

INSTITUTO POLITÉCNICO NACIONAL

Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas

Programación avanzada

Práctica 4: Interfaces Gráficas de Usuario.
Controles Básicos

Adalberto Cabrera Vazquez

Boleta: 2023640791

Ingeniería Mecatrónica, Grupo 2MV7

29 de Marzo del 2024

1 Objetivo

Crear aplicaciones con interfaces gráficas de usuario (GUI) utilizando los controles básicos.

2 Introducción

2.1 Interfaces gráficas de usuario (GUI):

Una graphical user interface o GUI es una interfaz que se puede utilizar para controlar PC, tabletas y otros dispositivos. Las GUI utilizan elementos gráficos como iconos, menús e imágenes para facilitar el manejo del usuario humano. Tanto los sistemas operativos como las aplicaciones utilizan una interfaz gráfica de usuario. De hecho, casi todos los programas para usuarios finales de hoy en día vienen con esta interfaz.

La GUI es una interfaz de usuario que permite a los usuarios comunicarse con el ordenador. Suele estar basada en la interacción a través del ratón y el teclado (aunque el control a través de gestos es cada vez más común): al mover el ratón, el puntero se desplaza en la pantalla. La señal del dispositivo se transmite al ordenador, que luego la traduce en un movimiento equivalente en la pantalla. Por ejemplo, si un usuario hace clic en un determinado icono de programa en el menú, se ejecuta la instrucción correspondiente y se abre el programa.

La GUI es, por lo tanto, una especie de traductor en la comunicación entre el humano y la máquina. Sin la GUI, tendrías que utilizar la línea de comandos para llamar a programas y aplicaciones.

2.2 Qt Designer:

QT Designer es un programa que permite desarrollar interfaces gráficas de usuario (GUI). Permittiéndonos componer y personalizar las ventanas o cuadros de diálogo para su empleo en QGIS (por ejemplo). Qt se encuentra disponible para sistemas tipo unix (Linux, BSDs, Unix), Para Apple Mac OS X y Windows.

3 Desarrollo

Crear una aplicacion que permita realizar las 4 operaciones básicas: suma, resta, division y multipli-cacion.

Primero debir realizar el diseño de la calculadora (reutilice la que vimos en clase) y fui dandole nombre a cada uno de los botones.

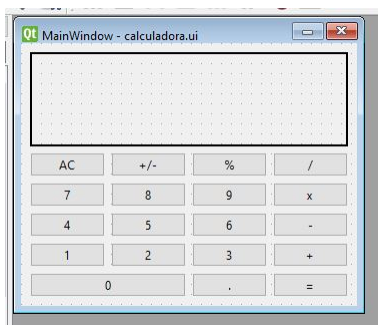


Figure 1: GUI de la Calculadora

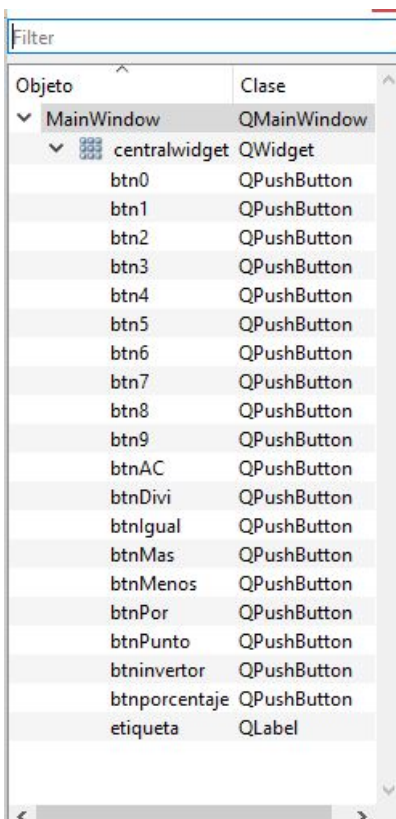


Figure 2: Botones y sus nombres

Ahora nos pasaremos al Visual Studio Code para realizar la parte de programación de la calculadora. Primero debemos de llamar a las librerías necesarias. Ahora vamos a a crear la clase MainWindow que

```
from PyQt6.QtGui import QKeyEvent
from Ui_calculadora import *
from PyQt6.QtWidgets import QMainWindow
from PyQt6 import QtCore
import sys
```

Figure 3: Librerías

hereda de las super clases QMainWindow y Ui.Mainwindow para poder así tener todos sus atributos. Esto se hace para que modifiquemos otro archivo de python en lugar del original que nos brinda Qt Designer.

Las líneas de código:

self.'boton_que_quiera'.connect(self.'funcion_definida') (1)

sirve para que cada que damos click en el boton definido, haga cierta acción predeterminada por una función que mas adelante serán definidas. Estos lo usaremos en cada boton de nuestra calculadora como muestra la siguiente imagen.

```

class MainWindow(QMainWindow, Ui_MainWindow):
    def __init__(self, *parent, **flags) -> None:
        super().__init__(*parent, **flags)
        self.setupUi(self)

        self.btn0.clicked.connect(self.setNumero)
        self.btn1.clicked.connect(self.setNumero)
        self.btn2.clicked.connect(self.setNumero)
        self.btn3.clicked.connect(self.setNumero)
        self.btn4.clicked.connect(self.setNumero)
        self.btn5.clicked.connect(self.setNumero)
        self.btn6.clicked.connect(self.setNumero)
        self.btn7.clicked.connect(self.setNumero)
        self.btn8.clicked.connect(self.setNumero)
        self.btn9.clicked.connect(self.setNumero)

        self.btnPunto.clicked.connect(self.setPunto)

        self.btnAC.clicked.connect(self.setCero)

        self.btnMas.clicked.connect(self.setMas)
        self.btnMenos.clicked.connect(self.setMenos)
        self.btnPor.clicked.connect(self.setMultiplicacion)
        self.btnDivi.clicked.connect(self.setDivision)
        self.btnporcentaje.clicked.connect(self.setPorcentaje)
        self.btninvertedor.clicked.connect(self.setInv)
        self.btnIgual.clicked.connect(self.setIgual)

```

Figure 4: Sub clase MainWindow

Y empezamos con las funciones para cada boton.

SetCero: Sirve para el boton de AC cuando queremos limpiar la pantalla de la calculadora.

```

def setCero(self):
    self.etiqueta.setText("")

```

Figure 5: Boton de AC

SetMenos: Sirve para poner el menos pero primero evalua que no haya ningun operando antes para que lo pueda escribir y agrega un '-' al final del texto.

```
def setMenos(self):
    cadena = self.etiqueta.text()
    if cadena.find('+')!=-1 or cadena.find('-')!=-1 or cadena.find('*')!=-1 or cadena.find('/')!=-1 or cadena.find('%')!=-1:
        pass
    else:
        cadena = cadena + "-"
        self.etiqueta.setText(cadena)
```

Figure 6: Boton de menos

SetMas: Funciona de la misma manera que el Set Menos pero aqui agrega un '+'

```
def setMas(self):
    cadena = self.etiqueta.text()
    if cadena.find('+')!=-1 or cadena.find('-')!=-1 or cadena.find('*')!=-1 or cadena.find('/')!=-1 or cadena.find('%')!=-1:
        pass
    else:
        cadena = cadena + "+"
        self.etiqueta.setText(cadena)
```

Figure 7: Boton más

SetMultiplicación: Al igual que los anteriores, hace lo mismo pero agrega un '*'

```
def setMultiplicacion(self):
    cadena = self.etiqueta.text()
    if cadena.find('+')!=-1 or cadena.find('-')!=-1 or cadena.find('*')!=-1 or cadena.find('/')!=-1 or cadena.find('%')!=-1:
        pass
    else:
        cadena = cadena + "*"
        self.etiqueta.setText(cadena)
```

Figure 8: Boton multiplicacion

SetDivision: Igual que las anteriores y agregar un '/'

```
def setDivision(self):
    cadena = self.etiqueta.text()
    if cadena.find('+')!=-1 or cadena.find('-')!=-1 or cadena.find('*')!=-1 or cadena.find('/')!=-1 or
        pass
    else:
        cadena = cadena + "/"
        self.etiqueta.setText(cadena)
```

Figure 9: Boton división

SetPorcentaje: Agrega el '%'

```
def setPorcentaje(self):
    cadena = self.etiqueta.text()
    if cadena.find('+')!=-1 or cadena.find('-')!=-1 or cadena.find('*')!=-1 or cadena.find('/')!=-1 or cadena.
        pass
    else:
        cadena = cadena + "%"
        self.etiqueta.setText(cadena)
```

Figure 10: Boton porcentaje

SetInv: Agrega el '(inv)'

```
def setInv(self):
    cadena=self.etiqueta.text()
    if cadena.find('*')!=-1 or cadena.find('/')!=-1 or cadena.find('%')!=-1 or cadena.find('(+/-)')!=-1:
        pass
    else:
        cadena=cadena+' (inv)'
        self.etiqueta.setText(cadena)
```

Figure 11: Boton inversa

SetPunto: Esta cambia un poco ya que com tal la calculadora no te deja cometer el error de meter mas de un punto decimal en cada numero asi que primero evalua si hay un punto decimal, si hay un punto decimal y no hay algun operador entonces no te deja escribirlo, pero si hay un punto decimal y un operador entonces si te deja escribirlo. Y por ultimo si hay un punto decimal, un operador y otro punto decimal, entonces ya no te deja escribir otro mas.

```

def setPunto(self):
    cadena = self.etiqueta.text()

    if (cadena.find('+')== -1 or cadena.find('-')== -1 or cadena.find('*')== -1 or cadena.find('/')== -1):
        cadena = cadena + "."
        self.etiqueta.setText(cadena)

    elif cadena.find('+') != -1 :
        posicion = cadena.find('+')
        if cadena.find('.', posicion) == -1:
            cadena = cadena + "."
            self.etiqueta.setText(cadena)

    elif cadena.find('-') != -1:
        posicion = cadena.find('-')
        if cadena.find('.', posicion) == -1:
            cadena = cadena + "."

```

Figure 12: Boton punto

SetNumero: Este es solamente para irle metiendo los numeros.

```

def setNumero(self):
    cadena = self.etiqueta.text() + self.sender().text()
    '''flotante = float(cadena)
    if flotante.is_integer():
        entero = int(flottante)
        cadena = str(entero)
    else:
        cadena = str(flottante)'''

    self.etiqueta.setText(cadena)
    print(cadena)

```

Figure 13: Boton de los numeros

SetIgual: Este es para que la calculadora arroje el resultado de la operación. Aquí lo que hice es que primero busque en la cadena de texto el operador que este, si es una suma entonces suma, si es una resta entonces resta y así respectivamente. Ya que encuentra al operador entonces lo que hace es que al string que esta antes del operador los convierte en un numero flotante y al que esta despues del operador los convierte en otro numero flotante para hacer la respectiva operación.


```

def setIgual(self):
    cadena=self.etiqueta.text()

    if cadena.find('+')!=-1 :
        indice_operador = cadena.find('+')
        # Extraer los operandos
        operando1 = float(cadena[:indice_operador])
        operando2 = float(cadena[indice_operador + 1:])
        self.etiqueta.setText(str(operando1+operando2))

    elif cadena.find('-')!=-1 and cadena.find('inv')==-1 and cadena.find('/')==1:
        indice_operador = cadena.find('-')
        # Extraer los operandos
        operando1 = float(cadena[:indice_operador])
        operando2 = float(cadena[indice_operador + 1:])
        self.etiqueta.setText(str(operando1-operando2))

    elif cadena.find('*')!=-1:
        indice_operador = cadena.find('*')
        # Extraer los operandos
        operando1 = float(cadena[:indice_operador])
        operando2 = float(cadena[indice_operador + 1:])
        self.etiqueta.setText(str(operando1*operando2))

```

```

    elif cadena.find('/')!=-1 :
        indice_operador = cadena.find('/')
        # Extraer los operandos
        operando1 = float(cadena[:indice_operador])
        operando2 = float(cadena[indice_operador + 1:])
        if operando2==0:
            self.etiqueta.setText('Syntax Error')
        else:
            self.etiqueta.setText(str(operando1/operando2))

    elif cadena.find('%')!=-1:
        indice_operador = cadena.find('%')
        # Extraer los operandos
        operando1 = float(cadena[:indice_operador])
        operando2 = float(cadena[indice_operador + 1:])
        self.etiqueta.setText(str((operando1*operando2)/100))

    elif cadena.find('inv')!=-1 :
        indice_operador = cadena.find('inv')
        # Extraer los operandos
        operando1 = float(cadena[:indice_operador])

        if operando1>0:
            self.etiqueta.setText(str(operando1*(-1)))
        else:
            self.etiqueta.setText(str(operando1*(-1)))

```

Figure 14: Boton igual

Las siguientes líneas de código son para que se vea la pantalla.

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

Figure 15: Pantalla

3.1 Ejecución

Las siguientes imágenes son algunas operaciones que se realizaron (Multiplicación con puntos decimales, sacarle el negativo a un número y un porcentaje).



Figure 16: Multiplicacion con puntos decimales

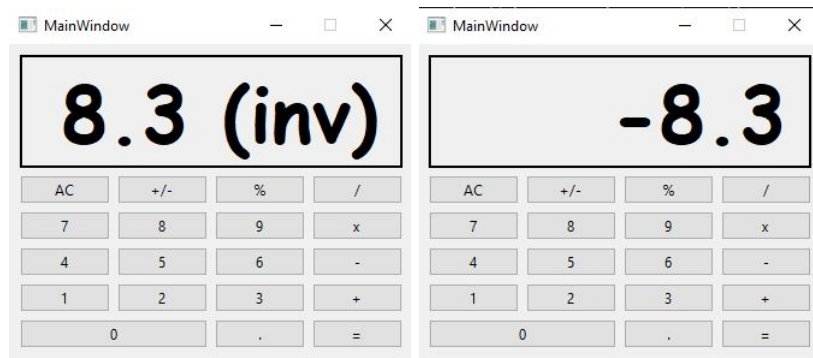


Figure 17: Inversa

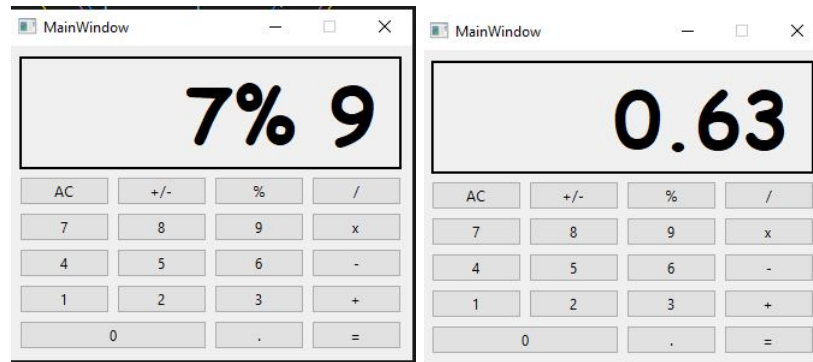


Figure 18: Porcentaje

4 Conclusión

En esta practica creé una interfaz gráfica de usuario (GUI) que fue una calculadora y pude hacer uso de los controles principale como son los botones y las etiquetas para escribir texto.

5 Bibliografia

- Escribir ecuaciones en LaTeX — Manualdelatex.com. (n.d.). <https://manualdelatex.com/tutoriales/ecuaciones>
- Equipo editorial de IONOS. (2021, January 28). ¿Qué es una interfaz gráfica de usuario (GUI)? IONOS Digital Guide. <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/que-es-una-gui/>
- Alonso, D. (2020, March 18). Cómo crear formularios para QGIS con Qt Designer. MappingGIS. <https://mappinggis.com/2018/12/como-crear-formularios-para-qgis-con-qt-designer/#::~:~:text=QT%20Designer%20es%20un%20programa,Mac%20S%20X%20y%20Windows.>