

Actividad 07

(QFileDialog)

Alcaraz Valdivia Marcos Fernando

Seminario de Solución de Problemas de Algoritmia

Lineamientos de evaluación

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

AAAAAAAAAAAA

Satisfacción 10/10

Desarrollo

Captura de pantalla previo a generar respaldo

MainWindow

Archivo

Particulas

Origen X (0-500) 11

Origen Y (0-500) 22

Destino X (0-500) 55

Destino Y (0-500) 66

Velocidad (KM/h) 770

RED (0-255) 8

GREEN (0-255) 87

BLUE (0-255) 45

Insertar al Inicio Insertar al Final

Mostrar

Particula

ID : 2
Origen X : 98
Origen Y : 87
Destino X: 65
Destino Y: 54
Velocidad: 32
Red : 123
Green : 12
Blue : 123
Distancia: 46.66904755831214

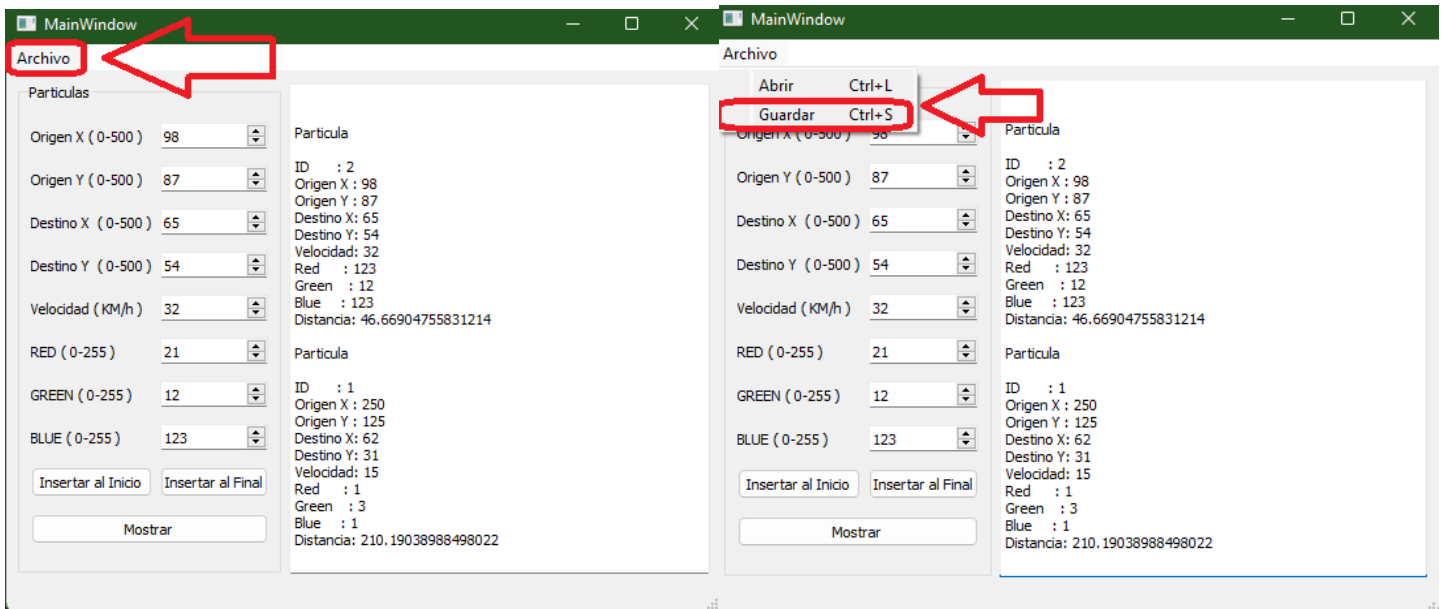
Particula

ID : 1
Origen X : 250
Origen Y : 125
Destino X: 62
Destino Y: 31
Velocidad: 15
Red : 1
Green : 3
Blue : 1
Distancia: 210.19038988498022

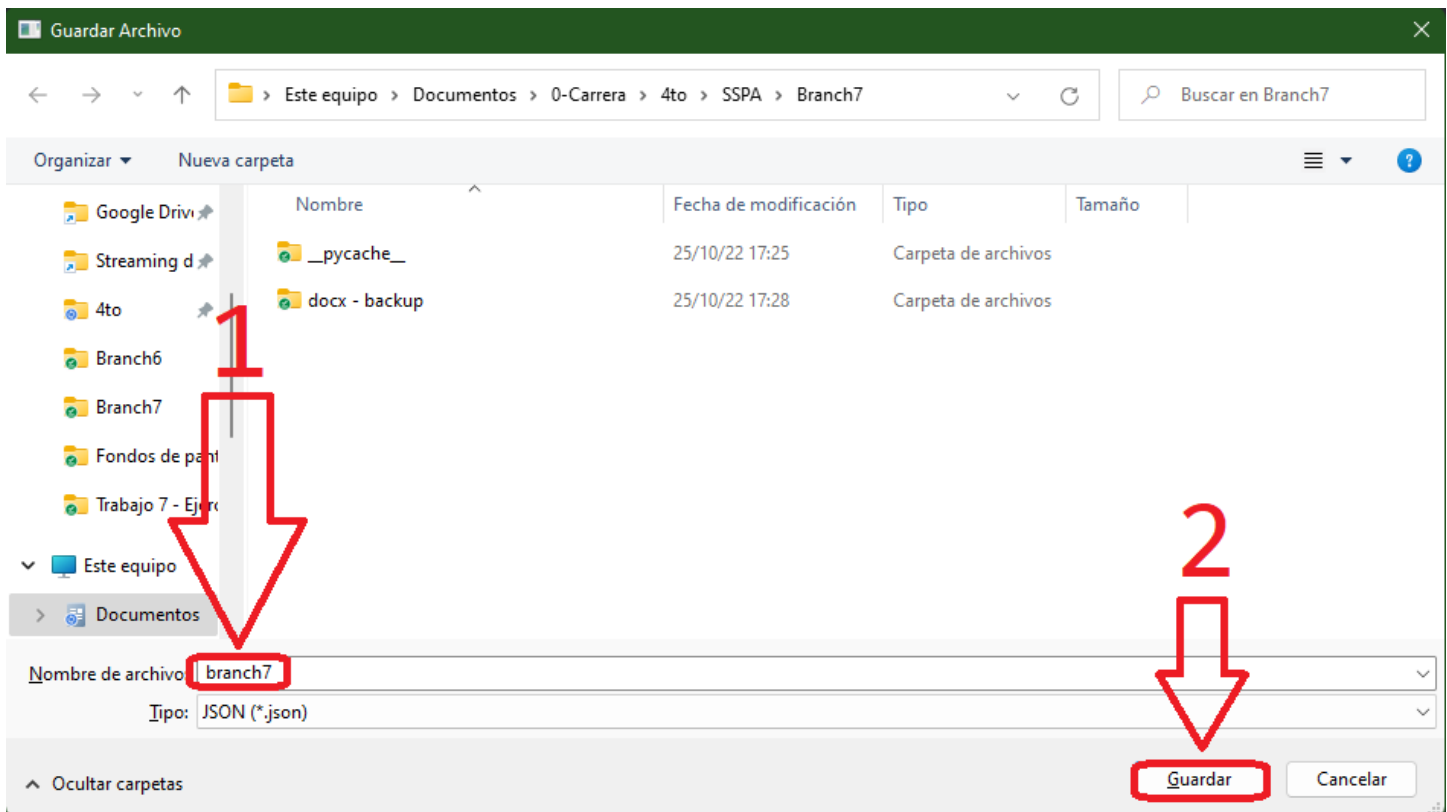
Particula

ID : 3
Origen X : 11
Origen Y : 22
Destino X: 55
Destino Y: 66
Velocidad: 770
Red : 45
Green : 87
Blue : 45
Distancia: 62.22539674441618

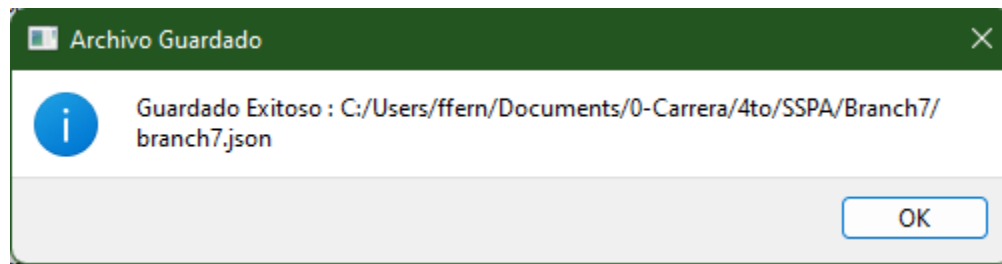
Pasos para realizar en la interfaz para realizar un respaldo



Se selecciona una ruta y nombre del archivo



Recibe el mensaje de respuesta al guardar el archivo



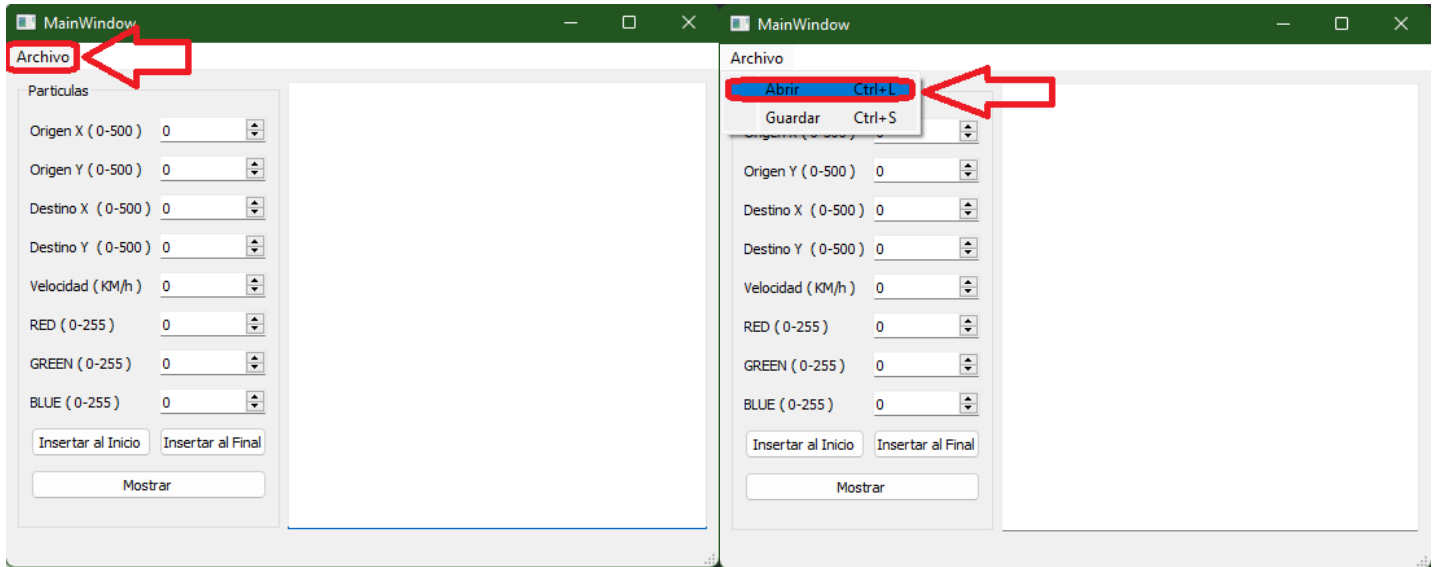
Contenido del archivo JSON

A screenshot of a code editor window showing the content of a JSON file named 'branch7.json'. The editor has a menu bar (File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window) and a toolbar. The file path in the title bar is 'C:\Users\ffern\Documents\0-Carrera\4to\SSPA\Branch7\bra...'. The JSON content is as follows:

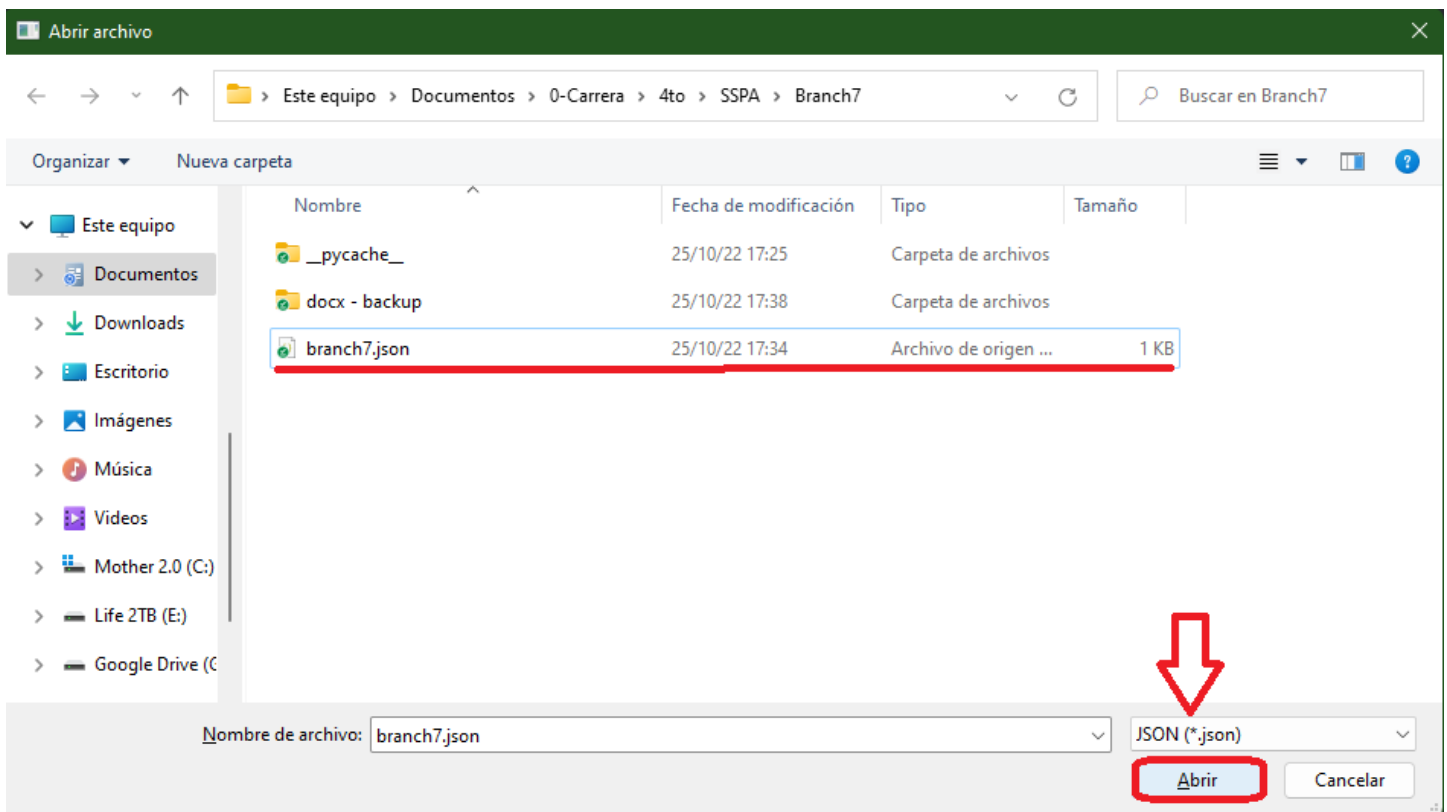
```
1  [
2    {
3      "id": 2,
4      "origen_x": 98,
5      "origen_y": 87,
6      "destino_x": 65,
7      "destino_y": 54,
8      "velocidad": 32,
9      "red": 21,
10     "green": 12,
11     "blue": 123
12   },
13   {
14     "id": 1,
15     "origen_x": 250,
16     "origen_y": 125,
17     "destino_x": 62,
18     "destino_y": 31,
19     "velocidad": 15,
20     "red": 7,
21     "green": 3,
22     "blue": 1
23   },
24   {
25     "id": 3,
26     "origen_x": 11,
27     "origen_y": 22,
28     "destino_x": 55,
29     "destino_y": 66,
30     "velocidad": 770,
31     "red": 8,
32     "green": 87,
33     "blue": 45
34   }
35 ]
```

The status bar at the bottom shows 'Ln: 22 Col: 14 Pos: 473', 'Windows (CR LF)', 'UTF-8', and 'INS'.

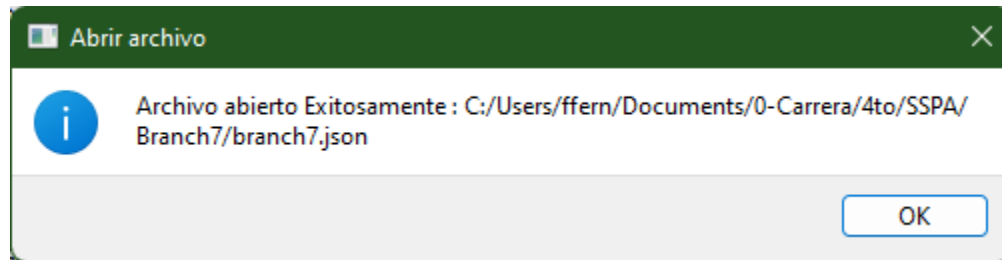
Pasos para abrir archivo



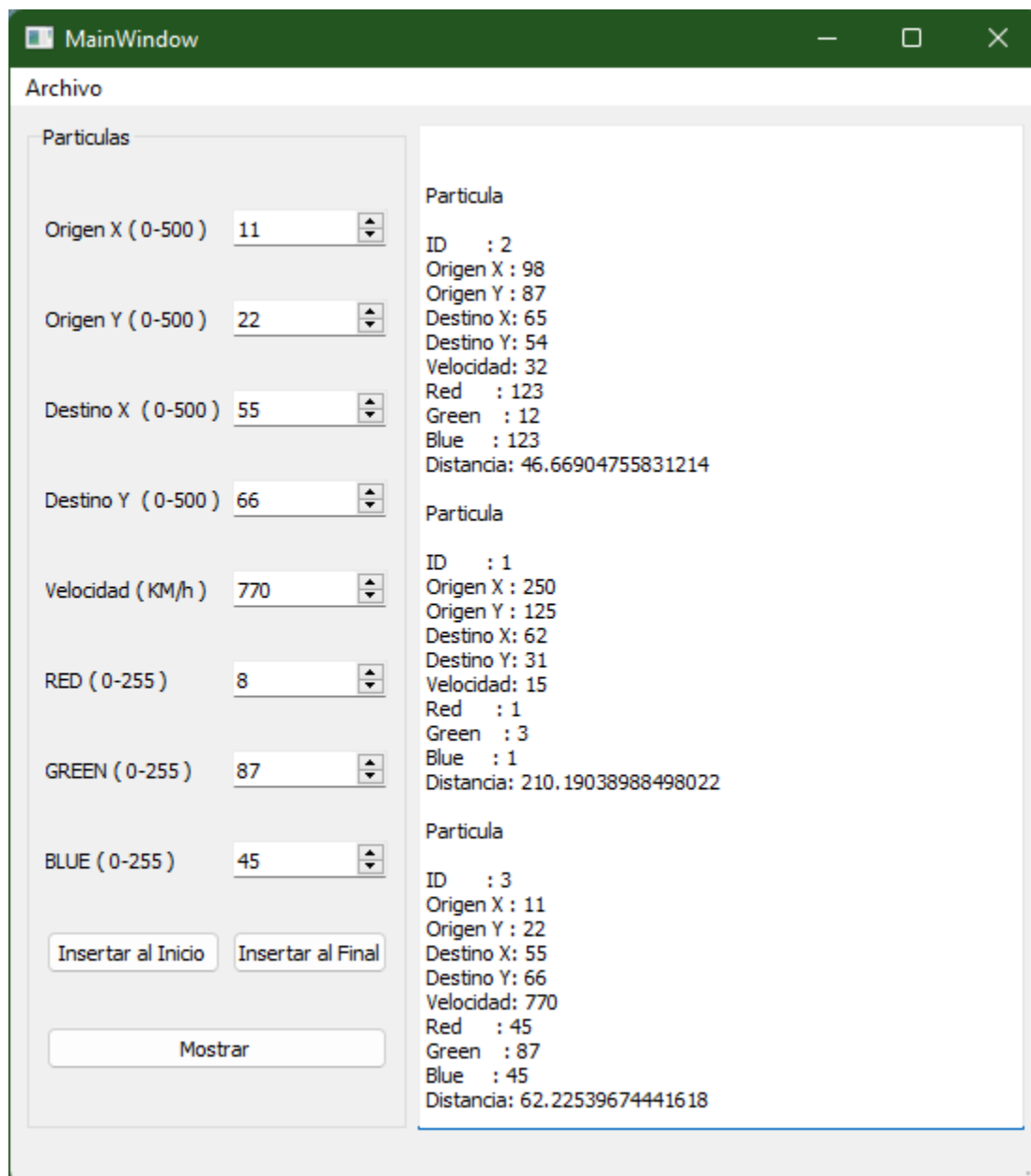
Seleccionar archivo JSON



Recibir mensaje al abrir el archivo



Captura de pantalla después de realizar el respaldo



Conclusiones

Usar Python es divertido.

Todo lo que hacía en C++ era gran numero de veces más complicado que en Python. Dios, leer archivos JSON con solo una librería que Python ya tiene por defecto, leer y escribir archivos de manera tan simple, ***recorrer listas completas con solo un ciclo for***, es increíble.

Estoy consiente de que he sido capaz de realizar todo este proyecto hasta ahora gracias a Michel Davalos Boites, pero debo decir que es realmente más fácil resolver un problema en Python que en C++.

Referencias

PySide2 - QFileDialog (Qt for Python)(IV)

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

Código

Se mostrarán solo los archivos modificados *particula.py*

```
# Archivo particula.py
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(origen_x, origen_y, destino_x, destino_y)

    def __str__(self):
        return ('\n\nParticula\n' +
            '\nID      : ' + str(self.__id) +
            '\nOrigen X : ' + str(self.__origen_x) +
            '\nOrigen Y : ' + str(self.__origen_y) +
            '\nDestino X: ' + str(self.__destino_x) +
            '\nDestino Y: ' + str(self.__destino_y) +
            '\nVelocidad: ' + str(self.__velocidad) +
            '\nRed      : ' + str(self.__red) +
            '\nGreen    : ' + str(self.__green) +
            '\nBlue     : ' + str(self.__blue) +
            '\nDistancia: ' + str(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,
        }
```

manager.py

```
# Archivo manager.py
from particula import Particula
import json
```

```
class Manager:
```

```

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

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

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

def imprimir(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

```

mainwindow.py

```

from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from manager import Manager
from particula import Particula

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.manager = Manager()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.id = 0

        # Cuando el botón pushbutton es presionado, ejecuta la función click_agregar
        # self.ui.mostrar.clicked.connect(self.click_mostrar)
        self.ui.insertar_inicio.clicked.connect(self.click_insertar_inicio)
        self.ui.insertar_final.clicked.connect(self.click_insertar_final)

```

```

self.ui.mostrar.clicked.connect(self.click_mostrar)

self.ui.accionAbrir.triggered.connect(self.accionAbrirArchivo)
self.ui.accionGuardar.triggered.connect(self.accionGuardarArchivo)

@Slot()
def accionAbrirArchivo(self):
    ubicacion = QFileDialog.getOpenFileName(
        self,
        "Abrir archivo",
        ".",
        "JSON (*.json)"
    )[0]
    if self.manager.abrir(ubicacion):
        self.click_mostrar()
        QMessageBox.information(self, "Abrir archivo", "Archivo abierto Exitosamente : "
+ ubicacion)
    else:
        QMessageBox.critical(self, "Error", "No se puede abrir el archivo : " +
ubicacion)

@Slot()
def accionGuardarArchivo(self):
    ubicacion = QFileDialog.getSaveFileName(
        self,
        "Guardar Archivo",
        ".",
        "JSON (*.json)"
    )[0]
    if self.manager.guardar(ubicacion):
        QMessageBox.information(self, "Archivo Guardado", "Guardado Exitoso : " +
ubicacion)
    else:
        QMessageBox.critical(self, "Error", "Archivo no Guardado : " + ubicacion)

@Slot()
def click_insertar_inicio(self):
    self.id += 1
    aux = Particula(self.id, self.ui.ox.value(), self.ui.oy.value(), self.ui.dx.value(),
self.ui.dy.value(
    ), self.ui.velocidad.value(), self.ui.red.value(), self.ui.green.value(),
self.ui.blue.value())
    self.manager.agregarInicio(aux)
    self.click_mostrar()

@Slot()
def click_insertar_final(self):
    self.id += 1
    aux = Particula(self.id, self.ui.ox.value(), self.ui.oy.value(), self.ui.dx.value(),
self.ui.dy.value(
    ), self.ui.velocidad.value(), self.ui.red.value(), self.ui.green.value(),
self.ui.blue.value())
    self.manager.agregarFinal(aux)
    self.click_mostrar()

@Slot()
def click_mostrar(self):

```

```
self.ui.lista_particulas.clear()
self.ui.lista_particulas.insertPlainText(str(self.manager))
```

ui.mainwindow.py

```
# -*- 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(549, 405)
        self.accionGuardar = QAction(MainWindow)
        self.accionGuardar.setObjectName(u"accionGuardar")
        self.accionAbrir = QAction(MainWindow)
        self.accionAbrir.setObjectName(u"accionAbrir")
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.gridLayout_2 = QGridLayout(self.centralwidget)
        self.gridLayout_2.setObjectName(u"gridLayout_2")
        self.lista_particulas = QLineEdit(self.centralwidget)
        self.lista_particulas.setObjectName(u"lista_particulas")

        self.gridLayout_2.addWidget(self.lista_particulas, 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.green = QSpinBox(self.groupBox)
        self.green.setObjectName(u"green")
        self.green.setMaximum(255)

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

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

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

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

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

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

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

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

self.gridLayout.addWidget(self.label_5, 6, 0, 1, 1)

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

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

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

self.gridLayout.addWidget(self.label_3, 4, 0, 1, 1)

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

self.gridLayout.addWidget(self.label_2, 3, 0, 1, 1)

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

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

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

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

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

self.gridLayout.addWidget(self.label_4, 5, 0, 1, 1)

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

self.gridLayout.addWidget(self.label_6, 7, 0, 1, 1)

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

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

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

self.gridLayout.addWidget(self.label, 2, 0, 1, 1)
```

```

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

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

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

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

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

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

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

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

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

self.gridLayout.addWidget(self.ox, 0, 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, 549, 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.accionAbrir)
self.menuArchivo.addAction(self.accionGuardar)

self.retranslateUi(MainWindow)

QMetaObject.connectSlotsByName(MainWindow)
# setupUi

def retranslateUi(self, MainWindow):
    MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow", u"MainWindow",
None))
    self.accionGuardar.setText(QCoreApplication.translate("MainWindow", u"Guardar",
None))
    #if QT_CONFIG(shortcut)

```

```

        self.accionGuardar.setShortcut(QCoreApplication.translate("MainWindow", u"Ctrl+S",
None))
#endif // QT_CONFIG(shortcut)
        self.accionAbrir.setText(QCoreApplication.translate("MainWindow", u"Abrir", None))
#if QT_CONFIG(shortcut)
        self.accionAbrir.setShortcut(QCoreApplication.translate("MainWindow", u"Ctrl+L",
None))
#endif // QT_CONFIG(shortcut)
        self.groupBox.setTitle(QCoreApplication.translate("MainWindow", u"Particulas", None))
        self.label_7.setText(QCoreApplication.translate("MainWindow", u"Origen Y ( 0-500 )",
None))
        self.label_5.setText(QCoreApplication.translate("MainWindow", u"GREEN ( 0-255 )",
None))
        self.insertar_final.setText(QCoreApplication.translate("MainWindow", u"Insertar al
Final", None))
        self.label_3.setText(QCoreApplication.translate("MainWindow", u"Velocidad ( KM/h )",
None))
        self.label_2.setText(QCoreApplication.translate("MainWindow", u"Destino Y ( 0-500
)", None))
        self.mostrar.setText(QCoreApplication.translate("MainWindow", u"Mostrar", None))
        self.label_4.setText(QCoreApplication.translate("MainWindow", u"RED ( 0-255 )",
None))
        self.label_6.setText(QCoreApplication.translate("MainWindow", u"BLUE ( 0-255 )",
None))
        self.label.setText(QCoreApplication.translate("MainWindow", u"Destino X ( 0-500 )",
None))
        self.insertar_inicio.setText(QCoreApplication.translate("MainWindow", u"Insertar al
Inicio", None))
        self.label_8.setText(QCoreApplication.translate("MainWindow", u"Origen X ( 0-500 )",
None))
        self.menuArchivo.setTitle(QCoreApplication.translate("MainWindow", u"Archivo", None))
# retranslateUi

```