

ACTIVIDAD 08

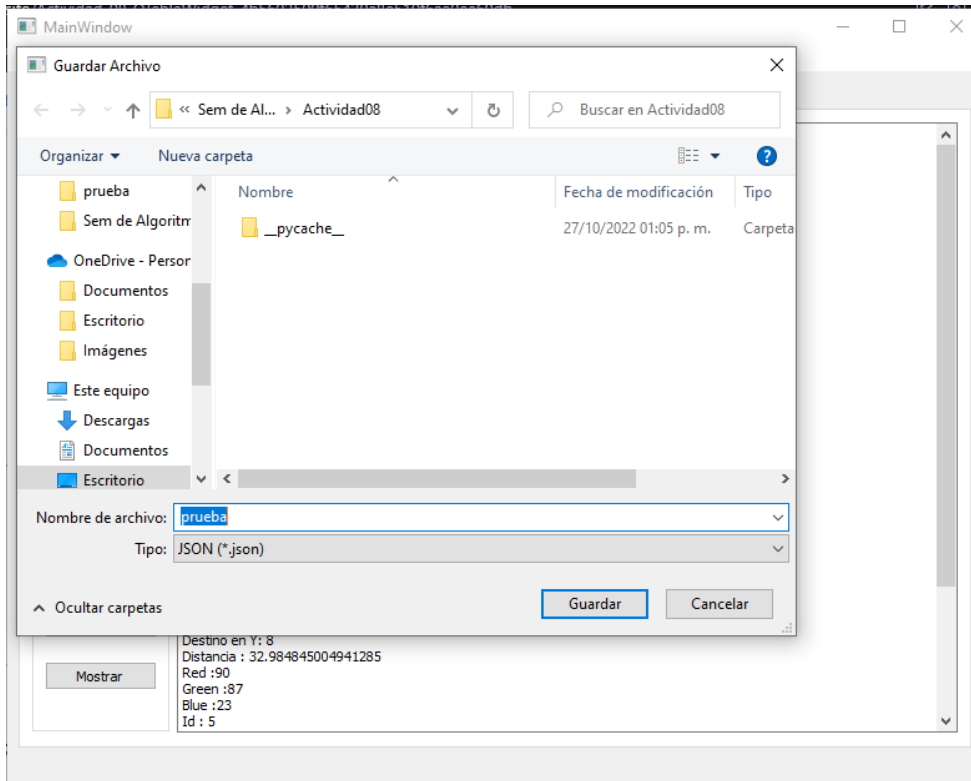
QTableWidget

Gomez Casillas Hector Samuel

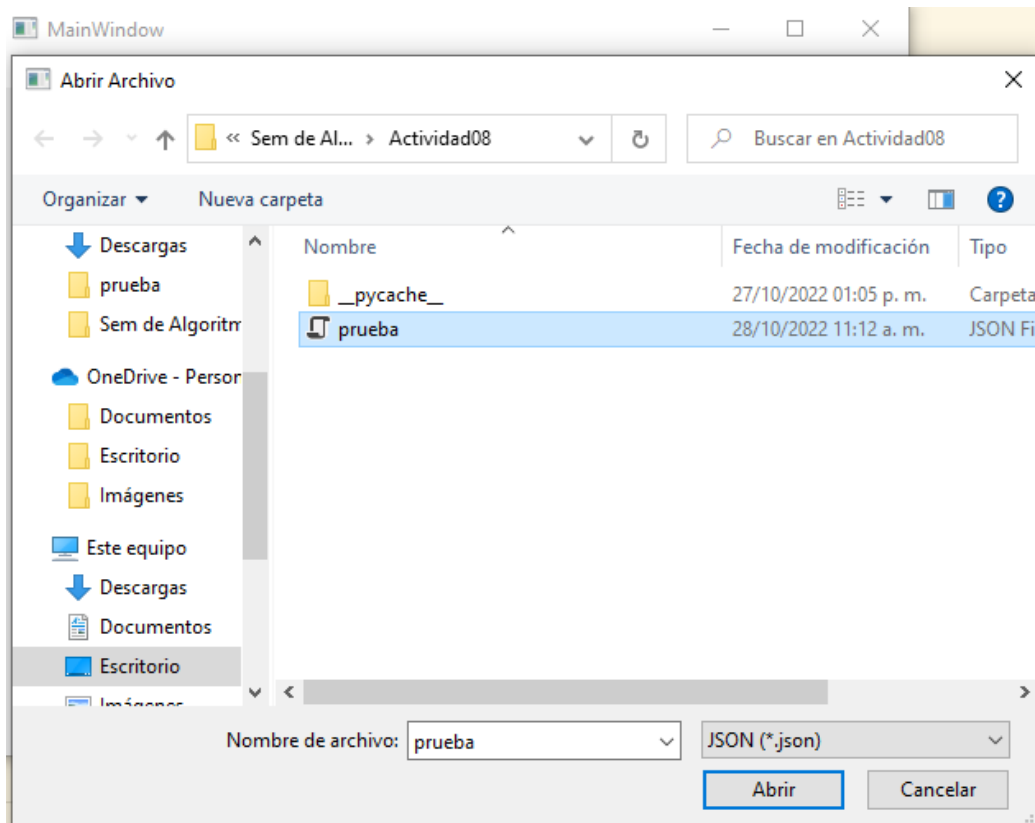
SEMINARIO DE SOLUCION DE PROBLEMAS DE ALGORITMIA

- 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.

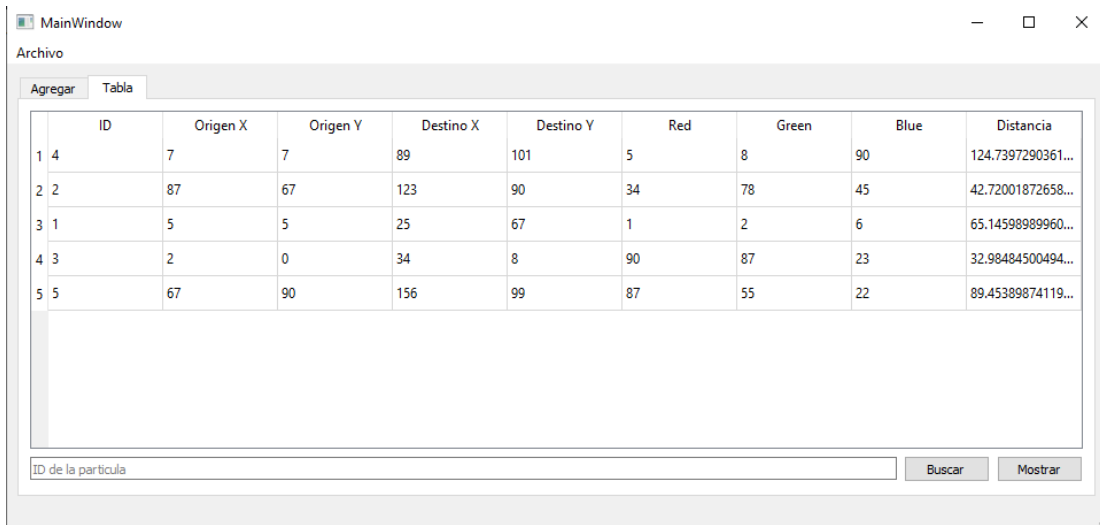
Captura en un archivo de 5 partículas “prueba.json”



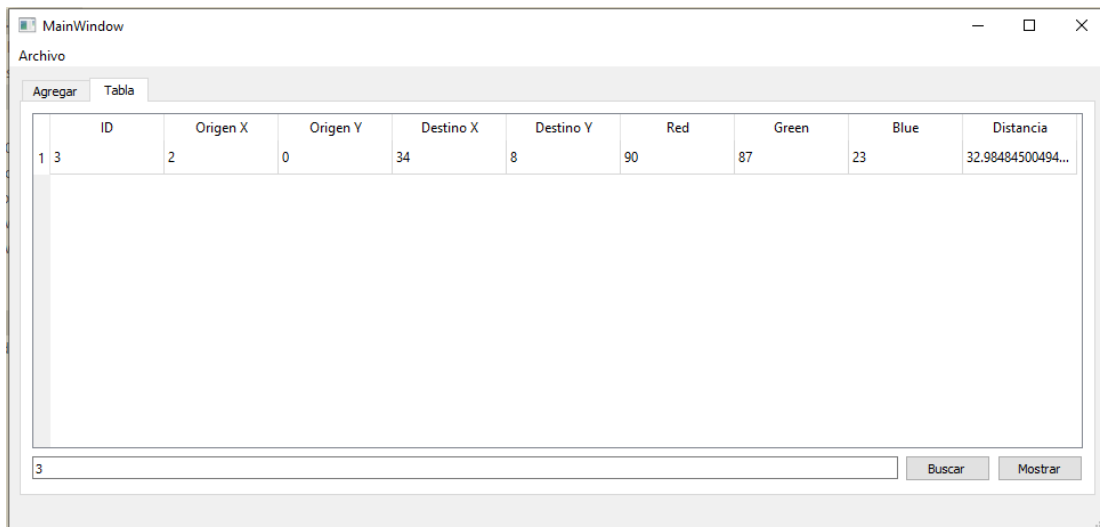
Recuperación del archivo “prueba.json”



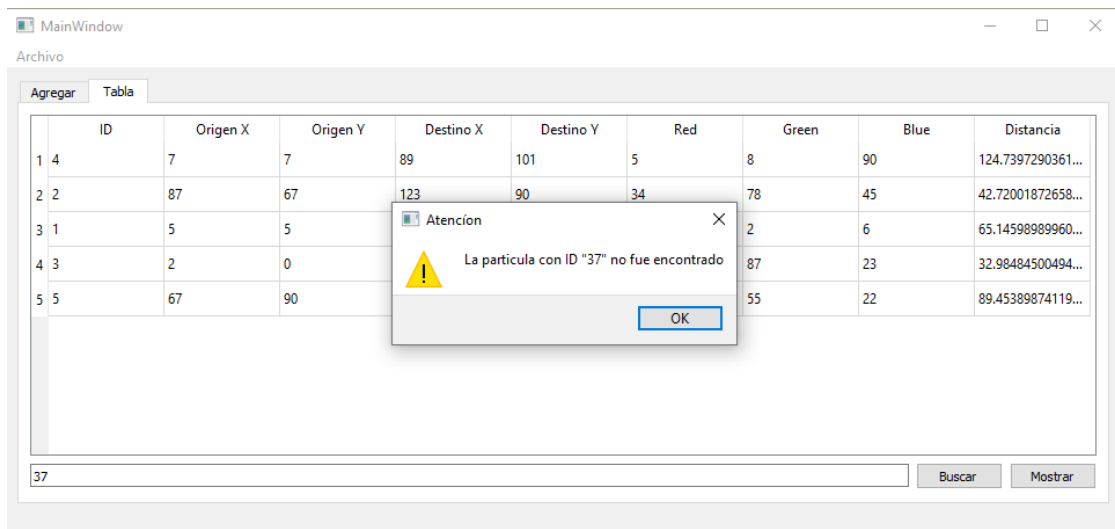
Muestra de las partículas en la tabla



Búsqueda de una partícula con ID que SI existe



Búsqueda de una partícula con ID que NO existe



CONCLUSIONES

Fue una actividad mas sencilla que las anteriores respecto a la complementación de la interfaz gráfica, agregué algo que no venia en el tutorial, que a la hora de buscar una partícula muestre el nombre de las columnas (ID,Origen,Destino,ect)., solamente a la hora de hacer la búsqueda había olvidado que como en el search_line() la entrada que introduzca el usuario es una string y necesitabas un entero para realizar la búsqueda, no me funcionaba, hasta que recordé que teníamos que convertirlo con int(), de ahí en mas todo salió bien.

REFERENCIAS

- PySide2 - Clases y Objetos (Qt for Python)(V) (MICHEL DAVALOS BOITES).
<https://www.youtube.com/watch?v=1yEpAHaiMxs>

Código “administradora.py”:

```
from partícula import Partícula
import json

class Administradora:
    def __init__(self):
        self.__partículas = []

    def agregar_final(self,partícula:Partícula):
        self.__partículas.append(partícula)

    def agregar_inicio(self,partícula:Partícula):
        self.__partículas.insert(0,partícula)

    def mostrar(self):
        for partícula in self.__partículas:
            print(partícula)

    def __str__(self):
        return "".join(
            str(partícula) for partícula in self.__partículas
        )

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

    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
    except:
        return 0
        #json.dump()

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

```

Codigo “algoritmos.py”:

```

import math

def distancia_euclidiana(x_1, y_1, x_2, y_2):
    a = (x_2 - x_1)*(x_2 - x_1)
    b = (y_2 - y_1)*(y_2 - y_1)

    c = a + b

    distancia = math.sqrt(c)

    return distancia

```

Codigo “mainwindow.py”:

```
from PySide2.QtWidgets import
QMainWindow,QFileDialog,QMessageBox,QTableWidgetItem
from ui_mainwindow import Ui_MainWindow
from administradora import Administradora
from particula import Particula
from PySide2.QtCore import Slot

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

        self.administrador = Administradora()

        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.Agregar_final.clicked.connect(self.agregar_final)
        self.ui.Agregar_Inicio.clicked.connect(self.agregar_inicio)
        self.ui.Mostrar.clicked.connect(self.ver)

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

        self.ui.view_button.clicked.connect(self.mostrar_tabla)
        self.ui.search_button.clicked.connect(self.buscar_tabla)

    @Slot()
    def buscar_tabla(self):
        id = self.ui.search_line.text()

        encontrado = False

        for particula in self.administrador:

            if int(id) == particula.id:
                self.ui.table.clear()
                self.ui.table.setRowCount(1)
                headers = ["ID","Origen X","Origen Y","Destino X","Destino
Y","Red","Green","Blue","Distancia"]
                self.ui.table.setHorizontalHeaderLabels(headers)
```

```

        id_widget = QTableWidgetItem(str(particula.id))
        origenx_widget = QTableWidgetItem(str(particula.origen_x))
        origeny_widget = QTableWidgetItem(str(particula.origen_y))
        destinox_widget = QTableWidgetItem(str(particula.destino_x))
        destinoy_widget = QTableWidgetItem(str(particula.destino_y))
        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.table.setItem(0,0,id_widget)
        self.ui.table.setItem(0,1,origenx_widget)
        self.ui.table.setItem(0,2,origeny_widget)
        self.ui.table.setItem(0,3,destinox_widget)
        self.ui.table.setItem(0,4,destinoy_widget)
        self.ui.table.setItem(0,5,red_widget)
        self.ui.table.setItem(0,6,green_widget)
        self.ui.table.setItem(0,7,blue_widget)
        self.ui.table.setItem(0,8,distancia_widget)

        encontrado = True

        return

    if not encontrado:
        QMessageBox.warning(self, 'Atención', f'La particula con ID "{id}"
no fue encontrado')

@Slot()
def mostrar_tabla(self):
    self.ui.table.setColumnCount(9)
    headers = ["ID", "Origen X", "Origen Y", "Destino X", "Destino
Y", "Red", "Green", "Blue", "Distancia"]
    self.ui.table.setHorizontalHeaderLabels(headers)

    self.ui.table.setRowCount(len(self.administrador))

    row = 0
    for particula in self.administrador:

```

```

        id_widget = QTableWidgetItem(str(particula.id))
        origenx_widget = QTableWidgetItem(str(particula.origen_x))
        origeny_widget = QTableWidgetItem(str(particula.origen_y))
        destinox_widget = QTableWidgetItem(str(particula.destino_x))
        destinoy_widget = QTableWidgetItem(str(particula.destino_y))
        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.table.setItem(row,0,id_widget)
        self.ui.table.setItem(row,1,origenx_widget)
        self.ui.table.setItem(row,2,origeny_widget)
        self.ui.table.setItem(row,3,destinox_widget)
        self.ui.table.setItem(row,4,destinoy_widget)
        self.ui.table.setItem(row,5,red_widget)
        self.ui.table.setItem(row,6,green_widget)
        self.ui.table.setItem(row,7,blue_widget)
        self.ui.table.setItem(row,8,distancia_widget)

        row += 1

    @Slot()
    def action_abrir_archivo(self):
        ubicacion = QFileDialog.getOpenFileName(self, 'Abrir
Archivo', '.', 'JSON (*.json)')[0]
        if self.administrador.abrir(ubicacion):
            QMessageBox.information(self, "Exitó", "Se abrió el archivo de " +
ubicacion)
        else:
            QMessageBox.information(self, "Error", "No se pudo abrir el
archivo de " + ubicacion)

    @Slot()
    def action_guardar_archivo(self):
        ubicacion = QFileDialog.getSaveFileName(self, 'Guardar
Archivo', '.', 'JSON (*.json)')[0]
        if self.administrador.guardar(ubicacion):
            QMessageBox.information(self, "Exitó", "Se creó el archivo con
éxito en " + ubicacion)
        else:
            QMessageBox.information(self, "Error", "No se pudo crear el
archivo en " + ubicacion)

```



```

@Slot()
def ver(self):
    self.ui.Print.clear()
    self.ui.Print.insertPlainText(str(self.administrador))

@Slot()
def agregar_final(self):
    ID = self.ui.ID_spinBox.value()
    OrigenX = self.ui.OrigenX_spinBox.value()
    OrigenY = self.ui.OrigenY_spinBox.value()
    DestinoX = self.ui.DestinoX_spinBox.value()
    DestinoY = self.ui.DestinoY_spinBox.value()
    Red = self.ui.Red_spinBox.value()
    Green = self.ui.Green_spinBox.value()
    Blue = self.ui.Blue_spinBox.value()

    particula1 =
Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY,Red,Green,Blue)
    self.administrador.agregar_final(particula1)

@Slot()
def agregar_inicio(self):
    ID = self.ui.ID_spinBox.value()
    OrigenX = self.ui.OrigenX_spinBox.value()
    OrigenY = self.ui.OrigenY_spinBox.value()
    DestinoX = self.ui.DestinoX_spinBox.value()
    DestinoY = self.ui.DestinoY_spinBox.value()
    Red = self.ui.Red_spinBox.value()
    Green = self.ui.Green_spinBox.value()
    Blue = self.ui.Blue_spinBox.value()

    particula1 =
Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY,Red,Green,Blue)
    self.administrador.agregar_inicio(particula1)

```

Codigo "particula.py":

```
from PySide2.QtWidgets import
QMainWindow,QFileDialog,QMessageBox,QTableWidgetItem
from ui_mainwindow import Ui_MainWindow
from administradora import Administradora
from particula import Particula
from PySide2.QtCore import Slot

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

        self.administrador = Administradora()

        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.Agregar_final.clicked.connect(self.agregar_final)
        self.ui.Agregar_Inicio.clicked.connect(self.agregar_inicio)
        self.ui.Mostrar.clicked.connect(self.ver)

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

        self.ui.view_button.clicked.connect(self.mostrar_tabla)
        self.ui.search_button.clicked.connect(self.buscar_tabla)

    @Slot()
    def buscar_tabla(self):
        id = self.ui.search_line.text()

        encontrado = False

        for particula in self.administrador:

            if int(id) == particula.id:
                self.ui.table.clear()
                self.ui.table.setRowCount(1)
                headers = ["ID","Origen X","Origen Y","Destino X","Destino
Y","Red","Green","Blue","Distancia"]
                self.ui.table.setHorizontalHeaderLabels(headers)
```

```

        id_widget = QTableWidgetItem(str(particula.id))
        origenx_widget = QTableWidgetItem(str(particula.origen_x))
        origeny_widget = QTableWidgetItem(str(particula.origen_y))
        destinox_widget = QTableWidgetItem(str(particula.destino_x))
        destinoy_widget = QTableWidgetItem(str(particula.destino_y))
        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.table.setItem(0,0,id_widget)
        self.ui.table.setItem(0,1,origenx_widget)
        self.ui.table.setItem(0,2,origeny_widget)
        self.ui.table.setItem(0,3,destinox_widget)
        self.ui.table.setItem(0,4,destinoy_widget)
        self.ui.table.setItem(0,5,red_widget)
        self.ui.table.setItem(0,6,green_widget)
        self.ui.table.setItem(0,7,blue_widget)
        self.ui.table.setItem(0,8,distancia_widget)

        encontrado = True

        return

    if not encontrado:
        QMessageBox.warning(self, 'Atención', f'La particula con ID "{id}"
no fue encontrado')

    @Slot()
    def mostrar_tabla(self):
        self.ui.table.setColumnCount(9)
        headers = ["ID", "Origen X", "Origen Y", "Destino X", "Destino
Y", "Red", "Green", "Blue", "Distancia"]
        self.ui.table.setHorizontalHeaderLabels(headers)

        self.ui.table.setRowCount(len(self.administrador))

        row = 0
        for particula in self.administrador:

```

```

        id_widget = QTableWidgetItem(str(particula.id))
        origenx_widget = QTableWidgetItem(str(particula.origen_x))
        origeny_widget = QTableWidgetItem(str(particula.origen_y))
        destinox_widget = QTableWidgetItem(str(particula.destino_x))
        destinoy_widget = QTableWidgetItem(str(particula.destino_y))
        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.table.setItem(row,0,id_widget)
        self.ui.table.setItem(row,1,origenx_widget)
        self.ui.table.setItem(row,2,origeny_widget)
        self.ui.table.setItem(row,3,destinox_widget)
        self.ui.table.setItem(row,4,destinoy_widget)
        self.ui.table.setItem(row,5,red_widget)
        self.ui.table.setItem(row,6,green_widget)
        self.ui.table.setItem(row,7,blue_widget)
        self.ui.table.setItem(row,8,distancia_widget)

        row += 1

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

    @Slot()
    def action_guardar_archivo(self):
        ubicacion = QFileDialog.getSaveFileName(self, 'Guardar
Archivo', '.', 'JSON (*.json)')[0]
        if self.administrador.guardar(ubicacion):
            QMessageBox.information(self, "Exito", "Se creó el archivo con
exito en " + ubicacion)
        else:
            QMessageBox.information(self, "Error", "No se pudo crear el
archivo en " + ubicacion)

```

```

@Slot()
def ver(self):
    self.ui.Print.clear()
    self.ui.Print.insertPlainText(str(self.administrador))

@Slot()
def agregar_final(self):
    ID = self.ui.ID_spinBox.value()
    OrigenX = self.ui.OrigenX_spinBox.value()
    OrigenY = self.ui.OrigenY_spinBox.value()
    DestinoX = self.ui.DestinoX_spinBox.value()
    DestinoY = self.ui.DestinoY_spinBox.value()
    Red = self.ui.Red_spinBox.value()
    Green = self.ui.Green_spinBox.value()
    Blue = self.ui.Blue_spinBox.value()

    particula1 =
Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY,Red,Green,Blue)
    self.administrador.agregar_final(particula1)

@Slot()
def agregar_inicio(self):
    ID = self.ui.ID_spinBox.value()
    OrigenX = self.ui.OrigenX_spinBox.value()
    OrigenY = self.ui.OrigenY_spinBox.value()
    DestinoX = self.ui.DestinoX_spinBox.value()
    DestinoY = self.ui.DestinoY_spinBox.value()
    Red = self.ui.Red_spinBox.value()
    Green = self.ui.Green_spinBox.value()
    Blue = self.ui.Blue_spinBox.value()

    particula1 =
Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY,Red,Green,Blue)
    self.administrador.agregar_inicio(particula1)

```

Codigo "prueba.py":

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

app = QApplication()

window = MainWindow()

window.show()

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

Codigo "ui_mainwindow.py":

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

#####
####
## Form generated from reading UI file 'mainwindow2.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(553, 426)
        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_3 = QGridLayout(self.centralwidget)
```

```
self.gridLayout_3.setObjectName(u"gridLayout_3")
self.tabWidget = QTabWidget(self.centralwidget)
self.tabWidget.setObjectName(u"tabWidget")
self.tab = QWidget()
self.tab.setObjectName(u"tab")
self.gridLayout_2 = QGridLayout(self.tab)
self.gridLayout_2.setObjectName(u"gridLayout_2")
self.groupBox = QGroupBox(self.tab)
self.groupBox.setObjectName(u"groupBox")
self.gridLayout = QGridLayout(self.groupBox)
self.gridLayout.setObjectName(u"gridLayout")
self.label_5 = QLabel(self.groupBox)
self.label_5.setObjectName(u"label_5")

self.gridLayout.addWidget(self.label_5, 5, 0, 1, 1)

self.Red_spinBox = QSpinBox(self.groupBox)
self.Red_spinBox.setObjectName(u"Red_spinBox")

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

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

self.gridLayout.addWidget(self.label_2, 6, 0, 1, 1)

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

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

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

self.gridLayout.addWidget(self.label_4, 4, 0, 1, 1)

self.label0 = QLabel(self.groupBox)
self.label0.setObjectName(u"label0")

self.gridLayout.addWidget(self.label0, 1, 0, 1, 1)

self.Origeny_spinBox = QSpinBox(self.groupBox)
self.Origeny_spinBox.setObjectName(u"Origeny_spinBox")
self.Origeny_spinBox.setMaximum(999)
```

```
self.gridLayout.addWidget(self.OrigemY_spinBox, 3, 1, 1, 1)

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

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

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

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

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

self.gridLayout.addWidget(self.label_8, 8, 0, 1, 1)

self.ID_spinBox = QSpinBox(self.groupBox)
self.ID_spinBox.setObjectName(u"ID_spinBox")
self.ID_spinBox.setMaximum(999)

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

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

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

self.OrigemX_spinBox = QSpinBox(self.groupBox)
self.OrigemX_spinBox.setObjectName(u"OrigemX_spinBox")
self.OrigemX_spinBox.setMaximum(999)

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

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

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

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

self.gridLayout.addWidget(self.label_3, 3, 0, 1, 1)
```



```
self.Green_spinBox = QSpinBox(self.groupBox)
self.Green_spinBox.setObjectName(u"Green_spinBox")

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

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

self.gridLayout.addWidget(self.Agregar_Inicio, 9, 0, 1, 2)

self.Blue_spinBox = QSpinBox(self.groupBox)
self.Blue_spinBox.setObjectName(u"Blue_spinBox")

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

self.label_7 = QLabel(self.groupBox)
self.label_7.setObjectName(u"label_7")

self.gridLayout.addWidget(self.label_7, 7, 0, 1, 1)

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

self.Print = QPlainTextEdit(self.tab)
self.Print.setObjectName(u"Print")

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

self.tabWidget.addTab(self.tab, "")
self.tab_2 = QWidget()
self.tab_2.setObjectName(u"tab_2")
self.gridLayout_4 = QGridLayout(self.tab_2)
self.gridLayout_4.setObjectName(u"gridLayout_4")
self.search_line = QLineEdit(self.tab_2)
self.search_line.setObjectName(u"search_line")

self.gridLayout_4.addWidget(self.search_line, 1, 0, 1, 1)

self.search_button = QPushButton(self.tab_2)
self.search_button.setObjectName(u"search_button")

self.gridLayout_4.addWidget(self.search_button, 1, 1, 1, 1)

self.view_button = QPushButton(self.tab_2)
```

```

        self.view_button.setObjectName(u"view_button")

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

        self.table = QTableWidgetItem(self.tab_2)
        self.table.setObjectName(u"table")

        self.gridLayout_4.addWidget(self.table, 0, 0, 1, 3)

        self.tabWidget.addTab(self.tab_2, "")

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

MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QMenuBar(MainWindow)
self.menubar.setObjectName(u"menubar")
self.menubar.setGeometry(QRect(0, 0, 553, 21))
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)

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("MainWindo
w", u"Ctrl+D", None))
#endif // QT_CONFIG(shortcut)
        self.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Particulas", None))
        self.label_5.setText(QCoreApplication.translate("MainWindow",
u"Destino Y", None))
        self.label_2.setText(QCoreApplication.translate("MainWindow",
u"Red", None))
        self.label_4.setText(QCoreApplication.translate("MainWindow",
u"Destino X", None))
        self.label0.setText(QCoreApplication.translate("MainWindow",
u"Origen X", None))
        self.Agregar_final.setText(QCoreApplication.translate("MainWindow",
u"Agregar al final", None))
        self.label_8.setText(QCoreApplication.translate("MainWindow",
u"Blue", None))
        self.label.setText(QCoreApplication.translate("MainWindow", u"ID:",
None))
        self.Mostrar.setText(QCoreApplication.translate("MainWindow",
u"Mostrar", None))
        self.label_3.setText(QCoreApplication.translate("MainWindow",
u"Origen Y", None))
        self.Agregar_Inicio.setText(QCoreApplication.translate("MainWindow",
u"Agregar al inicio", None))
        self.label_7.setText(QCoreApplication.translate("MainWindow",
u"Green", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab),
QCoreApplication.translate("MainWindow", u"Agregar", None))
        self.search_line.setPlaceholderText(QCoreApplication.translate("Main
Window", u"ID de la particula", None))
        self.search_button.setText(QCoreApplication.translate("MainWindow",
u"Buscar", None))
        self.view_button.setText(QCoreApplication.translate("MainWindow",
u"Mostrar", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2),
QCoreApplication.translate("MainWindow", u"Tabla", None))
        self.menuArchivo.setTitle(QCoreApplication.translate("MainWindow",
u"Archivo", None))
        # retranslateUi

```