Actividad 08 // (QTableWidget)

Sámano Juárez Juan Jesús.

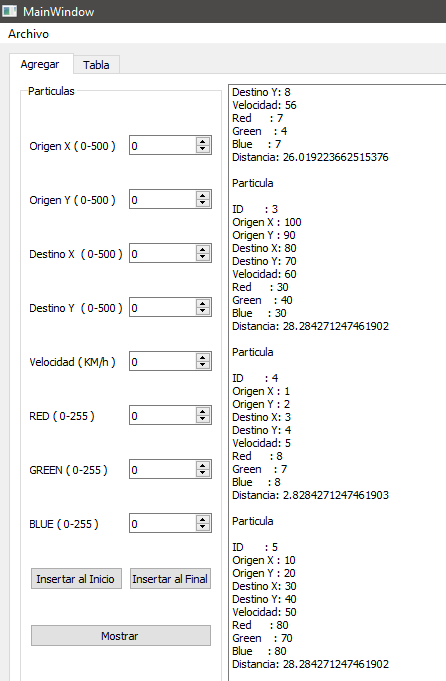
Seminario de Solución de Problemas de Algoritmia.

**Lineamiento de evaluación.**

* [ ] El reporte está en formato Google Docs o PDF.
* [ ] El reporte sigue las pautas del [Formato de Actividades](https://www.notion.so/Formato-de-Actividades-d098e1ce3a62423ca47c3661c5d93444) .
* [ ] El reporte tiene desarrollada todas las pautas del [Formato de Actividades](https://www.notion.so/Formato-de-Actividades-d098e1ce3a62423ca47c3661c5d93444).
* [ ] 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.

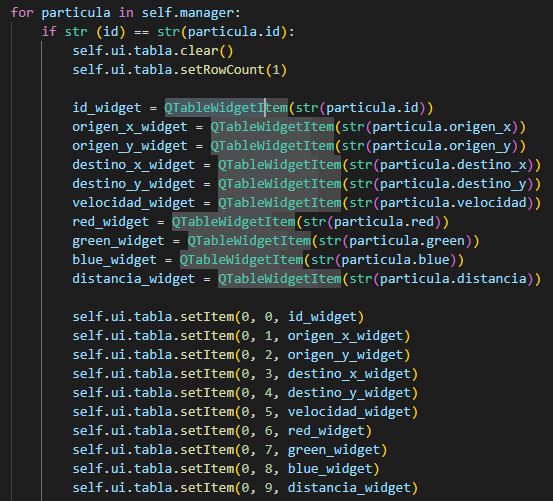
**Desarrollo.**

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)

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

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



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.)

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

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.)

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**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("MainWindow", u"ID de particula", None))

        self.buscar\_pushButton.setText(QCoreApplication.translate("MainWindow", u"Buscar", None))

        self.mostrar\_tabla\_pushButton.setText(QCoreApplication.translate("MainWindow", 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