

Actividad 07 (QFileDialog)



RAFAEL ARTURO GUTIERREZ CRUZ

Seminario de Solucion de Problemas de Algoritmia

Lineamientos de evaluación

- [] 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 la captura de pantalla de las partículas con el método mostrar() previo a generar el respaldo.
- [] Se muestran capturas de pantallas de los pasos que se realizan en la interfaz para generar el respaldo.
- [] Se muestra el contenido del archivo *.json*.
- [] Se muestran capturas de pantallas de los pasos que se realizan en la interfaz para abrir el archivo de respaldo *.json*.
- [] Se muestra la captura de pantalla de las partículas con el método mostrar() después de abrir el respaldo.

Desarrollo

MainWindow

File

Particula

Destino X: 0

Destino Y: 0

Velocidad: 0

Red 0

Green 0

Blue 0

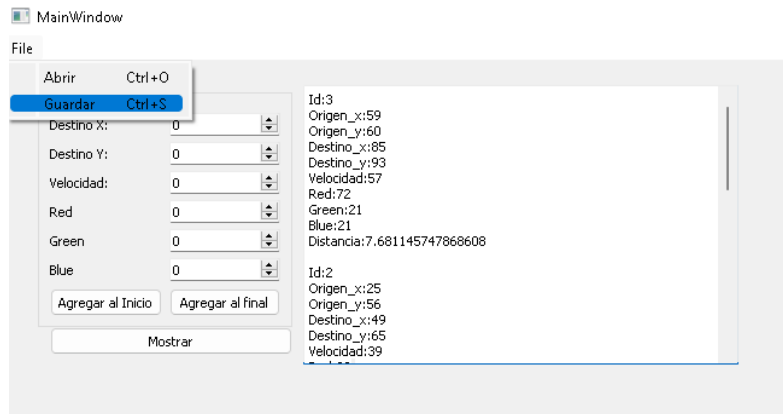
Agregar al Inicio Agregar al final

Mostrar

Id:3
Origen_x:59
Origen_y:60
Destino_x:85
Destino_y:93
Velocidad:57
Red:72
Green:21
Blue:21
Distancia:7.681145747868608

Id:2
Origen_x:25
Origen_y:56
Destino_x:49
Destino_y:65
Velocidad:39

Partículas antes de guardar en pruebas.json

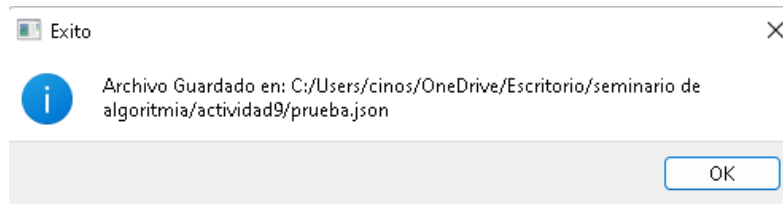
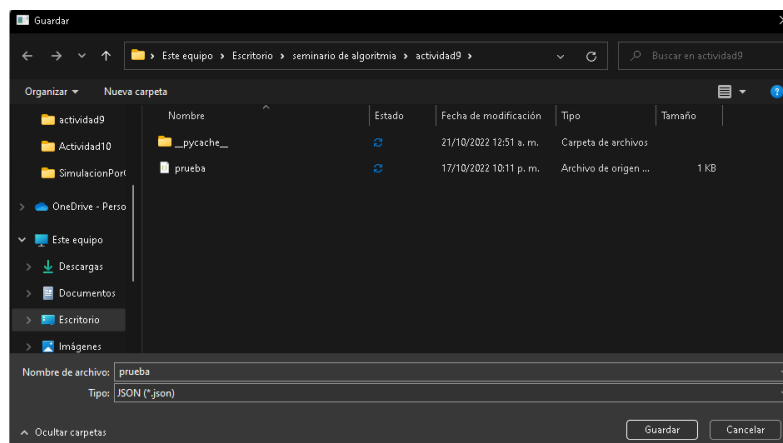


Pasos para guardar las partículas en pruebas.json:

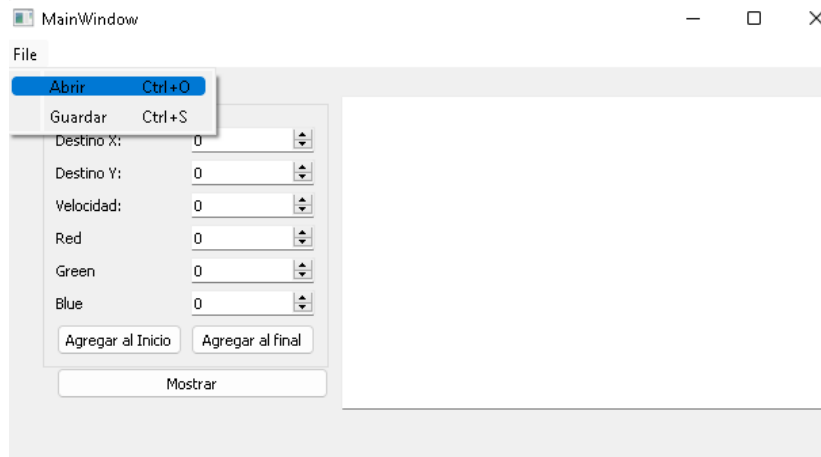
1-. Dirígete a la pestaña “File” y elige Guardar

2-. Elige la ubicación y el nombre del archivo

y cliquea en guardar



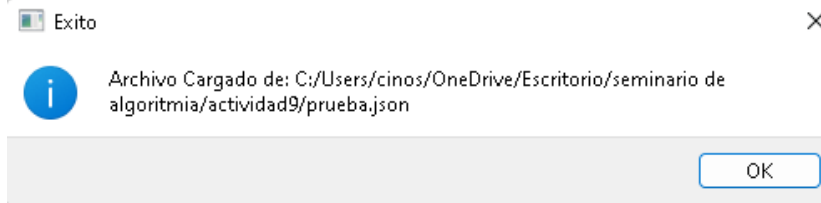
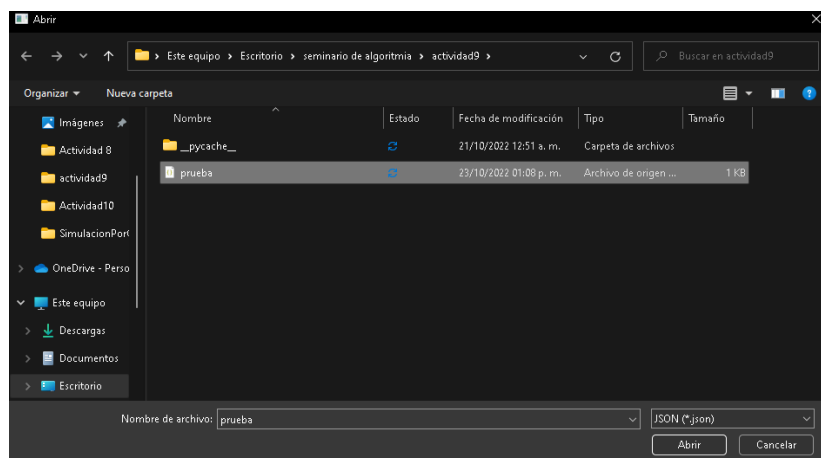
Contenido del Archivo de prueba.json



Pasos para recuperar las paraticulas

1-. Selecciona la opción abrir en la pestaña de “File”

2-. Selecciona el archivo que deseas cargar.



Conclusiones

No tuve muchas complicaciones, siguiendo el tutorial y con uno que otro error de dedo pude desarrollar la actividad.

Referencias

MICHEL DAVALOS BOITES. (2020, 22 octubre). *PySide2 - QFileDialog (Qt for Python)(IV)* [Video]. YouTube. Recuperado 23 de octubre de 2022, de <https://www.youtube.com/watch?v=HRY8QvXmcDM>

Código

algoritmos.py

```
import math

def distancia_euclidiana(x_1, y_1, x_2, y_2):

    valor1 = x_1 - y_1
    valor1**2

    valor2 = x_2 - y_2
    valor2**2

    return math.sqrt(valor1+valor2)
```

main.py

```
from PySide2.QtWidgets import QApplication
from mainwindow import MainWindow
from scipy.optimize import linprog
import sys

app =QApplication()

window = MainWindow()

window.show()

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

mainwindow.py

```
from base64 import decodebytes
from contextlib import redirect_stderr
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from particle_adminstrator import administrador
from particulas import Particula

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

```

        self.administrador = administrador()

        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

self.ui.agregarFinal_pushButton.clicked.connect(self.click_agregar)

self.ui.AgragrInicio_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('Abriendo')

            ubicacion = QFileDialog.getOpenFileName(self, 'Abrir', '.',
'JSON (*.json)')[0]

            if self.administrador.abrir(ubicacion):
                QMessageBox.information(self, "Exito", "Archivo Cargado de:
" + ubicacion)
            else:
                QMessageBox.critical(self, "Error", "No se pudo cargar el
archivo")

        @Slot()
        def action_guardar_archivo(self):
            # print('Guardando')

            ubicacion = QFileDialog.getSaveFileName(self, 'Guardar', '.',
'JSON (*.json)')[0]

            print(ubicacion)

            if self.administrador.guardar(ubicacion):
                QMessageBox.information(self, "Exito", "Archivo Guardado en:
" + ubicacion)
            else:
                QMessageBox.critical(self, "Error", "No se pudo guardar el
archivo")

        @Slot()

```

```

def click_mostrar(self):
    # self.administrador.mostrar()
    self.ui.salida.insertPlainText(str(self.administrador))

@Slot()
def click_agregar_inicio(self):
    desX = self.ui.DesX_spinBox.value()
    desY = self.ui.DesY_spinBox_2.value()
    velocidad = self.ui.Velocidad_spinBox_3.value()
    red = self.ui.Red_spinBox_4.value()
    green = self.ui.Green_spinBox_5.value()
    blue = self.ui.Blue_spinBox_6.value()

    Particle = Particula(0, destino_x=desX, destino_y=desY,
velocidad=velocidad, red=red, green=green, blue=blue)
    self.administrador.agregar_inicio(Particle)

@Slot()
def click_agregar(self):
    desX = self.ui.DesX_spinBox.value()
    desY = self.ui.DesY_spinBox_2.value()
    velocidad = self.ui.Velocidad_spinBox_3.value()
    red = self.ui.Red_spinBox_4.value()
    green = self.ui.Green_spinBox_5.value()
    blue = self.ui.Blue_spinBox_6.value()

    Particle = Particula(0, destino_x=desX, destino_y=desY,
velocidad=velocidad, red=red, green=green, blue=blue)
    self.administrador.agregar_final(Particle)

```

particle_administrador.py

```

from particulas import Particula
import json

class administrador:
    def __init__(self):
        self.__particles = []

    def agregar_final(self, particle:Particula):
        self.__particles.append(particle)

```

```

def agregar_incio(self, particle:Particula):
    self.__particles.insert(0,particle)

def mostrar(self):
    for particle in self.__particles:
        print(particle)

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

def guardar(self, ubicacion):
    try:

        with open(ubicacion, 'w') as file:
            lista = [particle.to_dict() for particle in
self.__particles]
            print(lista)
            json.dump(lista, file, indent=5)
        return 1
    except:
        return 0

def abrir(self, ubicacion):
    try:
        with open(ubicacion, 'r') as file:
            lista = json.load(file)
            self.__particles = [Particula(**particle)for particle
in lista]

        return 1
    except:
        return 0

```

particulas.py

```

import json
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(destino_x, origen_x,
destino_y, origen_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(624, 309)
    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.groupBox = QGroupBox(self.centralwidget)
    self.groupBox.setObjectName(u"groupBox")
    self.groupBox.setGeometry(QRect(30, 20, 211, 201))
    self.gridLayout = QGridLayout(self.groupBox)
    self.gridLayout.setObjectName(u"gridLayout")
    self.splitter_2 = QSplitter(self.groupBox)
    self.splitter_2.setObjectName(u"splitter_2")
    self.splitter_2.setOrientation(Qt.Vertical)
    self.label = QLabel(self.splitter_2)
    self.label.setObjectName(u"label")
    self.splitter_2.addWidget(self.label)
    self.label_2 = QLabel(self.splitter_2)
    self.label_2.setObjectName(u"label_2")
    self.splitter_2.addWidget(self.label_2)
    self.label_3 = QLabel(self.splitter_2)
    self.label_3.setObjectName(u"label_3")
    self.splitter_2.addWidget(self.label_3)
    self.label_4 = QLabel(self.splitter_2)
    self.label_4.setObjectName(u"label_4")
    self.splitter_2.addWidget(self.label_4)
    self.label_5 = QLabel(self.splitter_2)
    self.label_5.setObjectName(u"label_5")
    self.splitter_2.addWidget(self.label_5)
    self.label_6 = QLabel(self.splitter_2)
    self.label_6.setObjectName(u"label_6")
    self.splitter_2.addWidget(self.label_6)

    self.gridLayout.addWidget(self.splitter_2, 1, 0, 1, 1)

    self.AgragrInicio_pushButton = QPushButton(self.groupBox)

self.AgragrInicio_pushButton.setObjectName(u"AgragrInicio_pushButton")
```

```

self.gridLayout.addWidget(self.AgragrInicio_pushButton, 2, 0,
1, 1)

self.splitter = QSplitter(self.groupBox)
self.splitter.setObjectName(u"splitter")
self.splitter.setOrientation(Qt.Vertical)
self.splitter.setOpaqueResize(False)
self.splitter.setChildrenCollapsible(True)
self.DesX_spinBox = QSpinBox(self.splitter)
self.DesX_spinBox.setObjectName(u"DesX_spinBox")
self.DesX_spinBox.setMaximum(500)
self.splitter.addWidget(self.DesX_spinBox)
self.DesY_spinBox_2 = QSpinBox(self.splitter)
self.DesY_spinBox_2.setObjectName(u"DesY_spinBox_2")
self.DesY_spinBox_2.setMaximum(500)
self.splitter.addWidget(self.DesY_spinBox_2)
self.Velocidad_spinBox_3 = QSpinBox(self.splitter)
self.Velocidad_spinBox_3.setObjectName(u"Velocidad_spinBox_3")
self.Velocidad_spinBox_3.setMaximum(1000)
self.splitter.addWidget(self.Velocidad_spinBox_3)
self.Red_spinBox_4 = QSpinBox(self.splitter)
self.Red_spinBox_4.setObjectName(u"Red_spinBox_4")
self.Red_spinBox_4.setMaximum(255)
self.splitter.addWidget(self.Red_spinBox_4)
self.Green_spinBox_5 = QSpinBox(self.splitter)
self.Green_spinBox_5.setObjectName(u"Green_spinBox_5")
self.Green_spinBox_5.setMaximum(255)
self.splitter.addWidget(self.Green_spinBox_5)
self.Blue_spinBox_6 = QSpinBox(self.splitter)
self.Blue_spinBox_6.setObjectName(u"Blue_spinBox_6")
self.Blue_spinBox_6.setMaximum(255)
self.splitter.addWidget(self.Blue_spinBox_6)

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

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

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

self.Mostrar_pushButton = QPushButton(self.centralwidget)

```

```

        self.Mostrar_pushButton.setObjectName(u"Mostrar_pushButton")
        self.Mostrar_pushButton.setGeometry(QRect(40, 220, 201, 23))
        self.salida = QTextEdit(self.centralwidget)
        self.salida.setObjectName(u"salida")
        self.salida.setGeometry(QRect(250, 20, 361, 231))
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QMenuBar(MainWindow)
        self.menubar.setObjectName(u"menubar")
        self.menubar.setGeometry(QRect(0, 0, 624, 21))
        self.menuFile = QMenu(self.menubar)
        self.menuFile.setObjectName(u"menuFile")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QStatusBar(MainWindow)
        self.statusbar.setObjectName(u"statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.menubar.addAction(self.menuFile.menuAction())
        self.menuFile.addAction(self.actionAbrir)
        self.menuFile.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.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Particula", None))
        self.label.setText(QCoreApplication.translate("MainWindow",
u"Destino X:", None))
        self.label_2.setText(QCoreApplication.translate("MainWindow",
u"Destino Y:", None))
        self.label_3.setText(QCoreApplication.translate("MainWindow",
u"Velocidad:", None))
        self.label_4.setText(QCoreApplication.translate("MainWindow",
u"Red", None))
        self.label_5.setText(QCoreApplication.translate("MainWindow",
u"Green", None))
        self.label_6.setText(QCoreApplication.translate("MainWindow",
u"Blue", None))

self.AgragrInicio_pushButton.setText(QCoreApplication.translate("MainWi
ndow", u"Agregar al Inicio", None))

self.agregarFinal_pushButton.setText(QCoreApplication.translate("MainWi
ndow", u"Agregar al final", None))

self.Mostrar_pushButton.setText(QCoreApplication.translate("MainWindow"
, u"Mostrar", None))
        self.menuFile.setTitle(QCoreApplication.translate("MainWindow",
u"File", None))
        # retranslateUi
```