

## **ACTIVIDAD 07**

Cruz Collazo Wendy Paola

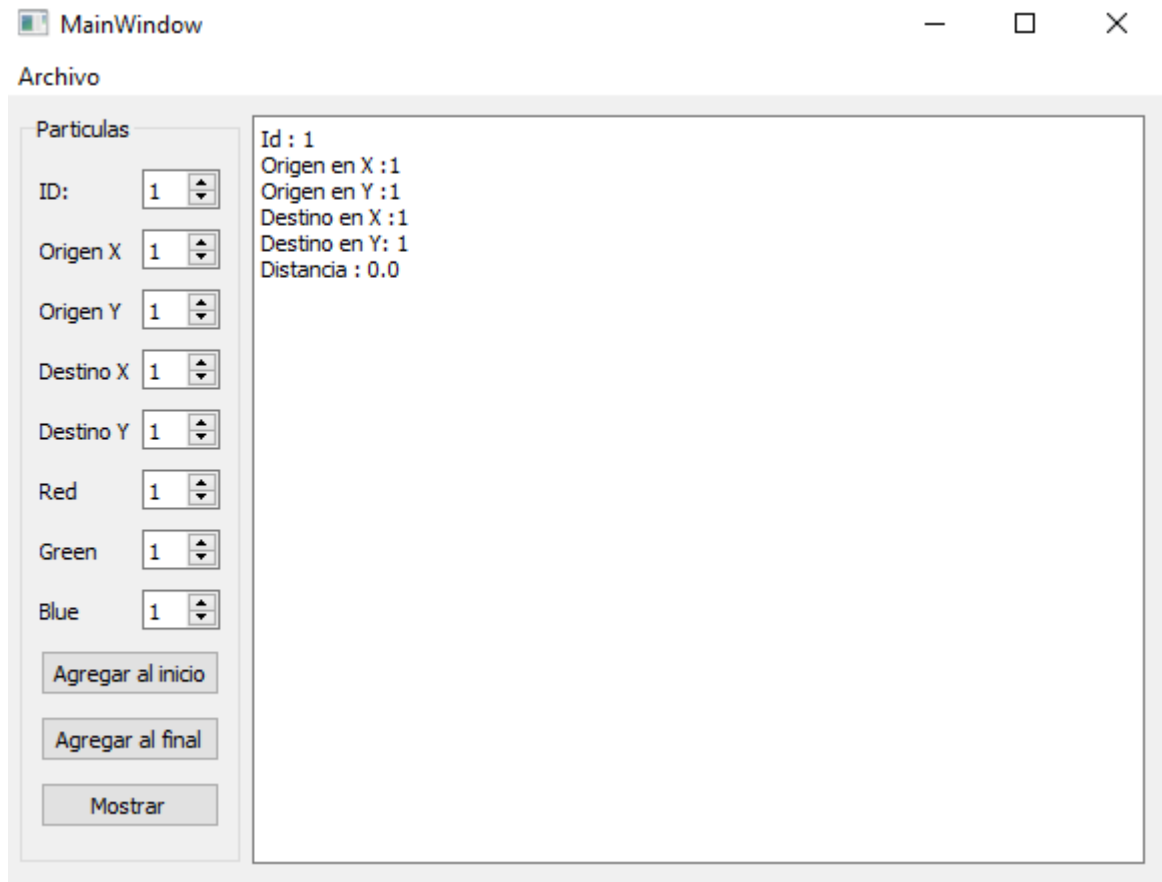
### SEMINARIO DE SOLUCION DE PROBLEMAS DE ALGORITMIA

#### Lineamientos de Evaluación:

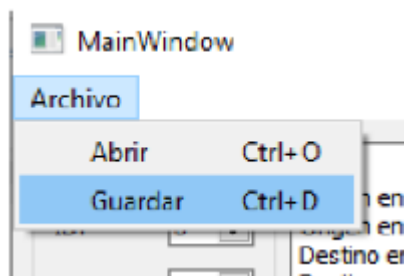
- ✓ El reporte esta en Formato Google Docs o PDF.
- ✓ El reporte sigue las pautas del Formato de Actividades.
- ✓ El reporte tiene desarrolladas todas las pautas del Formato de Actividades.
- ✓ Se muestra la captura de pantalla de las partículas con el método mostrar () previo a generar el respaldo.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

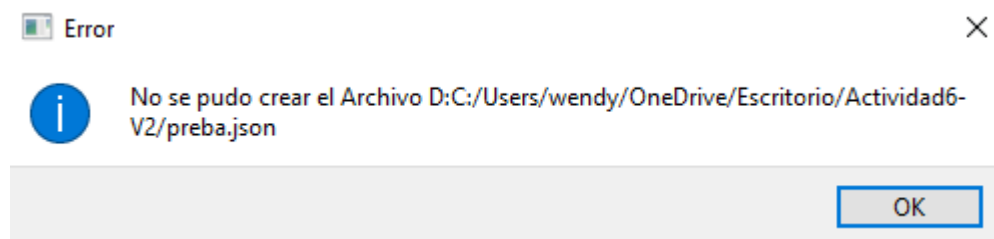
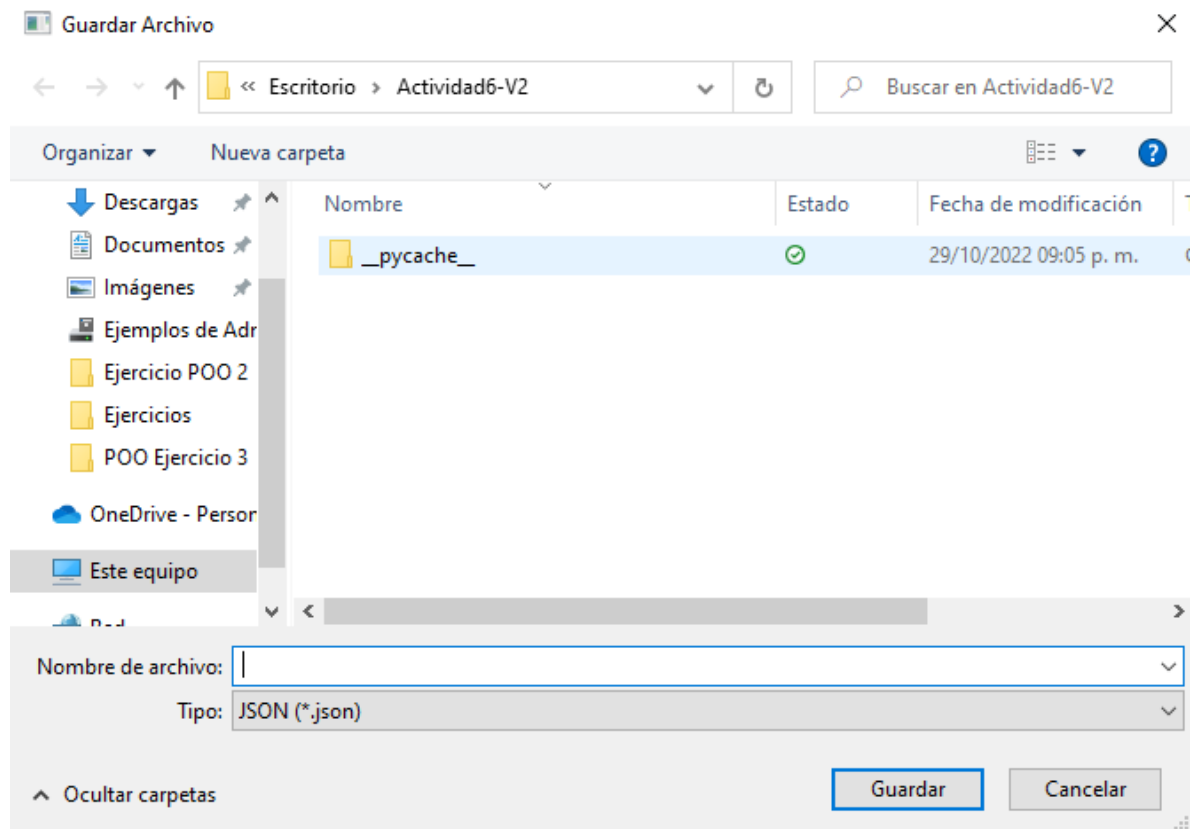
## ○ Desarrollo

Captura de Pantalla con el Metodo de Mostrar.



Captura de Pantalla de los pasos que necesitamos realizar en la Interfaz para hacer el respaldo.



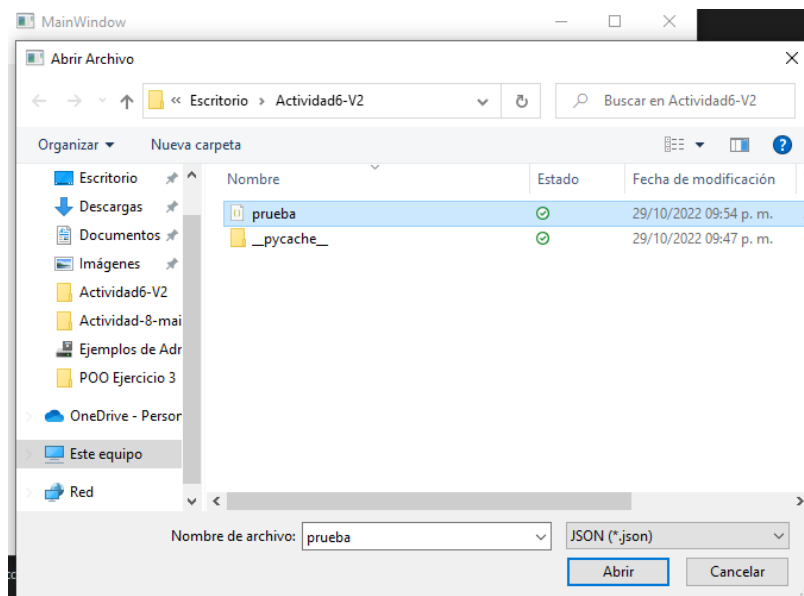
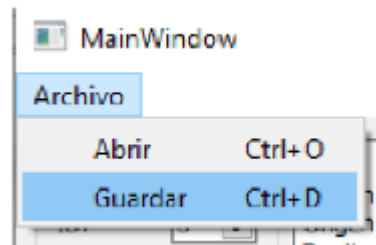


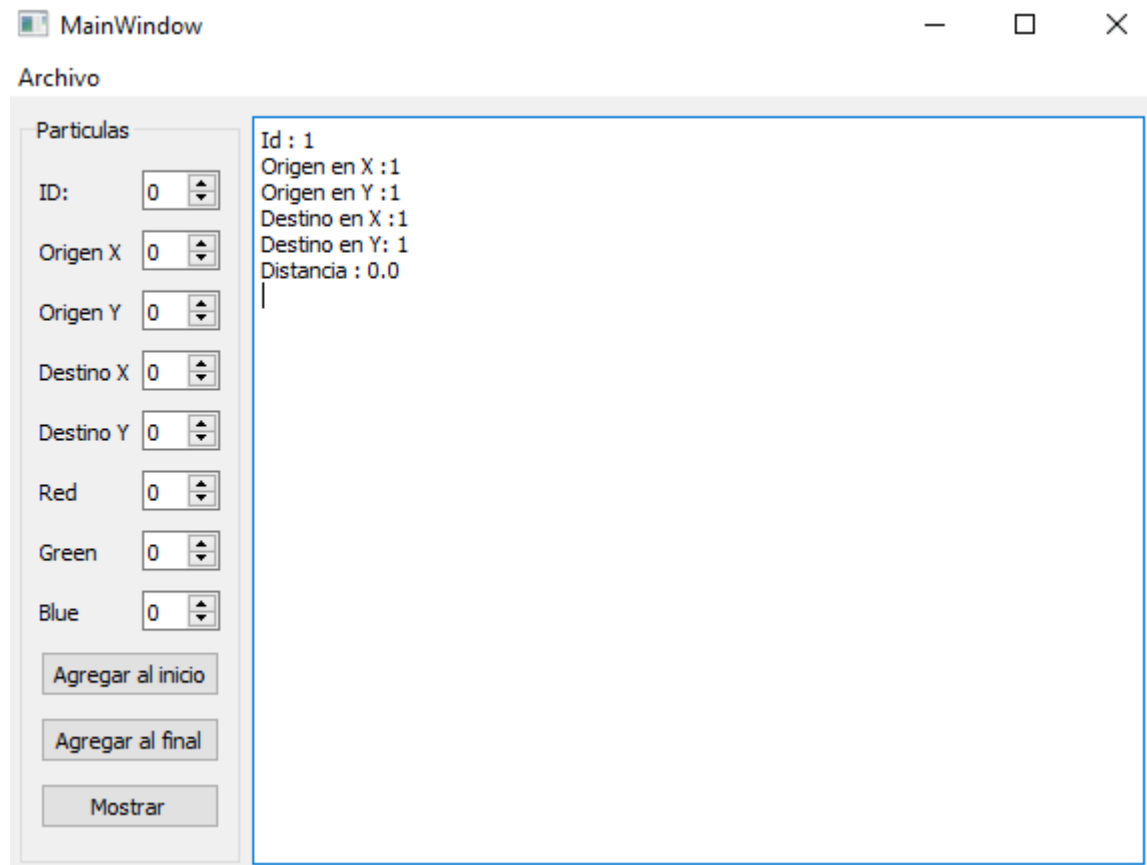
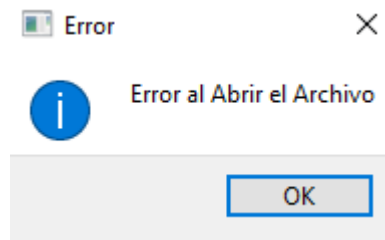
Archivo ,json



```
mainwindow.py  prueba.json x
{} prueba.json > {} 0 > # blue
1  [
2    {
3      "id": 1,
4      "origen_x": 1,
5      "origen_y": 1,
6      "destino_x": 1,
7      "destino_y": 1,
8      "red": 1,
9      "green": 1,
10     "blue": 1
11   }
12 ]
```

Para recuperar la información





## ○ Conclusiones

Se me facilito ya un poco mas el poder hacer el programa, solo que al momento de correr la interfaz cuando quiero guardar las partículas que inserte me sale el error como si algo estuviera mal, pero al momento de checarlo en el .json está bien. Y de igual en el de abrir.

- Referencias Bibliográficas

- PySide2- QFileDialog (Qt for Python)(IV)

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

- 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(580, 433)
○         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_2 = QGridLayout(self.centralwidget)
○         self.gridLayout_2.setObjectName(u"gridLayout_2")
○         self.groupBox = QGroupBox(self.centralwidget)
○         self.groupBox.setObjectName(u"groupBox")
○         self.gridLayout = QGridLayout(self.groupBox)
○         self.gridLayout.setObjectName(u"gridLayout")
○         self.label_5 = QLabel(self.groupBox)
```

```
o     self.label_5.setObjectName(u"label_5")
o
o     self.gridLayout.addWidget(self.label_5, 5, 0, 1, 1)
o
o     self.Red_spinBox = QSpinBox(self.groupBox)
o     self.Red_spinBox.setObjectName(u"Red_spinBox")
o
o     self.gridLayout.addWidget(self.Red_spinBox, 6, 1, 1, 1)
o
o     self.label_2 = QLabel(self.groupBox)
o     self.label_2.setObjectName(u"label_2")
o
o     self.gridLayout.addWidget(self.label_2, 6, 0, 1, 1)
o
o     self.DestinoY_spinBox = QSpinBox(self.groupBox)
o     self.DestinoY_spinBox.setObjectName(u"DestinoY_spinBox")
o     self.DestinoY_spinBox.setMaximum(255)
o
o     self.gridLayout.addWidget(self.DestinoY_spinBox, 5, 1, 1, 1)
o
o     self.label_4 = QLabel(self.groupBox)
o     self.label_4.setObjectName(u"label_4")
o
o     self.gridLayout.addWidget(self.label_4, 4, 0, 1, 1)
o
o     self.label0 = QLabel(self.groupBox)
o     self.label0.setObjectName(u"label0")
o
o     self.gridLayout.addWidget(self.label0, 1, 0, 1, 1)
o
o     self.Origeny_spinBox = QSpinBox(self.groupBox)
o     self.Origeny_spinBox.setObjectName(u"Origeny_spinBox")
o     self.Origeny_spinBox.setMaximum(999)
o
o     self.gridLayout.addWidget(self.Origeny_spinBox, 3, 1, 1, 1)
o
o     self.Agregar_final = QPushButton(self.groupBox)
o     self.Agregar_final.setObjectName(u"Agregar_final")
o
o     self.gridLayout.addWidget(self.Agregar_final, 10, 0, 1, 2)
o
o     self.DestinoX_spinBox = QSpinBox(self.groupBox)
o     self.DestinoX_spinBox.setObjectName(u"DestinoX_spinBox")
o     self.DestinoX_spinBox.setMaximum(255)
o
```

```
o         self.gridLayout.addWidget(self.DestinoX_spinBox, 4, 1, 1, 1)
o
o         self.label_8 = QLabel(self.groupBox)
o         self.label_8.setObjectName(u"label_8")
o
o         self.gridLayout.addWidget(self.label_8, 8, 0, 1, 1)
o
o         self.ID_spinBox = QSpinBox(self.groupBox)
o         self.ID_spinBox.setObjectName(u"ID_spinBox")
o         self.ID_spinBox.setMaximum(999)
o
o         self.gridLayout.addWidget(self.ID_spinBox, 0, 1, 1, 1)
o
o         self.label = QLabel(self.groupBox)
o         self.label.setObjectName(u"label")
o
o         self.gridLayout.addWidget(self.label, 0, 0, 1, 1)
o
o         self.OriginX_spinBox = QSpinBox(self.groupBox)
o         self.OriginX_spinBox.setObjectName(u"OriginX_spinBox")
o         self.OriginX_spinBox.setMaximum(999)
o
o         self.gridLayout.addWidget(self.OriginX_spinBox, 1, 1, 1, 1)
o
o         self.Mostrar = QPushButton(self.groupBox)
o         self.Mostrar.setObjectName(u"Mostrar")
o
o         self.gridLayout.addWidget(self.Mostrar, 11, 0, 1, 2)
o
o         self.label_3 = QLabel(self.groupBox)
o         self.label_3.setObjectName(u"label_3")
o
o         self.gridLayout.addWidget(self.label_3, 3, 0, 1, 1)
o
o         self.Green_spinBox = QSpinBox(self.groupBox)
o         self.Green_spinBox.setObjectName(u"Green_spinBox")
o
o         self.gridLayout.addWidget(self.Green_spinBox, 7, 1, 1, 1)
o
o         self.Agregar_Inicio = QPushButton(self.groupBox)
o         self.Agregar_Inicio.setObjectName(u"Agregar_Inicio")
o
o         self.gridLayout.addWidget(self.Agregar_Inicio, 9, 0, 1, 2)
o
o         self.Blue_spinBox = QSpinBox(self.groupBox)
```



```

o         self.Blue_spinBox.setObjectName(u"Blue_spinBox")
o
o         self.gridLayout.addWidget(self.Blue_spinBox, 8, 1, 1, 1)
o
o         self.label_7 = QLabel(self.groupBox)
o         self.label_7.setObjectName(u"label_7")
o
o         self.gridLayout.addWidget(self.label_7, 7, 0, 1, 1)
o
o         self.gridLayout_2.addWidget(self.groupBox, 0, 1, 1, 1)
o
o         self.Print = QPlainTextEdit(self.centralwidget)
o         self.Print.setObjectName(u"Print")
o
o         self.gridLayout_2.addWidget(self.Print, 0, 2, 1, 1)
o
o         MainWindow.setCentralWidget(self.centralwidget)
o         self.menubar = QMenuBar(MainWindow)
o         self.menubar.setObjectName(u"menubar")
o         self.menubar.setGeometry(QRect(0, 0, 580, 21))
o         self.menuArchivo = QMenu(self.menubar)
o         self.menuArchivo.setObjectName(u"menuArchivo")
o         MainWindow.setMenuBar(self.menubar)
o         self.statusbar = QStatusBar(MainWindow)
o         self.statusbar.setObjectName(u"statusbar")
o         MainWindow.setStatusBar(self.statusbar)
o
o         self.menubar.addAction(self.menuArchivo.menuAction())
o         self.menuArchivo.addAction(self.actionAbrir)
o         self.menuArchivo.addAction(self.actionGuardar)
o
o         self.retranslateUi(MainWindow)
o
o         QMetaObject.connectSlotsByName(MainWindow)
o     # setupUi
o
o     def retranslateUi(self, MainWindow):
o         MainWindow.setWindowTitle(QCoreApplication.translate("MainWind
ow", u"MainWindow", None))
o         self.actionAbrir.setText(QCoreApplication.translate("MainWindo
w", u"Abrir", None))
o         #if QT_CONFIG(shortcut)
o         self.actionAbrir.setShortcut(QCoreApplication.translate("MainW
indow", u"Ctrl+O", None))
o         #endif // QT_CONFIG(shortcut)

```

```

o         self.actionGuardar.setText(QCoreApplication.translate("MainWin
dow", u"Guardar", None))
o     #if QT_CONFIG(shortcut)
o         self.actionGuardar.setShortcut(QCoreApplication.translate("Mai
nWindow", u"Ctrl+S", None))
o     #endif // QT_CONFIG(shortcut)
o         self.groupBox.setTitle(QCoreApplication.translate("MainWindow"
, u"Particulas", None))
o         self.label_5.setText(QCoreApplication.translate("MainWindow",
u"Destino Y", None))
o         self.label_2.setText(QCoreApplication.translate("MainWindow",
u"Red", None))
o         self.label_4.setText(QCoreApplication.translate("MainWindow",
u"Destino X", None))
o         self.label0.setText(QCoreApplication.translate("MainWindow",
u"Origen X", None))
o         self.Agregar_final.setText(QCoreApplication.translate("MainWin
dow", u"Agregar al final", None))
o         self.label_8.setText(QCoreApplication.translate("MainWindow",
u"Blue", None))
o         self.label.setText(QCoreApplication.translate("MainWindow",
u"ID:", None))
o         self.Mostrar.setText(QCoreApplication.translate("MainWindow",
u"Mostrar", None))
o         self.label_3.setText(QCoreApplication.translate("MainWindow",
u"Origen Y", None))
o         self.Agregar_Inicio.setText(QCoreApplication.translate("MainWi
ndow", u"Agregar al inicio", None))
o         self.label_7.setText(QCoreApplication.translate("MainWindow",
u"Green", None))
o         self.menuArchivo.setTitle(QCoreApplication.translate("MainWind
ow", u"Archivo", None))
o         # retranslateUi
o

```

## o Main.py

```

o from PySide2.QtWidgets import QApplication
o from mainwindow import MainWindow
o import sys
o
o app = QApplication()
o
o window = MainWindow()
o
o window.show()

```

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

- `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):
        self.__id = id
        self.__origen_x = origen_x
        self.__origen_y = origen_y
        self.__destino_x = destino_x
        self.__destino_y = destino_y
        self.distancia =
distancia_euclidiana(origen_x, origen_y, destino_x, destino_y)

    def __str__(self):
        return('Id : ' + str(self.__id) + '\n' + 'Origen en X : ' +
str(self.__origen_x) + '\n' +
            'Origen en Y : ' + str(self.__origen_y) + '\n' + 'Destino en X
:' + str(self.__destino_x) + '\n' +
            'Destino en Y: ' + str(self.__destino_y) + '\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,
            "red": self.__red,
            "green": self.__green,
            "blue": self.__blue
        }
```

- `mainwindow.py`

- `from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox`
- `from ui_mainwindow import Ui_MainWindow`
- `from administradora import Administradora`

```

o from particula import Particula
o from PySide2.QtCore import Slot
o
o class MainWindow(QMainWindow):
o     def __init__(self):
o         super(MainWindow,self).__init__()
o
o         self.administrador = Administradora()
o
o         self.ui = Ui_MainWindow()
o         self.ui.setupUi(self)
o         self.ui.Agregar_final.clicked.connect(self.agregar_final)
o         self.ui.Agregar_Inicio.clicked.connect(self.agregar_inicio)
o         self.ui.Mostrar.clicked.connect(self.ver)
o         self.ui.actionAbrir.triggered.connect(self.action_abrir_archiv
o)
o         self.ui.actionGuardar.triggered.connect(self.action_guardar_ar
o chivo)
o
o         @Slot()
o         def action_abrir_archivo(self):
o             ubicacion = QFileDialog.getOpenFileName(self, 'Abrir
o Archivo','.', 'JSON (*.json)')[0]
o             if self.administrador.abrir(ubicacion):
o                 QMessageBox.information(self,"Exito", "Se abrio el Archivo
o de" + ubicacion)
o             else:
o                 QMessageBox.information(self,"Error","Error al Abrir el
o Archivo" + ubicacion)
o
o         @Slot()
o         def action_guardar_archivo(self):
o             ubicacion = QFileDialog.getSaveFileName(self, 'Guardar
o Archivo','.', 'JSON (*.json)')[0]
o             if self.administrador.guardar(ubicacion):
o                 QMessageBox.information(self, "Exito", "Se pudo Crear el
o archivo en" + ubicacion)
o             else:
o                 QMessageBox.information(self, "Error", "No se pudo crear
o el Archivo D:" + ubicacion)
o
o         @Slot()
o         def ver(self):
o             self.ui.Print.clear()
o             self.ui.Print.insertPlainText(str(self.administrador))

```

```

o
o
o @Slot()
o def agregar_final(self):
o     ID = self.ui.ID_spinBox.value()
o     OrigenX = self.ui.OrigenX_spinBox.value()
o     OrigenY = self.ui.OrigenY_spinBox.value()
o     DestinoX = self.ui.DestinoX_spinBox.value()
o     DestinoY = self.ui.DestinoY_spinBox.value()
o     Red = self.ui.Red_spinBox.value()
o     Green = self.ui.Green_spinBox.value()
o     Blue = self.ui.Blue_spinBox.value()
o
o
o     particula1 = Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY)
o     self.administrador.agregar_final(particula1)
o
o
o @Slot()
o def agregar_inicio(self):
o     ID = self.ui.ID_spinBox.value()
o     OrigenX = self.ui.OrigenX_spinBox.value()
o     OrigenY = self.ui.OrigenY_spinBox.value()
o     DestinoX = self.ui.DestinoX_spinBox.value()
o     DestinoY = self.ui.DestinoY_spinBox.value()
o     Red = self.ui.Red_spinBox.value()
o     Green = self.ui.Green_spinBox.value()
o     Blue = self.ui.Blue_spinBox.value()
o
o
o     particula1 = Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY)
o     self.administrador.agregar_inicio(particula1)
o
o

```

## o Algoritmos.py

```

o import math
o
o def distancia_euclidiana(x_1, y_1, x_2, y_2):
o     a = (x_2 - x_1)*(x_2 - x_1)
o     b = (y_2 - y_1)*(y_2 - y_1)
o
o     c = a + b
o
o     distancia = math.sqrt(c)
o
o     return distancia

```

- 
- 

- Administradora.py

```
from particula import Particula
import json

class Administradora:
    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 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
```