Actividad 07 - QFileDialog

JAIRO CAIN SANCHEZ ESTRADA// Luis Angel Elisea Graciano

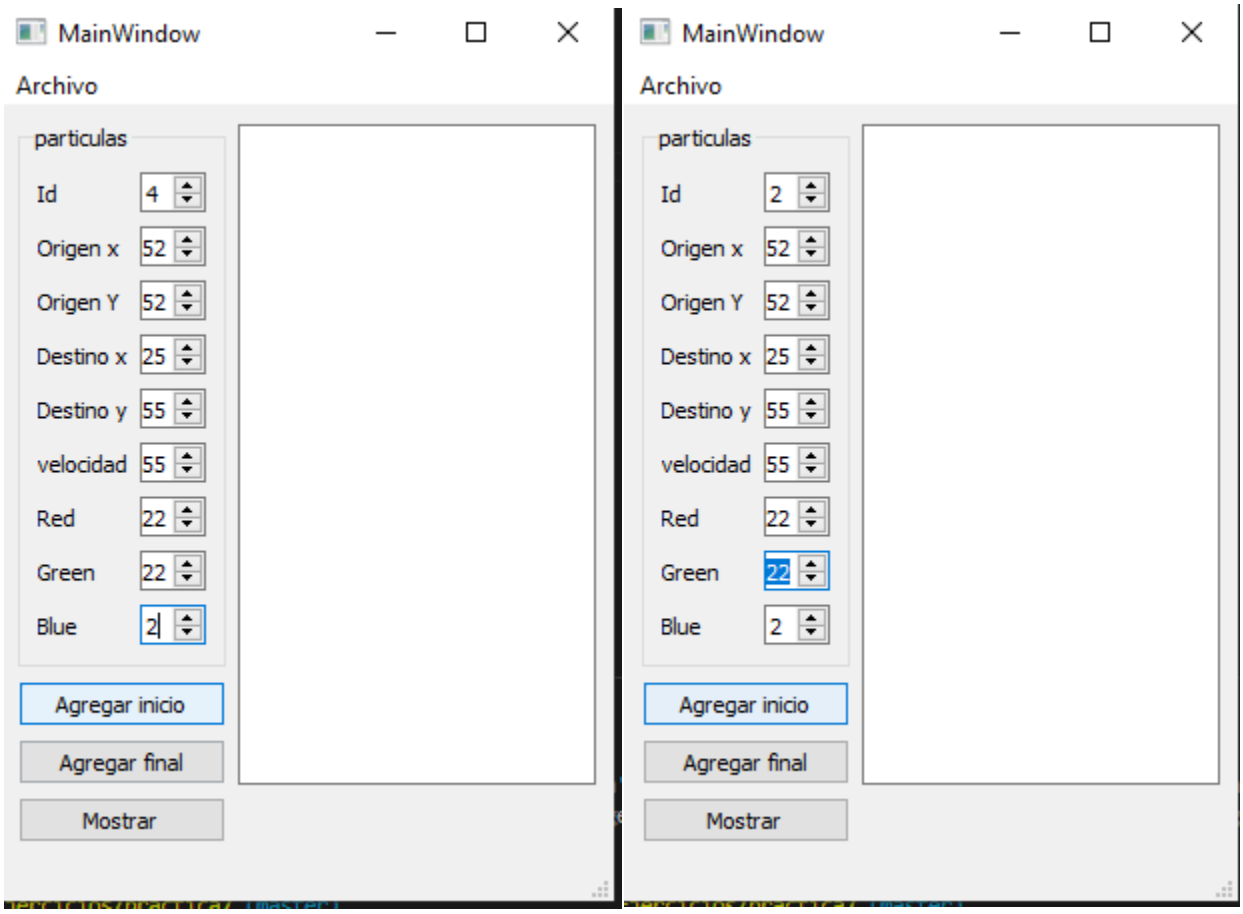
SEMINARIO DE SOLUCION DE PROBLEMAS DE ALGORITMIA

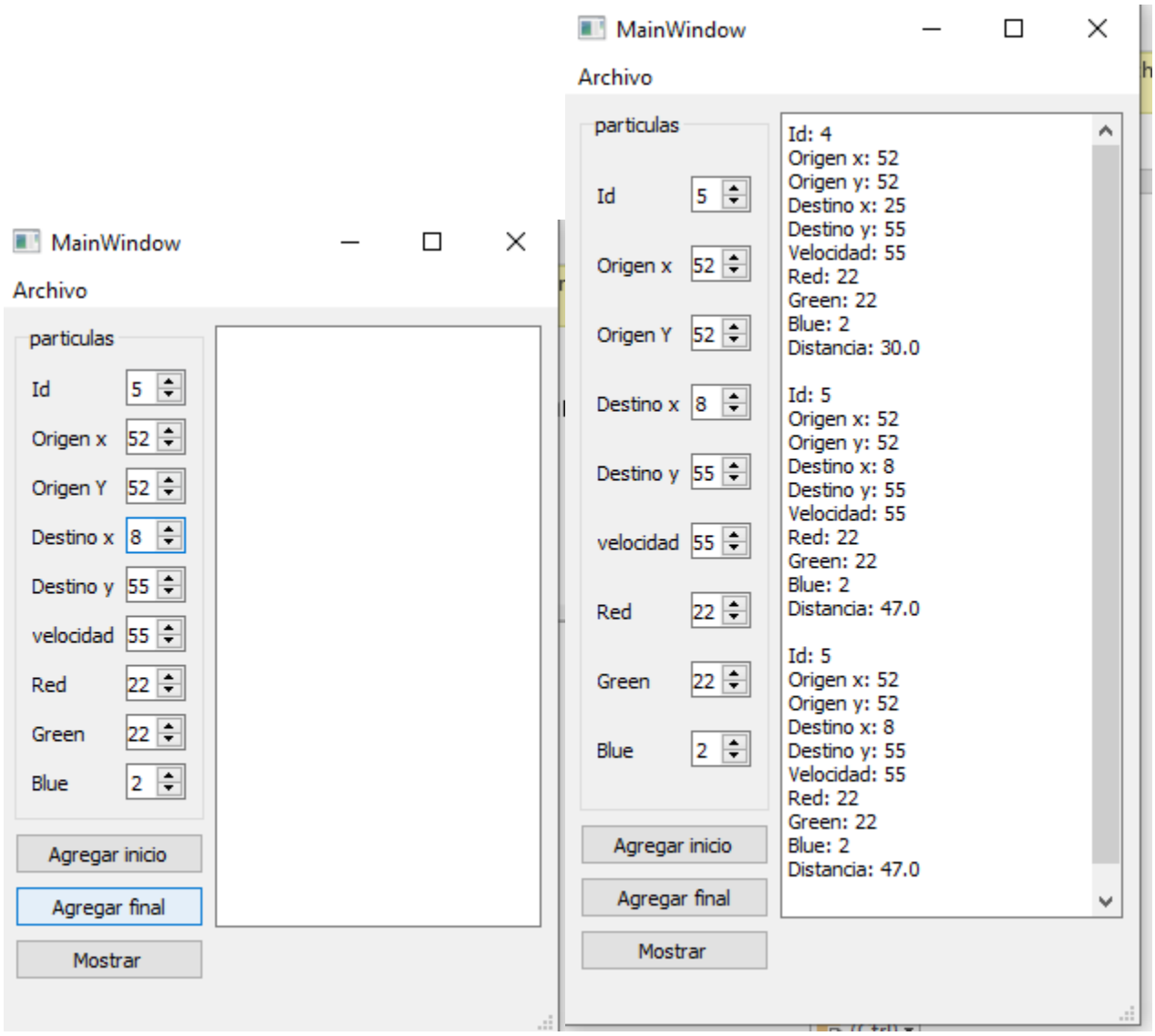
Lineamientos de evaluación

- Se agrego las opciones de guardar y abrir archivo
- Se implemento la lectura y escritura del archivo JSON
- Se modifico la interfaz y agrego el menu

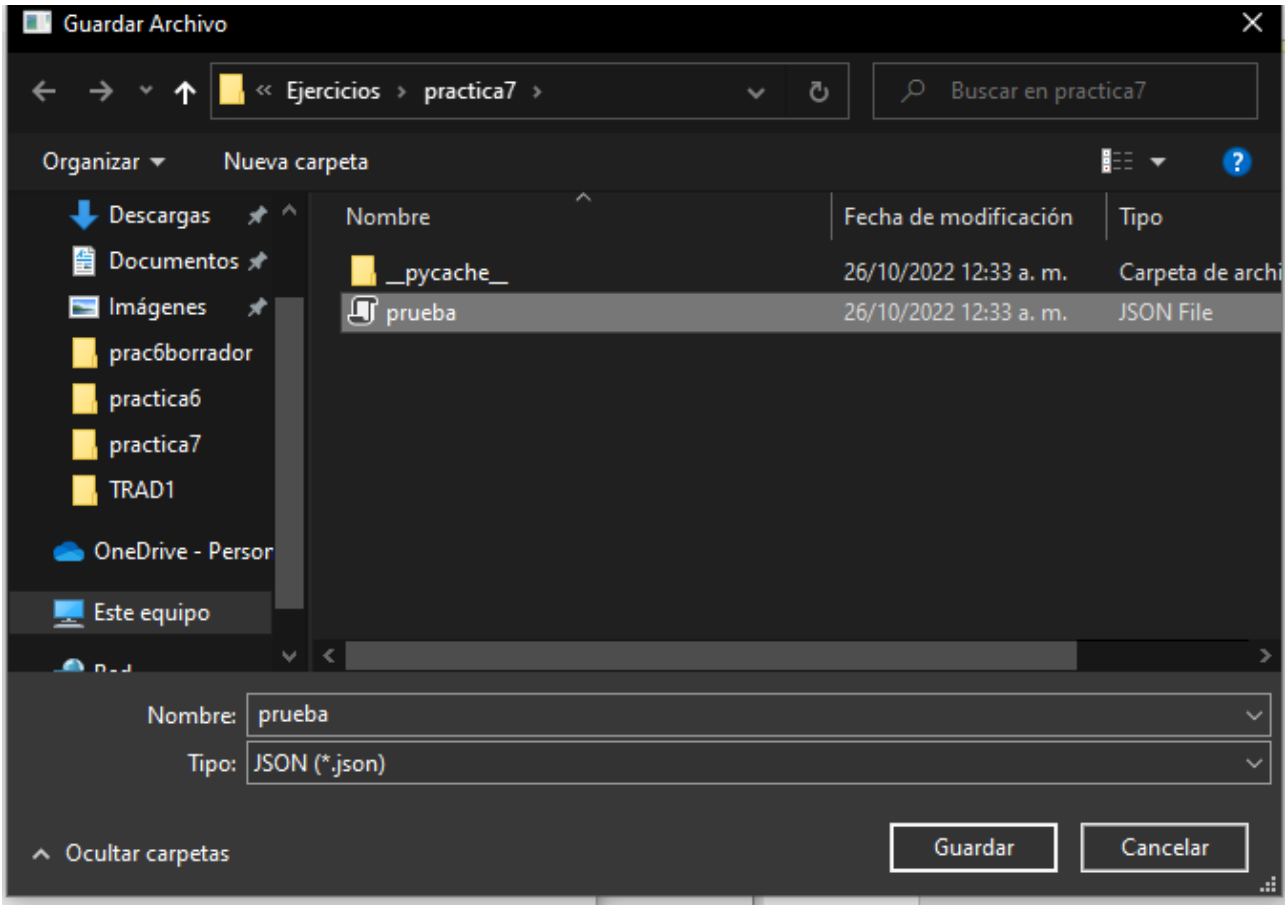
Desarrollo

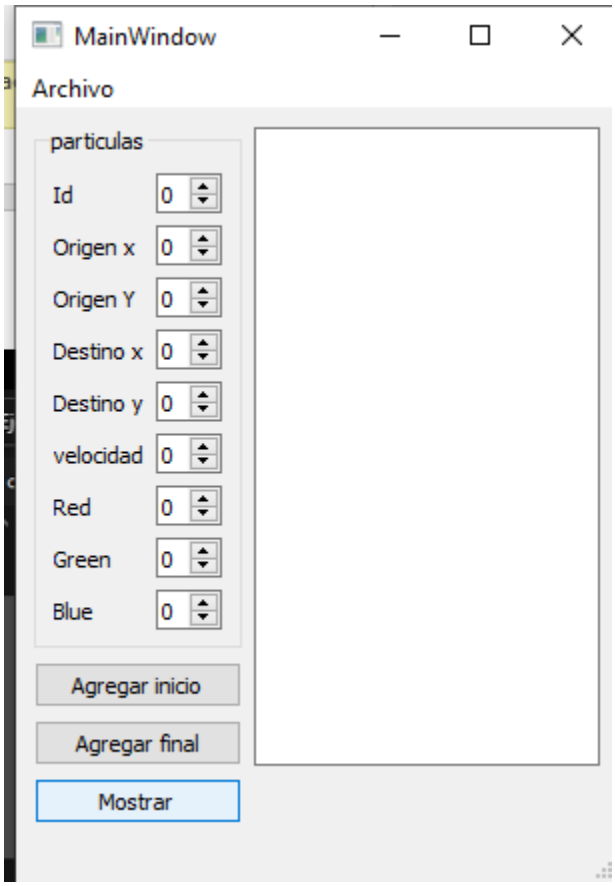
Implementamos el menú de Guardar y Abrir archivo para poder recuperar información de las particulas.
Se agregan particulas y posterior mente se muestran



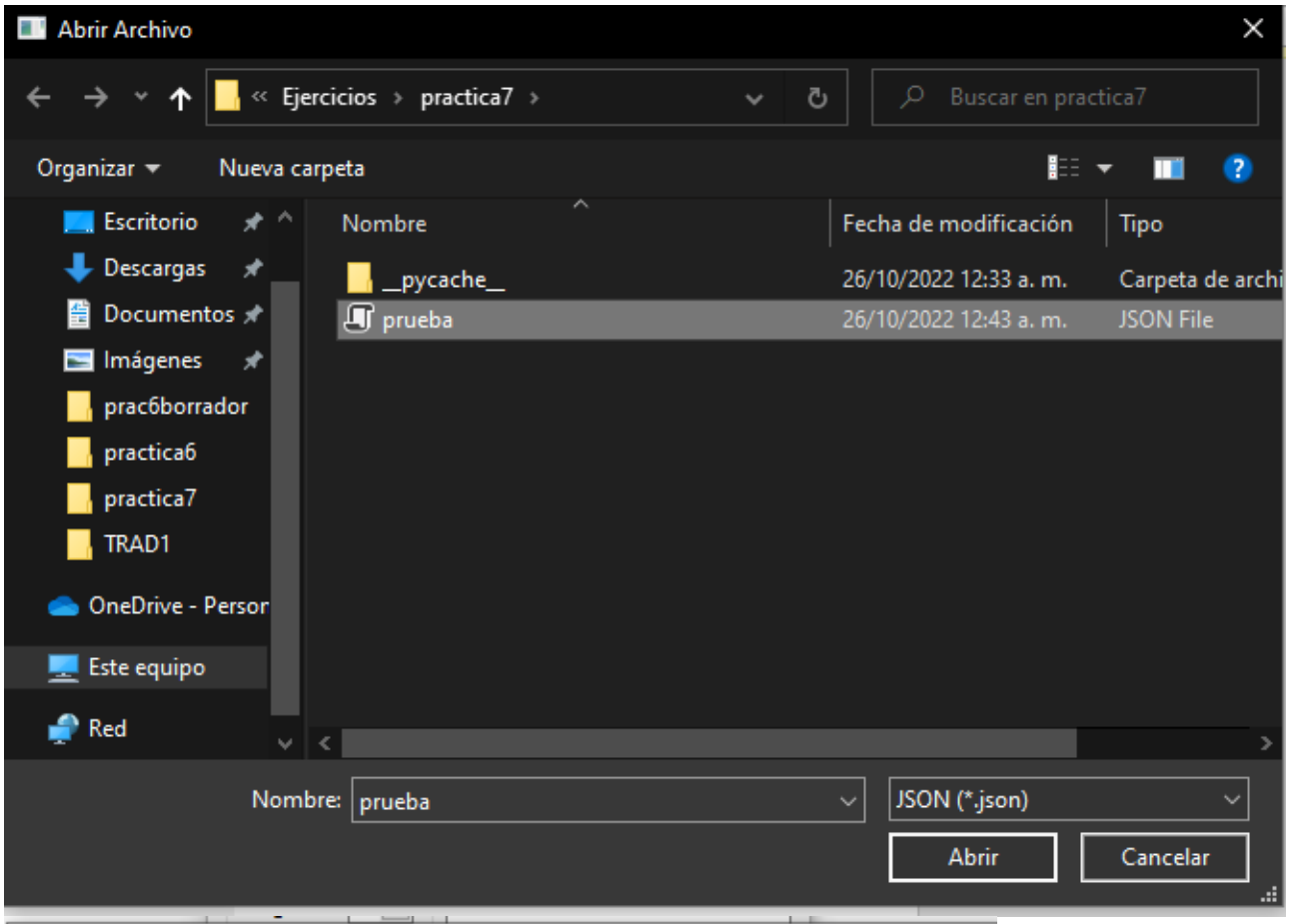


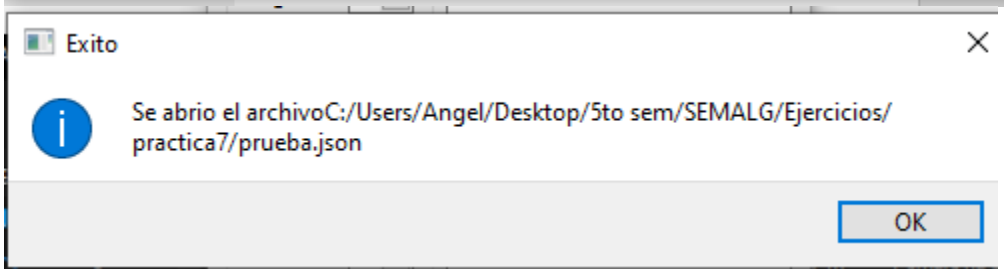
Se guarda el archivo en un documento .JSON

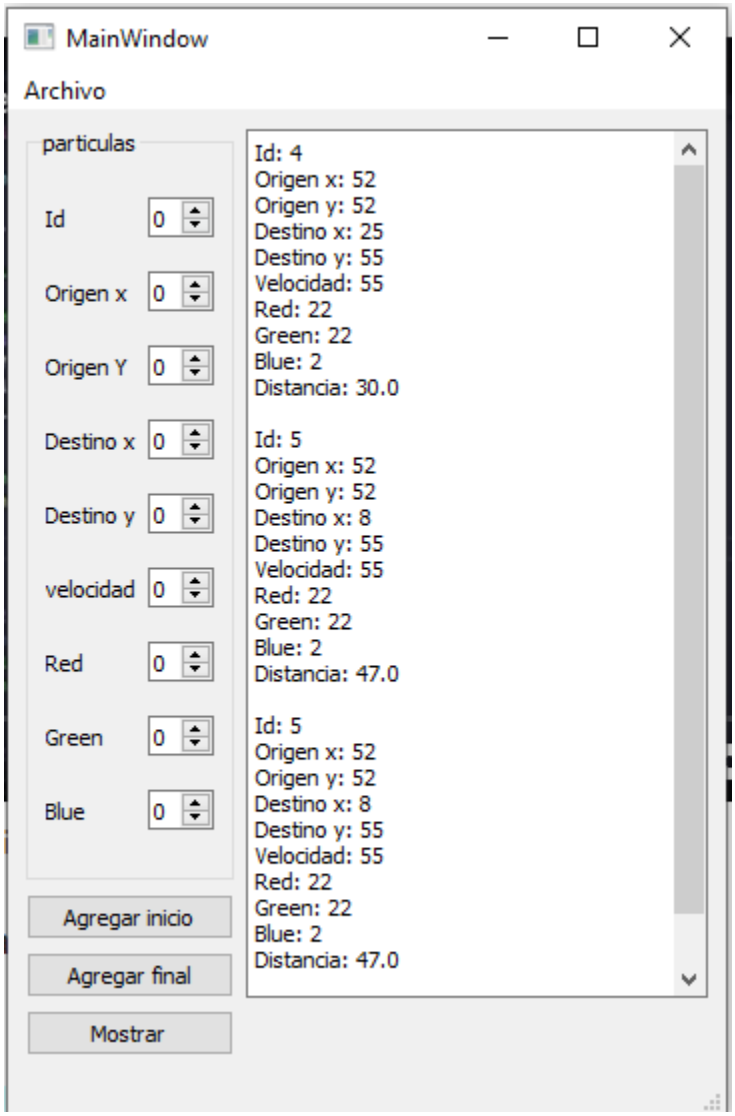

Guardar Archivo
Ejercicios › practica7 ›
Buscar en practica7
Organizar
Nueva carpeta
Descargas
Documentos
Imágenes
prac6borrador
practica6
practica7
TRAD1
OneDrive - Persor
Este equipo
Nombre
Fecha de modificación
Tipo
__pycache__
26/10/2022 12:33 a. m.
Carpeta de archi
prueba
26/10/2022 12:33 a. m.
JSON File
Nombre: prueba
Tipo: JSON (*.json)
Ocultar carpetas
Guardar
Cancelar



Se abre el documento .JSON y se muestran las particulas.


Abrir Archivo
Ejercicios › practica7 ›
Buscar en practica7
Organizar
Nueva carpeta
Escritorio
Descargas
Documentos
Imágenes
prac6borrador
practica6
practica7
TRAD1
OneDrive - Persor
Este equipo
Red
Nombre
Fecha de modificación
Tipo
__pycache__
26/10/2022 12:33 a. m.
Carpeta de archi
prueba
26/10/2022 12:43 a. m.
JSON File
Nombre:
prueba
JSON (*.json)
Abrir
Cancelar


Exito
Se abrio el archivoC:/Users/Angel/Desktop/5to sem/SEMALG/Ejercicios/
practica7/prueba.json
OK



Conclusiones

Me agrada la practica, pero tuve problemas al leer el archivo, la verdad no se cual fue el problema ni como lo solucione, solo fui moviendo cosas pero por ello tarde al entregar la actividad.

Referencias

PySide2 - QFileDialog (Qt for Python)(IV). (2020, October 22). https://youtu.be/HRY8QvXmcDM

Código 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,distancia=0.0):
        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, destino_x, origen_y,
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_dic(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,
```

```
            #'distancia':self.__distancia

        }
```

Algoritmos.py

```
from math import sqrt
def distancia_euclidiana(x_1, y_1, x_2, y_2):
    """ Calcula la distancia euclidiana
    Devuelve el resultado de la fórmula
    También se le conoce a la fórmula como:
    distancia entre dos puntos
    Parámetros:
    x_1 -- origen_x
    y_1 -- origen_y
    x_2 -- destino_x
    y_2 -- destino_y
    """
    return(sqrt(((x_2 - x_1)**2) + ((y_2 - y_1)**2)))
```

particulas.py

```
from particula import Particula
import json
class Particulas:
    def __init__(self):
        self.__particulas = []

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

    def agregar_final(self,particula:Particula):
        self.__particulas.append(particula)
```

```
    def mostrar(self):
        for particula in self.__particulas:
            print(particula)

    def __str__(self):
        return "".join(
            str(particula) + '\n' for particula in self.__particulas
        )
    def guardar(self,ubicacion):
        try:
            with open(ubicacion,"w") as archivo:
                lista=[particula.to_dic() for particula in self.__particulas]
                print(lista)
                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 particula import Particula
from particulas import Particulas

#pyside2-uic mainwindow.ui para pasar de .ui a python
class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()

        self.particulas= Particulas()
```

```
        self.ui=Ui_MainWindow()
        self.ui.setupUi(self)
        self.ui.Agregar_final_pushButton.clicked.connect(self.click_agregar)
        self.ui.Agregar_inicio_pushButton.clicked.connect(self.click_agregar_inicio
)
        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)
    @Slot()
    def action_abrir_archivo(self):
        #print("abrir")
        ubicacion=QFileDialog.getOpenFileName(
            self,
            "Abrir Archivo",
            ".",
            "JSON (*.json)"
        )[0]
        if self.particulas.abrir(ubicacion):
            QMessageBox.information(
                self,
                "Exito",
                "Se abrio el archivo" + ubicacion
            )
        else:
            QMessageBox.critical(
                self,
                "Error",
                "No se pudo abrir el archivo"
            )
    @Slot()
    def action_guardar_archivo(self):
        #print("guardar")
        ubicacion=QFileDialog.getSaveFileName(
            self,
            "Guardar Archivo",
            ".",
            "JSON (*.json)"

        )[0]
        print(ubicacion)
        if self.particulas.guardar(ubicacion):
            QMessageBox.information(
                self,
                "Exito",
                "Se pudo crear el archivo" + ubicacion
```

```
            )
        else:
            QMessageBox.critical(
                self,
                "Error",
                "No se pudo crear el archivo"
            )
    @Slot()
    def click_mostrar(self):
        self.ui.salida.clear()
        self.ui.salida.insertPlainText(str(self.particulas))

    @Slot()
    def click_agregar(self):
        id=self.ui.Id_spinBox.text()
        origen_x=self.ui.Origen_x_spinBox.value()
        origen_y=self.ui.Origen_y_spinBox.value()
        destino_x=self.ui.Destino_x_spinBox.value()
        destino_y=self.ui.Destino_y_spinBox.value()
        velocidad=self.ui.Velocidad_spinBox.value()
        red=self.ui.Red_spinBox.value()
        green=self.ui.Green_spinBox.value()
        blue=self.ui.Blue_spinBox.value()

        Particula1=Particula(id,origen_x,origen_y,destino_x,destino_y,velocidad,red
,green,blue)
        self.particulas.agregar_final(Particula1)

    @Slot()
    def click_agregar_inicio(self):
        id=self.ui.Id_spinBox.text()
        origen_x=self.ui.Origen_x_spinBox.value()
        origen_y=self.ui.Origen_y_spinBox.value()
        destino_x=self.ui.Destino_x_spinBox.value()
        destino_y=self.ui.Destino_y_spinBox.value()
        velocidad=self.ui.Velocidad_spinBox.value()
        red=self.ui.Red_spinBox.value()
        green=self.ui.Green_spinBox.value()
        blue=self.ui.Blue_spinBox.value()

        Particula1=Particula(id,origen_x,origen_y,destino_x,destino_y,velocidad,red
,green,blue)
        self.particulas.agregar_inicio(Particula1)
```

Main.py

```
from PySide2.QtWidgets import QApplication
from  mainwindow import MainWindow
#pyside2-uic mainwindow.ui para pasar de .ui a python
#pyside2-uic mainwindow.ui >ui_mainwindow.py
import sys
app=QApplication()
window=MainWindow()
window.show()
sys.exit(app.exec_())
```