Actividad 08 (QTableWidget)

RAFAEL ARTURO GUTIERREZ CRUZ

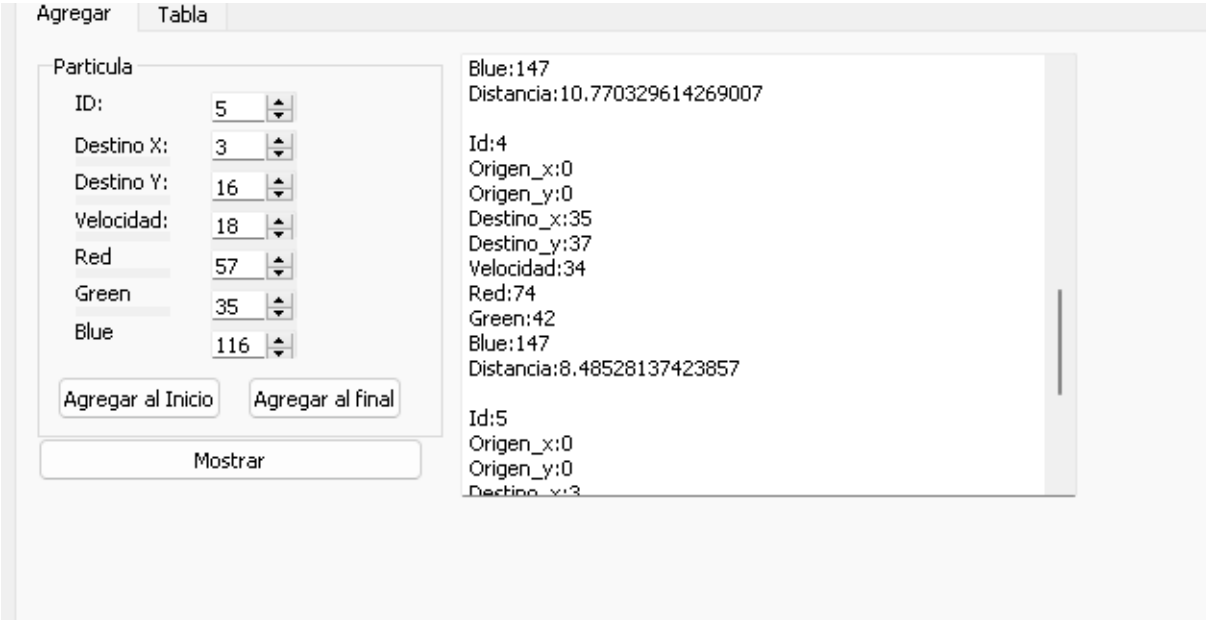
Seminario de Solucion de Problemas de Algoritmia

Lineamientos de evaluación

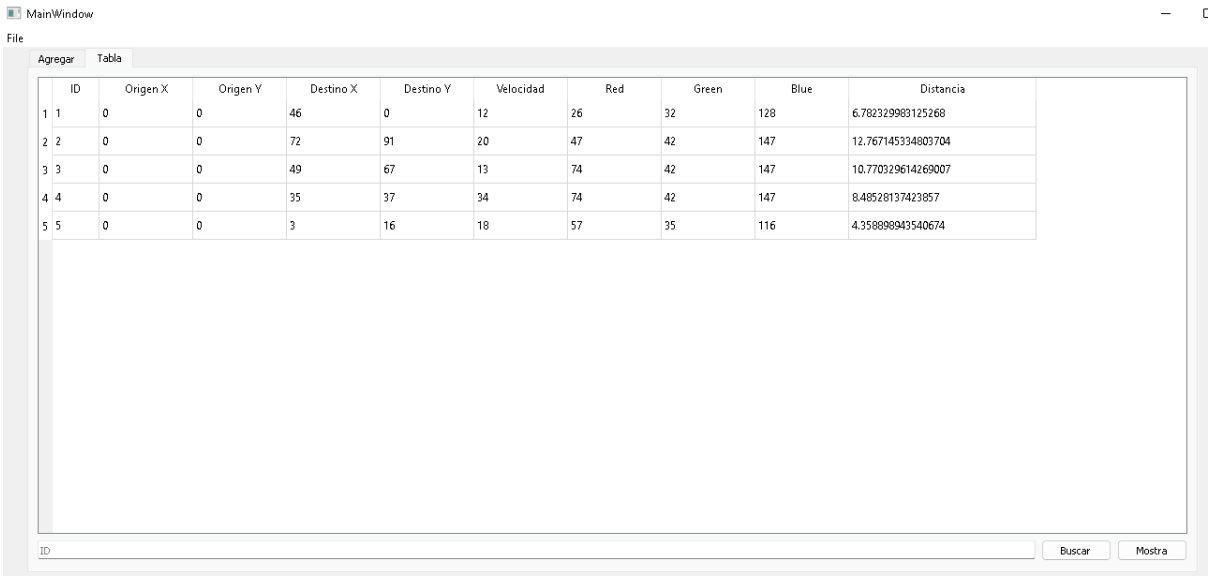
- [ ] El reporte está en formato Google Docs o PDF.
- [ ] El reporte sigue las pautas del Formato de Actividades .
- [ ] El reporte tiene desarrollada todas las pautas del Formato de Actividades.
- [ ] Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto a.
- [ ] Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto b.
- [ ] Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto c.
- [ ] Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto d.

Desarrollo

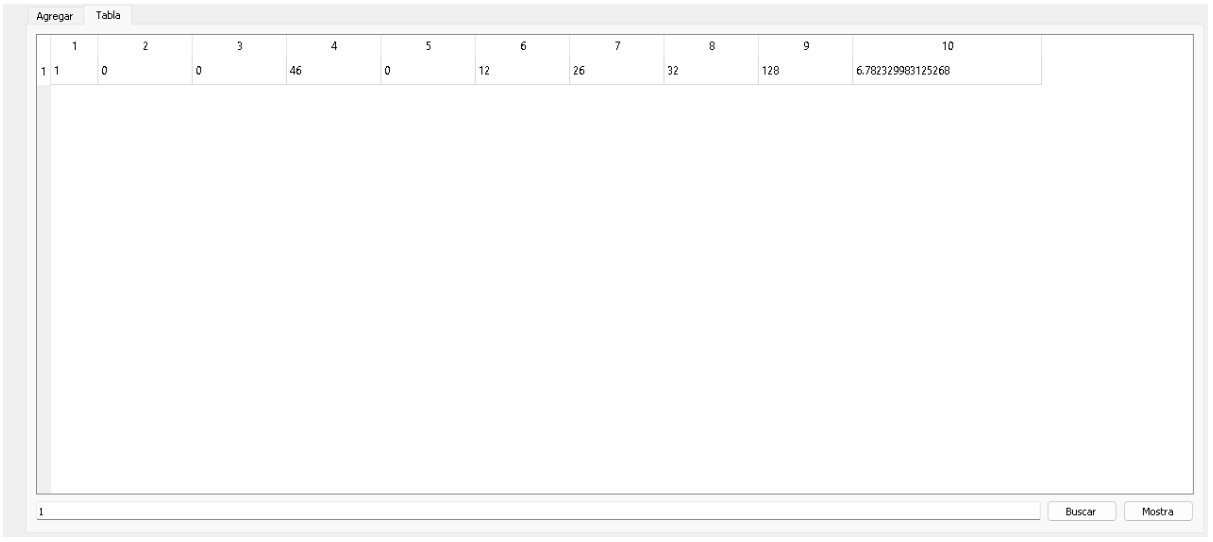
Introducción de las 5 partículas en el programa.



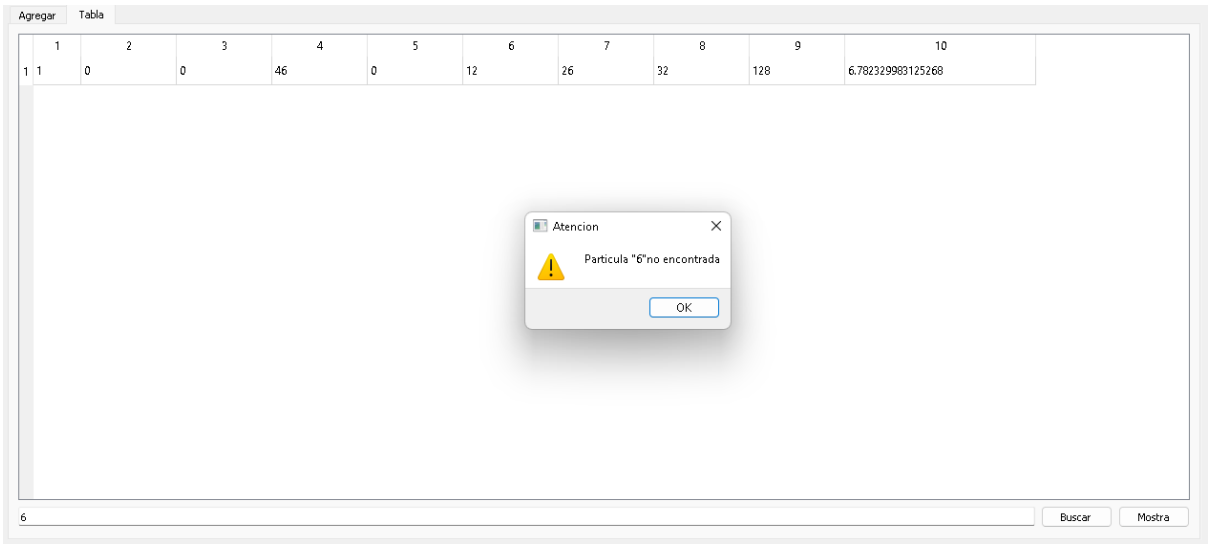
Partículas en el widget ”QTableWidget”



Búsqueda de partícula que esté en la tabla



Búsqueda de partícula que no esté en la tabla



Conclusiones

Tuve un problema de versiones de python con esta práctica, ya que la manera en la que se sobrecargaba del operador “__iter__” no me funcionaba como al video de referencia, tube que volver a hacer la instalación de python pero con una version mas antigua, mas precisamente la versión que utiliza el profesor en el video

Referencias

MICHEL DAVALOS BOITES. (2020b, octubre 29). *PySide2 - QTableWidget (Qt for Python)(V)* [Vídeo]. YouTube. Recuperado 23 de octubre de 2022, de https://www.youtube.com/watch?v=1yEpAHaiMxs

Código

algoritmo.py

```python
import math

def distancia_euclidiana(x_1, y_1, x_2, y_2):

    valor1 = x_1 - y_1
    valor1**2

    valor2 = x_2 - y_2
    valor2**2

    return math.sqrt(valor1+valor2)
```

mian.py

```python
from PySide2.QtWidgets import QApplication
from mainwindow import MainWindow
from scipy.optimize import linprog
import sys
```

```
app =QApplication()

window = MainWindow()

window.show()

sys.exit(app.exec_())
```

mainwindow.py

```
from base64 import decodebytes
from contextlib import redirect_stderr
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox,
QTableWidgetItem
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from particle_adminstrator import administrador
from particulas import Particula

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

        self.administrador = administrador()

        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

self.ui.agregarFinal_pushButton.clicked.connect(self.click_agregar)

self.ui.AgragrInicio_pushButton.clicked.connect(self.click_agregar_inic
io)
        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)


self.ui.Mostrar_Tabla_pushButton_2.clicked.connect(self.mostrar_tabla)
        self.ui.buscar_pushButton.clicked.connect(self.Buscar)
```

```
    @Slot()
    def Buscar(self):
        Codigo_Buscado = self.ui.Buscar_lineEdit.text()


        encontrado = False
        for particle in self.administrador:
            if Codigo_Buscado == str(particle.Codigo):
                self.ui.Tabla.clear()
                self.ui.Tabla.setRowCount(1)

                Codigo_Widget = QTableWidgetItem(str(particle.Codigo))
                OrigenesX_Widget =
QTableWidgetItem(str(particle.OrigenX))
                OrigenesY_Widget =
QTableWidgetItem(str(particle.OrigenY))
                DestinoX_Widget =
QTableWidgetItem(str(particle.DestinoX))
                DestinoY_Widget =
QTableWidgetItem(str(particle.DestinoY))
                Velocidad_Widget =
QTableWidgetItem(str(particle.Velocidad))
                Red_Widget = QTableWidgetItem(str(particle.Red))
                Green_Widget = QTableWidgetItem(str(particle.Green))
                Blue_Widget = QTableWidgetItem(str(particle.Blue))
                Distancia_Widget =
QTableWidgetItem(str(particle.Distancia))

                self.ui.Tabla.setItem(0, 0, Codigo_Widget)
                self.ui.Tabla.setItem(0, 1, OrigenesX_Widget)
                self.ui.Tabla.setItem(0, 2, OrigenesY_Widget)
                self.ui.Tabla.setItem(0, 3, DestinoX_Widget)
                self.ui.Tabla.setItem(0, 4, DestinoY_Widget)
                self.ui.Tabla.setItem(0, 5, Velocidad_Widget)
                self.ui.Tabla.setItem(0, 6, Red_Widget)
                self.ui.Tabla.setItem(0, 7, Green_Widget)
                self.ui.Tabla.setItem(0, 8, Blue_Widget)
                self.ui.Tabla.setItem(0, 9, Distancia_Widget)

                encontrado = True
                return
        print(Codigo_Buscado)
        if not encontrado:
```

```
            QMessageBox.warning(self,"Atencion", f'Particula
"{Codigo_Buscado}"no encontrada')



    @Slot()
    def mostrar_tabla(self):
        self.ui.Tabla.setColumnCount(10)
        headers = ["ID" ,"Origen X" ,"Origen Y" ,"Destino X" ,"Destino
Y" ,"Velocidad" ,"Red" ,"Green" ,"Blue" ,"Distancia"]
        self.ui.Tabla.setHorizontalHeaderLabels(headers)

        self.ui.Tabla.setRowCount(len(self.administrador))
        self.ui.Tabla.setColumnWidth(0,50)
        self.ui.Tabla.setColumnWidth(9,200)


        row = 0
        for particle in self.administrador:
            Codigo_Widget = QTableWidgetItem(str(particle.Codigo))
            OrigenesX_Widget = QTableWidgetItem(str(particle.OrigenX))
            OrigenesY_Widget = QTableWidgetItem(str(particle.OrigenY))
            DestinoX_Widget = QTableWidgetItem(str(particle.DestinoX))
            DestinoY_Widget = QTableWidgetItem(str(particle.DestinoY))
            Velocidad_Widget =
QTableWidgetItem(str(particle.Velocidad))
            Red_Widget = QTableWidgetItem(str(particle.Red))
            Green_Widget = QTableWidgetItem(str(particle.Green))
            Blue_Widget = QTableWidgetItem(str(particle.Blue))
            Distancia_Widget =
QTableWidgetItem(str(particle.Distancia))

            self.ui.Tabla.setItem(row, 0, Codigo_Widget)
            self.ui.Tabla.setItem(row, 1, OrigenesX_Widget)
            self.ui.Tabla.setItem(row, 2, OrigenesY_Widget)
            self.ui.Tabla.setItem(row, 3, DestinoX_Widget)
            self.ui.Tabla.setItem(row, 4, DestinoY_Widget)
            self.ui.Tabla.setItem(row, 5, Velocidad_Widget)
            self.ui.Tabla.setItem(row, 6, Red_Widget)
            self.ui.Tabla.setItem(row, 7, Green_Widget)
            self.ui.Tabla.setItem(row, 8, Blue_Widget)
            self.ui.Tabla.setItem(row, 9, Distancia_Widget)
```

```
            row += 1




    @Slot()
    def action_abrir_archivo(self):
        # print('Abriendo')
        ubicacion = QFileDialog.getOpenFileName(self, 'Abrir', '.',
'JSON (*.json)')[0]

        if self.administrador.abrir(ubicacion):
            QMessageBox.information(self, "Exito","Archivo Cargado de:
" + ubicacion)
        else:
            QMessageBox.critical(self, "Error", "No se pudo cargar el
archivo")

    @Slot()
    def action_guardar_archivo(self):
        # print('Guardando')
        ubicacion = QFileDialog.getSaveFileName(self, 'Guardar', '.',
'JSON (*.json)')[0]
        print(ubicacion)
        if self.administrador.guardar(ubicacion):
            QMessageBox.information(self, "Exito","Archivo Guardado en:
" + ubicacion)
        else:
            QMessageBox.critical(self, "Error","No se pudo guardar el
archivo")

    @Slot()
    def click_mostrar(self):
        # self.administrador.mostrar()
        self.ui.salida.insertPlainText(str(self.administrador))


    @Slot()
    def click_agregar_inicio(self):
        codigo = self.ui.ID_pinBox.value()
        desX = self.ui.DesX_pinBox.value()
        desY = self.ui.DesY_spinBox_2.value()
```

```
        velocidad = self.ui.Velocidad_spinBox_3.value()
        red = self.ui.Red_spinBox_4.value()
        green = self.ui.Green_spinBox_5.value()
        blue = self.ui.Blue_spinBox_6.value()

        Particle = Particula(id=codigo, destino_x=desX, destino_y=desY,
velocidad=velocidad, red=red, green=green, blue=blue)
        self.administrador.agregar_incio(Particle)

    @Slot()
    def click_agregar(self):
        codigo = self.ui.ID_pinBox.value()
        desX = self.ui.DesX_pinBox.value()
        desY = self.ui.DesY_spinBox_2.value()
        velocidad = self.ui.Velocidad_spinBox_3.value()
        red = self.ui.Red_spinBox_4.value()
        green = self.ui.Green_spinBox_5.value()
        blue = self.ui.Blue_spinBox_6.value()

        Particle = Particula(id=codigo, destino_x=desX, destino_y=desY,
velocidad=velocidad, red=red, green=green, blue=blue)
        self.administrador.agregar_final(Particle)
```

particle_administrator.py

```
from particulas import Particula
import json

class administrador:
    def __init__(self):
        self.__particles = []

    def agregar_final(self, particle:Particula):
        self.__particles.append(particle)

    def agregar_incio(self, particle:Particula):
        self.__particles.insert(0,particle)

    def mostrar(self):
        for particle in self.__particles:
            print(particle)

    def __str__(self):
```

```
        return "".join(
            str(particle) + '\n' for particle in self.__particles
        )
    def guardar(self, ubicacion):
        try:

            with open(ubicacion, 'w') as file:
                lista = [particle.to_dict() for particle in
self.__particles]
                print(lista)
                json.dump(lista, file, indent=5)
            return 1
        except:
            return 0

    def __len__(self):
        return len(self.__particles)

    def __iter__(self):
        self.cont = 0

        return self

    def __next__(self):
        if self.cont < len(self.__particles):
            particle = self.__particles[self.cont]
            self.cont += 1
            return particle
        else:
            raise StopIteration

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

particulas.py

```
import json
```

```python
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 = distancia_euclidiana(destino_x, origen_x,
destino_y, origen_y)

    def __str__(self):
        return(
            'Id:' + 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'
        )

    @property
    def Codigo(self):
        return self.__id

    @property
    def OrigenX(self):
        return self.__origen_x

    @property
    def OrigenY(self):
        return self.__origen_y
```

```
    @property
    def DestinoX(self):
        return self.__destino_x

    @property
    def DestinoY(self):
        return self.__destino_y

    @property
    def Velocidad(self):
        return self.__velocidad

    @property
    def Red(self):
        return self.__red

    @property
    def Green(self):
        return self.__green

    @property
    def Blue(self):
        return self.__blue

    @property
    def Distancia(self):
        return self.__distancia

    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
        }
```

ui_mainwindow.py

```python
# -*- 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.objectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.resize(1359, 630)
        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.tabWidget = QTabWidget(self.centralwidget)
        self.tabWidget.setObjectName(u"tabWidget")
        self.tabWidget.setGeometry(QRect(30, 0, 1251, 561))
        self.tab = QWidget()
        self.tab.setObjectName(u"tab")
        self.salida = QPlainTextEdit(self.tab)
        self.salida.setObjectName(u"salida")
        self.salida.setGeometry(QRect(230, 10, 321, 231))
        self.Mostrar_pushButton = QPushButton(self.tab)
        self.Mostrar_pushButton.setObjectName(u"Mostrar_pushButton")
        self.Mostrar_pushButton.setGeometry(QRect(10, 210, 201, 23))
        self.groupBox = QGroupBox(self.tab)
        self.groupBox.setObjectName(u"groupBox")
        self.groupBox.setGeometry(QRect(10, 10, 211, 201))
```

```
        self.splitter_2 = QSplitter(self.groupBox)
        self.splitter_2.setObjectName(u"splitter_2")
        self.splitter_2.setGeometry(QRect(20, 40, 49, 111))
        self.splitter_2.setOrientation(Qt.Vertical)
        self.label = QLabel(self.splitter_2)
        self.label.setObjectName(u"label")
        self.splitter_2.addWidget(self.label)
        self.label_2 = QLabel(self.splitter_2)
        self.label_2.setObjectName(u"label_2")
        self.splitter_2.addWidget(self.label_2)
        self.label_3 = QLabel(self.splitter_2)
        self.label_3.setObjectName(u"label_3")
        self.splitter_2.addWidget(self.label_3)
        self.label_4 = QLabel(self.splitter_2)
        self.label_4.setObjectName(u"label_4")
        self.splitter_2.addWidget(self.label_4)
        self.label_5 = QLabel(self.splitter_2)
        self.label_5.setObjectName(u"label_5")
        self.splitter_2.addWidget(self.label_5)
        self.label_6 = QLabel(self.splitter_2)
        self.label_6.setObjectName(u"label_6")
        self.splitter_2.addWidget(self.label_6)
        self.AgragrInicio_pushButton = QPushButton(self.groupBox)

self.AgragrInicio_pushButton.setObjectName(u"AgragrInicio_pushButton")
        self.AgragrInicio_pushButton.setGeometry(QRect(10, 168, 86,
23))
        self.splitter = QSplitter(self.groupBox)
        self.splitter.setObjectName(u"splitter")
        self.splitter.setGeometry(QRect(90, 40, 45, 120))
        self.splitter.setOrientation(Qt.Vertical)
        self.splitter.setOpaqueResize(False)
        self.splitter.setChildrenCollapsible(True)
        self.DesX_pinBox = QSpinBox(self.splitter)
        self.DesX_pinBox.setObjectName(u"DesX_pinBox")
        self.DesX_pinBox.setMaximum(500)
        self.splitter.addWidget(self.DesX_pinBox)
        self.DesY_spinBox_2 = QSpinBox(self.splitter)
        self.DesY_spinBox_2.setObjectName(u"DesY_spinBox_2")
        self.DesY_spinBox_2.setMaximum(500)
        self.splitter.addWidget(self.DesY_spinBox_2)
        self.Velocidad_spinBox_3 = QSpinBox(self.splitter)
        self.Velocidad_spinBox_3.setObjectName(u"Velocidad_spinBox_3")
```

```
        self.Velocidad_spinBox_3.setMaximum(1000)
        self.splitter.addWidget(self.Velocidad_spinBox_3)
        self.Red_spinBox_4 = QSpinBox(self.splitter)
        self.Red_spinBox_4.setObjectName(u"Red_spinBox_4")
        self.Red_spinBox_4.setMaximum(255)
        self.splitter.addWidget(self.Red_spinBox_4)
        self.Green_spinBox_5 = QSpinBox(self.splitter)
        self.Green_spinBox_5.setObjectName(u"Green_spinBox_5")
        self.Green_spinBox_5.setMaximum(255)
        self.splitter.addWidget(self.Green_spinBox_5)
        self.Blue_spinBox_6 = QSpinBox(self.splitter)
        self.Blue_spinBox_6.setObjectName(u"Blue_spinBox_6")
        self.Blue_spinBox_6.setMaximum(255)
        self.splitter.addWidget(self.Blue_spinBox_6)
        self.agregarFinal_pushButton = QPushButton(self.groupBox)

self.agregarFinal_pushButton.setObjectName(u"agregarFinal_pushButton")
        self.agregarFinal_pushButton.setGeometry(QRect(109, 168, 81,
23))
        self.label_7 = QLabel(self.groupBox)
        self.label_7.setObjectName(u"label_7")
        self.label_7.setGeometry(QRect(20, 20, 49, 11))
        self.ID_pinBox = QSpinBox(self.groupBox)
        self.ID_pinBox.setObjectName(u"ID_pinBox")
        self.ID_pinBox.setGeometry(QRect(90, 20, 45, 16))
        self.ID_pinBox.setMaximum(500000)
        self.tabWidget.addTab(self.tab, "")
        self.tab_2 = QWidget()
        self.tab_2.setObjectName(u"tab_2")
        self.gridLayout = QGridLayout(self.tab_2)
        self.gridLayout.setObjectName(u"gridLayout")
        self.Tabla = QTableWidget(self.tab_2)
        self.Tabla.setObjectName(u"Tabla")

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

        self.Buscar_lineEdit = QLineEdit(self.tab_2)
        self.Buscar_lineEdit.setObjectName(u"Buscar_lineEdit")

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

        self.buscar_pushButton = QPushButton(self.tab_2)
        self.buscar_pushButton.setObjectName(u"buscar_pushButton")
```

```
        self.gridLayout.addWidget(self.buscar_pushButton, 1, 1, 1, 1)

        self.Mostrar_Tabla_pushButton_2 = QPushButton(self.tab_2)

self.Mostrar_Tabla_pushButton_2.setObjectName(u"Mostrar_Tabla_pushButto
n_2")

        self.gridLayout.addWidget(self.Mostrar_Tabla_pushButton_2, 1,
2, 1, 1)

        self.tabWidget.addTab(self.tab_2, "")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QMenuBar(MainWindow)
        self.menubar.setObjectName(u"menubar")
        self.menubar.setGeometry(QRect(0, 0, 1359, 21))
        self.menuFile = QMenu(self.menubar)
        self.menuFile.setObjectName(u"menuFile")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QStatusBar(MainWindow)
        self.statusbar.setObjectName(u"statusbar")
        MainWindow.setStatusBar(self.statusbar)

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

        self.retranslateUi(MainWindow)

        self.tabWidget.setCurrentIndex(1)


        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.Mostrar_pushButton.setText(QCoreApplication.translate("MainWindow"
, u"Mostrar", None))
        self.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Particula", None))
        self.label.setText(QCoreApplication.translate("MainWindow",
u"Destino X:", None))
        self.label_2.setText(QCoreApplication.translate("MainWindow",
u"Destino Y:", None))
        self.label_3.setText(QCoreApplication.translate("MainWindow",
u"Velocidad:", None))
        self.label_4.setText(QCoreApplication.translate("MainWindow",
u"Red", None))
        self.label_5.setText(QCoreApplication.translate("MainWindow",
u"Green", None))
        self.label_6.setText(QCoreApplication.translate("MainWindow",
u"Blue", None))

self.AgragrInicio_pushButton.setText(QCoreApplication.translate("MainWi
ndow", u"Agregar al Inicio", None))

self.agregarFinal_pushButton.setText(QCoreApplication.translate("MainWi
ndow", u"Agregar al final", None))
        self.label_7.setText(QCoreApplication.translate("MainWindow",
u"ID:", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab),
QCoreApplication.translate("MainWindow", u"Agregar", None))

self.Buscar_lineEdit.setPlaceholderText(QCoreApplication.translate("Mai
nWindow", u"ID", None))
```

```
self.buscar_pushButton.setText(QCoreApplication.translate("MainWindow",
u"Buscar", None))

self.Mostrar_Tabla_pushButton_2.setText(QCoreApplication.translate("Mai
nWindow", u"Mostra", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2),
QCoreApplication.translate("MainWindow", u"Tabla", None))
        self.menuFile.setTitle(QCoreApplication.translate("MainWindow",
u"File", None))
    # retranslateUi
```