

Actividad 11 – Puntos más cercanos (Fuerza Bruta)

Roberto Haro González

Seminario de solución de problemas de algoritmia

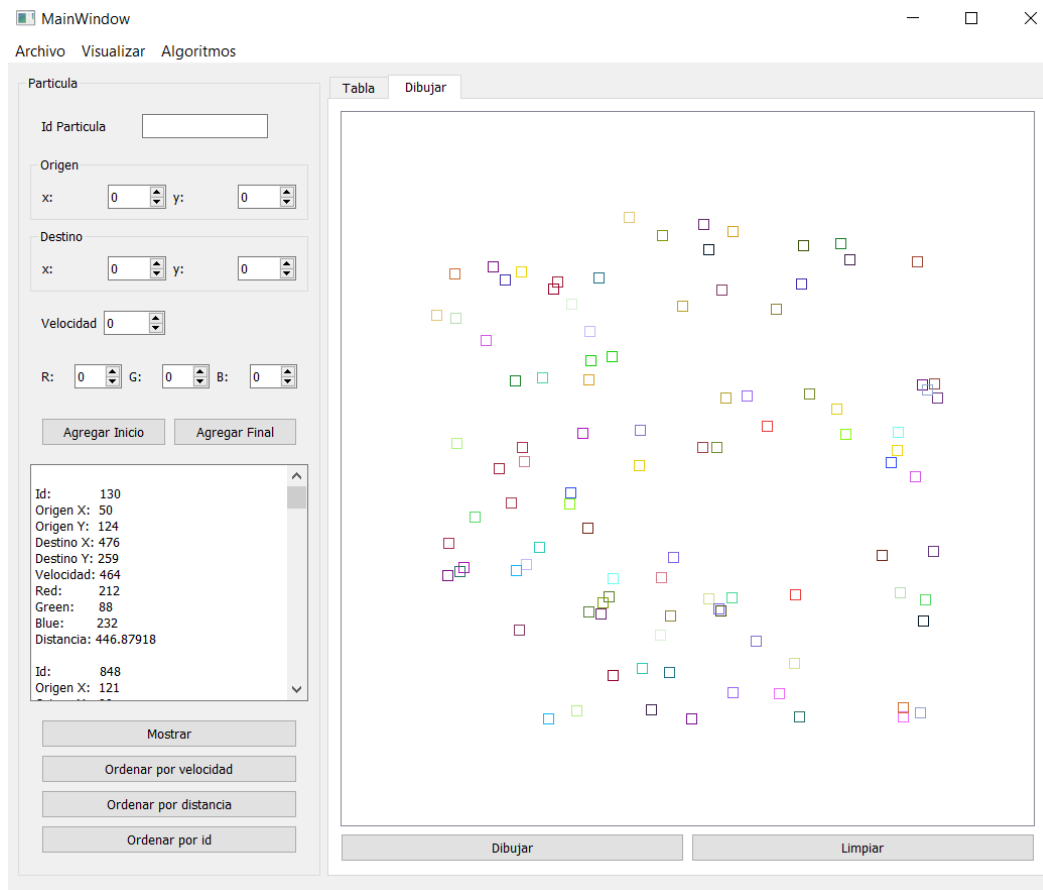
Lineamientos de evaluación

- ☑ El reporte está en formato Google Docs o PDF.
- ☑ El reporte sigue las pautas del Formato de Actividades.
- ☑ El reporte tiene desarrollada todas las pautas del Formato de Actividades
- ☑ Integra al proyecto de la interfaz gráfica desarrollada en la [Actividad 10 (sort)] el algoritmo de fuerza bruta visto en clase con las siguientes características:
 - ☑ Botón o acción para mostrar únicamente los puntos (x, y) de las partículas en el `QScene`.
 - ☑ Botón o acción para llamar al algoritmo de fuerza bruta para calcular los puntos más cercanos y mostrar en el `QScene` las líneas dibujadas a sus puntos más cercanos.
- ☑ Toma capturas de pantalla de la ejecución mostrando la visualización de los puntos en el `QScene` así como del resultado del algoritmo de fuerza bruta en el `QScene`.

Desarrollo

Habiendo analizado el video de ejemplificación el primer pasó que tomé fue rediseñar la interfaz debido a las implicaciones practicas que tiene, después de ello prepare las acciones necesarias en la interfaz y me prepararía para programar según lo visto en la clase grabada hasta conseguir los resultados esperados al crear una clase para almacenar los puntos, la respectiva lista e ir comparando los puntos y cuando hubiesen 2 como puntos mas cercanos se guardaran en una lista para luego ser impresos en pantalla y así quedaron implementados ambos métodos.

Partículas (puntos):



Partículas (puntos más cercanos):

MainWindow

Archivo Visualizar Algoritmos

Partícula

Id Partícula

Origen

x: y:

Destino

x: y:

Velocidad

R: G: B:

Agregar Inicio Agregar Final

Id: 130
Origen X: 50
Origen Y: 124
Destino X: 476
Destino Y: 259
Velocidad: 464
Red: 212
Green: 88
Blue: 232
Distancia: 446.87918

Id: 848
Origen X: 121

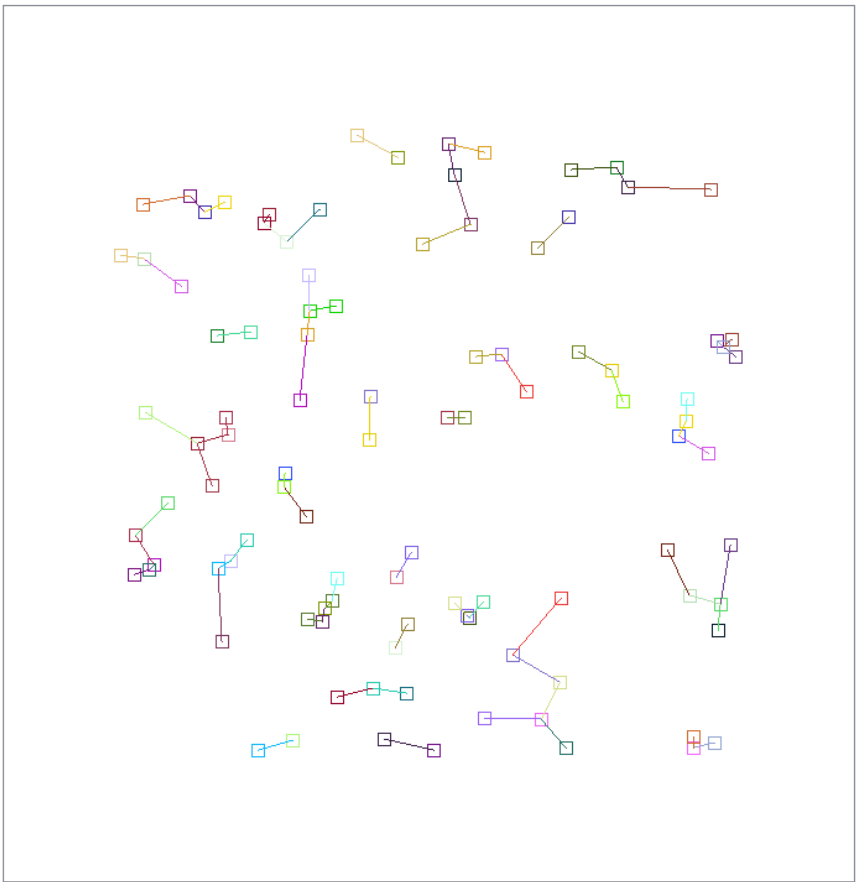
Mostrar

Ordenar por velocidad

Ordenar por distancia

Ordenar por id

Tabla Dibujar



Dibujar Limpiar

Conclusiones

Este programa como parte de mi semestre es un orgullo para mí gracias a todo el aprendizaje que llevó de la mano el aprender interfaces gráficas, en tema algoritmos etc. La verdad casi no había practicado ni siquiera las clases en Python y si en un futuro cercano me atrevo a implementar más algoritmos para él, seguro que lo haré

Referencias

MICHEL DAVALOS BOITES. (2021, 18 octubre). *Actividad 11 (Fuerza Bruta)*. YouTube.

<https://www.youtube.com/watch?v=hPARP71VEH0>

MICHEL DAVALOS BOITES. (2022, 8 abril). *Clase #11 del Seminario de Problemas de*

Algoritmia (08.abr.2022). YouTube. <https://www.youtube.com/watch?v=W6RdCvpPVUs>

Código

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))
```

lista partícula

```
from .particula import Particula
from .particula import Punto

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 __len__(self):
        return len(self.__particulas)

    def __iter__(self):
        self.cont = 0
        return self

    def __next__(self):
        if self.cont < len(self.__particulas):
            particula = self.__particulas[self.cont]
```

```

        self.cont += 1
        return particula
    else:
        raise StopIteration

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

def sort_by_id(self):
    self.__particulas.sort(key=lambda Particula: Particula.id)

def sort_by_distancia(self):
    self.__particulas.sort(key=lambda Particula: Particula.distancia,
reverse=True)

def sort_by_velocidad(self):
    self.__particulas.sort(key=lambda Particula: Particula.velocidad)

def get_puntos(self):
    puntos = []
    for particula in self.__particulas:
        punto01 = Punto(particula.origen_x, particula.origen_y, particula.red,
particula.green, particula.blue)
        punto02 = Punto(particula.destino_x, particula.destino_y,
particula.red, particula.green, particula.blue)
        puntos.append(punto01)
        puntos.append(punto02)
    return puntos

```

partícula

```
from .algoritmos import distancia_euclidiana
from random import randint

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 = randint(0,999) if id == '' else 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'
        )

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

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

    @property
    def origen_y(self):
```

```

        return self.__origen_y

@property
def destino_x(self):
    return self.__destino_x

@property
def destino_y(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
    }

```

```
class Punto:
```



```

def __init__(self, x=0, y=0, red=0, green=0, blue=0):
    self.x = x
    self.y = y
    self.red = red
    self.green = green
    self.blue = blue

```

main

```

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

app = QApplication()
window = MainWindow()
window.show()

sys.exit(app.exec_())

```

mainwindow

```

from operator import truediv
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox,
QTableWidgetItem, QGraphicsScene
from PySide2.QtGui import QPen, QColor, QTransform
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from particulas.listaparticula import ListaParticulas
from particulas.particula import Particula
from particulas.particula import Punto
from particulas.algoritmos import distancia_euclidiana

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

        self.myListaParticulas = ListaParticulas()
        self.puntos = []
        self.puntos_cercanos = []

        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.mostrar_pushButton.clicked.connect(self.mostrar_tabla)

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

self.ui.buscar_pushButton.clicked.connect(self.buscar_particula)

self.ui.dibujar_pushButton.clicked.connect(self.dibujar_grafos)
self.ui.limpiar_pushButton.clicked.connect(self.limpiar)

self.ui.actionPuntos.triggered.connect(self.dibujar_puntos)
self.ui.actionGrafos.triggered.connect(self.dibujar_grafos)
self.ui.actionPuntos_mas_cercanos.triggered.connect(self.calcular_puntos_ma
s_cercanos)

self.scene = QGraphicsScene()
self.ui.graphicsView.setScene(self.scene)

self.ui.ordenar_por_id_pushButton.clicked.connect(self.sort_by_id)
self.ui.ordenar_por_distancia_pushButton.clicked.connect(self.sort_by_dista
ncia)
self.ui.ordenar_por_velocidad_pushButton.clicked.connect(self.sort_by_veloc
idad)

@Slot()
def buscar_particula(self):
    id = self.ui.buscar_lineEdit.text()

    encontrado = False
    for particula in self.myListaParticulas:

        if int(id) == particula.id:
            self.ui.tabla_tableWidget.clear()
            self.ui.tabla_tableWidget.setColumnCount(10)
            headers = ["Id", "Origen X", "Origen Y ", "Destino X", "Destino Y",
"Velocidad", "Red", "Green", "Blue", "Distancia"]
            self.ui.tabla_tableWidget.setHorizontalHeaderLabels(headers)
            self.ui.tabla_tableWidget.setRowCount(1)

            id_widget = QTableWidgetItem(str(particula.id))
            origen_x_widget = QTableWidgetItem(str(particula.origen_x))
            origen_y_widget = QTableWidgetItem(str(particula.origen_y))
            destino_x_widget = QTableWidgetItem(str(particula.destino_x))
            destino_y_widget = QTableWidgetItem(str(particula.destino_y))

```

```

        velocidad_widget = QTableWidgetItem(str(particula.velocidad))
        green_widget = QTableWidgetItem(str(particula.green))
        red_widget = QTableWidgetItem(str(particula.red))
        blue_widget = QTableWidgetItem(str(particula.blue))
        distancia_widget = QTableWidgetItem(str(particula.distancia))

        self.ui.tabla_tableWidget.setItem(0, 0, id_widget)
        self.ui.tabla_tableWidget.setItem(0, 1, origen_x_widget)
        self.ui.tabla_tableWidget.setItem(0, 2, origen_y_widget)
        self.ui.tabla_tableWidget.setItem(0, 3, destino_x_widget)
        self.ui.tabla_tableWidget.setItem(0, 4, destino_y_widget)
        self.ui.tabla_tableWidget.setItem(0, 5, velocidad_widget)
        self.ui.tabla_tableWidget.setItem(0, 6, green_widget)
        self.ui.tabla_tableWidget.setItem(0, 7, red_widget)
        self.ui.tabla_tableWidget.setItem(0, 8, blue_widget)
        self.ui.tabla_tableWidget.setItem(0, 9, distancia_widget)

        encontrado = True
        return
    if not encontrado:
        QMessageBox.warning(
            self,
            "Atención",
            f'La partícula con la id: "{id}" no fue encontrada'
        )

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

        self.ui.tabla_tableWidget.setRowCount(len(self.myListaParticulas))

        row = 0
        for particula in self.myListaParticulas:
            id_widget = QTableWidgetItem(str(particula.id))
            origen_x_widget = QTableWidgetItem(str(particula.origen_x))
            origen_y_widget = QTableWidgetItem(str(particula.origen_y))
            destino_x_widget = QTableWidgetItem(str(particula.destino_x))
            destino_y_widget = QTableWidgetItem(str(particula.destino_y))
            velocidad_widget = QTableWidgetItem(str(particula.velocidad))
            red_widget = QTableWidgetItem(str(particula.red))
            green_widget = QTableWidgetItem(str(particula.green))

```

```

        blue_widget = QTableWidgetItem(str(particula.blue))
        distancia_widget = QTableWidgetItem(str(particula.distancia))

        self.ui.tabla_tableWidget.setItem(row, 0, id_widget)
        self.ui.tabla_tableWidget.setItem(row, 1, origen_x_widget)
        self.ui.tabla_tableWidget.setItem(row, 2, origen_y_widget)
        self.ui.tabla_tableWidget.setItem(row, 3, destino_x_widget)
        self.ui.tabla_tableWidget.setItem(row, 4, destino_y_widget)
        self.ui.tabla_tableWidget.setItem(row, 5, velocidad_widget)
        self.ui.tabla_tableWidget.setItem(row, 7, green_widget)
        self.ui.tabla_tableWidget.setItem(row, 6, red_widget)
        self.ui.tabla_tableWidget.setItem(row, 8, blue_widget)
        self.ui.tabla_tableWidget.setItem(row, 9, distancia_widget)

        row += 1

@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_lineEdit.text()
    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_lineEdit.text()
    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)

#DIBUJAR
@Slot()
def dibujar_grafos(self):
    self.scene.clear()
    pen = QPen()
    pen.setWidth(2)

    for particula in self.myListaParticulas:
        r = particula.red
        g = particula.green
        b = particula.blue

        color = QColor(r,g,b)
        pen.setColor(color)

        origen_x = particula.origen_x
        origen_y = particula.origen_y
        destino_x = particula.destino_x
        destino_y = particula.destino_y

        self.scene.addEllipse(origen_x,origen_y,3,3, pen)
        self.scene.addEllipse(destino_x,destino_y,3,3,pen)
        self.scene.addLine(origen_x+3,origen_y+3,destino_x,destino_y,pen)

@Slot()
def limpiar(self):
    self.scene.clear()
    self.myListaParticulas = ListaParticulas()
    self.puntos = []
    self.puntos_cercanos = []

@Slot()
def sort_by_id(self):
    self.myListaParticulas.sort_by_id()

@Slot()
def sort_by_distancia(self):
    self.myListaParticulas.sort_by_distancia()

```

```

@Slot()
def sort_by_velocidad(self):
    self.myListaParticulas.sort_by_velocidad()

@Slot()
def dibujar_puntos(self):
    self.scene.clear()
    self.puntos = self.myListaParticulas.get_puntos()
    for punto in self.puntos:
        x = punto.x
        y = punto.y
        red = punto.red
        green = punto.green
        blue = punto.blue
        color = QColor(red, green, blue)
        pen = QPen()
        pen.setColor(color)

        self.scene.addRect(x, y, 10, 10, pen)

@Slot()
def dibujar_puntos_mas_cercanos(self):
    for punto01, punto02 in self.puntos_cercanos:
        pen = QPen()
        color = QColor(punto01.red, punto01.green, punto01.blue)
        pen.setColor(color)
        self.scene.addLine(punto01.x+5, punto01.y+5, punto02.x+5, punto02.y+5,
pen)

@Slot()
def calcular_puntos_mas_cercanos(self):
    for punto01 in self.puntos:
        distMin = 1000
        punto = Punto
        for todos in self.puntos:
            if punto01 == todos:
                continue
            dist = distancia_euclidiana(punto01.x, punto01.y, todos.x, todos.y)
            if dist < distMin:
                distMin = dist
                punto = todos
        self.puntos_cercanos.append([punto01, punto])

```

```
self.dibujar_puntos_mas_cercanos()
```

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(1046, 868)
        self.actionAbrir = QAction(MainWindow)
        self.actionAbrir.setObjectName(u"actionAbrir")
        self.actionGuardar = QAction(MainWindow)
        self.actionGuardar.setObjectName(u"actionGuardar")
        self.actionPuntos = QAction(MainWindow)
        self.actionPuntos.setObjectName(u"actionPuntos")
        self.actionPuntos_mas_cercanos = QAction(MainWindow)
        self.actionPuntos_mas_cercanos.setObjectName(u"actionPuntos_mas_cercanos")
        self.actionGrafos = QAction(MainWindow)
        self.actionGrafos.setObjectName(u"actionGrafos")
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.gridLayout = QGridLayout(self.centralwidget)
        self.gridLayout.setObjectName(u"gridLayout")
        self.groupBox = QGroupBox(self.centralwidget)
        self.groupBox.setObjectName(u"groupBox")
        self.groupBox.setMaximumSize(QSize(300, 16777215))
        self.verticalLayout_2 = QVBoxLayout(self.groupBox)
        self.verticalLayout_2.setObjectName(u"verticalLayout_2")
        self.widget_6 = QWidget(self.groupBox)
        self.widget_6.setObjectName(u"widget_6")
        self.horizontalLayout_5 = QHBoxLayout(self.widget_6)
```



```

self.horizontalLayout_5.setObjectName(u"horizontalLayout_5")
self.label_7 = QLabel(self.widget_6)
self.label_7.setObjectName(u"label_7")

self.horizontalLayout_5.addWidget(self.label_7)

self.horizontalSpacer_2 = QSpacerItem(40, 20, QSizePolicy.Expanding,
QSizePolicy.Minimum)

self.horizontalLayout_5.addItem(self.horizontalSpacer_2)

self.id_lineEdit = QLineEdit(self.widget_6)
self.id_lineEdit.setObjectName(u"id_lineEdit")

self.horizontalLayout_5.addWidget(self.id_lineEdit)

self.horizontalSpacer = QSpacerItem(40, 20, QSizePolicy.Expanding,
QSizePolicy.Minimum)

self.horizontalLayout_5.addItem(self.horizontalSpacer)

self.verticalLayout_2.addWidget(self.widget_6)

self.groupBox_2 = QGroupBox(self.groupBox)
self.groupBox_2.setObjectName(u"groupBox_2")
self.horizontalLayout_3 = QHBoxLayout(self.groupBox_2)
self.horizontalLayout_3.setObjectName(u"horizontalLayout_3")
self.label_8 = QLabel(self.groupBox_2)
self.label_8.setObjectName(u"label_8")

self.horizontalLayout_3.addWidget(self.label_8)

self.origenX_spinBox = QSpinBox(self.groupBox_2)
self.origenX_spinBox.setObjectName(u"origenX_spinBox")
self.origenX_spinBox.setMaximum(500)

self.horizontalLayout_3.addWidget(self.origenX_spinBox)

self.label_9 = QLabel(self.groupBox_2)
self.label_9.setObjectName(u"label_9")

self.horizontalLayout_3.addWidget(self.label_9)

self.origenY_spinBox = QSpinBox(self.groupBox_2)
self.origenY_spinBox.setObjectName(u"origenY_spinBox")

```

```

self.origenY_spinBox.setMaximum(500)

self.horizontalLayout_3.addWidget(self.origenY_spinBox)

self.verticalLayout_2.addWidget(self.groupBox_2)

self.groupBox_3 = QGroupBox(self.groupBox)
self.groupBox_3.setObjectName(u"groupBox_3")
self.horizontalLayout_4 = QHBoxLayout(self.groupBox_3)
self.horizontalLayout_4.setObjectName(u"horizontalLayout_4")
self.label = QLabel(self.groupBox_3)
self.label.setObjectName(u"label")

self.horizontalLayout_4.addWidget(self.label)

self.destinoX_spinBox = QSpinBox(self.groupBox_3)
self.destinoX_spinBox.setObjectName(u"destinoX_spinBox")
self.destinoX_spinBox.setMaximum(500)

self.horizontalLayout_4.addWidget(self.destinoX_spinBox)

self.label_2 = QLabel(self.groupBox_3)
self.label_2.setObjectName(u"label_2")

self.horizontalLayout_4.addWidget(self.label_2)

self.destinoY_spinBox = QSpinBox(self.groupBox_3)
self.destinoY_spinBox.setObjectName(u"destinoY_spinBox")
self.destinoY_spinBox.setMaximum(500)

self.horizontalLayout_4.addWidget(self.destinoY_spinBox)

self.verticalLayout_2.addWidget(self.groupBox_3)

self.widget_7 = QWidget(self.groupBox)
self.widget_7.setObjectName(u"widget_7")
self.horizontalLayout_6 = QHBoxLayout(self.widget_7)
self.horizontalLayout_6.setObjectName(u"horizontalLayout_6")
self.label_3 = QLabel(self.widget_7)
self.label_3.setObjectName(u"label_3")

self.horizontalLayout_6.addWidget(self.label_3)

self.velocidad_spinBox = QSpinBox(self.widget_7)

```

```

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

self.horizontalLayout_6.addWidget(self.velocidad_spinBox)

self.horizontalSpacer_3 = QSpacerItem(40, 20, QSizePolicy.Expanding,
QSizePolicy.Minimum)

self.horizontalLayout_6.addItem(self.horizontalSpacer_3)

self.verticalLayout_2.addWidget(self.widget_7)

self.widget = QWidget(self.groupBox)
self.widget.setObjectName(u"widget")
self.horizontalLayout = QHBoxLayout(self.widget)
self.horizontalLayout.setObjectName(u"horizontalLayout")
self.label_4 = QLabel(self.widget)
self.label_4.setObjectName(u"label_4")

self.horizontalLayout.addWidget(self.label_4)

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

self.horizontalLayout.addWidget(self.red_spinBox)

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

self.horizontalLayout.addWidget(self.label_5)

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

self.horizontalLayout.addWidget(self.green_spinBox)

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

self.horizontalLayout.addWidget(self.label_6)

self.blue__spinBox = QSpinBox(self.widget)

```

```

self.blue__spinBox.setObjectName(u"blue__spinBox")
self.blue__spinBox.setMaximum(255)

self.horizontalLayout.addWidget(self.blue__spinBox)

self.verticalLayout_2.addWidget(self.widget)

self.widget_3 = QWidget(self.groupBox)
self.widget_3.setObjectName(u"widget_3")
self.horizontalLayout_2 = QHBoxLayout(self.widget_3)
self.horizontalLayout_2.setObjectName(u"horizontalLayout_2")
self.agregar_inicio_pushButton = QPushButton(self.widget_3)
self.agregar_inicio_pushButton.setObjectName(u"agregar_inicio_pushButton")

self.horizontalLayout_2.addWidget(self.agregar_inicio_pushButton)

self.agregar_final_pushButton = QPushButton(self.widget_3)
self.agregar_final_pushButton.setObjectName(u"agregar_final_pushButton")

self.horizontalLayout_2.addWidget(self.agregar_final_pushButton)

self.verticalLayout_2.addWidget(self.widget_3)

self.out_plainTextEdit = QPlainTextEdit(self.groupBox)
self.out_plainTextEdit.setObjectName(u"out_plainTextEdit")
self.out_plainTextEdit.setMinimumSize(QSize(100, 100))

self.verticalLayout_2.addWidget(self.out_plainTextEdit)

self.widget_2 = QWidget(self.groupBox)
self.widget_2.setObjectName(u"widget_2")
self.verticalLayout = QVBoxLayout(self.widget_2)
self.verticalLayout.setObjectName(u"verticalLayout")
self.mostrar_pushButton = QPushButton(self.widget_2)
self.mostrar_pushButton.setObjectName(u"mostrar_pushButton")

self.verticalLayout.addWidget(self.mostrar_pushButton)

self.ordenar_por_velocidad_pushButton = QPushButton(self.widget_2)
self.ordenar_por_velocidad_pushButton.setObjectName(u"ordenar_por_velocidad_pushButton")

self.verticalLayout.addWidget(self.ordenar_por_velocidad_pushButton)

```

```

self.ordenar_por_distancia_pushButton = QPushButton(self.widget_2)
self.ordenar_por_distancia_pushButton.setObjectName(u"ordenar_por_distancia_pushButton")

self.verticalLayout.addWidget(self.ordenar_por_distancia_pushButton)

self.ordenar_por_id_pushButton = QPushButton(self.widget_2)
self.ordenar_por_id_pushButton.setObjectName(u"ordenar_por_id_pushButton")

self.verticalLayout.addWidget(self.ordenar_por_id_pushButton)

self.verticalLayout_2.addWidget(self.widget_2)

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

self.tabWidget = QTabWidget(self.centralwidget)
self.tabWidget.setObjectName(u"tabWidget")
self.tabla_tab = QWidget()
self.tabla_tab.setObjectName(u"tabla_tab")
self.gridLayout_3 = QGridLayout(self.tabla_tab)
self.gridLayout_3.setObjectName(u"gridLayout_3")
self.tabla_tableWidget = QTableWidget(self.tabla_tab)
self.tabla_tableWidget.setObjectName(u"tabla_tableWidget")

self.gridLayout_3.addWidget(self.tabla_tableWidget, 0, 0, 1, 3)

self.buscar_pushButton = QPushButton(self.tabla_tab)
self.buscar_pushButton.setObjectName(u"buscar_pushButton")

self.gridLayout_3.addWidget(self.buscar_pushButton, 1, 1, 1, 2)

self.buscar_lineEdit = QLineEdit(self.tabla_tab)
self.buscar_lineEdit.setObjectName(u"buscar_lineEdit")

self.gridLayout_3.addWidget(self.buscar_lineEdit, 1, 0, 1, 1)

self.tabWidget.addTab(self.tabla_tab, "")
self.tab_5 = QWidget()
self.tab_5.setObjectName(u"tab_5")
self.gridLayout_4 = QGridLayout(self.tab_5)
self.gridLayout_4.setObjectName(u"gridLayout_4")
self.graphicsView = QGraphicsView(self.tab_5)
self.graphicsView.setObjectName(u"graphicsView")

```

```

self.gridLayout_4.addWidget(self.graphicsView, 0, 0, 1, 2)

self.limpiar_pushButton = QPushButton(self.tab_5)
self.limpiar_pushButton.setObjectName(u"limpiar_pushButton")

self.gridLayout_4.addWidget(self.limpiar_pushButton, 2, 1, 1, 1)

self.dibujar_pushButton = QPushButton(self.tab_5)
self.dibujar_pushButton.setObjectName(u"dibujar_pushButton")

self.gridLayout_4.addWidget(self.dibujar_pushButton, 2, 0, 1, 1)

self.tabWidget.addTab(self.tab_5, "")

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

MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QMenuBar(MainWindow)
self.menubar.setObjectName(u"menubar")
self.menubar.setGeometry(QRect(0, 0, 1046, 26))
self.menuArchivo = QMenu(self.menubar)
self.menuArchivo.setObjectName(u"menuArchivo")
self.menuAlgotirmos = QMenu(self.menubar)
self.menuAlgotirmos.setObjectName(u"menuAlgotirmos")
self.menuAlgoritmos = QMenu(self.menubar)
self.menuAlgoritmos.setObjectName(u"menuAlgoritmos")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QStatusBar(MainWindow)
self.statusbar.setObjectName(u"statusbar")
MainWindow.setStatusBar(self.statusbar)

self.menubar.addAction(self.menuArchivo.menuAction())
self.menubar.addAction(self.menuAlgotirmos.menuAction())
self.menubar.addAction(self.menuAlgoritmos.menuAction())
self.menuArchivo.addAction(self.actionAbrir)
self.menuArchivo.addAction(self.actionGuardar)
self.menuAlgotirmos.addAction(self.actionPuntos)
self.menuAlgotirmos.addAction(self.actionGrafos)
self.menuAlgoritmos.addAction(self.actionPuntos_mas_cercanos)

self.retranslateUi(MainWindow)

self.tabWidget.setCurrentIndex(0)

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.actionPuntos.setText(QCoreApplication.translate("MainWindow",
u"Puntos", None))
    #if QT_CONFIG(shortcut)
        self.actionPuntos.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+P", None))
    #endif // QT_CONFIG(shortcut)
    self.actionPuntos_mas_cercanos.setText(QCoreApplication.translate("MainWind
ow", u"Puntos m\u00e1s cercanos", None))
    self.actionGrafos.setText(QCoreApplication.translate("MainWindow",
u"Grafos", None))
    #if QT_CONFIG(shortcut)
        self.actionGrafos.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+G", None))
    #endif // QT_CONFIG(shortcut)
    self.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Particula", None))
    self.label_7.setText(QCoreApplication.translate("MainWindow", u"Id
Particula", None))
    self.groupBox_2.setTitle(QCoreApplication.translate("MainWindow",
u"Origen", None))
    self.label_8.setText(QCoreApplication.translate("MainWindow", u"x:", None))
    self.label_9.setText(QCoreApplication.translate("MainWindow", u"y:", None))
    self.groupBox_3.setTitle(QCoreApplication.translate("MainWindow",
u"Destino", None))
    self.label.setText(QCoreApplication.translate("MainWindow", u"x:", None))
    self.label_2.setText(QCoreApplication.translate("MainWindow", u"y:", None))
    self.label_3.setText(QCoreApplication.translate("MainWindow", u"Velocidad",
None))

```

```

        self.label_4.setText(QCoreApplication.translate("MainWindow", u"R:", None))
        self.label_5.setText(QCoreApplication.translate("MainWindow", u"G:", None))
        self.label_6.setText(QCoreApplication.translate("MainWindow", u"B:", None))
        self.agregar_inicio_pushButton.setText(QCoreApplication.translate("MainWind
ow", u"Agregar Inicio", None))
        self.agregar_final_pushButton.setText(QCoreApplication.translate("MainWindo
w", u"Agregar Final", None))
        self.mostrar_pushButton.setText(QCoreApplication.translate("MainWindow",
u"Mostrar", None))
        self.ordenar_por_velocidad_pushButton.setText(QCoreApplication.translate("M
ainWindow", u"Ordenar por velocidad", None))
        self.ordenar_por_distancia_pushButton.setText(QCoreApplication.translate("M
ainWindow", u"Ordenar por distancia", None))
        self.ordenar_por_id_pushButton.setText(QCoreApplication.translate("MainWind
ow", u"Ordenar por id", None))
        self.buscar_pushButton.setText(QCoreApplication.translate("MainWindow",
u"Buscar", None))
        self.buscar_lineEdit.setText("")
        self.buscar_lineEdit.setPlaceholderText(QCoreApplication.translate("MainWin
dow", u"Id de la part\u00e9cula", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tabla_tab),
QCoreApplication.translate("MainWindow", u"Tabla", None))
        self.limpiar_pushButton.setText(QCoreApplication.translate("MainWindow",
u"Limpiar", None))
        self.dibujar_pushButton.setText(QCoreApplication.translate("MainWindow",
u"Dibujar", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_5),
QCoreApplication.translate("MainWindow", u"Dibujar", None))
        self.menuArchivo.setTitle(QCoreApplication.translate("MainWindow",
u"Archivo", None))
        self.menuAlgoritmos.setTitle(QCoreApplication.translate("MainWindow",
u"Visualizar", None))
        self.menuAlgoritmos.setTitle(QCoreApplication.translate("MainWindow",
u"Algoritmos", None))
        # retranslateUi

```