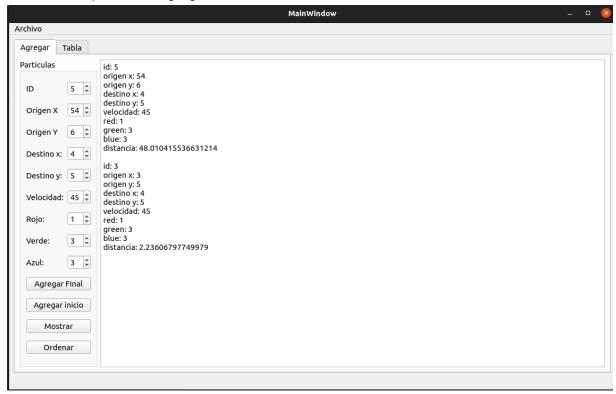
Actividad 10:Sort

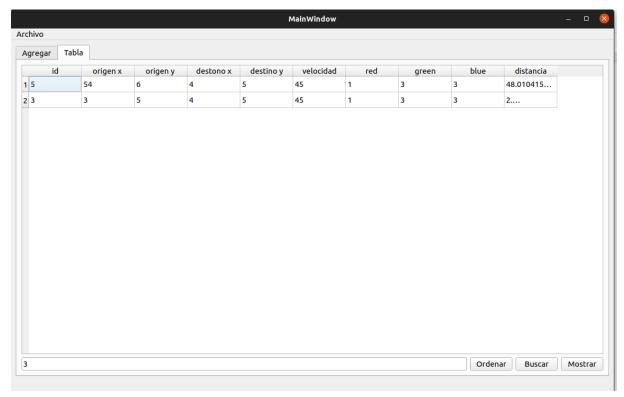
Ramirez Orduna Geovanni

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 captura de pantalla de las partículas del antes y después de ser ordenadas por id de manera ascendente tanto en el QPlainTextEdit como en el QTableWidget.
- [] Se muestra captura de pantalla de las partículas del antes y después de ser ordenadas por distancia de manera descendente tanto en el QPlainTextEdit como en el QTableWidget.
- [] Se muestra captura de pantalla de las partículas del antes y después de ser ordenadas por velocidad de manera ascendente tanto en el QPlainTextEdit como en el QTableWidget.

Tenemos las partículas agregadas





Tenemos los datos mostrados en la tabla

Conclusión:En esta práctica pude conocer cómo se realiza el método sort para asi poder ordenar de forma ascendente como descendente.

Referencias:

https://www.youtube.com/watch?v=0NZajLIy5qQ&t=1037s

```
from PySide2.QtWidgets import QMainWindow, QGraphicsScene
from ui mainwindow import Ui MainWindow
from PySide2.QtGui import QPen, QColor, QTransform
from PySide2.QtCore import Slot
from random import randint
class MainWindow(QMainWindow):
       super(MainWindow, self). init ()
       self.ui=Ui MainWindow()
       self.ui.setupUi(self)
       self.ui.Draw Button.clicked.connect(self.draw)
       self.scene=QGraphicsScene()
       self.ui.graphicsView.setScene(self.scene)
  def wheelEvent(self, event):
       if event.delta() >0:
           self.ui.graphicsView.scale(1.2,1.2)
           self.ui.graphicsView.scale(0.8,0.8)
  def draw(self):
      pen=QPen()
      pen.setWidth(2)
           r=randint(0,255)
          g=randint(0,255)
           b=randint(0,255)
           color=QColor(r,g,b)
           origen x=randint(0,500)
           origen y=randint(0,500)
           destino x=randint(0,500)
```

```
destino_y=randint(0,500)

    pen.setColor(color)
    self.scene.addEllipse(origen_x,origen_y,3,3,pen)
    self.scene.addEllipse(destino_x,destino_y,3,3,pen)
    self.scene.addLine(origen_x+3,

origen_y+3,destino_x,destino_y,pen)

@Slot()
def clean(self):
    self.scene.clear()
```

mainWindow

```
import PySide2
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox
from PySide2.QtCore import Slot
from ui mainwindow import Ui MainWindow
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)
self.ui.Agre Inicio Buttom.clicked.connect(self.click agregar inicio)
      self.ui.Mostrar buttom.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):
      ubicacion=QFileDialog.getOpenFileName(
          self,
       if self.admin.open(ubicacion):
           QMessageBox.information(
               self,
```

```
else:
            self,
@Slot()
def action guardar(self):
   ubicacion= QFileDialog.getSaveFileName(
   print( ubicacion)
   self.admin.save(ubicacion)
   if self.admin.save(ubicacion):
        QMessageBox.information(
            self,
        QMessageBox.critical(
            self,
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()
  def click mostrar(self):
           self.ui.salida.clear()
           self.ui.salida.insertPlainText(str(self.admin))
```