Actividad 08 // (QTableWidget)

Sámano Juárez Juan Jesús.

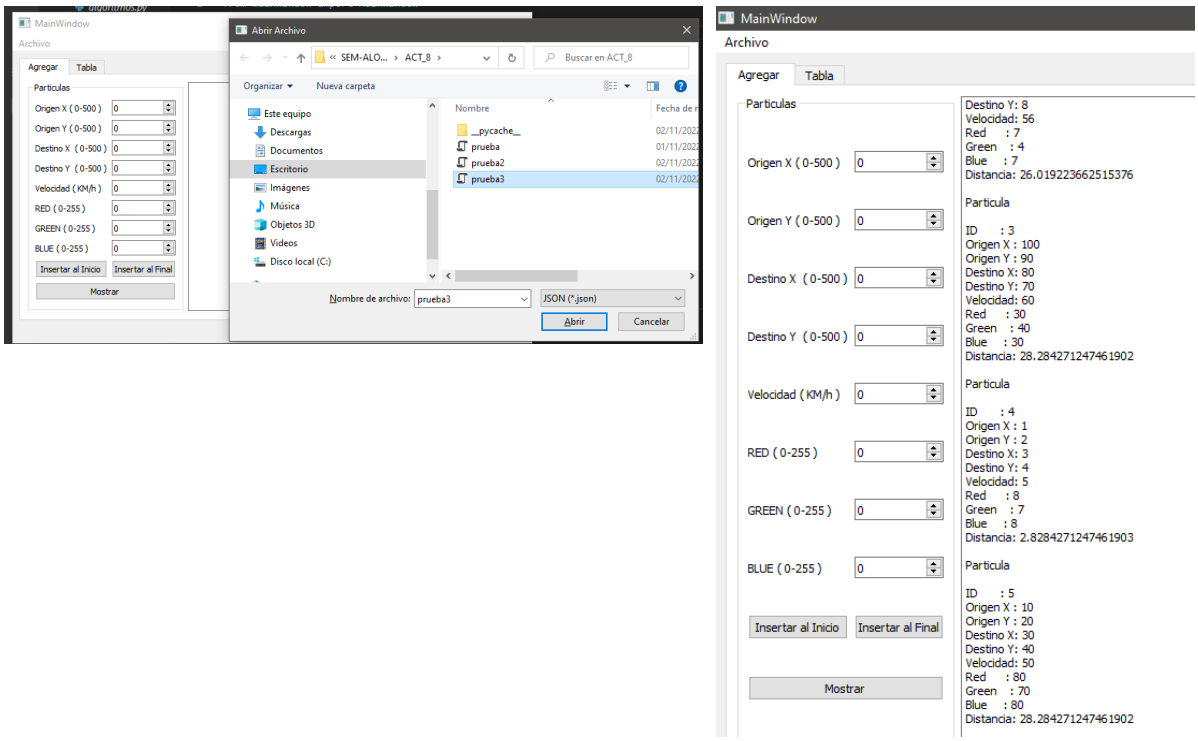
Seminario de Solución de Problemas de Algoritmia.

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 captura de pantalla de lo que se pide en el punto 2. sub punto a.
- [ ] Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto b.
- [ ] Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto c.
- [ ] Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto d.

Desarrollo.

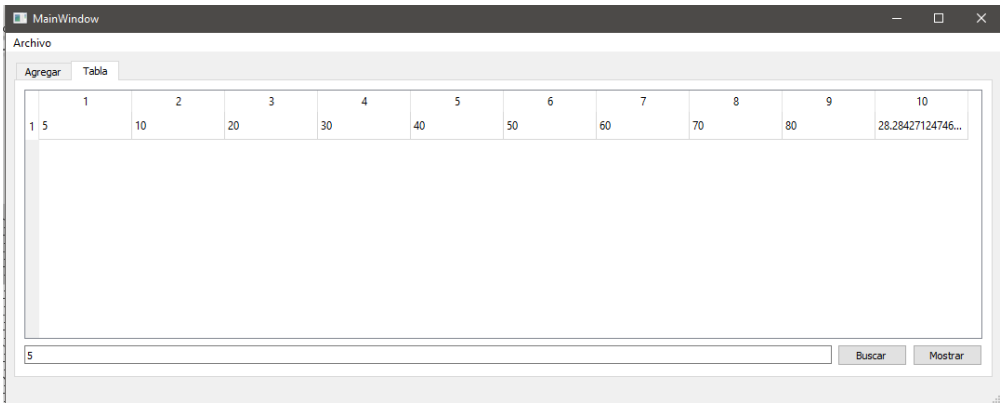
Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto a. (Agrega o recupera un respaldo de al menos 5 partículas)



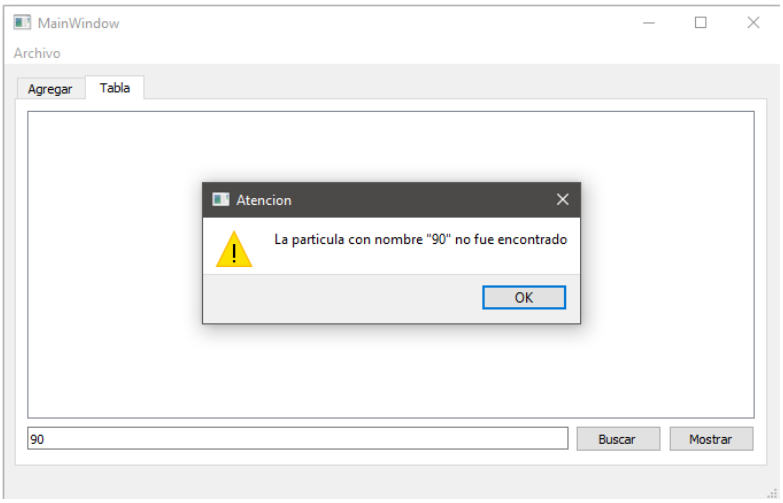
Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto b.(Muestra las partículas en el `QTableWidget` .)

```
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))
        origen_x_widget = QTableWidgetItem(str(particula.origen_x))
        origen_y_widget = QTableWidgetItem(str(particula.origen_y))
        destino_x_widget = QTableWidgetItem(str(particula.destino_x))
        destino_y_widget = QTableWidgetItem(str(particula.destino_y))
        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, origen_x_widget)
        self.ui.tabla.setItem(0, 2, origen_y_widget)
        self.ui.tabla.setItem(0, 3, destino_x_widget)
        self.ui.tabla.setItem(0, 4, destino_y_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)
```

Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto c.(Realiza una búsqueda de una partícula con un `id` existente.)



Se muestra captura de pantalla de lo que se pide en el punto 2. sub punto d.(Realiza una búsqueda de una partícula con un `id` no existente.)



Conclusiones.

Al principio todo fue sumamente fácil, hasta llegar a la parte donde tenia que insertar el id en la ventana llamada tabla para poder desplegar o mostrar la información de las partículas ingresadas previamente, ya que en el video se guardaba como tipo texto así que se tuve que cambar a str para que se pudiera encontrar la id de las partículas.

Referencias.

Michel Davalos Boites.[MICHEL DAVALOS BOITES](20/10/2022) PySide2 - QTableWidget (Qt for Python)(V)[Archivo de video]. https://www.youtube.com/watch?v=1yEpAHaiMxs&t=2425s&ab_channel=MICHELDAVALOSBOITES

Código.

main.py

```
from PySide2.QtWidgets import QPushButton, QApplication
from mainwindow import MainWindow
import sys

#Aplicación de QT
app = QApplication()
#Crear objeto
window = MainWindow()
#Hacer visible el elemento Botón
window.show()
#Qt loop
sys.exit(app.exec_())
```

mainwindow.py

```
from ast import Str
from math import fabs
from multiprocessing import managers
from sqlite3 import Row
from PySide2.QtWidgets import QMainWindow, QFileDialog, QMessageBox,
QTableWidgetItem
from PySide2.QtCore import Slot
```

```
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.mostrar.clicked.connect(self.click_mostrar)

        self.ui.actionAbrir.triggered.connect(self.action_abrir_archivo)
        self.ui.actionGuardar.triggered.connect(self.action_guardar_archivo)

        self.ui.mostrar_tabla_pushButton.clicked.connect(self.mostrar_tabla)
        self.ui.buscar_pushButton.clicked.connect(self.buscar_titulo)

    @Slot()
    def buscar_titulo(self):
        id = self.ui.buscar_lineEdit.text() #str
        encontrado = 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))
                origen_x_widget = QTableWidgetItem(str(particula.origen_x))
                origen_y_widget = QTableWidgetItem(str(particula.origen_y))
                destino_x_widget = QTableWidgetItem(str(particula.destino_x))
                destino_y_widget = QTableWidgetItem(str(particula.destino_y))
                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, origen_x_widget)
            self.ui.tabla.setItem(0, 2, origen_y_widget)
            self.ui.tabla.setItem(0, 3, destino_x_widget)
            self.ui.tabla.setItem(0, 4, destino_y_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)

            encontrado = True
            return
    if not encontrado:
        QMessageBox.warning(
            self,
            "Atencion",
            f'La particula con nombre "{id}" no fue encontrado'
            )




@Slot()
def mostrar_tabla(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))

    row = 0
    for particula in self.manager:
        id_widget = QTableWidgetItem(str(particula.id))
        origen_x_widget = QTableWidgetItem(str(particula.origen_x))
        origen_y_widget = QTableWidgetItem(str(particula.origen_y))
        destino_x_widget = QTableWidgetItem(str(particula.destino_x))
        destino_y_widget = QTableWidgetItem(str(particula.destino_y))
        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(row, 0, id_widget)
            self.ui.tabla.setItem(row, 1, origen_x_widget)
            self.ui.tabla.setItem(row, 2, origen_y_widget)
            self.ui.tabla.setItem(row, 3, destino_x_widget)
            self.ui.tabla.setItem(row, 4, destino_y_widget)
            self.ui.tabla.setItem(row, 5, velocidad_widget)
            self.ui.tabla.setItem(row, 6, red_widget)
            self.ui.tabla.setItem(row, 7, green_widget)
            self.ui.tabla.setItem(row, 8, blue_widget)
            self.ui.tabla.setItem(row, 9, distancia_widget)

            row += 1



    #Funcion que es llamada por x razón que imprime Click en Terminal.
    @Slot()
    # def click_mostrar(self):
    #     a

    @Slot()
    def action_abrir_archivo(self):
        #print("Abrir_archivo")
        ubicacion = QFileDialog.getOpenFileName(
            self,
            'Abrir Archivo',
            '.',
            'JSON (*.json)'
        ) [0]
        if self.manager.abrir(ubicacion):
            QMessageBox.information(
                self,
                "Éxito",
                "Se abrió el archivo" + ubicacion
            )
        else:
            QMessageBox.critical(
                self,
                "Error",
                "Error al abrir el archivo" + ubicacion
            )
```

```
    @Slot()
    def action_guardar_archivo(self):
        #print("guardar_archivo")
        ubicacion = QFileDialog.getSaveFileName(
            self,
            'Guardar Archivo',
            '.',
            'JSON (*.json)'
        )[0]
        print(ubicacion)
        if self.manager.guardar(ubicacion):
            QMessageBox.information(
                self,
                "Exito",
                "Se pudo crear el archivi" + ubicacion
            )
        else:
            QMessageBox.critical(
                self,
                "Error",
                "No se pudo crear el archivo" + ubicacion
            )


    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()

    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()

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

manager.py

```
import imp
from turtle import st
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 __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]
                print(lista)
                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
```

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)
                )

    @property
    def id(self):
        return self.__id

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

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

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

    @property
    def destino_y(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

    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
        }
```

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.objectName():
            MainWindow.setObjectName(u"MainWindow")
        MainWindow.resize(675, 400)
        self.actionGuardar = QAction(MainWindow)
        self.actionGuardar.setObjectName(u"actionGuardar")
        self.actionAbrir = QAction(MainWindow)
        self.actionAbrir.setObjectName(u"actionAbrir")
        self.centralwidget = QWidget(MainWindow)
        self.centralwidget.setObjectName(u"centralwidget")
        self.gridLayout_3 = QGridLayout(self.centralwidget)
        self.gridLayout_3.setObjectName(u"gridLayout_3")
        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.groupBox = QGroupBox(self.tab)
        self.groupBox.setObjectName(u"groupBox")
        self.gridLayout = QGridLayout(self.groupBox)
        self.gridLayout.setObjectName(u"gridLayout")
        self.green = QSpinBox(self.groupBox)
        self.green.setObjectName(u"green")
        self.green.setMaximum(255)

        self.gridLayout.addWidget(self.green, 6, 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_7 = QLabel(self.groupBox)
        self.label_7.setObjectName(u"label_7")

        self.gridLayout.addWidget(self.label_7, 1, 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.label_5 = QLabel(self.groupBox)
        self.label_5.setObjectName(u"label_5")

        self.gridLayout.addWidget(self.label_5, 6, 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.label_3 = QLabel(self.groupBox)
        self.label_3.setObjectName(u"label_3")

        self.gridLayout.addWidget(self.label_3, 4, 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.red = QSpinBox(self.groupBox)
        self.red.setObjectName(u"red")
        self.red.setMaximum(255)

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

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

        self.gridLayout.addWidget(self.mostrar, 9, 0, 1, 2)

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

        self.gridLayout.addWidget(self.label_4, 5, 0, 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.velocidad = QSpinBox(self.groupBox)
        self.velocidad.setObjectName(u"velocidad")
        self.velocidad.setMaximum(999)
```

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

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

        self.gridLayout.addWidget(self.label, 2, 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.insertar_inicio = QPushButton(self.groupBox)
        self.insertar_inicio.setObjectName(u"insertar_inicio")

        self.gridLayout.addWidget(self.insertar_inicio, 8, 0, 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.oy = QSpinBox(self.groupBox)
        self.oy.setObjectName(u"oy")
        self.oy.setMaximum(500)

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

        self.lista_particulas = QPlainTextEdit(self.tab)
        self.lista_particulas.setObjectName(u"lista_particulas")

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

        self.tabWidget.addTab(self.tab, "")
```

```
        self.tab_2 = QWidget()
        self.tab_2.setObjectName(u"tab_2")
        self.gridLayout_4 = QGridLayout(self.tab_2)
        self.gridLayout_4.setObjectName(u"gridLayout_4")
        self.tabla = QTableWidget(self.tab_2)
        self.tabla.setObjectName(u"tabla")

        self.gridLayout_4.addWidget(self.tabla, 0, 0, 1, 3)

        self.buscar_lineEdit = QLineEdit(self.tab_2)
        self.buscar_lineEdit.setObjectName(u"buscar_lineEdit")

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

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

        self.gridLayout_4.addWidget(self.buscar_pushButton, 1, 1, 1, 1)

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

        self.gridLayout_4.addWidget(self.mostrar_tabla_pushButton, 1, 2, 1, 1)

        self.tabWidget.addTab(self.tab_2, "")

        self.gridLayout_3.addWidget(self.tabWidget, 0, 0, 1, 1)

        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QMenuBar(MainWindow)
        self.menubar.setObjectName(u"menubar")
        self.menubar.setGeometry(QRect(0, 0, 675, 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.actionGuardar)
        self.menuArchivo.addAction(self.actionAbrir)

        self.retranslateUi(MainWindow)
```

```
        self.tabWidget.setCurrentIndex(1)


        QMetaObject.connectSlotsByName(MainWindow)
    # setupUi

    def retranslateUi(self, MainWindow):
        MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow",
u"MainWindow", None))
        self.actionGuardar.setText(QCoreApplication.translate("MainWindow",
u"Guardar", None))
#if QT_CONFIG(shortcut)
        self.actionGuardar.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+S", None))
#endif // QT_CONFIG(shortcut)
        self.actionAbrir.setText(QCoreApplication.translate("MainWindow",
u"Abrir", None))
#if QT_CONFIG(shortcut)
        self.actionAbrir.setShortcut(QCoreApplication.translate("MainWindow",
u"Ctrl+O", None))
#endif // QT_CONFIG(shortcut)
        self.groupBox.setTitle(QCoreApplication.translate("MainWindow",
u"Particulas", None))
        self.label_7.setText(QCoreApplication.translate("MainWindow", u"Origen Y
( 0-500 )", None))
        self.label_5.setText(QCoreApplication.translate("MainWindow", u"GREEN (
0-255 )", None))
        self.insertar_final.setText(QCoreApplication.translate("MainWindow",
u"Insertar al Final", None))
        self.label_3.setText(QCoreApplication.translate("MainWindow", u"Velocidad
( KM/h )", None))
        self.label_2.setText(QCoreApplication.translate("MainWindow", u"Destino
Y  ( 0-500 )", None))
        self.mostrar.setText(QCoreApplication.translate("MainWindow", u"Mostrar",
None))
        self.label_4.setText(QCoreApplication.translate("MainWindow", u"RED ( 0-
255 )", None))
        self.label_6.setText(QCoreApplication.translate("MainWindow", u"BLUE ( 0-
255 )", None))
        self.label.setText(QCoreApplication.translate("MainWindow", u"Destino
X  ( 0-500 )", None))
        self.insertar_inicio.setText(QCoreApplication.translate("MainWindow",
u"Insertar al Inicio", None))
        self.label_8.setText(QCoreApplication.translate("MainWindow", u"Origen X
( 0-500 )", None))
```

```
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab),
QCoreApplication.translate("MainWindow", u"Agregar", None))
        self.buscar_lineEdit.setPlaceholderText(QCoreApplication.translate("MainW
indow", u"ID de particula", None))
        self.buscar_pushButton.setText(QCoreApplication.translate("MainWindow",
u"Buscar", None))
        self.mostrar_tabla_pushButton.setText(QCoreApplication.translate("MainWin
dow", u"Mostrar", None))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2),
QCoreApplication.translate("MainWindow", u"Tabla", None))
        self.menuArchivo.setTitle(QCoreApplication.translate("MainWindow",
u"Archivo", None))
    # retranslateUi
```