

ACTIVIDAD 06

Cruz Collazo Wendy Paola

Seminario de Solución de Problemas de Algoritmia.

Lineamientos de Evaluación:

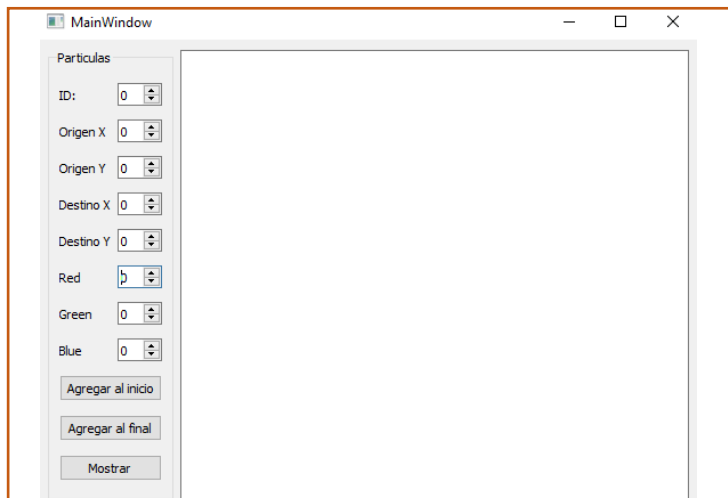
- ✓ El reporte esta en Formato Google Docs o PDF.
- ✓ El reporte sigue las pautas del Formato de Actividades.
- ✓ El reporte tiene desarrollada todas las pautas del Formato de Actividades.
- ✓ Se muestra la captura de pantalla de los datos antes de usar el botón para Agregar_Inicio() y la captura de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

▪ Desarrollo

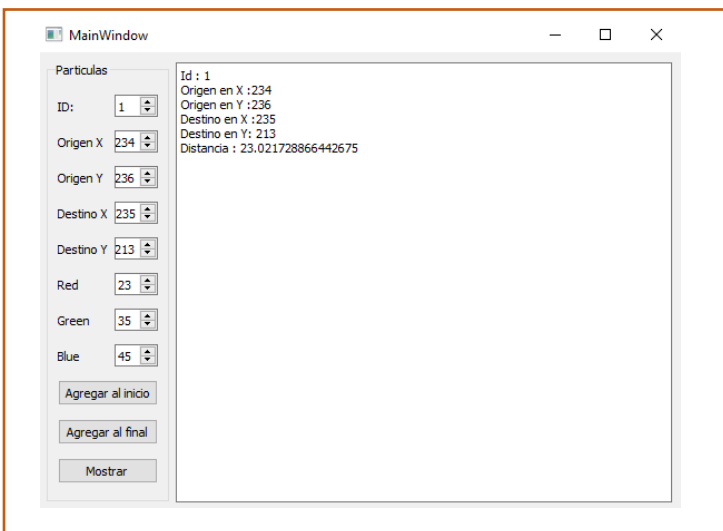
Comencé modificando la interfaz, agregándole 2 botones y un recuadro de Texto, modifiqué todos los botones de los spin box por el nombre de el label al que le correspondía.

Y en el visual studio comencé a adaptar el código y agregándole cosas para que funcionara con la iterfaz.

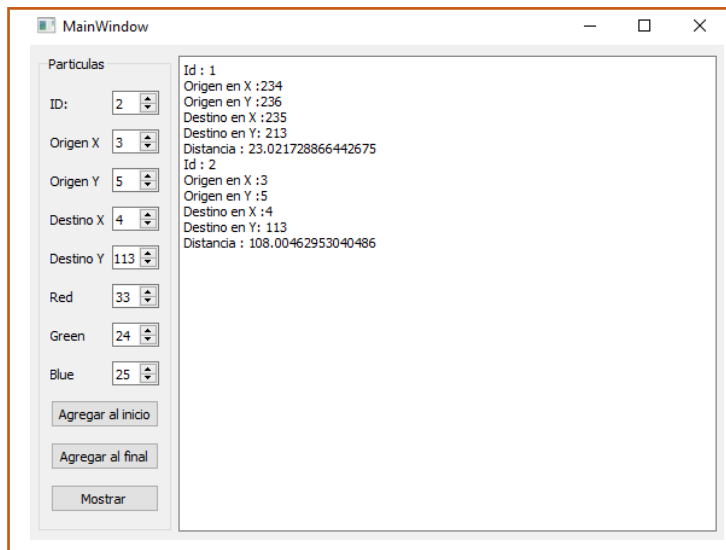
Los datos al Abrir la Interfaz:



Despues de Agregar un dato al Inicio:



Después de Agregar un dato al Final:



■ Conclusiones

En esta actividad si se me complico bastante ya que al momento de querer actualizar el mainwindow.ui me salían errores, y tuve que volver a crear otro que también tuvo errores ya que cuando tenia que importar la carpeta de Clases y Objetos no me funcionaba y pues la tuve que volver a hacer desde cero para que me funcionara.

■ Referencias Bibliográficas

✓ PySide2- QPlaitTextEdit (Qt for python)(III)

<https://www.youtube.com/watch?v=5TPKrKlAAU0>

- Algoritmos.py

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

- Administradora.py

```
from particula import Particula

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
        )
```

- **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')
```

- **MainWindow.py**

```
from PySide2.QtWidgets import QMainWindow
from ui_mainwindow import Ui_MainWindow
from administradora import Administradora
from particula import Particula
from PySide2.QtCore import Slot

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

        self.administrador = Administradora()

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

```

self.ui.Agregar_final.clicked.connect(self.agregar_final)
self.ui.Agregar_Inicio.clicked.connect(self.agregar_inicio)
self.ui.Mostrar.clicked.connect(self.ver)

@Slot()
def ver(self):
    self.ui.Print.clear()
    self.ui.Print.insertPlainText(str(self.administrador))

@Slot()
def agregar_final(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()
    Red = self.ui.Red_spinBox.value()
    Green = self.ui.Green_spinBox.value()
    Blue = self.ui.Blue_spinBox.value()

    particula1 = Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY)
    self.administrador.agregar_final(particula1)

@Slot()
def 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()
    Red = self.ui.Red_spinBox.value()
    Green = self.ui.Green_spinBox.value()
    Blue = self.ui.Blue_spinBox.value()

    particula1 = Particula(ID,OrigenX,OrigenY,DestinoX,DestinoY)
    self.administrador.agregar_inicio(particula1)

```