

# Actividad 10

## (Sort)

*Alcaraz Valdivia Marcos Fernando*

*Seminario de Solución de Problemas de Algoritmia*

### Lineamientos de evaluación

- El reporte está en formato Google Docs o PDF. (REALIZADO)
- El reporte sigue las pautas del Formato de Actividades. (REALIZADO)
- El reporte tiene desarrollada todas las pautas del Formato de Actividades. (REALIZADO)
- Se muestra captura de pantalla de las partículas del antes y después de ser ordenadas por id de manera ascendente tanto en el QTextEdit como en el QTableWidgetItem. (REALIZADO)
- Se muestra captura de pantalla de las partículas del antes y después de ser ordenadas por distancia de manera descendente tanto en el QTextEdit como en el QTableWidgetItem. (REALIZADO)
- Se muestra captura de pantalla de las partículas del antes y después de ser ordenadas por velocidad de manera ascendente tanto en el QTextEdit como en el QTableWidgetItem. (REALIZADO)

AAAAAAAAAAAA

*Satisfacción 10/10*

# Desarrollo

*Toma capturas de pantalla de las partículas del antes y el después de ser ordenadas por el **ID** de manera ascendente en el **QPlainTextEdit** y el **QTableWidget**:*

**QPlainText ANTES:**

MainWindow

Archivo

Agregar Partículas    Tabla de Partículas    Gráfico de Partículas

Partículas

Origen X ( 0-500 )    0

Origen Y ( 0-500 )    0

Destino X ( 0-500 )    0

Destino Y ( 0-500 )    0

Velocidad ( KM/h )    0

RED ( 0-255 )    0

GREEN ( 0-255 )    0

BLUE ( 0-255 )    0

Insertar al Inicio    Insertar al Final

Ordenar de forma Ascendente ID    Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

Partícula

ID : 5  
Origen X : 55  
Origen Y : 74  
Destino X : 98  
Destino Y : 54  
Velocidad: 452  
Red : 123  
Green : 45  
Blue : 123  
Distancia: 47.4236228055175

Partícula

ID : 3  
Origen X : 0  
Origen Y : 12  
Destino X : 55  
Destino Y : 78  
Velocidad: 452  
Red : 164  
Green : 200  
Blue : 164  
Distancia: 85.9127464349732

Partícula

ID : 2  
Origen X : 500  
Origen Y : 25  
Destino X : 345  
Destino Y : 465  
Velocidad: 525  
Red : 81  
Green : 72  
Blue : 81  
Distancia: 466.50294747193186

Partícula

ID : 1  
Origen X : 1  
Origen Y : 2  
Destino X : 3  
Destino Y : 4  
Velocidad: 5  
Red : 8  
Green : 7  
Blue : 8  
Distancia: 2.8284271247461903

Partícula

ID : 4  
Origen X : 55  
Origen Y : 444  
Destino X : 366  
Destino Y : 222  
Velocidad: 452  
Red : 111  
Green : 200  
Blue : 111  
Distancia: 382.1060062338722

# QTableWidget ANTES:

MainWindow

Archivo

Agregar Particulas

Tabla de Particulas

Gráfico de Particulas

	ID	Origen X	Origen Y	Destino X	Destino Y	Velocidad	Red	Green	Blue	Distancia
1	5	55	74	98	54	452	78	45	123	47.4236228055175
2	3	0	12	55	78	452	255	200	164	85.9127464349732
3	2	500	25	345	465	525	61	72	81	466.50294747193186
4	1	1	2	3	4	5	6	7	8	2.8284271247461903
5	4	55	444	366	222	452	44	200	111	382.1060062338722

Escribe un ID

Buscar

Ordenar de forma Ascendente ID

Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

## QPlainText DESPUÉS:

MainWindow

Archivo

Agregar Particulas    Tabla de Particulas    Gráfico de Particulas

Particulas

Origen X ( 0-500 )    0

Origen Y ( 0-500 )    0

Destino X ( 0-500 )    0

Destino Y ( 0-500 )    0

Velocidad ( KM/h )    0

RED ( 0-255 )    0

GREEN ( 0-255 )    0

BLUE ( 0-255 )    0

Insertar al Inicio    Insertar al Final

Ordenar de forma Ascendente ID    Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

Particula

ID : 1  
Origen X : 1  
Origen Y : 2  
Destino X: 3  
Destino Y: 4  
Velocidad: 5  
Red : 8  
Green : 7  
Blue : 8  
Distancia: 2.8284271247461903

Particula

ID : 2  
Origen X : 500  
Origen Y : 25  
Destino X: 345  
Destino Y: 465  
Velocidad: 525  
Red : 81  
Green : 72  
Blue : 81  
Distancia: 466.50294747193186

Particula

ID : 3  
Origen X : 0  
Origen Y : 12  
Destino X: 55  
Destino Y: 78  
Velocidad: 452  
Red : 164  
Green : 200  
Blue : 164  
Distancia: 85.9127464349732

Particula

ID : 4  
Origen X : 55  
Origen Y : 444  
Destino X: 366  
Destino Y: 222  
Velocidad: 452  
Red : 111  
Green : 200  
Blue : 111  
Distancia: 382.1060062338722

Particula

ID : 5  
Origen X : 55  
Origen Y : 74  
Destino X: 98  
Destino Y: 54  
Velocidad: 452  
Red : 123  
Green : 45  
Blue : 123  
Distancia: 47.4236228055175

## QTableWidget DESPUÉS:

MainWindow

Archivo

Agregar Partículas   Tabla de Partículas   Gráfico de Partículas

	ID	Origen X	Origen Y	Destino X	Destino Y	Velocidad	Red	Green	Blue	Distancia
1	1	1	2	3	4	5	6	7	8	2.8284271247461903
2	2	500	25	345	465	525	61	72	81	466.50294747193186
3	3	0	12	55	78	452	255	200	164	85.9127464349732
4	4	55	444	366	222	452	44	200	111	382.1060062338722
5	5	55	74	98	54	452	78	45	123	47.4236228055175

Escribe un ID

Buscar

Ordenar de forma Ascendente ID   Ordenar de forma Ascendente Velocidad   Ordenar de forma Decendente Distancia

**Toma capturas de pantalla de las partículas del antes y el después de ser ordenadas por la *Distancia* de manera descendente en el *QPlainTextEdit* y el *QTableWidget*:**

***QPlainText* ANTES:**

MainWindow

Archivo

Agregar Particulas    Tabla de Particulas    Gráfico de Particulas

Particulas

Origen X ( 0-500 )    0

Origen Y ( 0-500 )    0

Destino X ( 0-500 )    0

Destino Y ( 0-500 )    0

Velocidad ( KM/h )    0

RED ( 0-255 )    0

GREEN ( 0-255 )    0

BLUE ( 0-255 )    0

Insertar al Inicio    Insertar al Final

Ordenar de forma Ascendente ID    Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

Particula

ID : 5  
Origen X : 55  
Origen Y : 74  
Destino X: 98  
Destino Y: 54  
Velocidad: 452  
Red : 123  
Green : 45  
Blue : 123  
Distancia: 47.4236228055175

Particula

ID : 3  
Origen X : 0  
Origen Y : 12  
Destino X: 55  
Destino Y: 78  
Velocidad: 452  
Red : 164  
Green : 200  
Blue : 164  
Distancia: 85.9127464349732

Particula

ID : 2  
Origen X : 500  
Origen Y : 25  
Destino X: 345  
Destino Y: 465  
Velocidad: 525  
Red : 81  
Green : 72  
Blue : 81  
Distancia: 466.50294747193186

Particula

ID : 1  
Origen X : 1  
Origen Y : 2  
Destino X: 3  
Destino Y: 4  
Velocidad: 5  
Red : 8  
Green : 7  
Blue : 8  
Distancia: 2.8284271247461903

Particula

ID : 4  
Origen X : 55  
Origen Y : 444  
Destino X: 366  
Destino Y: 222  
Velocidad: 452  
Red : 111  
Green : 200  
Blue : 111  
Distancia: 382.1060062338722

# QTableWidget ANTES:

MainWindow

Archivo

Agregar Particulas

Tabla de Particulas

Gráfico de Particulas

	ID	Origen X	Origen Y	Destino X	Destino Y	Velocidad	Red	Green	Blue	Distancia
1	5	55	74	98	54	452	78	45	123	47.4236228055175
2	3	0	12	55	78	452	255	200	164	85.9127464349732
3	2	500	25	345	465	525	61	72	81	466.50294747193186
4	1	1	2	3	4	5	6	7	8	2.8284271247461903
5	4	55	444	366	222	452	44	200	111	382.1060062338722

Escribe un ID

Buscar

Ordenar de forma Ascendente ID

Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

QPlainText DESPUÉS:

MainWindow

Archivo

Agregar Particulas

Tabla de Particulas

Gráfico de Particulas

Particulas

Origen X ( 0-500 )

0

Origen Y ( 0-500 )

0

Destino X ( 0-500 )

0

Destino Y ( 0-500 )

0

Velocidad ( KM/h )

0

RED ( 0-255 )

0

GREEN ( 0-255 )

0

BLUE ( 0-255 )

0

Insertar al Inicio

Insertar al Final

Ordenar de forma Ascendente ID

Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

Particula

ID : 2  
Origen X : 500  
Origen Y : 25  
Destino X: 345  
Destino Y: 465  
Velocidad: 525  
Red : 81  
Green : 72  
Blue : 81  
Distancia: 466.50294747193186

Particula

ID : 4  
Origen X : 55  
Origen Y : 444  
Destino X: 366  
Destino Y: 222  
Velocidad: 452  
Red : 111  
Green : 200  
Blue : 111  
Distancia: 382.1060062338722

Particula

ID : 3  
Origen X : 0  
Origen Y : 12  
Destino X: 55  
Destino Y: 78  
Velocidad: 452  
Red : 164  
Green : 200  
Blue : 164  
Distancia: 85.9127464349732

Particula

ID : 5  
Origen X : 55  
Origen Y : 74  
Destino X: 98  
Destino Y: 54  
Velocidad: 452  
Red : 123  
Green : 45  
Blue : 123  
Distancia: 47.4236228055175

Particula

ID : 1  
Origen X : 1  
Origen Y : 2  
Destino X: 3  
Destino Y: 4  
Velocidad: 5  
Red : 8  
Green : 7  
Blue : 8  
Distancia: 2.8284271247461903



## QTableWidget DESPUÉS:

MainWindow

— □ ×

Archivo

Agregar Partículas

Tabla de Partículas

Gráfico de Partículas

	ID	Origen X	Origen Y	Destino X	Destino Y	Velocidad	Red	Green	Blue	Distancia
1	2	500	25	345	465	525	61	72	81	466.50294747193186
2	4	55	444	366	222	452	44	200	111	382.1060062338722
3	3	0	12	55	78	452	255	200	164	85.9127464349732
4	5	55	74	98	54	452	78	45	123	47.4236228055175
5	1	1	2	3	4	5	6	7	8	2.8284271247461903

Escribe un ID

Buscar

Ordenar de forma Ascendente ID

Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

*Toma capturas de pantalla de las partículas del antes y el después de ser ordenadas por la **Velocidad** de manera ascendente en el **QPlainTextEdit** y el **QTableWidget**:*

**QPlainText ANTES:**

MainWindow

Archivo

Agregar Particulas   Tabla de Particulas   Gráfico de Particulas

Particulas

Origen X ( 0-500 )   0

Origen Y ( 0-500 )   0

Destino X ( 0-500 )   0

Destino Y ( 0-500 )   0

Velocidad ( KM/h )   0

RED ( 0-255 )   0

GREEN ( 0-255 )   0

BLUE ( 0-255 )   0

Insertar al Inicio   Insertar al Final

Ordenar de forma Ascendente ID   Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

Particula

ID : 5  
Origen X : 55  
Origen Y : 74  
Destino X : 98  
Destino Y : 54  
Velocidad: 452  
Red : 123  
Green : 45  
Blue : 123  
Distancia: 47.4236228055175

Particula

ID : 3  
Origen X : 0  
Origen Y : 12  
Destino X : 55  
Destino Y : 78  
Velocidad: 452  
Red : 164  
Green : 200  
Blue : 164  
Distancia: 85.9127464349732

Particula

ID : 2  
Origen X : 500  
Origen Y : 25  
Destino X : 345  
Destino Y : 465  
Velocidad: 525  
Red : 81  
Green : 72  
Blue : 81  
Distancia: 466.50294747193186

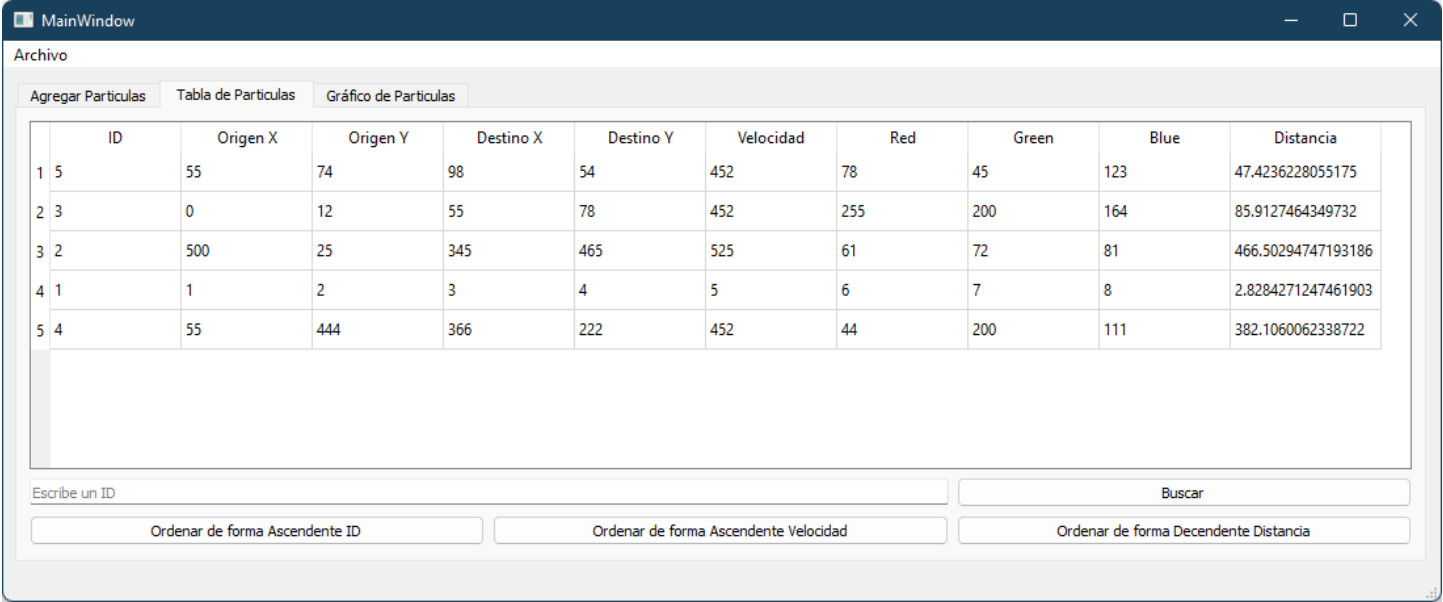
Particula

ID : 1  
Origen X : 1  
Origen Y : 2  
Destino X : 3  
Destino Y : 4  
Velocidad: 5  
Red : 8  
Green : 7  
Blue : 8  
Distancia: 2.8284271247461903

Particula

ID : 4  
Origen X : 55  
Origen Y : 444  
Destino X : 366  
Destino Y : 222  
Velocidad: 452  
Red : 111  
Green : 200  
Blue : 111  
Distancia: 382.1060062338722

# QTableWidget ANTES:



# QPlainText DESPUÉS:

MainWindow

Archivo

Agregar Particulas

Tabla de Particulas

Gráfico de Particulas

Particulas

Origen X ( 0-500 )

0

Origen Y ( 0-500 )

0

Destino X ( 0-500 )

0

Destino Y ( 0-500 )

0

Velocidad ( KM/h )

0

RED ( 0-255 )

0

GREEN ( 0-255 )

0

BLUE ( 0-255 )

0

Insertar al Inicio

Insertar al Final

Ordenar de forma Ascendente ID

Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

Particula

ID : 1  
Origen X : 1  
Origen Y : 2  
Destino X: 3  
Destino Y: 4  
Velocidad: 5  
Red : 8  
Green : 7  
Blue : 8  
Distancia: 2.8284271247461903

Particula

ID : 5  
Origen X : 55  
Origen Y : 74  
Destino X: 98  
Destino Y: 54  
Velocidad: 452  
Red : 123  
Green : 45  
Blue : 123  
Distancia: 47.4236228055175

Particula

ID : 3  
Origen X : 0  
Origen Y : 12  
Destino X: 55  
Destino Y: 78  
Velocidad: 452  
Red : 164  
Green : 200  
Blue : 164  
Distancia: 85.9127464349732

Particula

ID : 4  
Origen X : 55  
Origen Y : 444  
Destino X: 366  
Destino Y: 222  
Velocidad: 452  
Red : 111  
Green : 200  
Blue : 111  
Distancia: 382.1060062338722

Particula

ID : 2  
Origen X : 500  
Origen Y : 25  
Destino X: 345  
Destino Y: 465  
Velocidad: 525  
Red : 81  
Green : 72  
Blue : 81  
Distancia: 466.50294747193186

## QTableWidget DESPUÉS:

MainWindow

Archivo

Agregar Particulas

Tabla de Particulas

Gráfico de Particulas

	ID	Origen X	Origen Y	Destino X	Destino Y	Velocidad	Red	Green	Blue	Distancia
1	1	1	2	3	4	5	6	7	8	2.8284271247461903
2	5	55	74	98	54	452	78	45	123	47.4236228055175
3	3	0	12	55	78	452	255	200	164	85.9127464349732
4	4	55	444	366	222	452	44	200	111	382.1060062338722
5	2	500	25	345	465	525	61	72	81	466.50294747193186

Escribe un ID

Buscar

Ordenar de forma Ascendente ID

Ordenar de forma Ascendente Velocidad

Ordenar de forma Decendente Distancia

## **Conclusiones**

*Cada interacción con este programa es más divertido, porque me en esta ocasión me quedaron las ganas de ordenar de forma ascendente y descendente cada uno de los datos, pero bueno la tarea solo condicionaba el ID, Velocidad y Distancia.*

*(Detrás de PDFs estoy haciendo los Sort para cada dato)*

# ***Referencias***

Python - sort()

<https://youtu.be/0NZajLly5qQ>

# Código

## *mainwindow.py*

```
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox, QTableWidgetItem,
QGraphicsScene
from PySide2.QtCore import Slot
from PySide2.QtGui import QPen, QColor, QTransform
from ui_mainwindow import Ui_MainWindow
from manager import Manager
from particula import Particula

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.manager = Manager()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.id = 0

        # Cuando el botón pushbutton es presionado, ejecuta la función click_agregar
        # self.ui.mostrar.clicked.connect(self.click_mostrar)
        self.ui.insertar_inicio.clicked.connect(self.click_insertar_inicio)
        self.ui.insertar_final.clicked.connect(self.click_insertar_final)

        self.ui.OrdenarAscendenteID_pushbutton.clicked.connect(self.accionMostrarTablaSBID)
        self.ui.OrdenarAscendenteVelocidad_pushbutton.clicked.connect(self.accionMostrarTabla
SBS)
        self.ui.OrdenarDescendenteDistancia_pushbutton.clicked.connect(self.accionMostrarTabl
aSBD)

        self.ui.OrdenarAscendenteID_pushbutton_2.clicked.connect(self.accionMostrarTablaSBID)
        self.ui.OrdenarAscendenteVelocidad_pushbutton_2.clicked.connect(self.accionMostrarTab
laSBS)
        self.ui.OrdenarDescendenteDistancia_pushbutton_2.clicked.connect(self.accionMostrarTa
blaSBD)

        self.ui.accionAbrir.triggered.connect(self.accionAbrirArchivo)
        self.ui.accionGuardar.triggered.connect(self.accionGuardarArchivo)

        self.ui.Buscar_pushbutton.clicked.connect(self.accionBuscar)

        self.scene = QGraphicsScene()
        self.ui.graphicsView.setScene(self.scene)

    @Slot()
    def dibujar(self):
        pen = QPen()
```



```

for particula in self.manager:
    pen.setWidth(2)
    color = QColor(particula.red, particula.green, particula.blue)
    pen.setColor(color)
    self.scene.addEllipse(particula.origenX, particula.origenY, 5, 5, pen)
    self.scene.addEllipse(particula.destinoX, particula.destinoY, 5, 5, pen)
    self.scene.addLine(particula.origenX, particula.origenY, particula.destinoX,
particula.destinoY, pen)

```

@Slot()

```

def accionMostrarTabla(self):
    self.ui.tabla.setColumnCount(10)
    headers = ["ID", "Origen X", "Origen Y", "Destino X",
               "Destino Y", "Velocidad", "Red", "Green", "Blue", "Distancia"]
    self.ui.tabla.setHorizontalHeaderLabels(headers)

```

```

self.ui.tabla.setRowCount(len(self.manager))

```

```

fila = 0

```

```

for particula in self.manager:
    id_widget = QTableWidgetItem(str(particula.id))
    origenX_widget = QTableWidgetItem(str(particula.origenX))
    origenY_widget = QTableWidgetItem(str(particula.origenY))
    destinoX_widget = QTableWidgetItem(str(particula.destinoX))
    destinoY_widget = QTableWidgetItem(str(particula.destinoY))
    velocidad_widget = QTableWidgetItem(str(particula.velocidad))
    red_widget = QTableWidgetItem(str(particula.red))
    green_widget = QTableWidgetItem(str(particula.green))
    blue_widget = QTableWidgetItem(str(particula.blue))
    distancia_widget = QTableWidgetItem(str(particula.distancia))

```

```

self.ui.tabla.setItem(fila, 0, id_widget)
self.ui.tabla.setItem(fila, 1, origenX_widget)
self.ui.tabla.setItem(fila, 2, origenY_widget)
self.ui.tabla.setItem(fila, 3, destinoX_widget)
self.ui.tabla.setItem(fila, 4, destinoY_widget)
self.ui.tabla.setItem(fila, 5, velocidad_widget)
self.ui.tabla.setItem(fila, 6, red_widget)
self.ui.tabla.setItem(fila, 7, green_widget)
self.ui.tabla.setItem(fila, 8, blue_widget)
self.ui.tabla.setItem(fila, 9, distancia_widget)

```

```

fila += 1

```

@Slot()

```

def accionMostrarTablaSBID(self):
    self.ui.tabla.setColumnCount(10)
    headers = ["ID", "Origen X", "Origen Y", "Destino X",

```

```

        "Destino Y", "Velocidad", "Red", "Green", "Blue", "Distancia"]
self.ui.tabla.setHorizontalHeaderLabels(headers)

self.ui.tabla.setRowCount(len(self.manager))

fila = 0
array = []

for particula in self.manager:
    array.append(particula)

array.sort(key=lambda particula: particula.id, reverse=False)

self.ui.lista_particulas.clear()
for particula in array:
    self.ui.lista_particulas.insertPlainText(str(particula))

for particula in array:
    id_widget = QTableWidgetItem(str(particula.id))
    origenX_widget = QTableWidgetItem(str(particula.origenX))
    origenY_widget = QTableWidgetItem(str(particula.origenY))
    destinoX_widget = QTableWidgetItem(str(particula.destinoX))
    destinoY_widget = QTableWidgetItem(str(particula.destinoY))
    velocidad_widget = QTableWidgetItem(str(particula.velocidad))
    red_widget = QTableWidgetItem(str(particula.red))
    green_widget = QTableWidgetItem(str(particula.green))
    blue_widget = QTableWidgetItem(str(particula.blue))
    distancia_widget = QTableWidgetItem(str(particula.distancia))

    self.ui.tabla.setItem(fila, 0, id_widget)
    self.ui.tabla.setItem(fila, 1, origenX_widget)
    self.ui.tabla.setItem(fila, 2, origenY_widget)
    self.ui.tabla.setItem(fila, 3, destinoX_widget)
    self.ui.tabla.setItem(fila, 4, destinoY_widget)
    self.ui.tabla.setItem(fila, 5, velocidad_widget)
    self.ui.tabla.setItem(fila, 6, red_widget)
    self.ui.tabla.setItem(fila, 7, green_widget)
    self.ui.tabla.setItem(fila, 8, blue_widget)
    self.ui.tabla.setItem(fila, 9, distancia_widget)

    fila += 1

@Slot()
def accionMostrarTablaSBS(self):
    self.ui.tabla.setColumnCount(10)
    headers = ["ID", "Origen X", "Origen Y", "Destino X",
               "Destino Y", "Velocidad", "Red", "Green", "Blue", "Distancia"]
    self.ui.tabla.setHorizontalHeaderLabels(headers)

    self.ui.tabla.setRowCount(len(self.manager))

```

```

fila = 0
array = []

for particula in self.manager:
    array.append(particula)

array.sort(key=lambda particula: particula.velocidad, reverse=False)

self.ui.lista_particulas.clear()
for particula in array:
    self.ui.lista_particulas.insertPlainText(str(particula))

for particula in array:
    id_widget = QTableWidgetItem(str(particula.id))
    origenX_widget = QTableWidgetItem(str(particula.origenX))
    origenY_widget = QTableWidgetItem(str(particula.origenY))
    destinoX_widget = QTableWidgetItem(str(particula.destinoX))
    destinoY_widget = QTableWidgetItem(str(particula.destinoY))
    velocidad_widget = QTableWidgetItem(str(particula.velocidad))
    red_widget = QTableWidgetItem(str(particula.red))
    green_widget = QTableWidgetItem(str(particula.green))
    blue_widget = QTableWidgetItem(str(particula.blue))
    distancia_widget = QTableWidgetItem(str(particula.distancia))

    self.ui.tabla.setItem(fila, 0, id_widget)
    self.ui.tabla.setItem(fila, 1, origenX_widget)
    self.ui.tabla.setItem(fila, 2, origenY_widget)
    self.ui.tabla.setItem(fila, 3, destinoX_widget)
    self.ui.tabla.setItem(fila, 4, destinoY_widget)
    self.ui.tabla.setItem(fila, 5, velocidad_widget)
    self.ui.tabla.setItem(fila, 6, red_widget)
    self.ui.tabla.setItem(fila, 7, green_widget)
    self.ui.tabla.setItem(fila, 8, blue_widget)
    self.ui.tabla.setItem(fila, 9, distancia_widget)

    fila += 1

@Slot()
def accionMostrarTablaSBD(self):
    self.ui.tabla.setColumnCount(10)
    headers = ["ID", "Origen X", "Origen Y", "Destino X",
               "Destino Y", "Velocidad", "Red", "Green", "Blue", "Distancia"]
    self.ui.tabla.setHorizontalHeaderLabels(headers)

    self.ui.tabla.setRowCount(len(self.manager))

    fila = 0
    array = []

```

```

for particula in self.manager:
    array.append(particula)

array.sort(key=lambda particula: particula.distancia, reverse=True)

self.ui.lista_particulas.clear()
for particula in array:
    self.ui.lista_particulas.insertPlainText(str(particula))

for particula in array:
    id_widget = QTableWidgetItem(str(particula.id))
    origenX_widget = QTableWidgetItem(str(particula.origenX))
    origenY_widget = QTableWidgetItem(str(particula.origenY))
    destinoX_widget = QTableWidgetItem(str(particula.destinoX))
    destinoY_widget = QTableWidgetItem(str(particula.destinoY))
    velocidad_widget = QTableWidgetItem(str(particula.velocidad))
    red_widget = QTableWidgetItem(str(particula.red))
    green_widget = QTableWidgetItem(str(particula.green))
    blue_widget = QTableWidgetItem(str(particula.blue))
    distancia_widget = QTableWidgetItem(str(particula.distancia))

    self.ui.tabla.setItem(fila, 0, id_widget)
    self.ui.tabla.setItem(fila, 1, origenX_widget)
    self.ui.tabla.setItem(fila, 2, origenY_widget)
    self.ui.tabla.setItem(fila, 3, destinoX_widget)
    self.ui.tabla.setItem(fila, 4, destinoY_widget)
    self.ui.tabla.setItem(fila, 5, velocidad_widget)
    self.ui.tabla.setItem(fila, 6, red_widget)
    self.ui.tabla.setItem(fila, 7, green_widget)
    self.ui.tabla.setItem(fila, 8, blue_widget)
    self.ui.tabla.setItem(fila, 9, distancia_widget)

    fila += 1

@Slot()
def accionBuscar(self):
    id = self.ui.buscar_lineedit.text()
    bandera = False

    for particula in self.manager:
        if str(id) == str(particula.id):
            self.ui.tabla.clear()
            self.ui.tabla.setRowCount(1)

            id_widget = QTableWidgetItem(str(particula.id))
            origenX_widget = QTableWidgetItem(str(particula.origenX))
            origenY_widget = QTableWidgetItem(str(particula.origenY))
            destinoX_widget = QTableWidgetItem(str(particula.destinoX))
            destinoY_widget = QTableWidgetItem(str(particula.destinoY))
            velocidad_widget = QTableWidgetItem(str(particula.velocidad))

```

```

        red_widget = QTableWidgetItem(str(particula.red))
        green_widget = QTableWidgetItem(str(particula.green))
        blue_widget = QTableWidgetItem(str(particula.blue))
        distancia_widget = QTableWidgetItem(str(particula.distancia))

        self.ui.tabla.setItem(0, 0, id_widget)
        self.ui.tabla.setItem(0, 1, origenX_widget)
        self.ui.tabla.setItem(0, 2, origenY_widget)
        self.ui.tabla.setItem(0, 3, destinoX_widget)
        self.ui.tabla.setItem(0, 4, destinoY_widget)
        self.ui.tabla.setItem(0, 5, velocidad_widget)
        self.ui.tabla.setItem(0, 6, red_widget)
        self.ui.tabla.setItem(0, 7, green_widget)
        self.ui.tabla.setItem(0, 8, blue_widget)
        self.ui.tabla.setItem(0, 9, distancia_widget)

        bandera = True

        return

    if not bandera:
        QMessageBox.warning(self, "Particula no encontrada", f'La particula con el ID "{id}" no fue encontrada')

    @Slot()
    def accionAbrirArchivo(self):
        ubicacion = QFileDialog.getOpenFileName(
            self,
            "Abrir archivo",
            ".",
            "JSON (*.json)"
        )[0]
        if self.manager.abrir(ubicacion):
            self.click_mostrar()
            self.accionMostrarTabla()
            self.dibujar()
            QMessageBox.information(
                self, "Abrir archivo", "Archivo abierto Exitosamente : " + ubicacion)
        else:
            QMessageBox.critical(
                self, "Error", "No se puede abrir el archivo : " + ubicacion)

    @Slot()
    def accionGuardarArchivo(self):
        ubicacion = QFileDialog.getSaveFileName(
            self,
            "Guardar Archivo",
            ".",
            "JSON (*.json)"
        )[0]
        if self.manager.guardar(ubicacion):

```

```

        QMessageBox.information(
            self, "Archivo Guardado", "Guardado Exitoso : " + ubicacion)
    else:
        QMessageBox.critical(
            self, "Error", "Archivo no Guardado : " + ubicacion)

@Slot()
def click_insertar_inicio(self):
    self.id += 1
    aux = Particula(self.id, self.ui.ox.value(), self.ui.oy.value(), self.ui.dx.value(),
self.ui.dy.value(
    ), self.ui.velocidad.value(), self.ui.red.value(), self.ui.green.value(),
self.ui.blue.value())
    self.manager.agregarInicio(aux)
    self.click_mostrar()
    self.accionMostrarTabla()
    self.dibujar()

@Slot()
def click_insertar_final(self):
    self.id += 1
    aux = Particula(self.id, self.ui.ox.value(), self.ui.oy.value(), self.ui.dx.value(),
self.ui.dy.value(
    ), self.ui.velocidad.value(), self.ui.red.value(), self.ui.green.value(),
self.ui.blue.value())
    self.manager.agregarFinal(aux)
    self.click_mostrar()

@Slot()
def click_mostrar(self):
    self.ui.lista_particulas.clear()
    self.ui.lista_particulas.insertPlainText(str(self.manager))

```

# *ui\_mainwindow.py*

```
# -*- coding: utf-8 -*-

#####
## Form generated from reading UI file 'mainwindow.ui'
##
## Created by: Qt User Interface Compiler version 5.15.2
##
## WARNING! All changes made in this file will be lost when recompiling UI file!
#####

from PySide2.QtCore import *
from PySide2.QtGui import *
from PySide2.QtWidgets import *

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        if not MainWindow.setObjectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.resize(1093, 797)
        self.accionGuardar = QAction(MainWindow)
        self.accionGuardar.setObjectName(u"accionGuardar")
        self.accionAbrir = QAction(MainWindow)
        self.accionAbrir.setObjectName(u"accionAbrir")
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.gridLayout_4 = QGridLayout(self.centralwidget)
        self.gridLayout_4.setObjectName(u"gridLayout_4")
        self.tabWidget = QTabWidget(self.centralwidget)
        self.tabWidget.setObjectName(u"tabWidget")
        self.tab = QWidget()
        self.tab.setObjectName(u"tab")
        self.gridLayout_2 = QGridLayout(self.tab)
        self.gridLayout_2.setObjectName(u"gridLayout_2")
        self.lista_particulas = QLineEdit(self.tab)
        self.lista_particulas.setObjectName(u"lista_particulas")

        self.gridLayout_2.addWidget(self.lista_particulas, 0, 3, 1, 1)

        self.groupBox = QGroupBox(self.tab)
        self.groupBox.setObjectName(u"groupBox")
        self.gridLayout = QGridLayout(self.groupBox)
        self.gridLayout.setObjectName(u"gridLayout")
        self.label_3 = QLabel(self.groupBox)
        self.label_3.setObjectName(u"label_3")

        self.gridLayout.addWidget(self.label_3, 4, 0, 1, 1)
```

```

self.dx = QSpinBox(self.groupBox)
self.dx.setObjectName(u"dx")
self.dx.setMaximum(500)

self.gridLayout.addWidget(self.dx, 2, 1, 1, 1)

self.oy = QSpinBox(self.groupBox)
self.oy.setObjectName(u"oy")
self.oy.setMaximum(500)

self.gridLayout.addWidget(self.oy, 1, 1, 1, 1)

self.label_5 = QLabel(self.groupBox)
self.label_5.setObjectName(u"label_5")

self.gridLayout.addWidget(self.label_5, 6, 0, 1, 1)

self.blue = QSpinBox(self.groupBox)
self.blue.setObjectName(u"blue")
self.blue.setMaximum(255)

self.gridLayout.addWidget(self.blue, 7, 1, 1, 1)

self.insertar_inicio = QPushButton(self.groupBox)
self.insertar_inicio.setObjectName(u"insertar_inicio")

self.gridLayout.addWidget(self.insertar_inicio, 8, 0, 1, 1)

self.label = QLabel(self.groupBox)
self.label.setObjectName(u"label")

self.gridLayout.addWidget(self.label, 2, 0, 1, 1)

self.label_7 = QLabel(self.groupBox)
self.label_7.setObjectName(u"label_7")

self.gridLayout.addWidget(self.label_7, 1, 0, 1, 1)

self.velocidad = QSpinBox(self.groupBox)
self.velocidad.setObjectName(u"velocidad")
self.velocidad.setMaximum(999)

self.gridLayout.addWidget(self.velocidad, 4, 1, 1, 1)

self.OrdenarAscendenteID_pushbutton_2 = QPushButton(self.groupBox)
self.OrdenarAscendenteID_pushbutton_2.setObjectName(u"OrdenarAscendenteID_pushbutton_2")

self.gridLayout.addWidget(self.OrdenarAscendenteID_pushbutton_2, 9, 0, 1, 1)

```

2")



```
self.green = QSpinBox(self.groupBox)
self.green.setObjectName(u"green")
self.green.setMaximum(255)

self.gridLayout.addWidget(self.green, 6, 1, 1, 1)

self.red = QSpinBox(self.groupBox)
self.red.setObjectName(u"red")
self.red.setMaximum(255)

self.gridLayout.addWidget(self.red, 5, 1, 1, 1)

self.ox = QSpinBox(self.groupBox)
self.ox.setObjectName(u"ox")
self.ox.setMaximum(500)

self.gridLayout.addWidget(self.ox, 0, 1, 1, 1)

self.OrdenarAscendenteVelocidad_pushbutton_2 = QPushButton(self.groupBox)
self.OrdenarAscendenteVelocidad_pushbutton_2.setObjectName(u"OrdenarAscendenteVelocidad_pushbutton_2")

self.gridLayout.addWidget(self.OrdenarAscendenteVelocidad_pushbutton_2, 9, 1, 1, 1)

self.label_8 = QLabel(self.groupBox)
self.label_8.setObjectName(u"label_8")

self.gridLayout.addWidget(self.label_8, 0, 0, 1, 1)

self.label_2 = QLabel(self.groupBox)
self.label_2.setObjectName(u"label_2")

self.gridLayout.addWidget(self.label_2, 3, 0, 1, 1)

self.insertar_final = QPushButton(self.groupBox)
self.insertar_final.setObjectName(u"insertar_final")

self.gridLayout.addWidget(self.insertar_final, 8, 1, 1, 1)

self.dy = QSpinBox(self.groupBox)
self.dy.setObjectName(u"dy")
self.dy.setMaximum(500)

self.gridLayout.addWidget(self.dy, 3, 1, 1, 1)

self.label_6 = QLabel(self.groupBox)
self.label_6.setObjectName(u"label_6")

self.gridLayout.addWidget(self.label_6, 7, 0, 1, 1)
```

```

self.label_4 = QLabel(self.groupBox)
self.label_4.setObjectName(u"label_4")

self.gridLayout.addWidget(self.label_4, 5, 0, 1, 1)

self.OrdenarDescendenteDistancia_pushbutton_2 = QPushButton(self.groupBox)
self.OrdenarDescendenteDistancia_pushbutton_2.setObjectName(u"OrdenarDescendenteDistancia_pushbutton_2")

self.gridLayout.addWidget(self.OrdenarDescendenteDistancia_pushbutton_2, 10, 0, 1, 2)

self.gridLayout_2.addWidget(self.groupBox, 0, 0, 1, 1)

self.tabWidget.addTab(self.tab, "")
self.tab_2 = QWidget()
self.tab_2.setObjectName(u"tab_2")
self.gridLayout_3 = QGridLayout(self.tab_2)
self.gridLayout_3.setObjectName(u"gridLayout_3")
self.buscar_lineedit = QLineEdit(self.tab_2)
self.buscar_lineedit.setObjectName(u"buscar_lineedit")

self.gridLayout_3.addWidget(self.buscar_lineedit, 2, 0, 1, 2)

self.OrdenarAscendenteVelocidad_pushbutton = QPushButton(self.tab_2)
self.OrdenarAscendenteVelocidad_pushbutton.setObjectName(u"OrdenarAscendenteVelocidad_pushbutton")

self.gridLayout_3.addWidget(self.OrdenarAscendenteVelocidad_pushbutton, 4, 1, 1, 1)

self.OrdenarAscendenteID_pushbutton = QPushButton(self.tab_2)
self.OrdenarAscendenteID_pushbutton.setObjectName(u"OrdenarAscendenteID_pushbutton")

self.gridLayout_3.addWidget(self.OrdenarAscendenteID_pushbutton, 4, 0, 1, 1)

self.tabla = QTableWidget(self.tab_2)
self.tabla.setObjectName(u"tabla")

self.gridLayout_3.addWidget(self.tabla, 0, 0, 1, 3)

self.Buscar_pushbutton = QPushButton(self.tab_2)
self.Buscar_pushbutton.setObjectName(u"Buscar_pushbutton")

self.gridLayout_3.addWidget(self.Buscar_pushbutton, 2, 2, 1, 1)

self.OrdenarDescendenteDistancia_pushbutton = QPushButton(self.tab_2)
self.OrdenarDescendenteDistancia_pushbutton.setObjectName(u"OrdenarDescendenteDistancia_pushbutton")

self.gridLayout_3.addWidget(self.OrdenarDescendenteDistancia_pushbutton, 4, 2, 1, 1)

```

```

self.tabWidget.addTab(self.tab_2, "")
self.tab_3 = QWidget()
self.tab_3.setObjectName(u"tab_3")
self.graphicsView = QGraphicsView(self.tab_3)
self.graphicsView.setObjectName(u"graphicsView")
self.graphicsView.setGeometry(QRect(10, 10, 1051, 701))
self.tabWidget.addTab(self.tab_3, "")

```

```

self.gridLayout_4.addWidget(self.tabWidget, 1, 0, 1, 1)

```

```

MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QMenuBar(MainWindow)
self.menubar.setObjectName(u"menubar")
self.menubar.setGeometry(QRect(0, 0, 1093, 21))
self.menuArchivo = QMenu(self.menubar)
self.menuArchivo.setObjectName(u"menuArchivo")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QStatusBar(MainWindow)
self.statusbar.setObjectName(u"statusbar")
MainWindow.setStatusBar(self.statusbar)

```

```

self.menubar.addAction(self.menuArchivo.menuAction())
self.menuArchivo.addAction(self.accionAbrir)
self.menuArchivo.addAction(self.accionGuardar)

```

```

self.retranslateUi(MainWindow)

```

```

self.tabWidget.setCurrentIndex(1)

```

```

QMetaObject.connectSlotsByName(MainWindow)

```

```

# setupUi

```

```

def retranslateUi(self, MainWindow):
    MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow", u"MainWindow",
None))
    self.accionGuardar.setText(QCoreApplication.translate("MainWindow", u"Guardar",
None))
    #if QT_CONFIG(shortcut)
        self.accionGuardar.setShortcut(QCoreApplication.translate("MainWindow", u"Ctrl+S",
None))
    #endif // QT_CONFIG(shortcut)
    self.accionAbrir.setText(QCoreApplication.translate("MainWindow", u"Abrir", None))
    #if QT_CONFIG(shortcut)
        self.accionAbrir.setShortcut(QCoreApplication.translate("MainWindow", u"Ctrl+L",
None))
    #endif // QT_CONFIG(shortcut)
    self.groupBox.setTitle(QCoreApplication.translate("MainWindow", u"Particulas", None))

```

```

        self.label_3.setText(QCoreApplication.translate("MainWindow", u"Velocidad ( KM/h )",
None))
        self.label_5.setText(QCoreApplication.translate("MainWindow", u"GREEN ( 0-255 )",
None))
        self.insertar_inicio.setText(QCoreApplication.translate("MainWindow", u"Insertar al
Inicio", None))
        self.label.setText(QCoreApplication.translate("MainWindow", u"Destino X ( 0-500 )",
None))
        self.label_7.setText(QCoreApplication.translate("MainWindow", u"Origen Y ( 0-500 )",
None))
        self.OrdenarAscendenteID_pushbutton_2.setText(QCoreApplication.translate("MainWindow"
, u"Ordenar de forma Ascendente ID", None))
        self.OrdenarAscendenteVelocidad_pushbutton_2.setText(QCoreApplication.translate("Main
Window", u"Ordenar de forma Ascendente Velocidad", None))
        self.label_8.setText(QCoreApplication.translate("MainWindow", u"Origen X ( 0-500 )",
None))
        self.label_2.setText(QCoreApplication.translate("MainWindow", u"Destino Y ( 0-500
)", None))
        self.insertar_final.setText(QCoreApplication.translate("MainWindow", u"Insertar al
Final", None))
        self.label_6.setText(QCoreApplication.translate("MainWindow", u"BLUE ( 0-255 )",
None))
        self.label_4.setText(QCoreApplication.translate("MainWindow", u"RED ( 0-255 )",
None))
        self.OrdenarDescendenteDistancia_pushbutton_2.setText(QCoreApplication.translate("Mai
nWindow", u"Ordenar de forma Decendente Distancia", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab),
QCoreApplication.translate("MainWindow", u"Agregar Particulas", None))
        self.buscar_lineedit.setPlaceholderText(QCoreApplication.translate("MainWindow",
u"Escribe un ID", None))
        self.OrdenarAscendenteVelocidad_pushbutton.setText(QCoreApplication.translate("MainWi
ndow", u"Ordenar de forma Ascendente Velocidad", None))
        self.OrdenarAscendenteID_pushbutton.setText(QCoreApplication.translate("MainWindow",
u"Ordenar de forma Ascendente ID", None))
        self.Buscar_pushbutton.setText(QCoreApplication.translate("MainWindow", u"Buscar",
None))
        self.OrdenarDescendenteDistancia_pushbutton.setText(QCoreApplication.translate("MainW
indow", u"Ordenar de forma Decendente Distancia", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2),
QCoreApplication.translate("MainWindow", u"Tabla de Particulas", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_3),
QCoreApplication.translate("MainWindow", u"Gráfico de Particulas", None))
        self.menuArchivo.setTitle(QCoreApplication.translate("MainWindow", u"Archivo", None))
# retranslateUi

```

# *particula.py*

```
# Archivo 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,
    velocidad=0, red=0, green=0, blue=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.__velocidad = velocidad
        self.__red = red
        self.__green = green
        self.__blue = blue
        self.__distancia = distancia_euclidiana(origen_x, origen_y, destino_x, destino_y)

    def __str__(self):
        return ('\n\nParticula\n' +
            '\nID      : ' + str(self.__id) +
            '\nOrigen X : ' + str(self.__origen_x) +
            '\nOrigen Y : ' + str(self.__origen_y) +
            '\nDestino X: ' + str(self.__destino_x) +
            '\nDestino Y: ' + str(self.__destino_y) +
            '\nVelocidad: ' + str(self.__velocidad) +
            '\nRed      : ' + str(self.__blue) +
            '\nGreen    : ' + str(self.__green) +
            '\nBlue     : ' + str(self.__blue) +
            '\nDistancia: ' + str(self.__distancia)
        )

    def __lt__(self, other):
        return self.id < other.id

    def to_dict(self):
        return {
            "id": self.__id,
            "origen_x": self.__origen_x,
            "origen_y": self.__origen_y,
            "destino_x": self.__destino_x,
            "destino_y": self.__destino_y,
            "velocidad": self.__velocidad,
            "red": self.__red,
            "green": self.__green,
            "blue": self.__blue,
        }

@property
```

```
def id(self):
    return self.__id

@property
def origenX(self):
    return self.__origen_x

@property
def origenY(self):
    return self.__origen_y

@property
def destinoX(self):
    return self.__destino_x

@property
def destinoY(self):
    return self.__destino_y

@property
def velocidad(self):
    return self.__velocidad

@property
def red(self):
    return self.__red

@property
def green(self):
    return self.__green

@property
def blue(self):
    return self.__blue

@property
def distancia(self):
    return self.__distancia
```

# *manager.py*

```
# Archivo manager.py
from particula import Particula
import json

class Manager:

    def __init__(self):
        self.__particulas = []

    def agregarInicio(self, particula: Particula):
        self.__particulas.insert(0, particula)

    def agregarFinal(self, particula: Particula):
        self.__particulas.append(particula)

    def imprimir(self):
        for particula in self.__particulas:
            print(particula)

    def __str__(self):
        return "".join(
            str(particula) for particula in self.__particulas
        )

    def sortById(self):
        array = []

        for particula in self.__particulas:
            array.append(particula)

        array.sort(key=particula.__id, reverse=False)

        return "".join(
            str(particula) for particula in array
        )

    def sortBySpeed(self):
        array = []

        for particula in self.__particulas:
            array.append(particula)

        array.sort(key=particula.velocidad, reverse=False)

        return "".join(
            str(particula) for particula in array
        )
```

```

def sortByDistance(self):
    array = []

    for particula in self.__particulas:
        array.append(particula)

    array.sort(key=particula.distancia, reverse=True)

    return "".join(
        str(particula) for particula in array
    )

def __len__(self):
    return len(self.__particulas)

def __iter__(self):
    self.cont = 0
    return self

def __next__(self):
    if self.cont < len(self.__particulas):
        particula = self.__particulas[self.cont]
        self.cont += 1
        return particula
    else:
        raise StopIteration

def guardar(self, ubicacion):
    try:
        with open(ubicacion, "w") as archivo:
            lista = [particula.to_dict() for particula in self.__particulas]
            json.dump(lista, archivo, indent=5)
        return 1
    except:
        return 0

def abrir(self, ubicacion):
    try:
        with open(ubicacion, "r") as archivo:
            lista = json.load(archivo)
            self.__particulas = [Particula(**particula) for particula in lista]
        return 1
    except:
        return 0

```