = QtWidgets.QApplication(sys.argv)
    window = ParImparApp()
    window.show()
    sys.exit(app.exec_())
```





Area de un cuadrado



The screenshot shows the PyCharm IDE interface. On the left is the Project tool window displaying a file structure for a Python project named 'Pip_2025_1_eq_4'. Inside the 'Ejercicios' folder, there is a 'area_cuadrado' folder containing 'area_cuadrado.py' and 'area_cuadrado.ui'. The main editor window shows the Python code for a PyQt5 application. The code imports QtWidgets and uic, defines a class 'AreaCuadradoApp' that loads a UI file, and contains methods for calculating the area based on user input. A terminal window at the bottom shows the command 'python area_cuadrado.py'. To the right, a screenshot of the application's window titled 'MainWindow' is shown. It has a text input field containing '5', a button labeled 'Calcular area', and a result label below it showing 'Area: 25.00 cm²'.

```
from PyQt5 import QtWidgets, uic
import sys

class AreaCuadradoApp(QtWidgets.QMainWindow):
    def __init__(self):
        super(AreaCuadradoApp, self).__init__()
        # Cargar la interfaz desde el archivo .ui
        uic.loadUi('area_cuadrado.ui', self)

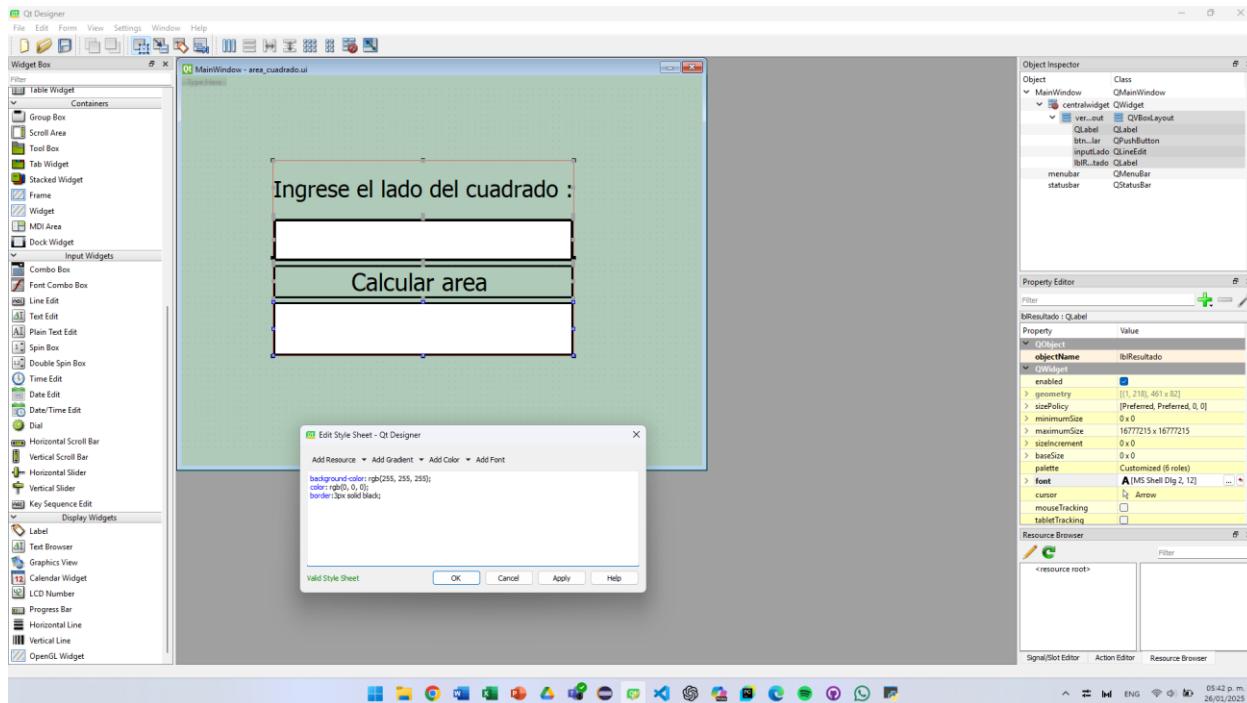
    # Conectar widgets
    self.inputLado = self.findChild(QtWidgets.QLineEdit, "inputLado")
    self.btnCalcular = self.findChild(QtWidgets.QPushButton, "btnCalcular")
    self.lblResultado = self.findChild(QtWidgets.QLabel, "lblResultado")

    # Conectar el botón a la función calcular_area
    self.btnCalcular.clicked.connect(self.calcular_area)

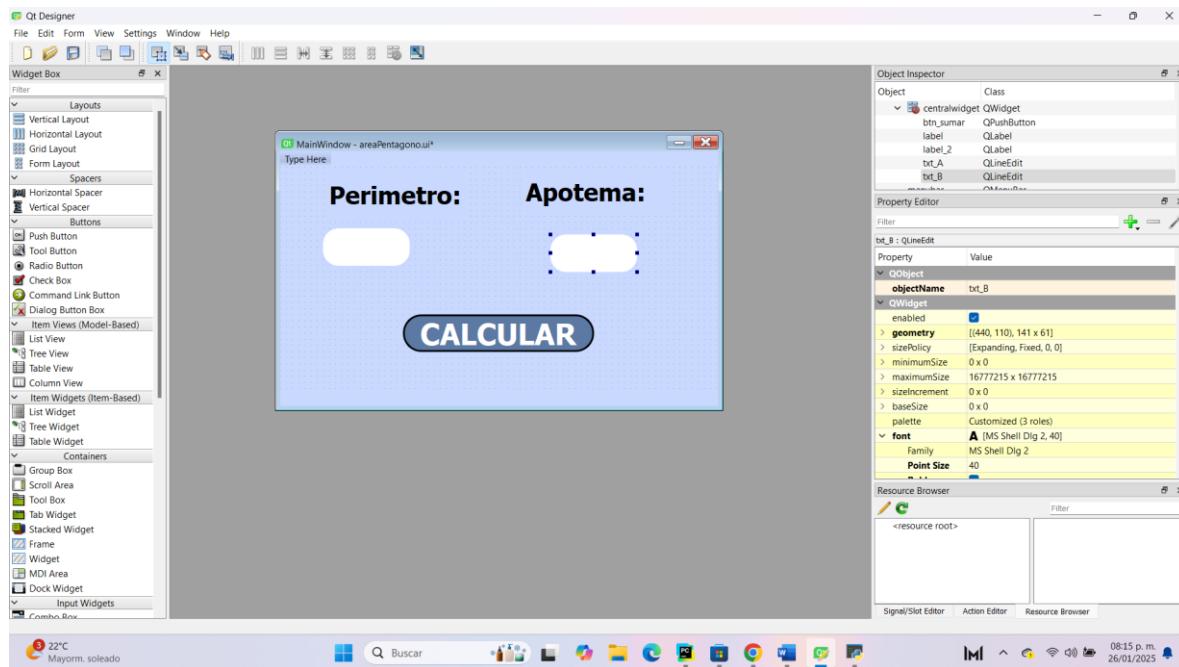
    def calcular_area(self):
        lado = float(self.inputLado.text())
        if lado < 0:
            self.lblResultado.setText("El lado debe ser positivo.")
        else:
            area = lado ** 2
            self.lblResultado.setText(f"Área: {area:.2f} cm2")
        except ValueError:
            self.lblResultado.setText("Por favor, introduce un número válido.")

if __name__ == "__main__":
    import sys

    app = QtWidgets.QApplication(sys.argv)
    window = AreaCuadradoApp()
    window.show()
    sys.exit(app.exec_())
```



Áreas de un pentágono



Project

- PIP_2025_1_Eq_0_J C:\Users\gilbe\Desktop\PIP_2025_1_Eq_0_J
- Formatos
- tarea
 - areaPentagono.py
 - areaPentagono.ui
 - mayorEdad.py
 - mayorEdad.ui
 - tablaMultiplicar.py
 - tablaMultiplicar.ui
- Unidad 1
 - P00_Introduccion.py
 - P00_Introduccion.ui
 - P01_EjemploConUnBoton.py
 - P01_EjemploConUnBoton.ui
 - P02_MensajeEmergenteBasico.py
 - P02_MensajeEmergenteBasico.ui
 - P03_CajaDeTextoLineEdit.py
 - P03_CajaDeTextoLineEdit.ui
 - P04_SumaDeDosNumeros.py
 - P04_SumaDeDosNumeros.ui
- External Libraries
- Scratches and Consoles

areaPentagono.py

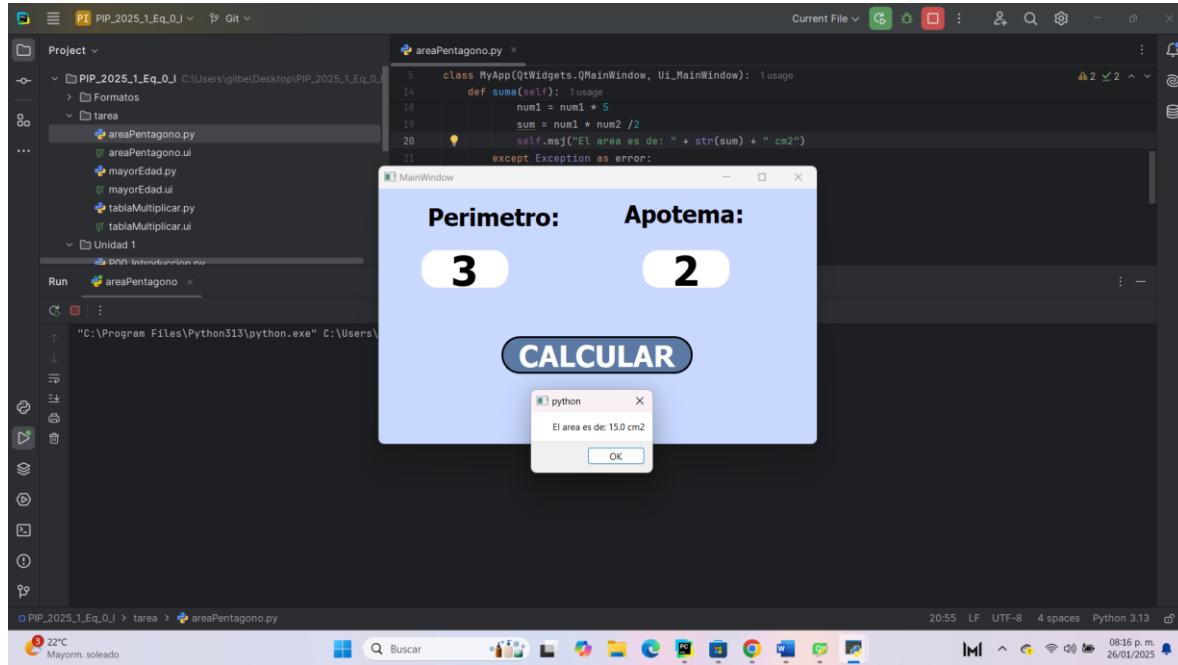
```
1 > import ...
2 qtCreatorFile = "AreaPentagono.ui" # Nombre del archivo aquí.
3 Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
4
5 class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
6     def __init__(self):
7         QtWidgets.QMainWindow.__init__(self)
8         Ui_MainWindow.__init__(self)
9         self.setupUi(self)
10        # Área de los Signals
11        self.btn_sumar.clicked.connect(self.suma)
12
13    # Área de los Slots
14    def suma(self):  Usage
15        try:
16            num1 = float(self.txt_A.text())
17            num2 = float(self.txt_B.text())
18            num1 = num1 * 5
19            sum = num1 * num2 / 2
20            self.msj("El área es de: " + str(sum) + " cm2")
21        except Exception as error:
22            print(error)
23
24    def msj(self, txt):  Usage
25        m = QtWidgets.QMessageBox()
26        m.setText(txt)
27        m.exec_()
28
29 if __name__ == "__main__":
30     app = QtWidgets.QApplication(sys.argv)
31     window = MyApp()
32     window.show()
33     sys.exit(app.exec_())
```

PIP_2025_1_Eq_0_J > tarea > areaPentagono.py

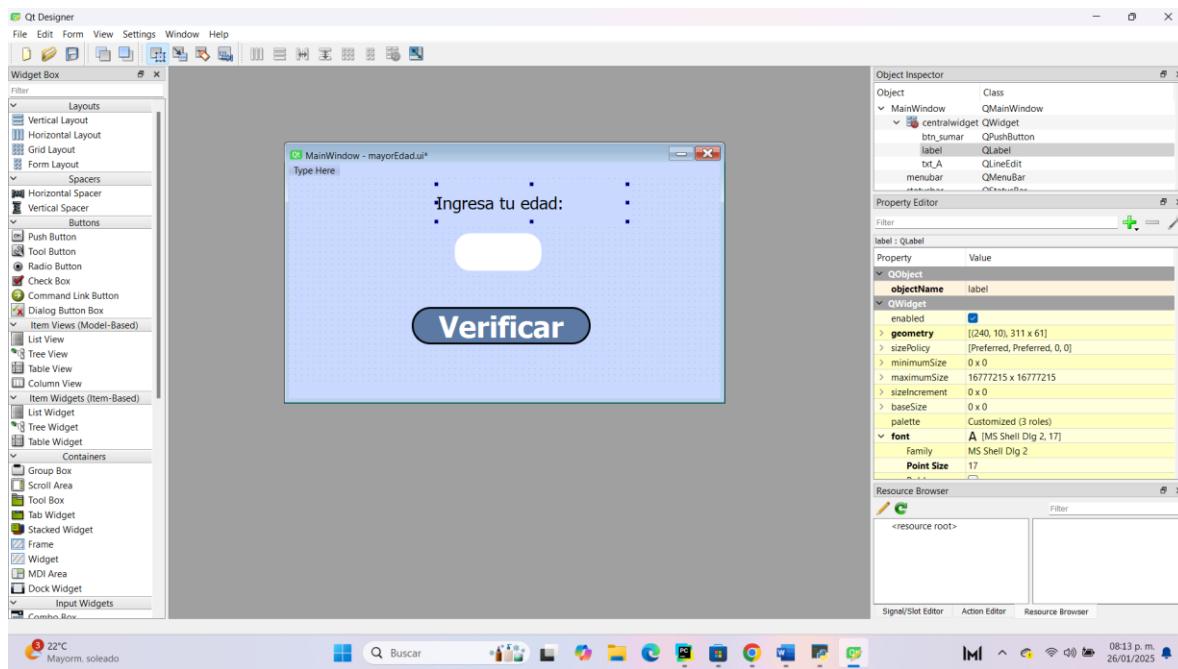
Current File ▾

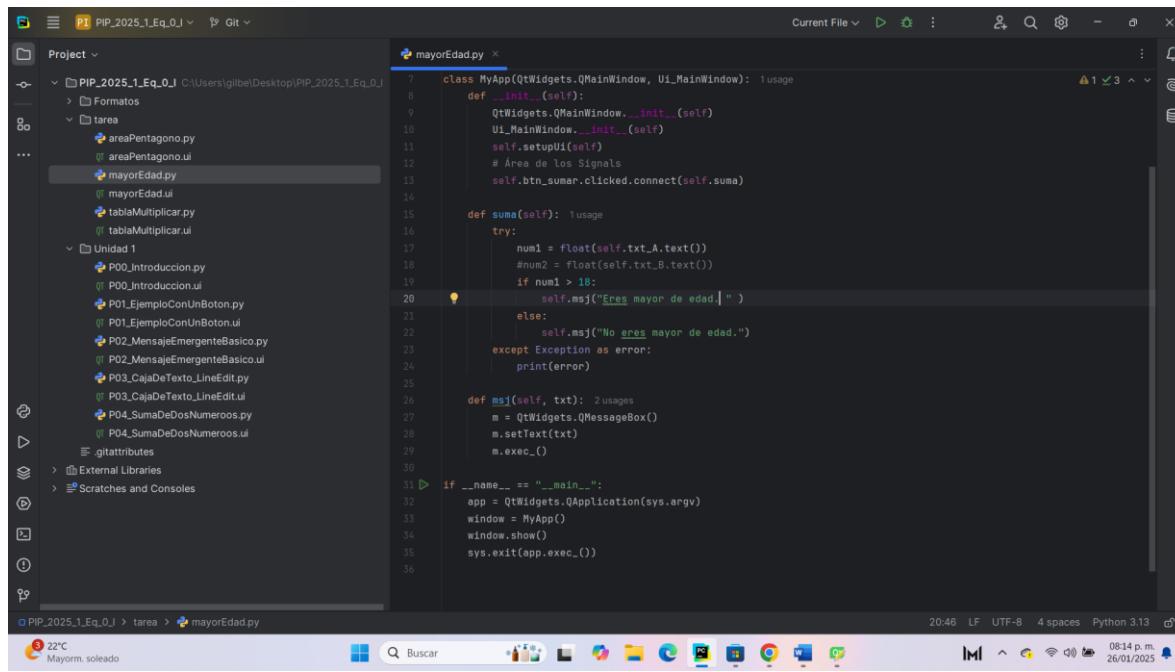
20:55 LF UTF-8 4 spaces Python 3.13

22°C Mayor, soleado 08:16 p. m. 26/01/2025



Comprobar si es mayor de edad una persona



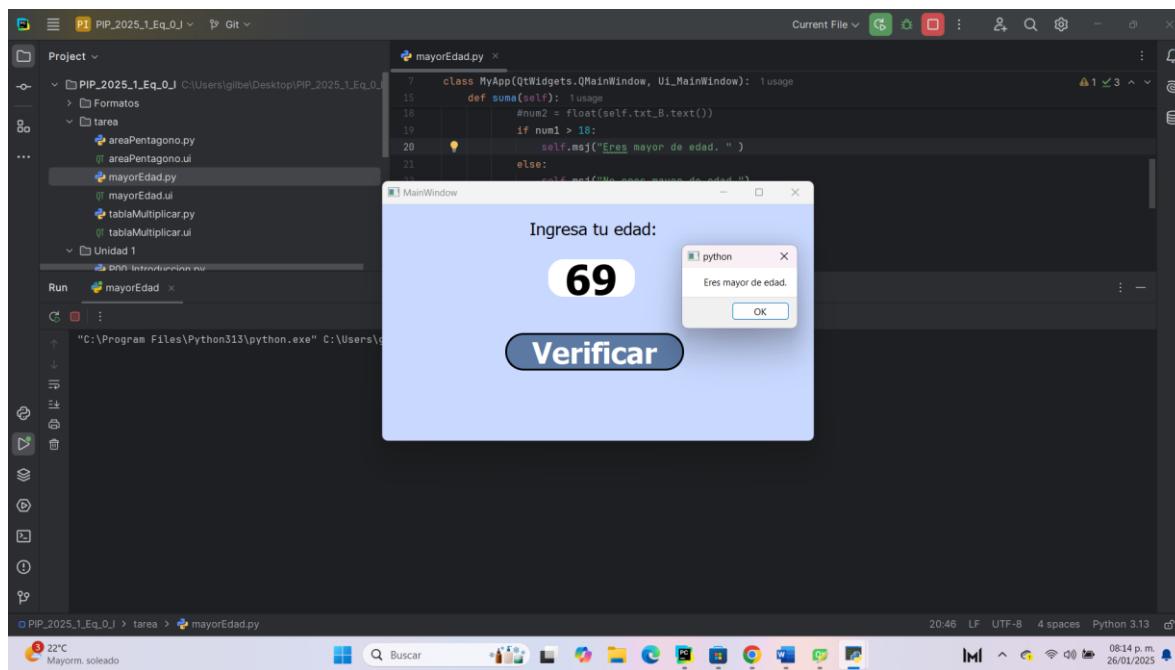


```
class MyApp(QtWidgets.QMainWindow, Ui_MainWindow): 1 usage
    def __init__(self):
        QtWidgets.QMainWindow.__init__(self)
        self.setupUi(self)
        # Área de los Signals
        self.btn_sumar.clicked.connect(self.suma)

    def suma(self): 1 usage
        try:
            num1 = float(self.txt_A.text())
            #num2 = float(self.txt_B.text())
            if num1 > 18:
                self.msj("Eres mayor de edad. ")
            else:
                self.msj("No eres mayor de edad. ")
        except Exception as error:
            print(error)

    def msj(self, txt): 2 usages
        m = QtWidgets.QMessageBox()
        m.setText(txt)
        m.exec_()

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    window = MyApp()
    window.show()
    sys.exit(app.exec_())
```



Ingresá tu edad:

69

Verificar

Eres mayor de edad.

```
class MyApp(QtWidgets.QMainWindow, Ui_MainWindow): 1 usage
    def suma(self): 1 usage
        #num2 = float(self.txt_B.text())
        if num1 > 18:
            self.msj("Eres mayor de edad. ")
        else:
            self.msj("No eres mayor de edad. ")
```



Imprimir en consola/MessageBox una tabla de multiplicar

The screenshot shows the Qt Designer interface. In the center, there is a window titled "Main Window - tablaMultiplicar.ui" with the following content:

```
Ingresá la tabla de multiplicar que deseás:  
CALCULAR
```

On the left, the "Widget Box" contains various UI components like Push Button, Radio Button, Check Box, etc. On the right, the "Object Inspector" lists the objects: MainWindow, centralwidget, btn_sumar, label, txt_A, and menubar. The "Property Editor" shows properties for the "label" object, including "font" set to "MS Shell Dlg 2" and "Point Size" set to 17. The "Resource Browser" is also visible.

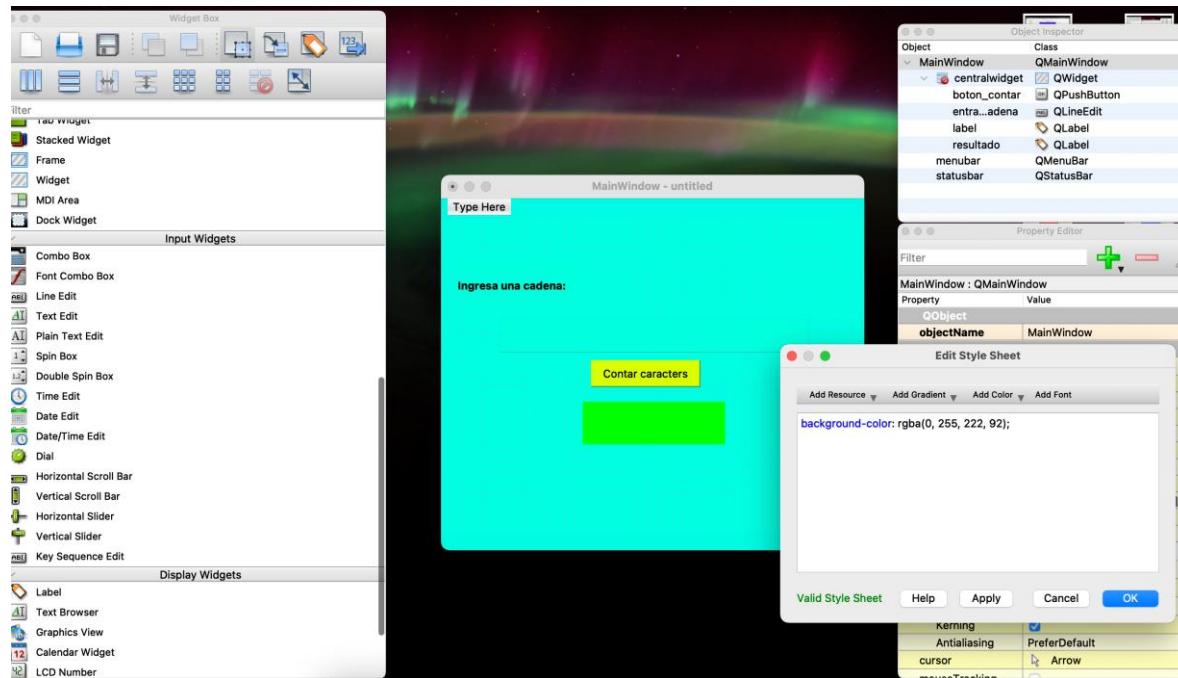
The screenshot shows the PyCharm IDE. The project structure on the left includes files like "areaPentagono.py", "mayorEdad.py", "tablaMultiplicar.py", and "tablaMultiplicar.ui". The code editor on the right contains the following Python script:

```
import sys
from PyQt5 import uic, QtWidgets
qtCreatorFile = "tablaMultiplicar.ui" # Nombre del archivo aquí.
Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
class MyApp(QtWidgets.QMainWindow, Ui_MainWindow):
    def __init__(self):
        QtWidgets.QMainWindow.__init__(self)
        Ui_MainWindow.__init__(self)
        self.setupUi(self)
        # Área de los Signals
        self.btn_sumar.clicked.connect(self.suma)

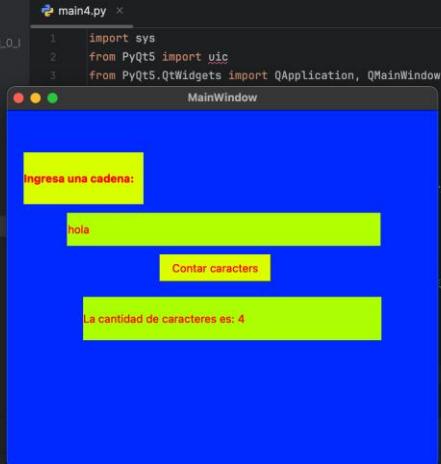
    # Área de los Slots
    def suma(self):
        try:
            num1 = int(self.txt_A.text())
            resultado = f"Tabla de multiplicar del {num1}\n"
            for i in range(1, 11):
                resultado += f"{num1} x {i} = {num1 * i}\n"
            self.msj(resultado)
        except ValueError:
            self.msj("Por favor, ingrese un número válido.")
        except Exception as error:
            print(error)

    def msj(self, txt):
        m = QtWidgets.QMessageBox()
        m.setText(txt)
        m.exec_()
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    myApp = MyApp()
    myApp.show()
    sys.exit(app.exec_())
```

cantidad de caracteres de una cadena



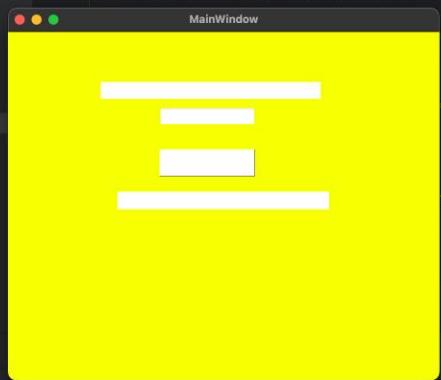
15



```
main4.py
1 import sys
2 from PyQt5 import uic
3 from PyQt5.QtWidgets import QApplication, QMainWindow
```

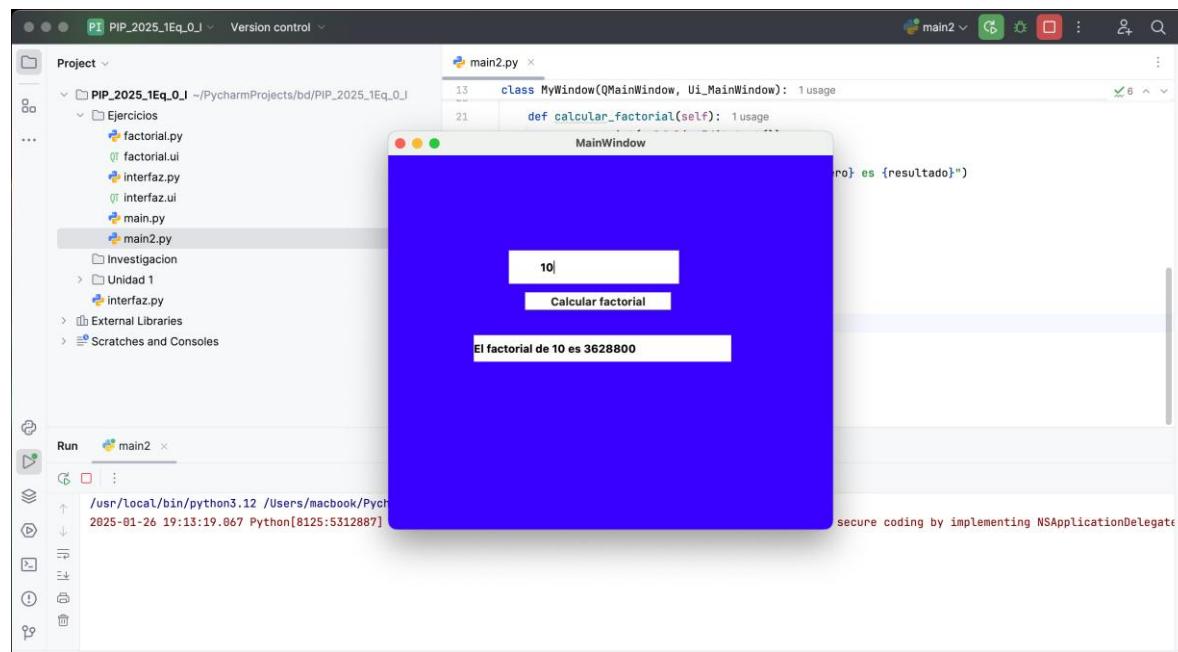
The screenshot shows a PyCharm interface with a project named "PIP_2025_1Eq_0_I". The "Ejercicios" folder contains several files: factorial.py, Interfaz.py, interfaz.ui, interfaz2.ui, main.py, main2.py, main3.py, and main4.py. The "Run" tab shows the application "main4" is selected. The terminal output shows the command run and a warning about secure coding. The application window has a title "MainWindow" and displays the text "Ingresa una cadena:" in a yellow box, "holo" in a green box, and "Contar caracteres" in a yellow box. Below that, it says "La cantidad de caracteres es: 4" in a yellow box.

Cuantas horas le restan al día para terminarse



```
main3.py
1 import sys
2 from PyQt5 import uic
3 from PyQt5.QtWidgets import QApplication, QMainWindow
```

The screenshot shows a PyCharm interface with a project named "PIP_2025_1Eq_0_I". The "Ejercicios" folder contains several files: factorial.py, Interfaz.py, interfaz.ui, interfaz2.ui, main.py, main2.py, main3.py, and ventana.ui. The "Run" tab shows the application "main3" is selected. The terminal output shows the command run and a warning about secure coding. The application window has a title "MainWindow" and displays four horizontal white bars of decreasing length.



Calcular la factorial de un número

