Actividad 08 – QTableWidget

Hernández Lomelí Diego Armando

Seminario de algoritmia 2022B D02

Lineamientos 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

Gráfico, Gráfico de dispersión

Descripción generada automáticamenteDebemos agrandar el componente y meter dentro del todos los componentes incluidos en trabajos anteriores dentro de **Tab1**, solo falta arrastrarlos componentes.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

En las propiedades editamos **currentTabText** para modificar el texto de la tabulación.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Incluimos el **TableWidget** en la segunda tabulación, para acceder a su espacio solamente debemos dar clic en su etiqueta

Texto

Descripción generada automáticamente

Lo agregamos y justo debajo también ponemos un **LineEdit** y 2 botones.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Después elegimos con click derecho en el espacio que estamos usando y le agregamos el **layout** “**lay out in a grid”**

El siguiente paso es cambiar etiquetas de texto y cambiar el nombre de los componentes individuales.

Este es el resultado.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Estos son todos los cambios que vamos a realizar en la ventana. Seguimos desde Python y para eso primeramente convertimos el archivo **.ui** a uno **.py.**



Y ya podemos ejecutar la ventana desde Python.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Toca agregar eventos a los botones que creamos anteriormente, empezaremos con el btn para mostrar, con el tendremos una vista a las partículas que hemos almacenado.



Este método nos genera las columnas de la tabla pero debemos implementar el método **len** a la lista de partículas para evitar errores en la compilación.

Texto

Descripción generada automáticamente

El siguiente método debe estar dentro de la clase **listaParticula**.

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Ahora podemos probar la funcionalidad del botón. Al presionar el botón para buscar, veremos que la tabla ha cambiado y ha agregado filas para la cantidad de partículas que respaldamos en la actividad anterior (3 en total)

Tabla

Descripción generada automáticamente

Texto

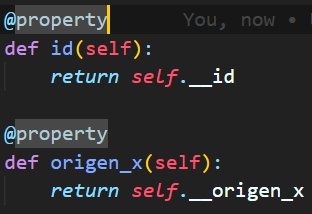
Descripción generada automáticamenteDebemos hacer que los objetos de la lista de partículas sean visibles desde acá, para ello debemos hacer que la instancia de **listaParticulas** sea iterable, esto es con propósito a no acceder directamente a la lista.

Dentro de la clase **listaParticula** agregamos la siguiente función.

La siguiente función estará dentro de la misma clase y nos devolverá el objeto que debería seguir en la iteración, esto sin acceder directamente a la lista **partículas**.

Texto

Descripción generada automáticamente

Antes de volver a **mainwindow.py** debemos agregar modificadores de acceso a la clase **particula** para poder leerlos desde otra clase.

Estos getters tendrán la siguiente forma:

Debemos repetir para cada campo que quedamos acceder desde una clase externa

Seguimos con la importación de la clase **QTableWidgetItem** en la clase **mainwindow.py** para empezar a rellenar de datos la tabla.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteRetomamos el método **mostrar\_tabla** desde donde lo dejamos y agregamos el código de la imagen, con cada **QTableWidgetItem** estamos generando una nueva celda.

Cuando usamos el método **setItem** de **QTable** decidimos en que fila y en que columna queremos agregar el objeto.

Si probamos el método después de haber cargado datos previamente guardados, tendremos el siguiente resultado, ya son visibles desde la lista.

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Lo que procede ahora es incluir el método de búsqueda mediante el id de la partícula

Texto

Descripción generada automáticamente

Parece ser muy extenso, pero por la cantidad de atributos toma mínimo 18 líneas solo para agregar una fila a la tabla. La búsqueda es línea entre los elementos del arreglo.

Ya podemos empezar a hacer una mejor prueba, para ello generaré un respaldo de 5 partículas y lo cargaré en la ventana, después haré una búsqueda con intención a ser exitosa y luego otra con intención a ser fallida.

1. Agregar o recuperar un respaldo de por lo menos 5 particulas.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Mostrar las partículas en el **QTableWidget.**

Tabla

Descripción generada automáticamente con confianza baja

1. Realizar la búsqueda de una partícula con un id existente.

Para este ejemplo buscaremos la partícula con el id **“2”.**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Solo muestra la partícula que coincide en la búsqueda, todas las demás no son vsibles.

1. Realizar la búsqueda de una partícula con un id no existente.

Para este ejemplo buscaremos las partícula con el id **“10”.**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Conclusiones

En esta actividad logre desarrollar los requerimientos con ayuda del material proporcionado, lo que me sorprendió en esta actividad fue la escritura de métodos que solo puedo suponer que son sobre escritos para poder trabajar con ellos (método len y next) por las características del lenguaje, sobre el manejo de la tabla me recuerda al manejo que se le da desde javax.swing.JTable, pero se usan más instancias diferentes para hacer que se desplieguen sus datos. En general fue una buena actividad para mejorar practicas en el acceso a atributos de una clase dentro de otra.

# Referencia

BOITES, M. D. (29 de Octubre de 2020). Obtenido de Youtube: https://www.youtube.com/watch?v=1yEpAHaiMxs

Código

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(779, 552)

*self*.actionAbrir = QAction(MainWindow)

*self*.actionAbrir.setObjectName(u"actionAbrir")

*self*.actionGuardar = QAction(MainWindow)

*self*.actionGuardar.setObjectName(u"actionGuardar")

*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 = QGridLayout(*self*.tab)

*self*.gridLayout.setObjectName(u"gridLayout")

*self*.groupBox = QGroupBox(*self*.tab)

*self*.groupBox.setObjectName(u"groupBox")

*self*.formLayout = QFormLayout(*self*.groupBox)

*self*.formLayout.setObjectName(u"formLayout")

*self*.label = QLabel(*self*.groupBox)

*self*.label.setObjectName(u"label")

*self*.formLayout.setWidget(4, QFormLayout.LabelRole, *self*.label)

*self*.spnnDestinoX = QSpinBox(*self*.groupBox)

*self*.spnnDestinoX.setObjectName(u"spnnDestinoX")

*self*.spnnDestinoX.setMaximum(500)

*self*.formLayout.setWidget(4, QFormLayout.FieldRole, *self*.spnnDestinoX)

*self*.label\_2 = QLabel(*self*.groupBox)

*self*.label\_2.setObjectName(u"label\_2")

*self*.formLayout.setWidget(5, QFormLayout.LabelRole, *self*.label\_2)

*self*.spnnDestinoY = QSpinBox(*self*.groupBox)

*self*.spnnDestinoY.setObjectName(u"spnnDestinoY")

*self*.spnnDestinoY.setMaximum(500)

*self*.formLayout.setWidget(5, QFormLayout.FieldRole, *self*.spnnDestinoY)

*self*.label\_3 = QLabel(*self*.groupBox)

*self*.label\_3.setObjectName(u"label\_3")

*self*.formLayout.setWidget(6, QFormLayout.LabelRole, *self*.label\_3)

*self*.label\_5 = QLabel(*self*.groupBox)

*self*.label\_5.setObjectName(u"label\_5")

*self*.formLayout.setWidget(7, QFormLayout.LabelRole, *self*.label\_5)

*self*.spnnRed = QSpinBox(*self*.groupBox)

*self*.spnnRed.setObjectName(u"spnnRed")

*self*.spnnRed.setMaximum(255)

*self*.formLayout.setWidget(7, QFormLayout.FieldRole, *self*.spnnRed)

*self*.label\_6 = QLabel(*self*.groupBox)

*self*.label\_6.setObjectName(u"label\_6")

*self*.formLayout.setWidget(8, QFormLayout.LabelRole, *self*.label\_6)

*self*.spnnGreen = QSpinBox(*self*.groupBox)

*self*.spnnGreen.setObjectName(u"spnnGreen")

*self*.spnnGreen.setMaximum(255)

*self*.formLayout.setWidget(8, QFormLayout.FieldRole, *self*.spnnGreen)

*self*.label\_7 = QLabel(*self*.groupBox)

*self*.label\_7.setObjectName(u"label\_7")

*self*.formLayout.setWidget(9, QFormLayout.LabelRole, *self*.label\_7)

*self*.spnnBlue = QSpinBox(*self*.groupBox)

*self*.spnnBlue.setObjectName(u"spnnBlue")

*self*.spnnBlue.setMaximum(255)

*self*.formLayout.setWidget(9, QFormLayout.FieldRole, *self*.spnnBlue)

*self*.btnAgregarFinal = QPushButton(*self*.groupBox)

*self*.btnAgregarFinal.setObjectName(u"btnAgregarFinal")

*self*.btnAgregarFinal.setCursor(QCursor(Qt.PointingHandCursor))

*self*.formLayout.setWidget(13, QFormLayout.SpanningRole, *self*.btnAgregarFinal)

*self*.label\_4 = QLabel(*self*.groupBox)

*self*.label\_4.setObjectName(u"label\_4")

*self*.formLayout.setWidget(0, QFormLayout.LabelRole, *self*.label\_4)

*self*.spnnOrigenX = QSpinBox(*self*.groupBox)

*self*.spnnOrigenX.setObjectName(u"spnnOrigenX")

*self*.spnnOrigenX.setMaximum(500)

*self*.formLayout.setWidget(0, QFormLayout.FieldRole, *self*.spnnOrigenX)

*self*.spnnOrigenY = QSpinBox(*self*.groupBox)

*self*.spnnOrigenY.setObjectName(u"spnnOrigenY")

*self*.spnnOrigenY.setMaximum(500)

*self*.formLayout.setWidget(2, QFormLayout.FieldRole, *self*.spnnOrigenY)

*self*.label\_8 = QLabel(*self*.groupBox)

*self*.label\_8.setObjectName(u"label\_8")

*self*.formLayout.setWidget(2, QFormLayout.LabelRole, *self*.label\_8)

*self*.spnnVelocidad = QSpinBox(*self*.groupBox)

*self*.spnnVelocidad.setObjectName(u"spnnVelocidad")

*self*.spnnVelocidad.setMaximum(500)

*self*.formLayout.setWidget(6, QFormLayout.FieldRole, *self*.spnnVelocidad)

*self*.btnAgregarInicio = QPushButton(*self*.groupBox)

*self*.btnAgregarInicio.setObjectName(u"btnAgregarInicio")

*self*.btnAgregarInicio.setCursor(QCursor(Qt.PointingHandCursor))

*self*.formLayout.setWidget(14, QFormLayout.SpanningRole, *self*.btnAgregarInicio)

*self*.btnMostrar = QPushButton(*self*.groupBox)

*self*.btnMostrar.setObjectName(u"btnMostrar")

*self*.btnMostrar.setCursor(QCursor(Qt.PointingHandCursor))

*self*.formLayout.setWidget(15, QFormLayout.SpanningRole, *self*.btnMostrar)

*self*.label\_9 = QLabel(*self*.groupBox)

*self*.label\_9.setObjectName(u"label\_9")

*self*.formLayout.setWidget(10, QFormLayout.LabelRole, *self*.label\_9)

*self*.spnnDistancia = QSpinBox(*self*.groupBox)

*self*.spnnDistancia.setObjectName(u"spnnDistancia")

*self*.spnnDistancia.setMaximum(255)

*self*.formLayout.setWidget(10, QFormLayout.FieldRole, *self*.spnnDistancia)

*self*.gridLayout.addWidget(*self*.groupBox, 0, 0, 1, 1)

*self*.plainTextEdit = QPlainTextEdit(*self*.tab)

*self*.plainTextEdit.setObjectName(u"plainTextEdit")

*self*.plainTextEdit.setMaximumSize(QSize(500, 16777215))

*self*.plainTextEdit.setFrameShadow(QFrame.Raised)

*self*.gridLayout.addWidget(*self*.plainTextEdit, 0, 1, 1, 1)

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

*self*.tab\_2 = QWidget()

*self*.tab\_2.setObjectName(u"tab\_2")

*self*.gridLayout\_2 = QGridLayout(*self*.tab\_2)

*self*.gridLayout\_2.setObjectName(u"gridLayout\_2")

*self*.tableParticulas = QTableWidget(*self*.tab\_2)

*self*.tableParticulas.setObjectName(u"tableParticulas")

*self*.gridLayout\_2.addWidget(*self*.tableParticulas, 0, 0, 1, 3)

*self*.searchEdit = QLineEdit(*self*.tab\_2)

*self*.searchEdit.setObjectName(u"searchEdit")

*self*.gridLayout\_2.addWidget(*self*.searchEdit, 1, 0, 1, 1)

*self*.buscar\_pushButton = QPushButton(*self*.tab\_2)

*self*.buscar\_pushButton.setObjectName(u"buscar\_pushButton")

*self*.gridLayout\_2.addWidget(*self*.buscar\_pushButton, 1, 1, 1, 1)

*self*.mostrar\_pushButton = QPushButton(*self*.tab\_2)

*self*.mostrar\_pushButton.setObjectName(u"mostrar\_pushButton")

*self*.gridLayout\_2.addWidget(*self*.mostrar\_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, 779, 26))

*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*.actionAbrir)

*self*.menuArchivo.addAction(*self*.actionGuardar)

*self*.retranslateUi(MainWindow)

*self*.tabWidget.setCurrentIndex(0)

        QMetaObject.connectSlotsByName(MainWindow)

*# setupUi*

    def retranslateUi(self, MainWindow):

        MainWindow.setWindowTitle(QCoreApplication.translate("MainWindow", u"MainWindow", None))

*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*.actionGuardar.setText(QCoreApplication.translate("MainWindow", u"Guardar", None))

*#if QT\_CONFIG(shortcut)*

*self*.actionGuardar.setShortcut(QCoreApplication.translate("MainWindow", u"Ctrl+G", None))

*#endif // QT\_CONFIG(shortcut)*

*self*.groupBox.setTitle(QCoreApplication.translate("MainWindow", u"GroupBox", None))

*self*.label.setText(QCoreApplication.translate("MainWindow", u"Destino X", None))

*self*.label\_2.setText(QCoreApplication.translate("MainWindow", u"Destino Y", None))

*self*.label\_3.setText(QCoreApplication.translate("MainWindow", u"Velocidad", None))

*self*.label\_5.setText(QCoreApplication.translate("MainWindow", u"Red", None))

*self*.label\_6.setText(QCoreApplication.translate("MainWindow", u"Green", None))

*self*.label\_7.setText(QCoreApplication.translate("MainWindow", u"Blue", None))

*self*.btnAgregarFinal.setText(QCoreApplication.translate("MainWindow", u"Agregar final", None))

*self*.label\_4.setText(QCoreApplication.translate("MainWindow", u"Origen X:", None))

*self*.label\_8.setText(QCoreApplication.translate("MainWindow", u"Origen Y: ", None))

*self*.btnAgregarInicio.setText(QCoreApplication.translate("MainWindow", u"Agregar inicio", None))

*self*.btnMostrar.setText(QCoreApplication.translate("MainWindow", u"Mostrar", None))

*self*.label\_9.setText(QCoreApplication.translate("MainWindow", u"Distancia:", None))

*self*.tabWidget.setTabText(*self*.tabWidget.indexOf(*self*.tab), QCoreApplication.translate("MainWindow", u"Agregar", None))

*self*.searchEdit.setPlaceholderText(QCoreApplication.translate("MainWindow", u"Id de particula", None))

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

*self*.mostrar\_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*

mainwindow.py

*from* ui\_mainwindow *import* Ui\_MainWindow, QFileDialog, QMessageBox, QTableWidgetItem

*from* PySide2.QtWidgets *import* QMainWindow

*from* PySide2.QtCore *import* Slot

*from* listaParticulas *import* listaParticula

*from* Particula *import* Particula

class MainWindow(QMainWindow):

    \_\_contador = 0

    def \_\_init\_\_(self):

        super(MainWindow, *self*).\_\_init\_\_()

*self*.\_\_lista = listaParticula()

*self*.ui = Ui\_MainWindow()

*self*.ui.setupUi(*self*)

*self*.ui.btnAgregarInicio.clicked.connect(*self*.click\_agregar\_inicio)

*self*.ui.btnAgregarFinal.clicked.connect(*self*.click\_agregar\_final)

*self*.ui.btnMostrar.clicked.connect(*self*.mostrar)

*""" Metodos para el menu de la ventana """*

*self*.ui.actionAbrir.triggered.connect(*self*.action\_abrir\_archivo)

*self*.ui.actionGuardar.triggered.connect(*self*.action\_guardar\_archivo)

*""" Metodos para trabajar con la tabla """*

*self*.ui.mostrar\_pushButton.clicked.connect(*self*.mostrar\_tabla)

*self*.ui.buscar\_pushButton.clicked.connect(*self*.buscar\_id\_tabla)

    @Slot()

    def mostrar\_tabla(self):

*self*.ui.tableParticulas.setColumnCount(10)

        headers = ["id", "origen\_x", "origen\_y", "destino\_x",

                   "destino\_y", "veloicidad", "red", "green", "blue", "distancia"]

*self*.ui.tableParticulas.setHorizontalHeaderLabels(headers)

*self*.ui.tableParticulas.setRowCount(len(*self*.\_\_lista))

*""" Empezamos a rellenar la tabla """*

        row = 0

*for* particula *in* *self*.\_\_lista:

            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))

            veloicidad\_widget = QTableWidgetItem(str(particula.veloicidad))

            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.tableParticulas.setItem(row, 0, id\_widget)

*self*.ui.tableParticulas.setItem(row, 1, origen\_x\_widget)

*self*.ui.tableParticulas.setItem(row, 2, origen\_y\_widget)

*self*.ui.tableParticulas.setItem(row, 3, destino\_x\_widget)

*self*.ui.tableParticulas.setItem(row, 4, destino\_y\_widget)

*self*.ui.tableParticulas.setItem(row, 5, veloicidad\_widget)

*self*.ui.tableParticulas.setItem(row, 6, red\_widget)

*self*.ui.tableParticulas.setItem(row, 7, green\_widget)

*self*.ui.tableParticulas.setItem(row, 8, blue\_widget)

*self*.ui.tableParticulas.setItem(row, 9, distancia\_widget)

            row += 1

    @Slot()

    def buscar\_id\_tabla(self):

        idBusqueda = *self*.ui.searchEdit.text()

*for* particula *in* *self*.\_\_lista:

*if* idBusqueda == str(particula.id):

*self*.ui.tableParticulas.clear()

*self*.ui.tableParticulas.setColumnCount(10)

                headers = ["id", "origen\_x", "origen\_y", "destino\_x",

                           "destino\_y", "veloicidad", "red", "green", "blue", "distancia"]

*self*.ui.tableParticulas.setHorizontalHeaderLabels(headers)

*self*.ui.tableParticulas.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))

                veloicidad\_widget = QTableWidgetItem(str(particula.veloicidad))

                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.tableParticulas.setItem(0, 0, id\_widget)

*self*.ui.tableParticulas.setItem(0, 1, origen\_x\_widget)

*self*.ui.tableParticulas.setItem(0, 2, origen\_y\_widget)

*self*.ui.tableParticulas.setItem(0, 3, destino\_x\_widget)

*self*.ui.tableParticulas.setItem(0, 4, destino\_y\_widget)

*self*.ui.tableParticulas.setItem(0, 5, veloicidad\_widget)

*self*.ui.tableParticulas.setItem(0, 6, red\_widget)

*self*.ui.tableParticulas.setItem(0, 7, green\_widget)

*self*.ui.tableParticulas.setItem(0, 8, blue\_widget)

*self*.ui.tableParticulas.setItem(0, 9, distancia\_widget)

*return*

        QMessageBox.warning(

*self*,

            "Error",

            f'No se ha encontrado una particula con el id: "{idBusqueda}"'

        )

*""" Generación de eventos para acciones del menu """*

    @Slot()

    def action\_guardar\_archivo(self):

        ubicacion = QFileDialog.getSaveFileName(

*self*,

            "Guardar archivo",

            ".",

            "JSON (\*.json)"

        )[0]

*if* *self*.\_\_lista.guardar(ubicacion):

            QMessageBox.information(

*self*,

                "Exito",

                ("Se pudo crear el archivo " + ubicacion)

            )

*else*:

            QMessageBox.critical(

*self*,

                "Error",

                ("No pudo crear el archivo " + ubicacion)

            )

    @Slot()

    def action\_abrir\_archivo(self):

        ubicacion = QFileDialog.getSaveFileName(

*self*,

            "Guardar archivo",

            ".",

            "JSON (\*.json)"

        )[0]

*if* *self*.\_\_lista.abrir(ubicacion):

            QMessageBox.information(

*self*,

                "Exito",

                ("Se pudo abrir el archivo " + ubicacion)

            )

*self*.ui.plainTextEdit.clear()

*self*.ui.plainTextEdit.insertPlainText(str(*self*.\_\_lista))

*else*:

            QMessageBox.critical(

*self*,

                "Error",

                ("No pudo abrir el archivo " + ubicacion)

            )

    @ Slot()

    def click\_agregar\_inicio(self):

*self*.\_\_lista.agregar\_inicio(*self*.procesarParticula())

*self*.\_\_contador += 1

    @ Slot()

    def click\_agregar\_final(self):

*self*.\_\_lista.agregar\_final(*self*.procesarParticula())

*self*.\_\_contador += 1

    @ Slot()

    def mostrar(self):

*self*.ui.plainTextEdit.clear()

*self*.ui.plainTextEdit.insertPlainText(str(*self*.\_\_lista))

    def procesarParticula(self):

*""" id, origen\_x, origen\_y, destino\_x, destino\_y, veloicidad, red, green, blue, distancia """*

*return* Particula(*self*.\_\_contador,

*self*.ui.spnnOrigenX.value(),

*self*.ui.spnnOrigenY.value(),

*self*.ui.spnnDestinoX.value(),

*self*.ui.spnnDestinoY.value(),

*self*.ui.spnnVelocidad.text(),

*self*.ui.spnnRed.value(),

*self*.ui.spnnBlue.value(),

*self*.ui.spnnGreen.value(),

*self*.ui.spnnDistancia.value())

particular.py

*from* algoritmos *import* distancia\_euclidiana

class Particula(object):

    \_\_id = 0

    \_\_origen\_x = 0

    \_\_origen\_y = 0

    \_\_destino\_x = 0

    \_\_destino\_y = 0

    \_\_veloicidad = 0

    \_\_red = 0

    \_\_green = 0

    \_\_blue = 0

    \_\_distancia = 0.0

    def \_\_init\_\_(self, id, origen\_x, origen\_y, destino\_x, destino\_y, veloicidad, red, green, blue, distancia):

*""" Propiedades de la clase """*

*self*.\_\_id = id

*self*.\_\_origen\_x = origen\_x

*self*.\_\_origen\_y = origen\_y

*self*.\_\_destino\_x = destino\_x

*self*.\_\_destino\_y = destino\_y

*self*.\_\_veloicidad = veloicidad

*self*.\_\_red = red

*self*.\_\_green = green

*self*.\_\_blue = blue

*self*.\_\_distancia = distancia

*""" Calculo de la distancia euclidiana """*

*self*.\_\_distancia = distancia\_euclidiana(

            origen\_x, origen\_y, destino\_x, destino\_y)

*""" Metodos geters """*

    @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 veloicidad(self):

*return* *self*.\_\_veloicidad

    @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 \_\_str\_\_(self):

*return* (

            "######################################\n"

            + "Id: " + str(*self*.\_\_id) + ",\n"

            + "Origen X: " + str(*self*.\_\_origen\_x) + ",\n"

            + "Origen Y: " + str(*self*.\_\_origen\_y) + ",\n"

            + "Destino X: " + str(*self*.\_\_destino\_x) + ",\n"

            + "Destino Y: " + str(*self*.\_\_destino\_y) + ",\n"

            + "Velocidad: " + str(*self*.\_\_veloicidad) + ",\n"

            + "Rojo: " + str(*self*.\_\_red) + ",\n"

            + "Verde: " + str(*self*.\_\_green) + ",\n"

            + "Azul: " + str(*self*.\_\_blue) + ",\n"

            + "Distancia: " + str(*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,

            "veloicidad": *self*.\_\_veloicidad,

            "red": *self*.\_\_red,

            "green": *self*.\_\_green,

            "blue": *self*.\_\_blue,

            "distancia": + *self*.\_\_distancia

        }

listaParticula.py

*import* json

*from* Particula *import* Particula

class listaParticula:

    def \_\_init\_\_(self):

*self*.\_\_particulas = []

    def agregar\_inicio(self, particula: Particula):

*self*.\_\_particulas.insert(0, particula)

    def agregar\_final(self, particula: Particula):

*self*.\_\_particulas.append(particula)

    def mostrar(self):

*for* particula *in* *self*.\_\_particulas:

            print(particula)

    def \_\_str\_\_(self):

*return* "".join(

            str(particula) *for* particula *in* *self*.\_\_particulas

        )

    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

    def \_\_len\_\_(self):

*return* len(*self*.\_\_particulas)

    def \_\_iter\_\_(self):

*self*.cont = 0

*return* *self*

    def \_\_next\_\_(self):

*if* *self*.cont < len(*self*.\_\_particulas):

*""" Asignamos la particula a devolver """*

            particula = *self*.\_\_particulas[*self*.cont]

*""" Incremenamos el contador """*

*self*.cont += 1

*return* particula

*""" detemos la iteración si se sobrepasa el tamaño de la lista """*

*raise* StopIteration

main.py

*from* PySide2.QtWidgets *import* QApplication

*from* mainwindow *import* MainWindow

*import* sys

app = QApplication()

window = MainWindow()

window.show()

sys.exit(app.exec\_())

Subida a gitHub

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8

$ git init

Initialized empty Git repository in C:/Users/Armando/Documents/GitHub/Actividad 8/.git/

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git status

On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git add .

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git status

On branch master

No commits yet

Untracked files:

(use "git add <file>..." to include in what will be committed)

src/

nothing added to commit but untracked files present (use "git add" to track)

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git add .

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: src/Main.py

new file: src/Particula.py

new file: src/\_\_pycache\_\_/Particula.cpython-310.pyc

new file: src/\_\_pycache\_\_/algoritmos.cpython-310.pyc

new file: src/\_\_pycache\_\_/listaParticulas.cpython-310.pyc

new file: src/\_\_pycache\_\_/mainwindow.cpython-310.pyc

new file: src/\_\_pycache\_\_/ui\_mainwindow.cpython-310.pyc

new file: src/algoritmos.py

new file: src/listaParticulas.py

new file: src/mainwindow.py

new file: src/mainwindow.ui

new file: src/ui\_mainwindow.py

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git commit -m "Subida ilnicial"

[master (root-commit) 6985313] Subida ilnicial

12 files changed, 648 insertions(+)

create mode 100644 src/Main.py

create mode 100644 src/Particula.py

create mode 100644 src/\_\_pycache\_\_/Particula.cpython-310.pyc

create mode 100644 src/\_\_pycache\_\_/algoritmos.cpython-310.pyc

create mode 100644 src/\_\_pycache\_\_/listaParticulas.cpython-310.pyc

create mode 100644 src/\_\_pycache\_\_/mainwindow.cpython-310.pyc

create mode 100644 src/\_\_pycache\_\_/ui\_mainwindow.cpython-310.pyc

create mode 100644 src/algoritmos.py

create mode 100644 src/listaParticulas.py

create mode 100644 src/mainwindow.py

create mode 100644 src/mainwindow.ui

create mode 100644 src/ui\_mainwindow.py

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git config --global user.email diego.hernandez7503@alumnos.udg.mx

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git config --global user.name Diego-Armando-H

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git commit -m "Subida ilnicial"

On branch master

nothing to commit, working tree clean

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (master)

$ git branch -M main

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (main)

$ git remote add origin https://github.com/Diego-Armando-H/Actividad\_8.git

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (main)

$ git push -u origin main

remote: Repository not found.

fatal: repository 'https://github.com/Diego-Armando-H/Actividad\_8.git/' not found

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (main)

$ git remote remove origin

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (main)

$ git remote add origin https://github.com/Diego-Armando-H/Actividad-8.git

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (main)

$ git push -u origin main

Enumerating objects: 16, done.

Counting objects: 100% (16/16), done.

Delta compression using up to 4 threads

Compressing objects: 100% (15/15), done.

Writing objects: 100% (16/16), 10.44 KiB | 1.74 MiB/s, done.

Total 16 (delta 0), reused 0 (delta 0), pack-reused 0

To https://github.com/Diego-Armando-H/Actividad-8.git

\* [new branch] main -> main

branch 'main' set up to track 'origin/main'.

Armando@Armando04 MINGW64 ~/Documents/GitHub/Actividad 8 (main)

$