

# Actividad 07 QFileDialog

Ramirez Orduna Giovanni

## 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.

tenemos el main principal

MainWindow

Archivo

Particulas

ID

0

Origen X

0

Origen Y

0

Destino x:

0

Destino y:

0

Velocidad:

0

Rojo:

0

Verde:

0

Azul:

0

Agregar Final

Agregar inicio

Mostrar

Agregamos datos

MainWindow

Archivo

Particulas

ID	23
Origen X	3
Origen Y	2
Destino x:	3
Destino y:	2
Velocidad:	3
Rojo:	4
Verde:	4
Azul:	5

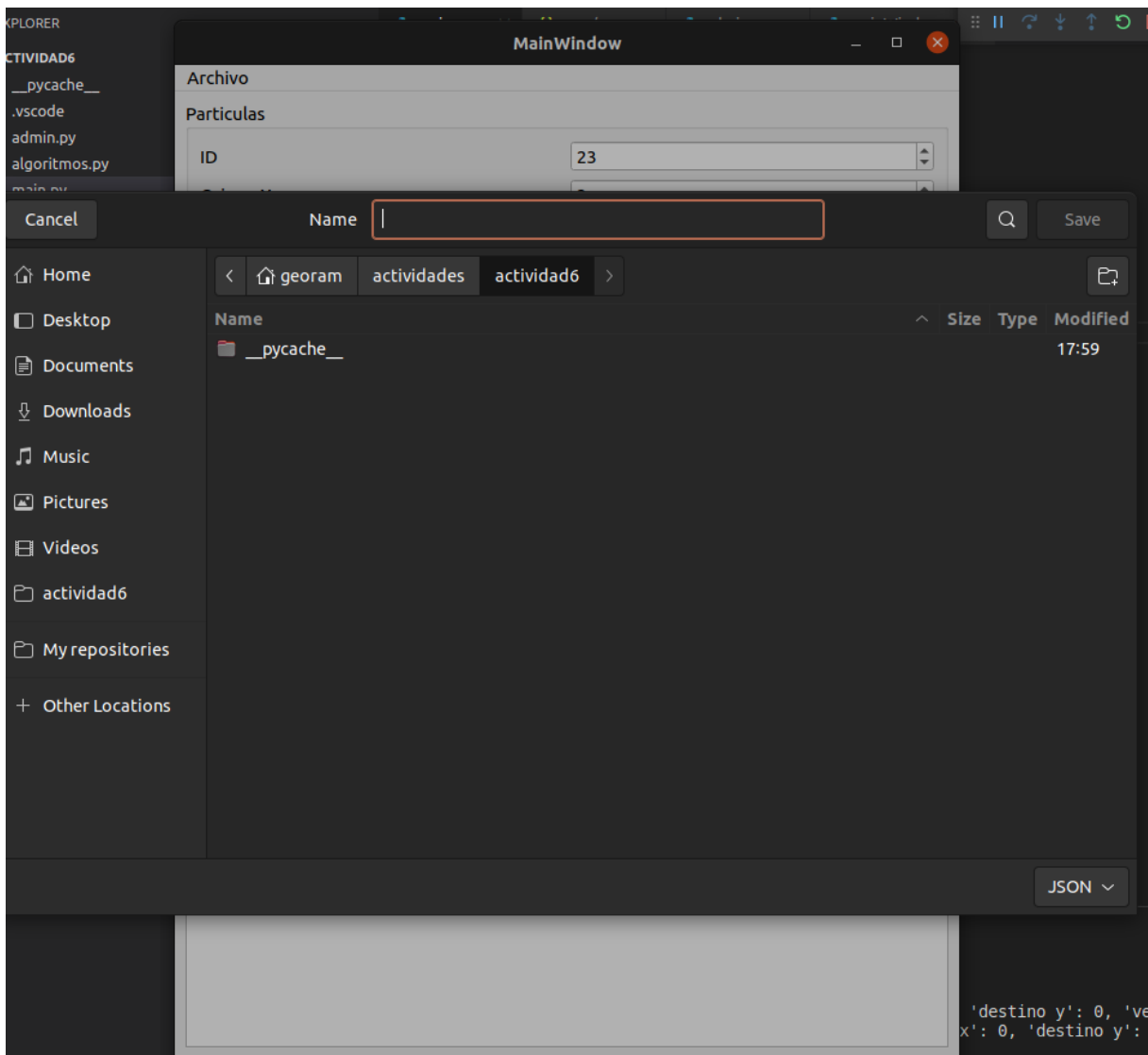
Agregar Final

Agregar inicio

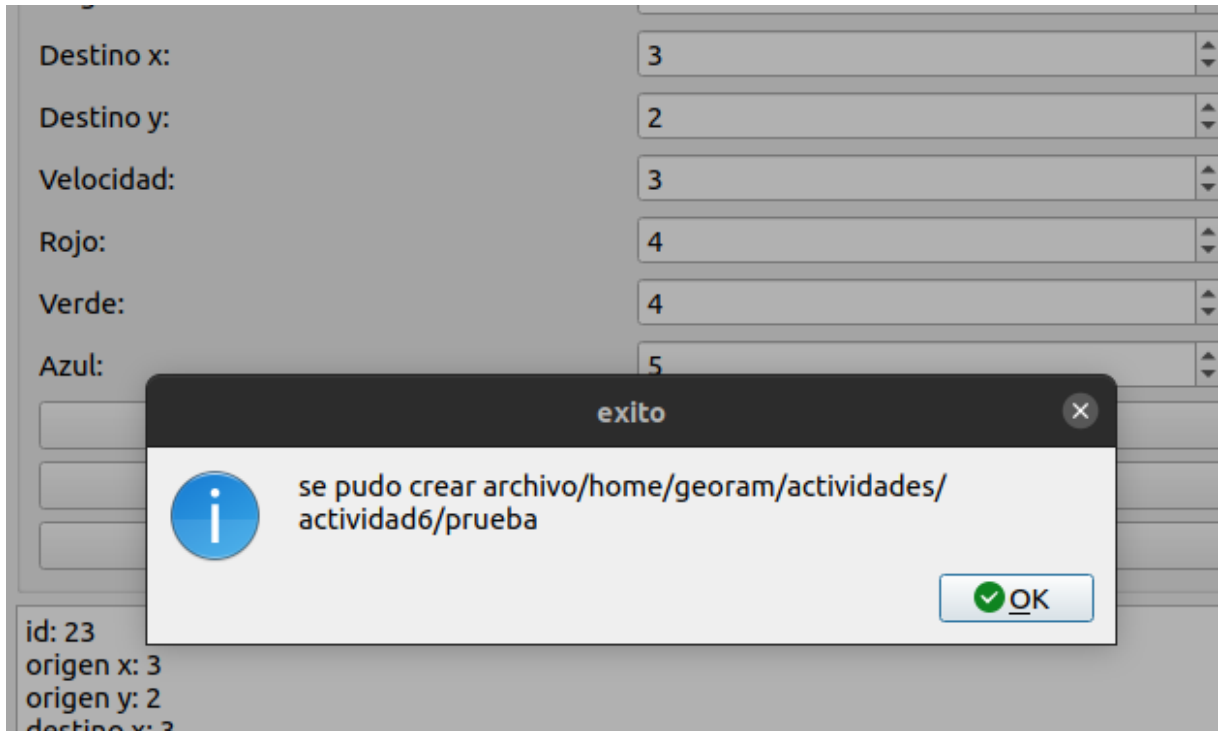
Mostrar

id: 23  
origen x: 3  
origen y: 2  
destino x: 3  
destino y: 2  
velocidad: 3  
red: 4  
green: 4  
blue: 5  
distancia: 1.4142135623730951

tenemos la ruta para guardar un archivo json



creamos el archivo hay un mensaje de confirmación que se creó



tenemos nuestro archivo con los datos que se capturaron

```
main.py  {} prueba  ×  admin.py  mainWindo
{} prueba > ...
1  [
2      {
3          "id": 23,
4          "origen x ": 3,
5          "origen y ": 2,
6          "destono x": 3,
7          "destino y": 2,
8          "velocidad": 3,
9          "red ": 4,
10         "green": 4,
11         "blue": 5,
12         "distancia": 1.4142135623730951
13     }
14 ]
```

### Conclusión

en esta actividad puede crear un respaldo de los datos capturados de mi interfaz gráfica con la extensión de json. esto puede ser muy práctico ya que tienes donde ir guardando los datos

### Referencias :

<https://www.youtube.com/watch?v=HRY8QvXmcDM&t=2668s>

## codigo

```
from simplejson import JSONEncoderForHTML
from particula import Particula
from algoritmos import distancia_euclidiana
import json

class Admin:
    def __init__(self):
        self.__almacen=[]

    def agregar_final(self, particula:Particula):
        self.__almacen.append(particula)

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

    def mostrar(self):
        for particula in self.__almacen:
            print(particula)

    def __str__(self):
        return "".join(
            str(admin)+ "\n" for admin in self.__almacen )

    def save(self, ubicacion):
        try:

            with open(ubicacion, 'w') as archivo:
                # archivo.write(str(self))
                lista= [Particula.to_dict() for Particula in
self.__almacen]
                print(lista)
                json.dump(lista, archivo, indent=5)
            return 1
        except:
            return 0

    def open(self, ubicacion):
        try:
            with open(ubicacion, "r") as archivo:
                lista=json.load(archivo)
                self.__almacen=[Particula(**Particula) for Particula in
lista ]
```

```
        return 1
    except:
        return 0
```



mainWindow

```
import PySide2
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox
from PySide2.QtCore import Slot
from ui_mainwindow import Ui_MainWindow
from admin import Admin
from particula import Particula
from algoritmos import distancia_euclidiana

class MainWIndow(QMainWindow):
    def __init__(self):
        super(MainWIndow, self).__init__()

        self.admin=Admin()

        self.ui=Ui_MainWindow()
        self.ui.setupUi(self) ##mete la intrerfaz
        self.ui.Agre_Final_button.clicked.connect(self.click_agregar)
##conectar clase a boton

self.ui.Agre_Inicio_Button.clicked.connect(self.click_agregar_inicio)
        self.ui.Mostrar_button.clicked.connect(self.click_mostrar)


        self.ui.actionAbrir.triggered.connect(self.action_abrir_archivo)
        self.ui.actionGuardar.triggered.connect(self.action_guardar)


    @Slot()
    def action_abrir_archivo(self):
        #print("abrir archivo")
        ubicacion=QFileDialog.getOpenFileName(
            self,
            "abrir archivo",
            ".",
            "JSON (*.json)"
        )[0]
        if self.admin.open(ubicacion):
            QMessageBox.information(
                self,
                "exito",
                "Se abrio el archivo"+ubicacion
            )
```

```

        else:
            QMessageBox.critical(
                self,
                "Error",
                "Error abrir archivo "
            )

@Slot()
def action_guardar(self):
    #print("guardar")
    ubicacion= QFileDialog.getSaveFileName(
        self,
        'Guardar Archivo',
        '.',
        'JSON (*.json)'
    )[0]
    print(ubicacion)
    self.admin.save(ubicacion)
    if self.admin.save(ubicacion):
        QMessageBox.information(
            self,
            "exito",
            "se pudo crear archivo"+ubicacion
        )
    else:
        QMessageBox.critical(
            self,
            "Error",
            "no se pudo crear archivo"
        )

@Slot()
def click_agregar(self):
    id=self.ui.Id_spinBox.value()
    OrigenX=self.ui.OrigenX_spinBox.value()
    OrigenY=self.ui.OrigenY_spinBox.value()
    DestinoX=self.ui.DestinoX_spinBox.value()
    DestinoY=self.ui.DestinoY_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()
        distancia=distacia_euclidiana

particula=Particula(id,OrigenX,OrigenY,DestinoX,DestinoY,Velocidad,
Red,Green,Blue,distancia)
        self.admin.agregar_final(particula)

        #print(OrigenX,OrigenY,
DestinoX,DestinoY,Velocidad,Red,Green,Blue)
        #self.ui.salida.insertPlainText(str(OrigenX) + str( OrigenY)
+ str(DestinoX) + str (DestinoY) + str (Velocidad) +str( Red)+
str(Green) + str(Blue))
    @Slot()
    def click_agregar_inicio(self):
        id=self.ui.Id_spinBox.value()
        OrigenX=self.ui.OrigenX_spinBox.value()
        OrigenY=self.ui.OrigenY_spinBox.value()
        DestinoX=self.ui.DestinoX_spinBox.value()
        DestinoY=self.ui.DestinoY_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()
        distancia=distacia_euclidiana

particula=Particula(id,OrigenX,OrigenY,DestinoX,DestinoY,Velocidad,
Red,Green,Blue,distancia)
        self.admin.agregar_inicio(particula)

    @Slot()
    def click_mostrar(self):
        # self.admin.mostrar()
        self.ui.salida.clear()
        self.ui.salida.insertPlainText(str(self.admin))

```