

Actividad 06 - QFileDialog

Roberto Haro González

Seminario de solución de problemas de algoritmia

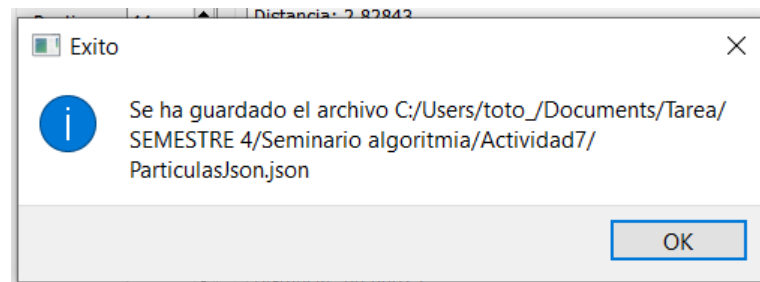
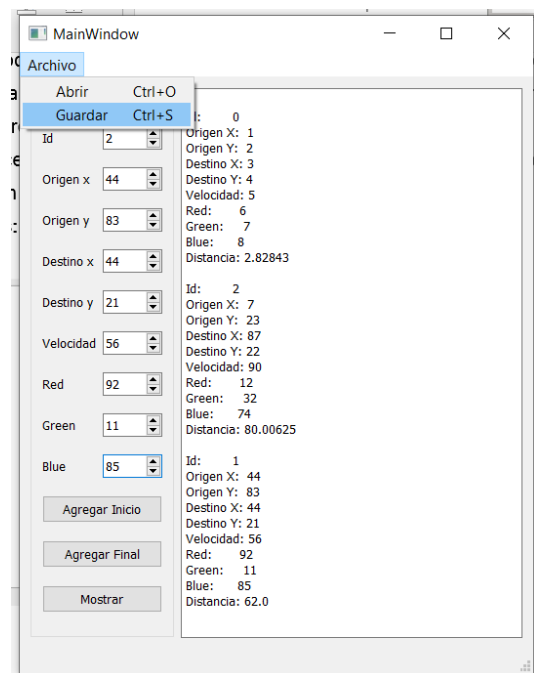
Lineamientos de evaluación

- ☑ El reporte está en formato Google Docs o PDF.
- ☑ El reporte sigue las pautas del Formato de Actividades.
- ☑ Se muestra la captura de pantalla de las partículas con el método `mostrar()` previo a generar el respaldo.
- ☑ Se muestran capturas de pantallas de los pasos que se realizan en la interfaz para generar el respaldo.
- ☑ Se muestra el contenido del archivo `*.json*`.
- ☑ Se muestran capturas de pantallas de los pasos que se realizan en la interfaz para abrir el archivo de respaldo `*.json*`.
- ☑ Se muestra la captura de pantalla de las partículas con el método `mostrar()` después de abrir el respaldo.

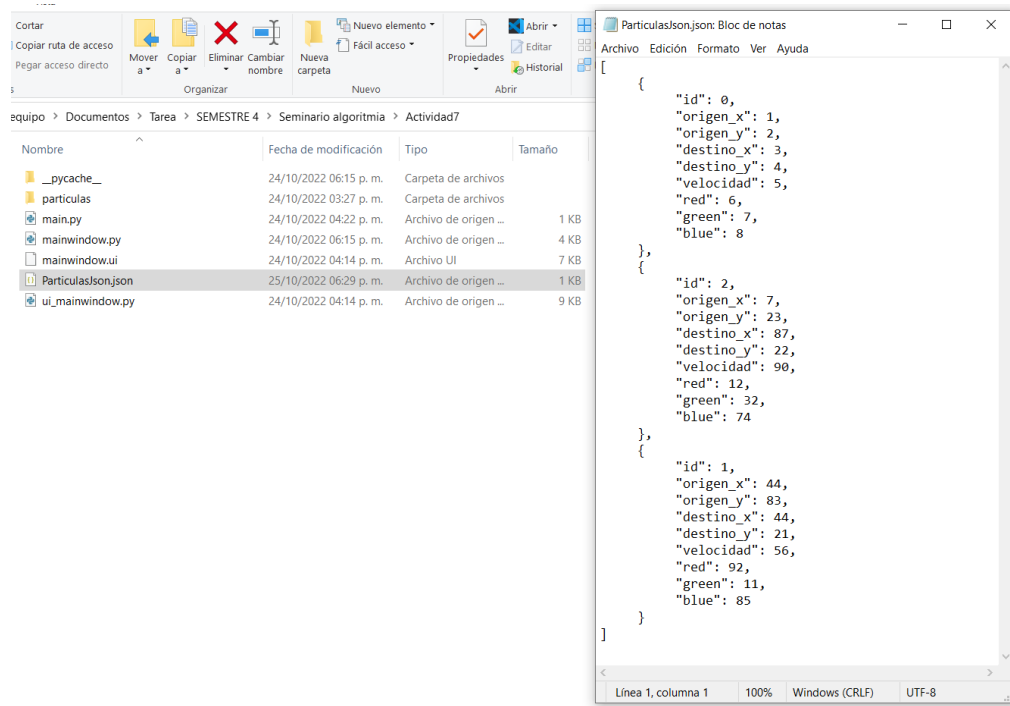
Desarrollo

Se desarrollaron el método de convertir a diccionario en la clase partícula y los métodos de guardado y apertura en la lista y quedó todo conectado en la UI, los cambios principales a la UI fueron el agregar los triggers para almacenar y recuperar los archivos, también utilice la función incorporada round para limitar la cantidad de decimales que aparecen en distancia.

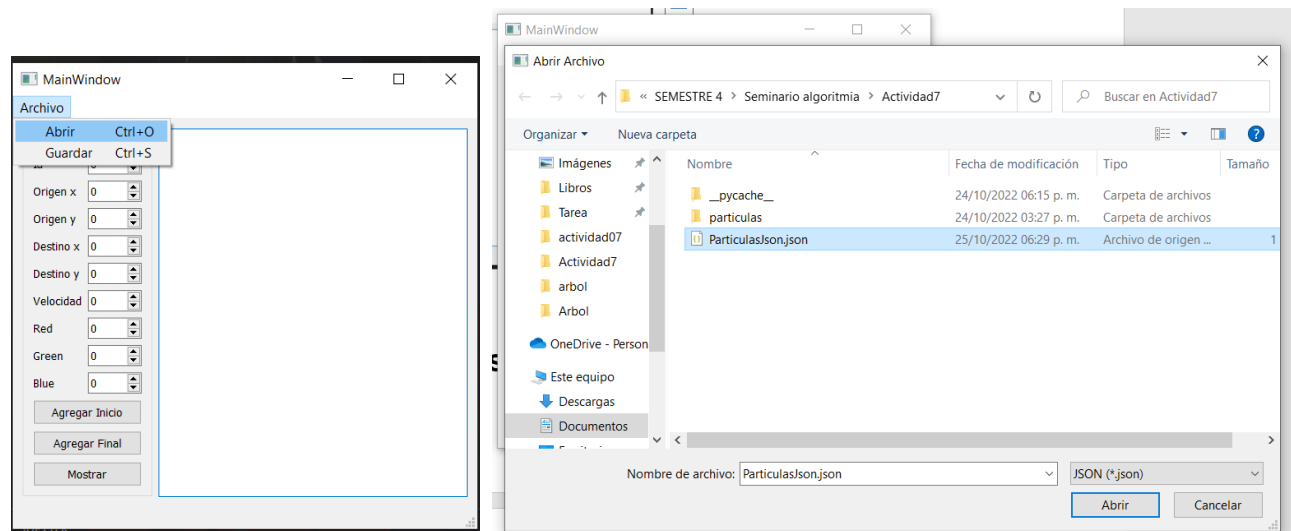
Guardamos 3 partículas:

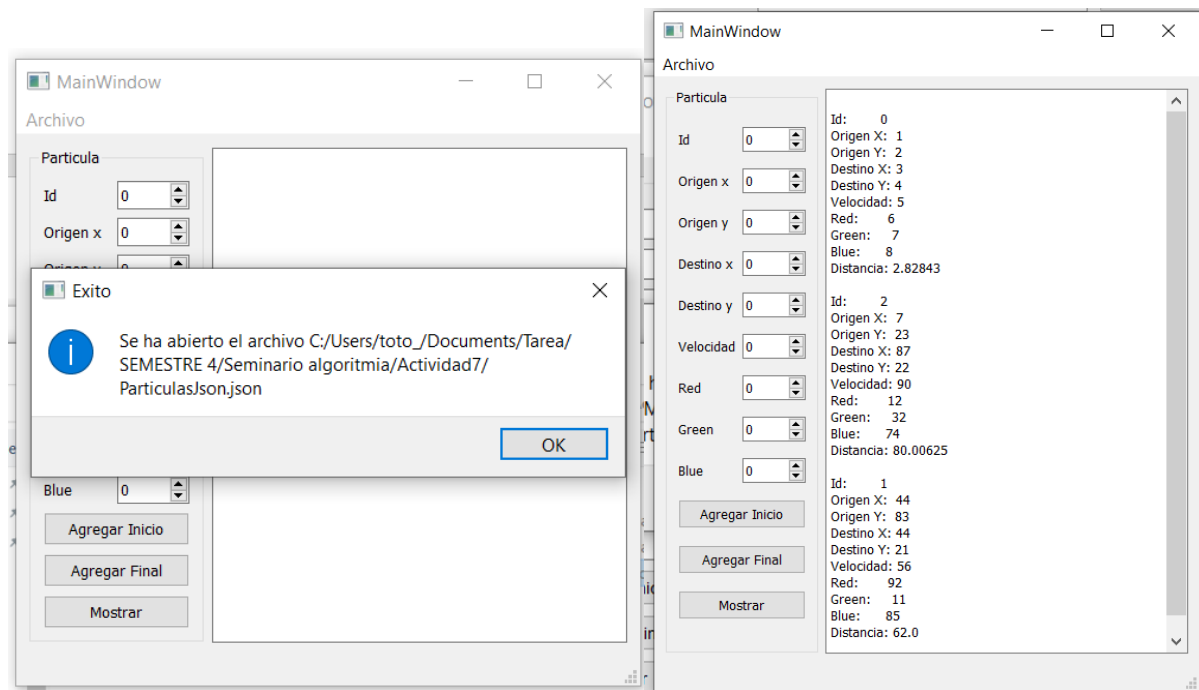


Archivo físico generado



Abrir archivo JSON:





Conclusiones

Ver como toma forma el programa con interfaz grafica no deja de ser impresionante cada vez mas con los nuevos elementos integrados, he de mencionar que durante el desarrollo de esta practica tuve un problema con el programa ya que dentro de mi lista tenia definidos a mis objetos tipo partícula con mayúscula y el programa en un punto dado surgía un error ya que me detectaba el objeto como una clase y la clase como un objeto según el caso y tuve que reestructurar esa parte y ver donde hacia referencia al objeto y donde hacia referencia a la clase.

Referencias

MICHEL DAVALOS BOITES. (2020b, octubre 22). *PySide2 - QFileDialog (Qt for Python)(IV)*.

YouTube. <https://www.youtube.com/watch?v=HRY8QvXmcDM>

Código

Main

```
from PySide2.QtWidgets import QApplication
from mainwindow import MainWindow
import sys

# Aplicación de Qt
app = QApplication()
# Se crea la ventana
window = MainWindow()
# Se hace visible la ventana
window.show()
sys.exit(app.exec_())
```

Algoritmos

```
import math

def distancia_euclidiana(x_1, y_1, x_2, y_2): return math.sqrt(math.pow((x_1-
x_2),2)+math.pow((y_2-y_1),2))
```

Partícula

```
from .algoritmos import distancia_euclidiana

class Particula:

    def __init__(self, id=0, origen_x=0, origen_y=0, destino_x=0, destino_y=0,
velocidad=0, red=0, green=0, blue=0):

        self.__id = id
        self.__origen_x = origen_x
        self.__origen_y = origen_y
        self.__destino_x = destino_x
        self.__destino_y = destino_y
        self.__velocidad = velocidad
        self.__red = red
        self.__green = green
        self.__blue = blue
        self.__distancia =
round(distancia_euclidiana(origen_x,origen_y,destino_x,destino_y),5)

    def __str__(self):
        return(
            '\nId:      ' + str(self.__id) + '\n' +
            'Origen X:  ' + str(self.__origen_x) + '\n' +
            'Origen Y:  ' + str(self.__origen_y) + '\n' +
            'Destino X: ' + str(self.__destino_x) + '\n' +
            'Destino Y: ' + str(self.__destino_y) + '\n' +
            'Velocidad: ' + str(self.__velocidad) + '\n' +
            'Red:       ' + str(self.__red) + '\n' +
            'Green:      ' + str(self.__green) + '\n' +
            'Blue:       ' + str(self.__blue) + '\n' +
            'Distancia: ' + str(self.__distancia) + '\n'
        )

    def to_dict(self):
        return {
            "id": self.__id,
            "origen_x": self.__origen_x,
            "origen_y": self.__origen_y,
            "destino_x": self.__destino_x,
            "destino_y": self.__destino_y,
            "velocidad": self.__velocidad,
            "red": self.__red,
            "green": self.__green,
            "blue": self.__blue
        }
```

```
}
```

Lista partícula

```
from .particula import Particula
import json

class ListaParticulas:
    def __init__(self):
        self.__particulas = []

    def agregar_final(self, particula:Particula):
        self.__particulas.append(particula)

    def agregar_inicio(self, particula:Particula):
        self.__particulas.insert(0,particula)

    def mostrar(self):
        for particula in self.__particulas:
            print(particula)

    def __str__(self):
        return "".join(
            str(particula) for particula in self.__particulas
        )

    def guardar(self, ubicacion):
        try:
            with open(ubicacion, 'w') as archivo:
                lista = [particula.to_dict() for particula in self.__particulas]
                json.dump(lista, archivo, indent=5)
            return 1
        except:
            return 0

    def abrir(self, ubicacion):
        try:
            with open(ubicacion, 'r') as archivo:
                lista = json.load(archivo)
                self.__particulas = [Particula(**particula) for particula in lista]
            return 1
        except:
            return 0
```


Main window

```
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from particulas.listaparticula import ListaParticulas
from particulas.particula import Particula

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()

        self.myListaParticulas = ListaParticulas()

        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.agregar_inicio_pushButton.clicked.connect(self.click_agregar_inicio)

        self.ui.agregar_final_pushButton.clicked.connect(self.click_agregar_final)
        self.ui.mostrar_pushButton.clicked.connect(self.click_mostrar)

        self.ui.actionAbrir.triggered.connect(self.action_abrir_archivo)
        self.ui.actionGuardar.triggered.connect(self.action_guardar_archivo)

    @Slot()
    def action_abrir_archivo(self):
        ubicacion = QFileDialog.getOpenFileName(
            self,
            'Abrir Archivo',
            '.',
            'JSON (*.json)'
        )[0]
        if self.myListaParticulas.abrir(ubicacion):
            QMessageBox.information(
                self,
                "Exito",
                "Se ha abierto el archivo " + ubicacion
            )
        else:
            QMessageBox.critical(
                self,
                "Error",
                "No se ha abierto el archivo " + ubicacion
            )
```

```

@Slot()
def action_guardar_archivo(self):
    #print('Guardar archivo')
    ubicacion = QFileDialog.getSaveFileName(
        self,
        'Guardar Archivo',
        '.',
        'JSON (*.json)'
    )[0]
    print(ubicacion)
    if self.myListaParticulas.guardar(ubicacion):
        QMessageBox.information(
            self,
            "Exito",
            "Se ha guardado el archivo " + ubicacion
        )
    else:
        QMessageBox.critical(
            self,
            "Error",
            "No se ha creado el archivo " + ubicacion
        )

@Slot()
def click_mostrar(self):
    self.ui.out_plainTextEdit.clear()
    self.ui.out_plainTextEdit.insertPlainText(str(self.myListaParticulas))

@Slot()
def click_agregar_inicio(self):
    myId = self.ui.id_spinBox.value()
    myOrigenX = self.ui.origenX_spinBox.value()
    myOrigenY = self.ui.origenY_spinBox.value()
    myDestinoX = self.ui.destinoX_spinBox.value()
    myDestinoY = self.ui.destinoY_spinBox.value()
    myVelocidad = self.ui.velocidad_spinBox.value()
    myRed = self.ui.red_spinBox.value()
    myGreen = self.ui.green_spinBox.value()
    myBlue = self.ui.blue__spinBox.value()

    myParticula = Particula(myId, myOrigenX, myOrigenY,
myDestinoX, myDestinoY, myVelocidad, myRed, myGreen, myBlue)
    self.myListaParticulas.agregar_inicio(myParticula)

@Slot()

```

```

def click_agregar_final(self):
    myId = self.ui.id_spinBox.value()
    myOrigenX = self.ui.origenX_spinBox.value()
    myOrigenY = self.ui.origenY_spinBox.value()
    myDestinoX = self.ui.destinoX_spinBox.value()
    myDestinoY = self.ui.destinoY_spinBox.value()
    myVelocidad = self.ui.velocidad_spinBox.value()
    myRed = self.ui.red_spinBox.value()
    myGreen = self.ui.green_spinBox.value()
    myBlue = self.ui.blue__spinBox.value()

    myParticula = Particula(myId, myOrigenX, myOrigenY,
myDestinoX,myDestinoY,myVelocidad,myRed,myGreen,myBlue)
    self.myListaParticulas.agregar_final(myParticula)

```

ui mainwindow

```

# -*- coding: utf-8 -*-

#####
## Form generated from reading UI file 'mainwindow.ui'
##
## Created by: Qt User Interface Compiler version 5.15.2
##
## WARNING! All changes made in this file will be lost when recompiling UI file!
#####

from PySide2.QtCore import *
from PySide2.QtGui import *
from PySide2.QtWidgets import *

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        if not MainWindow.setObjectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.resize(526, 486)
        self.actionAbrir = QAction(MainWindow)
        self.actionAbrir.setObjectName(u"actionAbrir")
        self.actionGuardar = QAction(MainWindow)
        self.actionGuardar.setObjectName(u"actionGuardar")
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.gridLayout_2 = QGridLayout(self.centralwidget)
        self.gridLayout_2.setObjectName(u"gridLayout_2")
        self.out_plainTextEdit = QPlainTextEdit(self.centralwidget)

```

```

self.out_plainTextEdit.setObjectName(u"out_plainTextEdit")
self.out_plainTextEdit.setMinimumSize(QSize(350, 350))

self.gridLayout_2.addWidget(self.out_plainTextEdit, 0, 1, 1, 1)

self.groupBox = QGroupBox(self.centralwidget)
self.groupBox.setObjectName(u"groupBox")
self.gridLayout = QGridLayout(self.groupBox)
self.gridLayout.setObjectName(u"gridLayout")
self.label_7 = QLabel(self.groupBox)
self.label_7.setObjectName(u"label_7")

self.gridLayout.addWidget(self.label_7, 0, 0, 1, 1)

self.label_8 = QLabel(self.groupBox)
self.label_8.setObjectName(u"label_8")

self.gridLayout.addWidget(self.label_8, 1, 0, 1, 1)

self.red_spinBox = QSpinBox(self.groupBox)
self.red_spinBox.setObjectName(u"red_spinBox")
self.red_spinBox.setMaximum(255)

self.gridLayout.addWidget(self.red_spinBox, 6, 1, 1, 1)

self.label_4 = QLabel(self.groupBox)
self.label_4.setObjectName(u"label_4")

self.gridLayout.addWidget(self.label_4, 6, 0, 1, 1)

self.label_2 = QLabel(self.groupBox)
self.label_2.setObjectName(u"label_2")

self.gridLayout.addWidget(self.label_2, 4, 0, 1, 1)

self.velocidad_spinBox = QSpinBox(self.groupBox)
self.velocidad_spinBox.setObjectName(u"velocidad_spinBox")
self.velocidad_spinBox.setMaximum(10000)
self.velocidad_spinBox.setDisplayIntegerBase(10)

self.gridLayout.addWidget(self.velocidad_spinBox, 5, 1, 1, 1)

self.mostrar_pushButton = QPushButton(self.groupBox)
self.mostrar_pushButton.setObjectName(u"mostrar_pushButton")

```

```

self.gridLayout.addWidget(self.mostrar_pushButton, 12, 0, 1, 2)

self.destinoX_spinBox = QSpinBox(self.groupBox)
self.destinoX_spinBox.setObjectName(u"destinoX_spinBox")
self.destinoX_spinBox.setMaximum(500)

self.gridLayout.addWidget(self.destinoX_spinBox, 3, 1, 1, 1)

self.agregar_final_pushButton = QPushButton(self.groupBox)
self.agregar_final_pushButton.setObjectName(u"agregar_final_pushButton")

self.gridLayout.addWidget(self.agregar_final_pushButton, 11, 0, 1, 2)

self.id_spinBox = QSpinBox(self.groupBox)
self.id_spinBox.setObjectName(u"id_spinBox")
self.id_spinBox.setMaximum(500)

self.gridLayout.addWidget(self.id_spinBox, 0, 1, 1, 1)

self.destinoY_spinBox = QSpinBox(self.groupBox)
self.destinoY_spinBox.setObjectName(u"destinoY_spinBox")
self.destinoY_spinBox.setMaximum(500)

self.gridLayout.addWidget(self.destinoY_spinBox, 4, 1, 1, 1)

self.label_6 = QLabel(self.groupBox)
self.label_6.setObjectName(u"label_6")

self.gridLayout.addWidget(self.label_6, 8, 0, 1, 1)

self.green_spinBox = QSpinBox(self.groupBox)
self.green_spinBox.setObjectName(u"green_spinBox")
self.green_spinBox.setMaximum(255)

self.gridLayout.addWidget(self.green_spinBox, 7, 1, 1, 1)

self.label_3 = QLabel(self.groupBox)
self.label_3.setObjectName(u"label_3")

self.gridLayout.addWidget(self.label_3, 5, 0, 1, 1)

self.agregar_inicio_pushButton = QPushButton(self.groupBox)
self.agregar_inicio_pushButton.setObjectName(u"agregar_inicio_pushButton")

self.gridLayout.addWidget(self.agregar_inicio_pushButton, 10, 0, 1, 2)

```

```

self.blue__spinBox = QSpinBox(self.groupBox)
self.blue__spinBox.setObjectName(u"blue__spinBox")
self.blue__spinBox.setMaximum(255)

self.gridLayout.addWidget(self.blue__spinBox, 8, 1, 1, 1)

self.label = QLabel(self.groupBox)
self.label.setObjectName(u"label")

self.gridLayout.addWidget(self.label, 3, 0, 1, 1)

self.label_5 = QLabel(self.groupBox)
self.label_5.setObjectName(u"label_5")

self.gridLayout.addWidget(self.label_5, 7, 0, 1, 1)

self.label_9 = QLabel(self.groupBox)
self.label_9.setObjectName(u"label_9")

self.gridLayout.addWidget(self.label_9, 2, 0, 1, 1)

self.origenX_spinBox = QSpinBox(self.groupBox)
self.origenX_spinBox.setObjectName(u"origenX_spinBox")
self.origenX_spinBox.setMaximum(500)

self.gridLayout.addWidget(self.origenX_spinBox, 1, 1, 1, 1)

self.origenY_spinBox = QSpinBox(self.groupBox)
self.origenY_spinBox.setObjectName(u"origenY_spinBox")
self.origenY_spinBox.setMaximum(500)

self.gridLayout.addWidget(self.origenY_spinBox, 2, 1, 1, 1)

self.gridLayout_2.addWidget(self.groupBox, 0, 0, 1, 1)

MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QMenuBar(MainWindow)
self.menubar.setObjectName(u"menubar")
self.menubar.setGeometry(QRect(0, 0, 526, 26))
self.menuArchivo = QMenu(self.menubar)
self.menuArchivo.setObjectName(u"menuArchivo")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QStatusBar(MainWindow)
self.statusbar.setObjectName(u"statusbar")

```

```

MainWindow.setStatusBar(self.statusbar)

self.menubar.addAction(self.menuArchivo.menuAction())
self.menuArchivo.addAction(self.actionAbrir)
self.menuArchivo.addAction(self.actionGuardar)

self.retranslateUi(MainWindow)

QMetaObject.connectSlotsByName(MainWindow)
# setupUi

def retranslateUi(self, MainWindow):
    MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow",
u"MainWindow", None))
    self.actionAbrir.setText(QCoreApplication.translate("MainWindow", u"Abrir",
None))
    #if QT_CONFIG(shortcut)
        self.actionAbrir.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+O", None))
    #endif // QT_CONFIG(shortcut)
    self.actionGuardar.setText(QCoreApplication.translate("MainWindow",
u"Guardar", None))
    #if QT_CONFIG(shortcut)
        self.actionGuardar.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+S", None))
    #endif // QT_CONFIG(shortcut)
    self.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Particula", None))
    self.label_7.setText(QCoreApplication.translate("MainWindow", u"Id", None))
    self.label_8.setText(QCoreApplication.translate("MainWindow", u"Origen x",
None))
    self.label_4.setText(QCoreApplication.translate("MainWindow", u"Red",
None))
    self.label_2.setText(QCoreApplication.translate("MainWindow", u"Destino y",
None))
    self.mostrar_pushButton.setText(QCoreApplication.translate("MainWindow",
u"Mostrar", None))
    self.agregar_final_pushButton.setText(QCoreApplication.translate("MainWindo
w", u"Agregar Final", None))
    self.label_6.setText(QCoreApplication.translate("MainWindow", u"Blue",
None))
    self.label_3.setText(QCoreApplication.translate("MainWindow", u"Velocidad",
None))
    self.agregar_inicio_pushButton.setText(QCoreApplication.translate("MainWind
ow", u"Agregar Inicio", None))

```

```
        self.label.setText(QCoreApplication.translate("MainWindow", u"Destino x",
None))
        self.label_5.setText(QCoreApplication.translate("MainWindow", u"Green",
None))
        self.label_9.setText(QCoreApplication.translate("MainWindow", u"Origen y",
None))
        self.menuArchivo.setTitle(QCoreApplication.translate("MainWindow",
u"Archivo", None))
        # retranslateUi
```