

Actividad 07 – Clases // QFileDialog

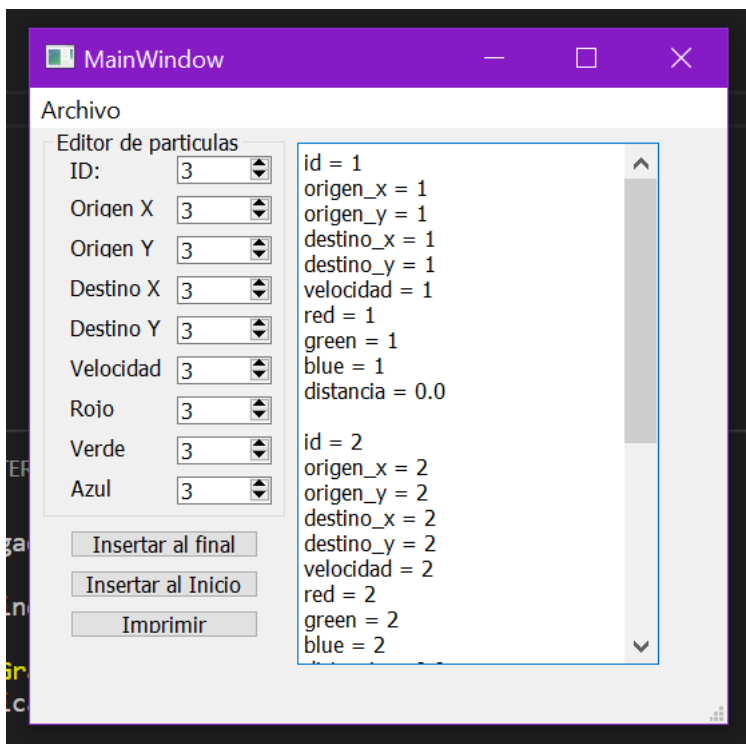
Michel Dávalos Boites // Alan Gabriel Bautista Chavira

Programación // Seminario de solución de problemas de algoritmia

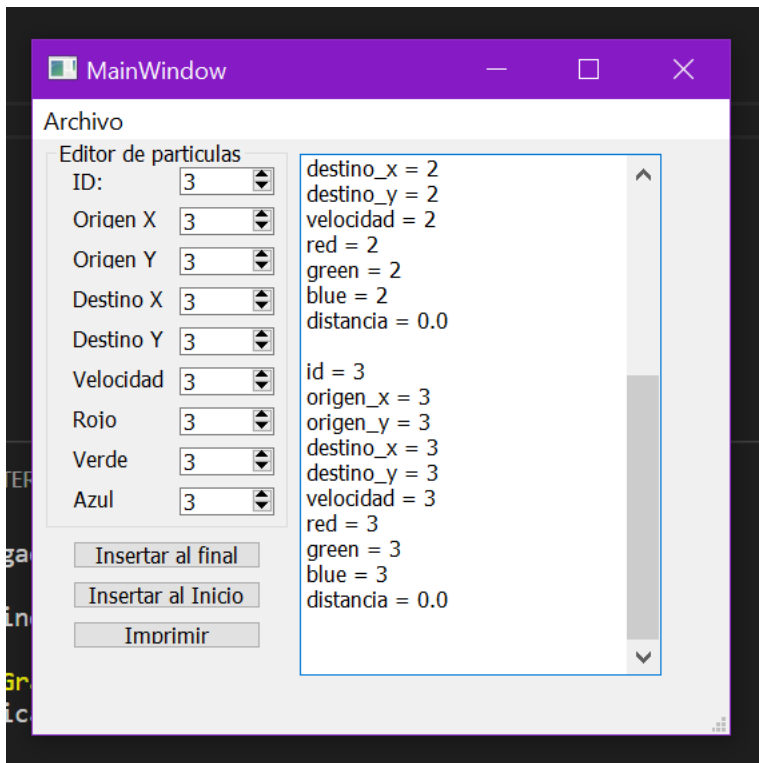
Lineamientos de evaluación:

- El reporte esta en formato Docs o PDF
- El reporte sigue las pautas del formato de actividades
- El reporte tiene desarrolladas todas las pautas del formato de actividades
- Se muestran capturas de las partículas con el método mostrar previo a crear el respaldo
- Se muestran capturas de los pasos para crear el respaldo
- Se muestra el contenido del archivo .json
- Se muestran capturas de los pasos para abrir el archivo de respaldo
- Se muestra la captura de pantalla de las partículas con el método mostrar después de abrir el respaldo

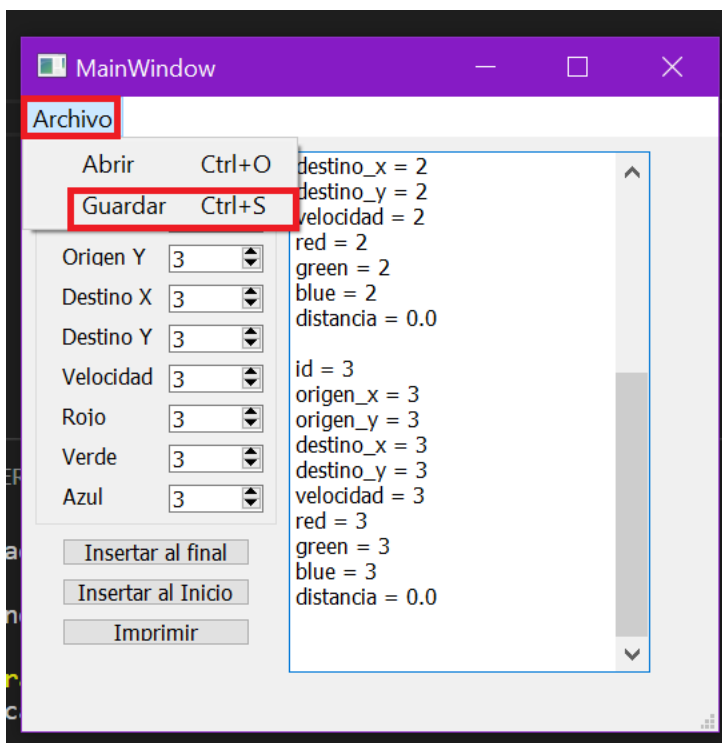
Captura de las inserciones de las partículas antes de crear el respaldo



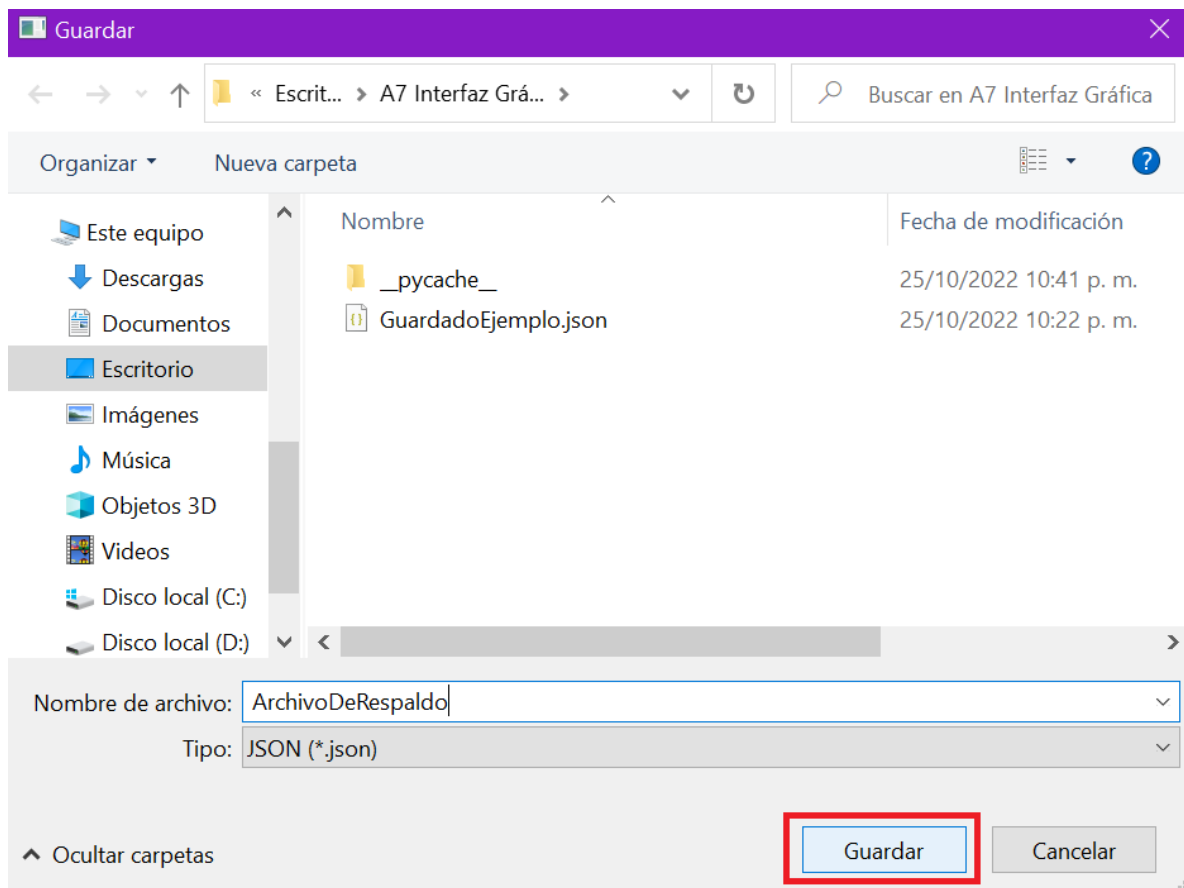
Se insertaron 3 partículas, con todos los campos en un numero del 1 al 3 para cada partícula



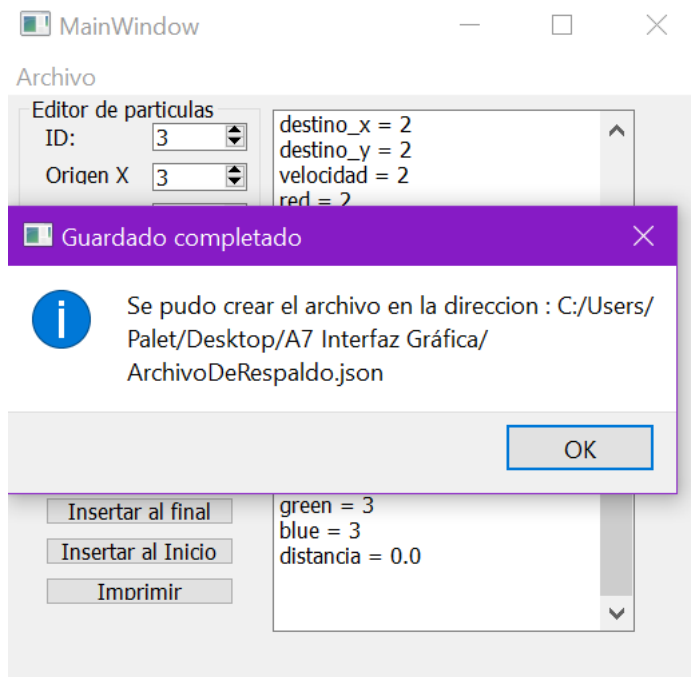
Captura de los pasos para crear un respaldo



(Puedes acceder a la función de guardar ya sea dando click en el menú o con el atajo Ctrl+S)



(Hay que ponerle nombre al archivo a crear y darle en guardar)



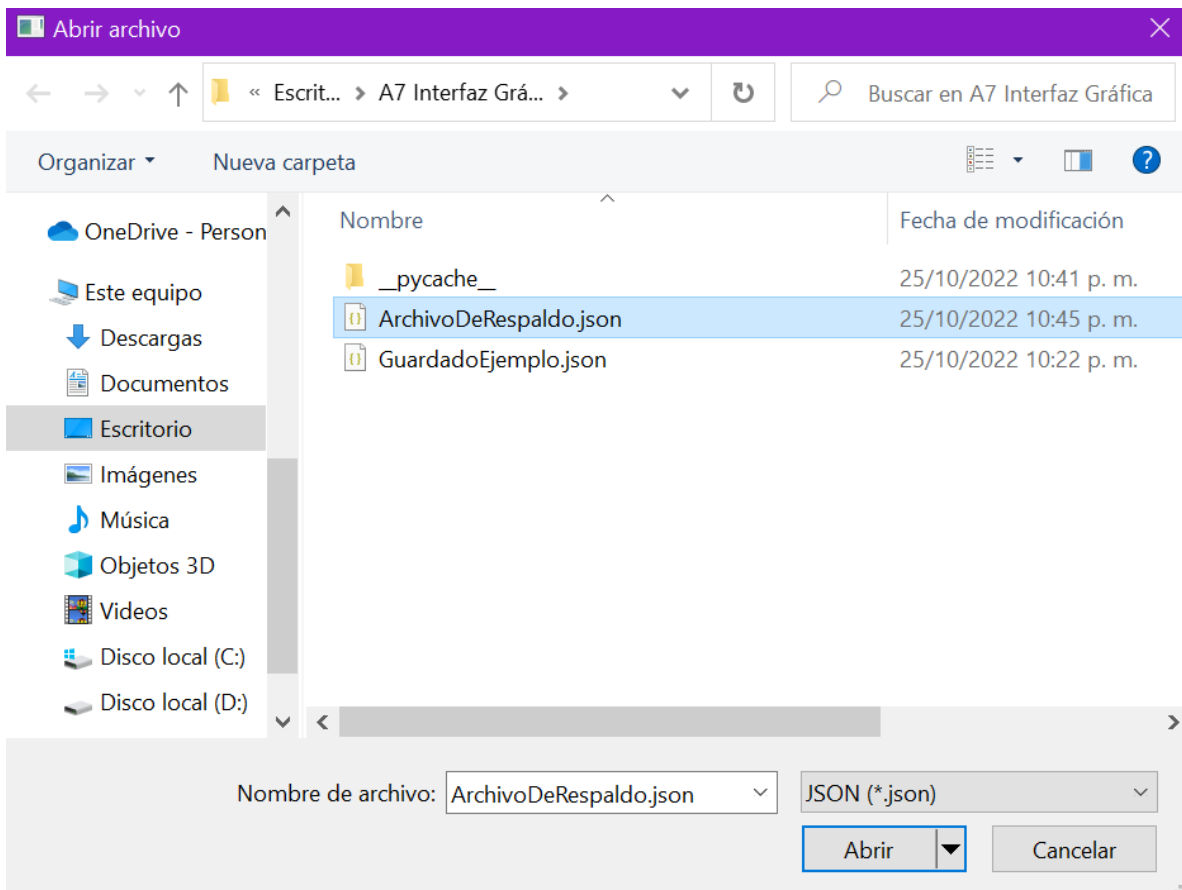
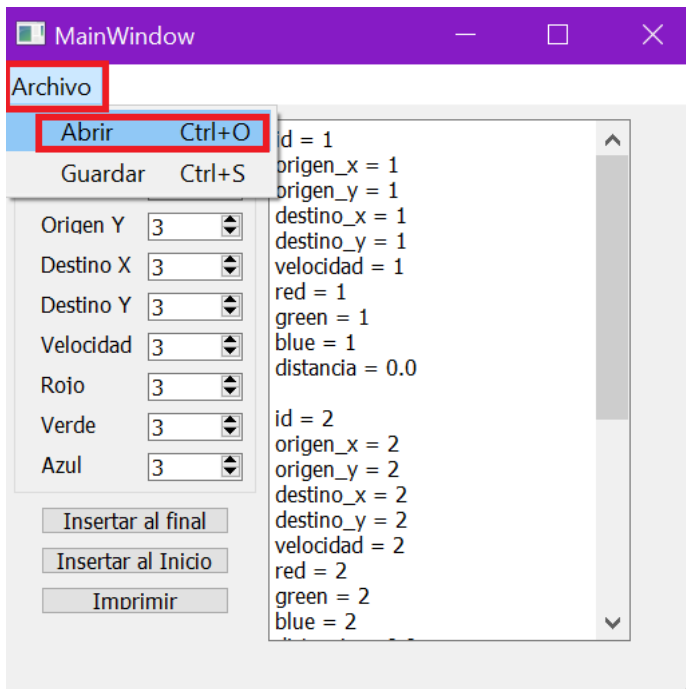
(Esto significa que se guardó correctamente)

Contenido del archivo json

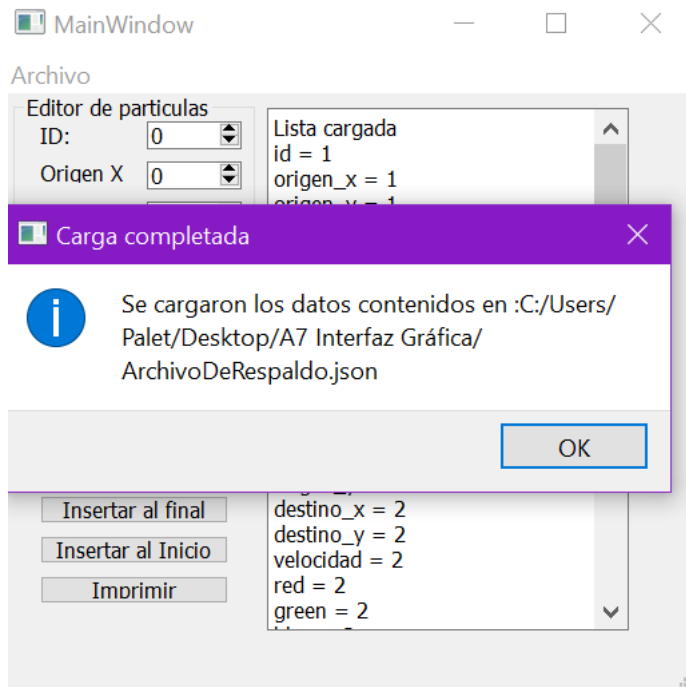
```
ArchivoDeRespaldo.json: Bloc de notas
Archivo Edición Formato Ver Ayuda
[
  {
    "id": 1,
    "origen_x": 1,
    "origen_y": 1,
    "destino_x": 1,
    "destino_y": 1,
    "velocidad": 1,
    "red": 1,
    "green": 1,
    "blue": 1
  },
  {
    "id": 2,
    "origen_x": 2,
    "origen_y": 2,
    "destino_x": 2,
    "destino_y": 2,
    "velocidad": 2,
    "red": 2,
    "green": 2,
    "blue": 2
  },
  {
    "id": 3,
    "origen_x": 3,
    "origen_y": 3,
    "destino_x": 3,
    "destino_y": 3,
    "velocidad": 3,
    "red": 3,
    "green": 3,
    "blue": 3
  }
]
```

(Exactamente los 3 registros que almacenamos)

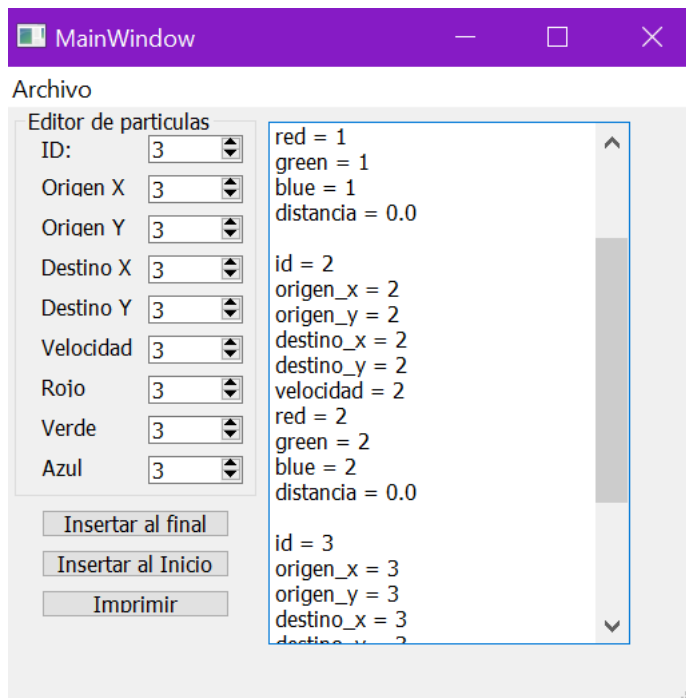
Capturas de los pasos para abrir un respaldo



(Eliges el archivo y le das a Abrir)



Captura de la lista que se cargó del archivo de respaldo



(Como se puede apreciar, estan las partículas con sus respectivos campos en 1, 2 y 3)

En conclusión, por lo que noté al hacer la actividad, guardar información en archivos y prácticamente cualquier implementación como una lista, es considerablemente más fácil en Python que en otros lenguajes. Además, aquí Qt hace prácticamente todo el trabajo.

Si tenemos algunos campos de un registro que se calculan con una función, podemos no guardar esos campos en el archivo a costa de que al momento de imprimir los datos se tenga que llamar a esa función, por cada registro que se imprima, tal es el caso del campo distancia en esta actividad. Mejora el tamaño de la información que se tiene que almacenar pero empeora el tiempo de ejecución

Código:

main.py

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

app = QApplication()

window = MainWindow()

window.show()

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

mainwindow.py

```
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox
from ui_mainwindow import Ui_MainWindow
from PySide2.QtCore import Slot
from Particulas import Particula
from Listas import Lista

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.lista = Lista()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.pushButton_2.clicked.connect(self.click_agregar)
        self.ui.pushButton.clicked.connect(self.click_final)
        self.ui.pushButton_3.clicked.connect(self.click_mostrar)
        self.ui.actionAbrir.triggered.connect(self.action_abrir_archivo)

        self.ui.actionGuardar.triggered.connect(self.action_guardar_archivo)
```

```

@Slot()
def action_abrir_archivo(self):

    ubicacion = QFileDialog.getOpenFileName(
        self,
        'Abrir archivo',
        '.',
        'JSON (*.json)'
    )[0]
    if self.lista.abrir_archivo(ubicacion):
        self.ui.plainTextEdit.clear()
        self.ui.plainTextEdit.insertPlainText("Lista cargada\n")
        self.ui.plainTextEdit.insertPlainText(str(self.lista))
        QMessageBox.information(
            self,
            "Carga completada",
            "Se cargaron los datos contenidos en :" + ubicacion
        )
    else:
        QMessageBox.critical(
            self,
            "Error",
            "Error al abrir el archivo" + ubicacion
        )

@Slot()
def action_guardar_archivo(self):
    ubicacion = QFileDialog.getSaveFileName(
        self,
        'Guardar',
        '.',
        'JSON (*.json)'
    )[0]
    print(ubicacion)
    if self.lista.guardar_archivo(ubicacion):
        QMessageBox.information(
            self,
            "Guardado completado",
            "Se pudo crear el archivo en la direccion : " + ubicacion,
        )
    else :
        QMessageBox.critical(
            self,
            "ERROR",
            "NO SE PUDO CREAR EL ARCHIVO" + ubicacion
        )

@Slot()
def click_mostrar(self):
    self.ui.plainTextEdit.clear()
    self.ui.plainTextEdit.insertPlainText(str(self.lista))

```



```

@Slot()
def click_agregar(self):
    id = self.ui.spinBox.value()
    origen_x = self.ui.spinBox_2.value()
    origen_y = self.ui.spinBox_3.value()
    destino_x = self.ui.spinBox_4.value()
    destino_y = self.ui.spinBox_5.value()
    velocidad = self.ui.spinBox_6.value()
    rojo = self.ui.spinBox_7.value()
    verde = self.ui.spinBox_8.value()
    azul = self.ui.spinBox_9.value()

    partícula =
Partícula(id,origen_x,origen_y,destino_x,destino_y,velocidad,rojo,verde,a
zul)

    self.lista.insertar_inicio(partícula)

@Slot()
def click_final(self):
    id = self.ui.spinBox.value()
    origen_x = self.ui.spinBox_2.value()
    origen_y = self.ui.spinBox_3.value()
    destino_x = self.ui.spinBox_4.value()
    destino_y = self.ui.spinBox_5.value()
    velocidad = self.ui.spinBox_6.value()
    rojo = self.ui.spinBox_7.value()
    verde = self.ui.spinBox_8.value()
    azul = self.ui.spinBox_9.value()

    partícula =
Partícula(id,origen_x,origen_y,destino_x,destino_y,velocidad,rojo,verde,a
zul)

    self.lista.insertar_final(partícula)

```

Listas.py

```

from Particulas import Particula
import json

class Lista:
    def __init__(self):
        self.__Lista = []

    def insertar_final(self, Particulas:Particula):
        self.__Lista.append(Particulas)

    def insertar_inicio(self, Particulas:Particula):
        self.__Lista.insert(0, Particulas)

    def imprimir(self):
        for Particulas in self.__Lista:

```

```

        print(Particulas)

    def __str__(self):
        return "".join(str(Particulas) + '\n' for Particulas in
self.__Lista)

    def guardar_archivo (self, ubicacion):
        try:
            with open(ubicacion, "w") as archivoLista:
                lista = [Particulas.to_dict() for Particulas in
self.__Lista]
                json.dump(lista, archivoLista, indent = 11)
            return 1
        except:
            return 0

    def abrir_archivo(self, ubicacion):
        try:
            with open(ubicacion, "r") as archivoLista:
                listaTemporal = json.load(archivoLista)
                self.__Lista = [Particula(**Particulas) for Particulas in
listaTemporal]
            return 1
        except:
            return 0

```

Algoritmos.py

```
import math
```

```

def distanciaEuclidiana(x1, y1, x2, y2):
    diferenciaX = pow((x2 - x1), 2)
    diferenciaY = pow((y2 - y1), 2)
    resultado = math.sqrt(diferenciaX + diferenciaY)
    return (resultado)

```

Particulas.py

```
from Algoritmos import distanciaEuclidiana
```

```

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, distancia =
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 = distanciaEuclidiana(origen_x, origen_y,
destino_x, destino_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'
        )

    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

```

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(522, 476)
        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.plainTextEdit = QPlainTextEdit(self.centralwidget)
        self.plainTextEdit.setObjectName(u"plainTextEdit")

```

```
self.plainTextEdit.setGeometry(QRect(200, 10, 271, 391))
self.pushButton = QPushButton(self.centralwidget)
self.pushButton.setObjectName(u"pushButton")
self.pushButton.setGeometry(QRect(30, 300, 141, 21))
self.pushButton_2 = QPushButton(self.centralwidget)
self.pushButton_2.setObjectName(u"pushButton_2")
self.pushButton_2.setGeometry(QRect(30, 330, 141, 21))
self.pushButton_3 = QPushButton(self.centralwidget)
self.pushButton_3.setObjectName(u"pushButton_3")
self.pushButton_3.setGeometry(QRect(30, 360, 141, 21))
self.groupBox = QGroupBox(self.centralwidget)
self.groupBox.setObjectName(u"groupBox")
self.groupBox.setGeometry(QRect(10, 0, 181, 291))
self.spinBox_9 = QSpinBox(self.groupBox)
self.spinBox_9.setObjectName(u"spinBox_9")
self.spinBox_9.setGeometry(QRect(100, 260, 71, 21))
self.spinBox_4 = QSpinBox(self.groupBox)
self.spinBox_4.setObjectName(u"spinBox_4")
self.spinBox_4.setGeometry(QRect(100, 110, 71, 21))
self.spinBox_5 = QSpinBox(self.groupBox)
self.spinBox_5.setObjectName(u"spinBox_5")
self.spinBox_5.setGeometry(QRect(100, 140, 71, 21))
self.label_2 = QLabel(self.groupBox)
self.label_2.setObjectName(u"label_2")
self.label_2.setGeometry(QRect(20, 50, 71, 16))
self.label_7 = QLabel(self.groupBox)
self.label_7.setObjectName(u"label_7")
self.label_7.setGeometry(QRect(20, 200, 71, 16))
self.spinBox = QSpinBox(self.groupBox)
self.spinBox.setObjectName(u"spinBox")
self.spinBox.setGeometry(QRect(100, 20, 71, 21))
self.spinBox_6 = QSpinBox(self.groupBox)
self.spinBox_6.setObjectName(u"spinBox_6")
self.spinBox_6.setGeometry(QRect(100, 170, 71, 21))
self.label_8 = QLabel(self.groupBox)
self.label_8.setObjectName(u"label_8")
self.label_8.setGeometry(QRect(20, 230, 71, 16))
self.spinBox_7 = QSpinBox(self.groupBox)
self.spinBox_7.setObjectName(u"spinBox_7")
self.spinBox_7.setGeometry(QRect(100, 200, 71, 21))
self.label_9 = QLabel(self.groupBox)
self.label_9.setObjectName(u"label_9")
self.label_9.setGeometry(QRect(20, 260, 71, 16))
self.label_5 = QLabel(self.groupBox)
self.label_5.setObjectName(u"label_5")
self.label_5.setGeometry(QRect(20, 140, 71, 16))
self.spinBox_8 = QSpinBox(self.groupBox)
self.spinBox_8.setObjectName(u"spinBox_8")
self.spinBox_8.setGeometry(QRect(100, 230, 71, 21))
self.label_6 = QLabel(self.groupBox)
self.label_6.setObjectName(u"label_6")
self.label_6.setGeometry(QRect(20, 170, 71, 16))
self.spinBox_3 = QSpinBox(self.groupBox)
```

```

self.spinBox_3.setObjectName(u"spinBox_3")
self.spinBox_3.setGeometry(QRect(100, 80, 71, 21))
self.label_4 = QLabel(self.groupBox)
self.label_4.setObjectName(u"label_4")
self.label_4.setGeometry(QRect(20, 110, 71, 16))
self.spinBox_2 = QSpinBox(self.groupBox)
self.spinBox_2.setObjectName(u"spinBox_2")
self.spinBox_2.setGeometry(QRect(100, 50, 71, 21))
self.label = QLabel(self.groupBox)
self.label.setObjectName(u"label")
self.label.setGeometry(QRect(20, 20, 71, 20))
self.label_3 = QLabel(self.groupBox)
self.label_3.setObjectName(u"label_3")
self.label_3.setGeometry(QRect(20, 80, 71, 16))
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QMenuBar(MainWindow)
self.menubar.setObjectName(u"menubar")
self.menubar.setGeometry(QRect(0, 0, 522, 22))
self.menuArchivo = QMenu(self.menubar)
self.menuArchivo.setObjectName(u"menuArchivo")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QStatusBar(MainWindow)
self.statusbar.setObjectName(u"statusbar")
MainWindow.setStatusBar(self.statusbar)

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

self.retranslateUi(MainWindow)

QMetaObject.connectSlotsByName(MainWindow)
# setupUi

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

self.actionAbrir.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+O", None))
    #endif // QT_CONFIG(shortcut)

self.actionGuardar.setText(QCoreApplication.translate("MainWindow",
u"Guardar", None))
    #if QT_CONFIG(shortcut)

self.actionGuardar.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+S", None))
    #endif // QT_CONFIG(shortcut)

```

```
        self.pushButton.setText(QCoreApplication.translate("MainWindow",
u"Insertar al final", None))

self.pushButton_2.setText(QCoreApplication.translate("MainWindow",
u"Insertar al Inicio", None))

self.pushButton_3.setText(QCoreApplication.translate("MainWindow",
u"Imprimir", None))
        self.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Editor de particulas", None))
        self.label_2.setText(QCoreApplication.translate("MainWindow",
u"Origen X", None))
        self.label_7.setText(QCoreApplication.translate("MainWindow",
u"Rojo", None))
        self.label_8.setText(QCoreApplication.translate("MainWindow",
u"Verde", None))
        self.label_9.setText(QCoreApplication.translate("MainWindow",
u"Azul", None))
        self.label_5.setText(QCoreApplication.translate("MainWindow",
u"Destino Y", None))
        self.label_6.setText(QCoreApplication.translate("MainWindow",
u"Velocidad", None))
        self.label_4.setText(QCoreApplication.translate("MainWindow",
u"Destino X", None))
        self.label.setText(QCoreApplication.translate("MainWindow",
u"ID:", None))
        self.label_3.setText(QCoreApplication.translate("MainWindow",
u"Origen Y", None))

self.menuArchivo.setTitle(QCoreApplication.translate("MainWindow",
u"Archivo", None))
        # retranslateUi
```