Actividad 06- QPlaintextEdit

Ramirez Orduna Geovanni

Lineamiento 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 los datos antes de usar el botón para agregar_inicio() y la captura de pantalla de mostrar partículas en el QPlainTextEdit después de haber agregado la Partícula.
- [] Se muestra la captura de pantalla de los datos antes de usar el botón para agregar_final() y la captura de pantalla de mostrar partículas en el QPlainTextEdit después de haber agregado la Partícula.

DESARROLLO

tenemos el main principal

```
main.py > ...
1  from PySide2.QtWidgets import QApplication
2  from mainWindow import MainWIndow
3  import sys
4
5  app=QApplication()
6
7  window=MainWIndow()
8  window.show()
9  sys.exit(app.exec_())
10
11
12
```

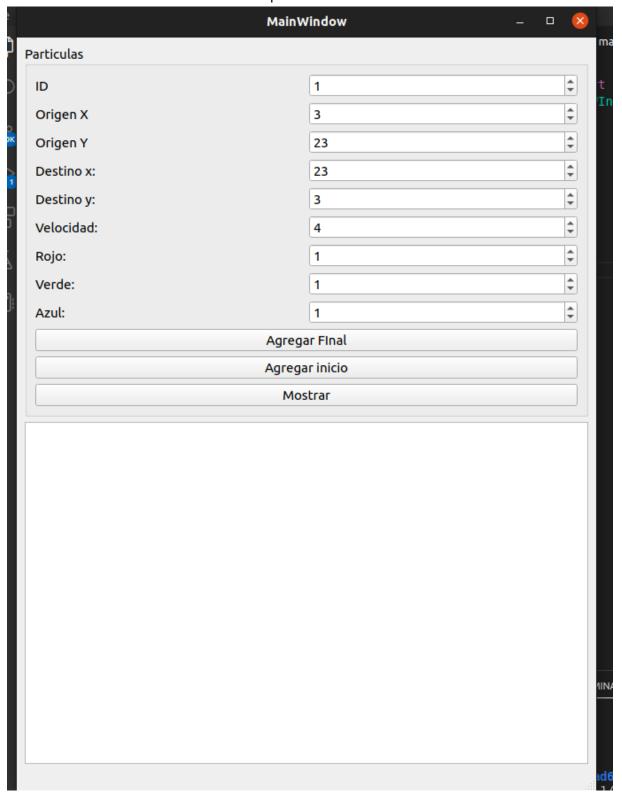
tenemos admin que importa la clase algoritmos para calcular la distancia

Tenemos la clase partícula

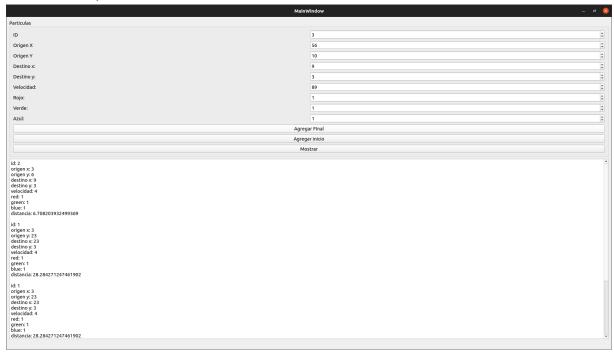
tenemos el mainWindow donde se desarrollo la conexión con la interfaz grafica

```
| Improved the content of the conten
```

tenemos la interfaz corriendo con una captura de datos



tenemos la impresión de los datos



CONCLUSIONES:

Tenemos la conexión de la interfaz gráfica para poder agregar datos de la clase particular así como calcular la distancia.

Referencias:

https://www.youtube.com/watch?v=5TPKrKIAAU0&feature=emb_title

Código

```
import PySide2
from PySide2.QtWidgets import QMainWindow
from PySide2.QtCore import Slot
from ui mainwindow import Ui MainWindow
from admin import Admin
from particula import Particula
from algoritmos import distacia euclidiana
class MainWIndow(QMainWindow):
  def init (self):
      self.admin=Admin()
      self.ui=Ui MainWindow()
       self.ui.setupUi(self) ##mete la intrerfaz
self.ui.Agre Final buttom.clicked.connect(self.click agregar)
##conectar clase a boton
self.ui.Agre Inicio Buttom.clicked.connect(self.click agregar inic
io)
       self.ui.Mostrar buttom.clicked.connect(self.click mostrar)
  @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)
```

```
@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()
          self.ui.salida.clear()
           self.ui.salida.insertPlainText(str(self.admin))
```