Actividad 06 - QPlainTextEdit

Arturo Sánchez Sánchez

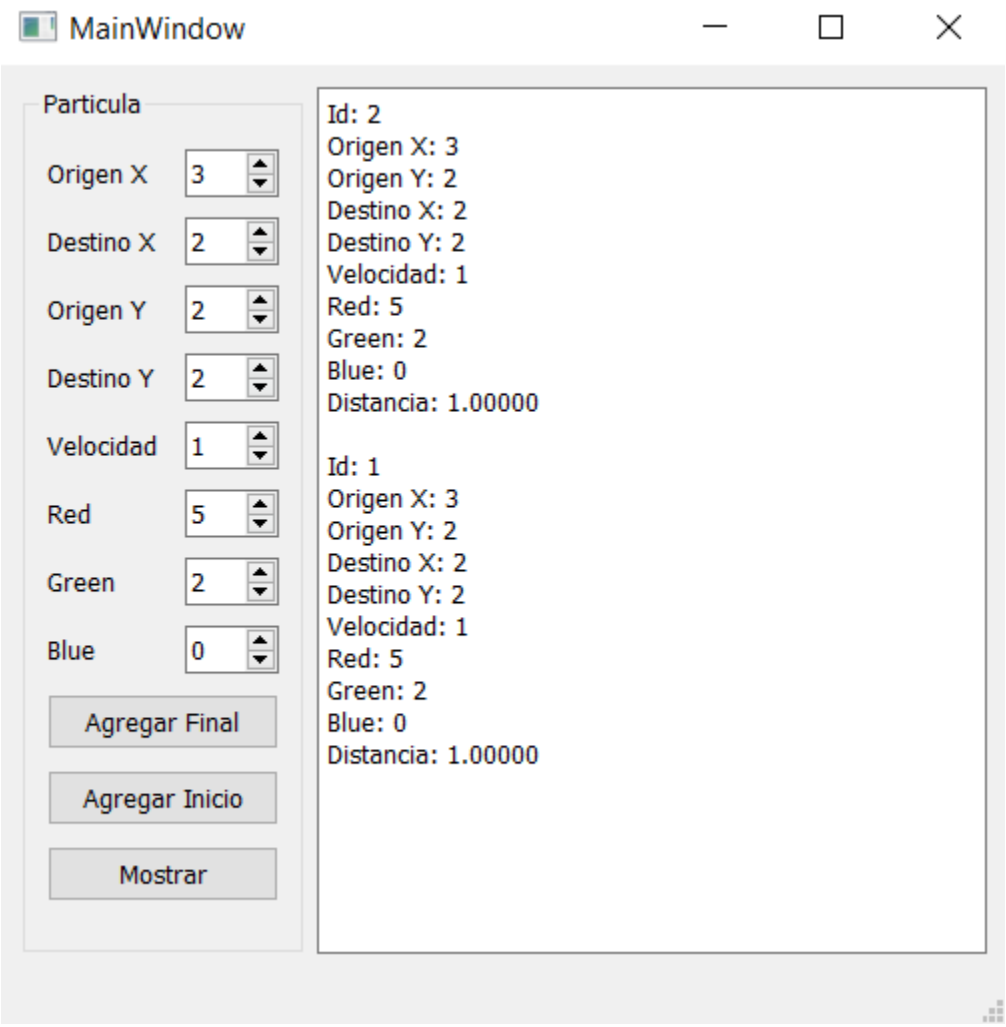
Seminario de Algoritmia

Lineamientos de evaluación

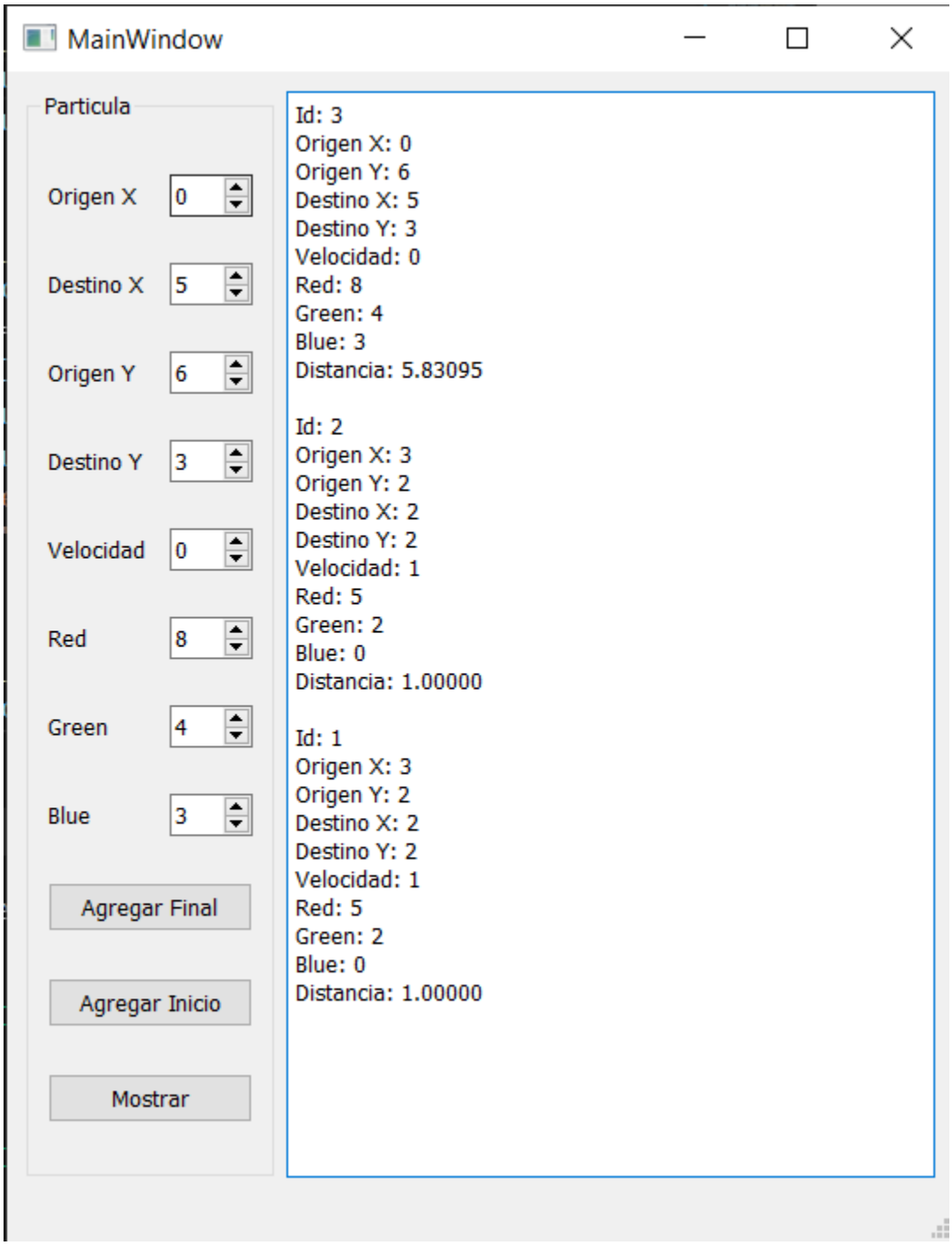
- El reporte está en formato Google Docs o PDF.
- El reporte sigue las pautas del Formato de Actividades .
- El reporte tiene desarrollada todas las pautas del Formato de Actividades.
- Se muestra la captura de pantalla de los datos antes de usar el botón para agregar_inicio() y la captura de pantalla del mostrar partículas en el QPlainTextEdit después de haber agregado la Particula.
- Se muestra la captura de pantalla de los datos antes de usar el botón para agregar_final() y la captura de pantalla del mostrar partículas en el QPlainTextEdit después de haber agregado la Particula.

Desarrollo

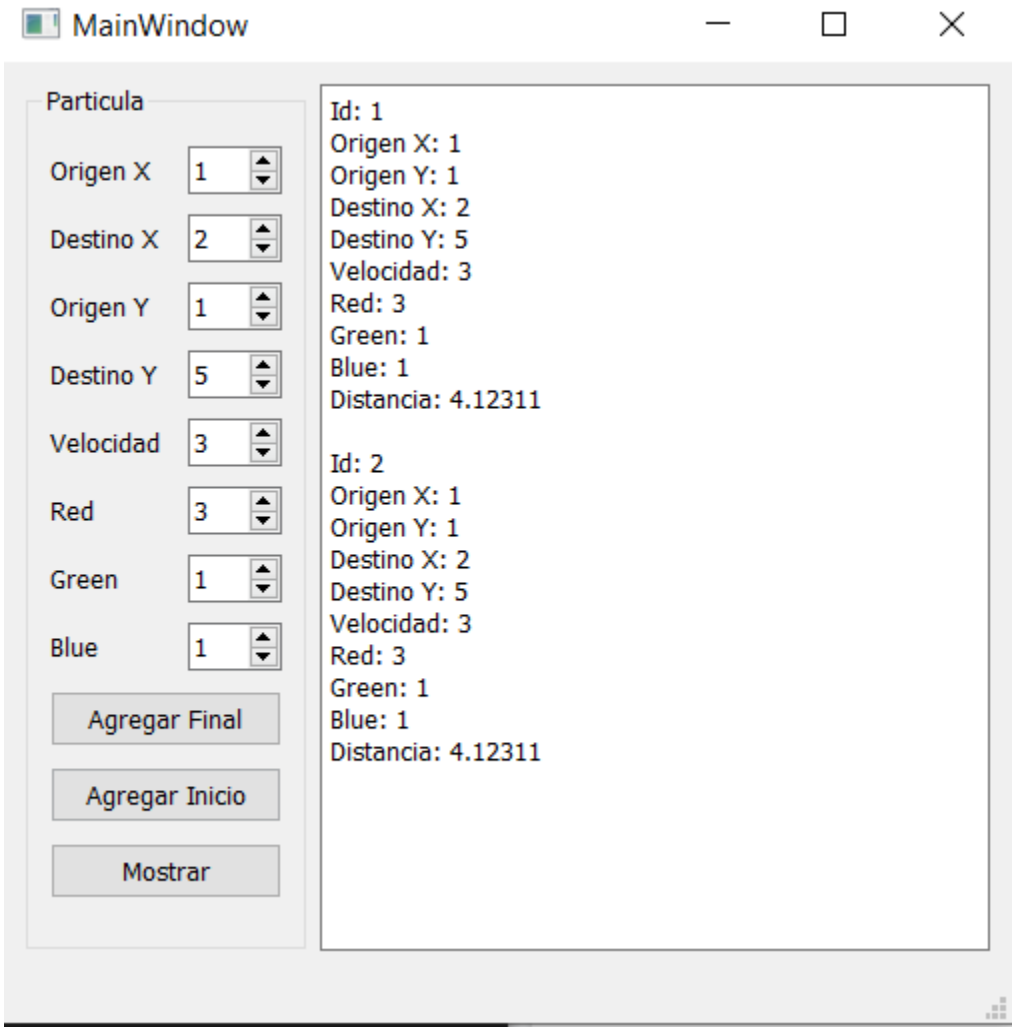
Primero creé la clase de partícula con sus atributos y los métodos requeridos, con el diseñador de interfaces después lo transformé a un archivo de python, por consiguiente creé un archivo para obtener los datos de cada botón y trabajar con ellos.



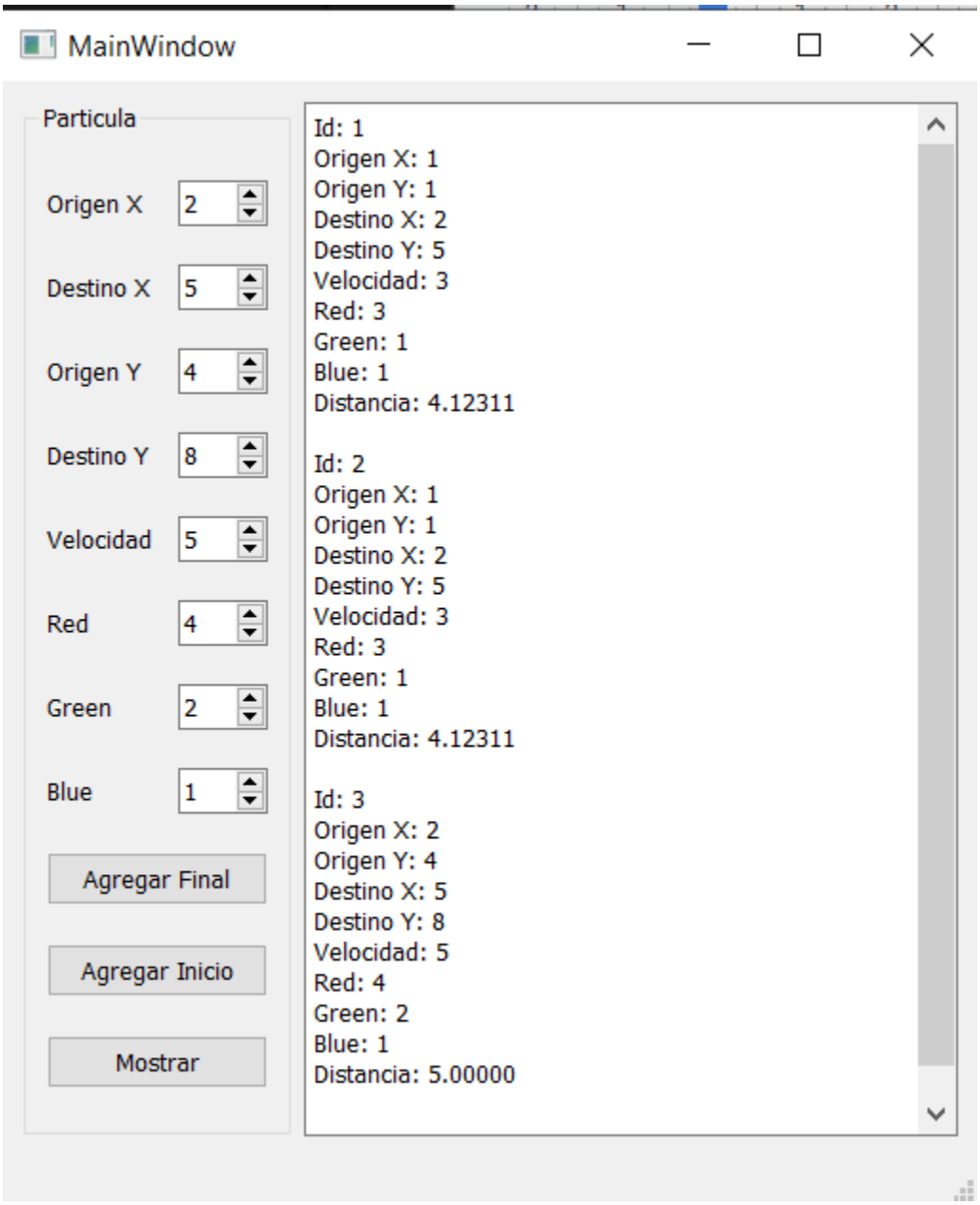
Captura de pantalla antes de usar el botón agregar inicio



Captura de pantalla después de usar el botón agregar inicio



Captura de pantalla antes de usar el el botón agregar final



Captura de pantalla despues de usar el botón agregar final

Conclusiones

Una actividad muy interesante en su totalidad, me gustó el uso de una interfaz y como

poco a poco va creciendo el nivel de las cosas que estamos haciendo, nos adentramos a un mundo de posibilidades una vez que sabemos usar las herramientas que nos da python para el uso de la programación orientada a objetos, espero que en un futuro podamos seguir trabajando con esto, creando cada vez más cosas teniendo como límite únicamente nuestra imaginación.

Referencias

undefined [MICHEL DAVALOS BOITES]. (2020, October 7). PySide2 - Introducción (Qt for Python)(I) [Video]. YouTube. Retrieved September 29, 2022, from https://www.youtube.com/watch?v=T0qJdF1fMqo&t=424s

Código

mainwindow.py

```
import imp

import re

from PySide2.QtWidgets import QMainWindow

from PySide2.QtCore import Slot

from ui_mainwindow import Ui_MainWindow

from particulasact.particula import Particula

from particulasact.index import Nodo, Lista_ligada



class MainWindow(QMainWindow):

    def __init__(self):

        super(MainWindow, self).__init__()

        self.lista_ligada = Lista_ligada()

        self.ui = Ui_MainWindow()

        self.ui.setupUi(self)

        self.ui.agregarFinal_pushButton.clicked.connect(

            self.click_agregarFinal)

        self.ui.agregarInicio_pushButton.clicked.connect(

            self.click_agregarInicio)

        self.ui.mostrar_pushButton.clicked.connect(self.click_mostrar)


    def creadorDeParticulas(self):

        destinoX = self.ui.destinoX_spinBox.value()

        origenX = self.ui.origenX_spinBox.value()

        destinoY = self.ui.destinoY_spinBox.value()
```

```
        origenY = self.ui.origenY_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()
        return Particula(self.lista_ligada.no_elements+1, origenX, origenY,
                         destinoX, destinoY, velocidad, red, green, blue)

    @Slot()
    def click_mostrar(self):
        self.ui.salida.clear()
        self.ui.salida.insertPlainText(str(self.lista_ligada))

    @Slot()
    def click_agregarFinal(self):
        particula = self.creadorDeParticulas()
        nodo = Nodo(particula)
        self.lista_ligada.agregar_final(nodo)
        self.ui.salida.clear()
        self.ui.salida.insertPlainText("Agregado al Final")
        """ self.ui.salida.insertPlainText(
            f"ID:{particula.id}\nOrigen X:{particula.origen_x}\nDestino X:
{particula.destino_x}\nOrigen Y:{particula.origen_y}\nDestino Y:
{particula.destino_y}\nVelocidad:
{particula.velocidad}\nDistancia:{particula.distancia}\nRed:
{particula.red}\nGreen: {particula.green}\nBlue: {particula.blue}")
 """
    @Slot()
    def click_agregarInicio(self):
```

```
        particula = self.creadorDeParticulas()
        nodo = Nodo(particula)
        self.lista_ligada.agregar_inicio(nodo)
        self.ui.salida.clear()
        self.ui.salida.insertPlainText("Agregado al Inicio")
```

index.py

```
from array import array
import string
from particulasact.particula import Particula



class Nodo():
    dato = None
    siguiente = None
    anterior = None

    def __init__(self, dato):
        self.dato = dato
        self.siguiente = None



class Lista_ligada():
    nodo_inicial = None
    nodo_final = None
    no_elements = 0
```

```
def __init__(self):
    self.nodo_inicial = None
    self.nodo_final = None

def agregar_inicio(self, nodo):
    if(self.no_elements == 0):
        self.nodo_inicial = nodo
        self.nodo_final = nodo
        self.no_elements = self.no_elements + 1
    else:
        temporal = self.nodo_inicial
        temporal.anterior = nodo
        nodo.siguiente = temporal
        self.nodo_inicial = nodo
        self.no_elements = self.no_elements + 1

def agregar_final(self, nodo):
    if(self.no_elements == 0):
        self.nodo_inicial = nodo
        self.nodo_final = nodo
        self.no_elements = self.no_elements + 1
    else:
        temporal = self.nodo_final
        temporal.siguiente = nodo
        nodo.anterior = temporal
        self.nodo_final = nodo
        self.no_elements = self.no_elements+1
```

```
def mostrar(self):
    temp = self.nodo_inicial
    while(temp):
        print(temp.dato)
        temp = temp.siguiente

def __str__(self):
    temp = self.nodo_inicial
    array = []
    while(temp):
        array.append(str(temp.dato))
        temp = temp.siguiente

    return "".join(array)
```