Actividad 05

Clases y Objetos

*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 la captura de pantalla de los datos antes de usar el método agregar\_inicio() y la captura de pantalla del método mostrar() después de haber utilizado el método agregar\_inicio(). *(Con duda)*
* Se muestra la captura de pantalla de los datos antes de usar el método agregar\_final() y la captura de pantalla del método mostrar() después de haber utilizado el método agregar\_final(). *(Con duda)*

***AAAAAAAAXX  
Satisfacción 8/10***

***Desarrollo***

***Valor de los datos antes de usar el método agregarInicio()***

******

***Método mostrar() después de haber utilizado el método agregarInicio()***

***Texto, Tabla

Descripción generada automáticamente***

***Valor de los datos antes de usar el método agregarFinal()***

******

***Método mostrar() después de haber utilizado el método agregarFinal()***

***Texto

Descripción generada automáticamente***

Una disculpa profesor, en realidad no comprendí correctamente las instrucciones de las capturas de pantalla y demostré lo que comprendí de ellas.

***Conclusiones***

El Recurso Informativo ha sido clave para la comprensión del desarrollo de la actividad, todo el desarrollo ha sido fluido hasta tener la necesidad de concatenar enteros a una cadena de caracteres, solucionado usando la función str(), posteriormente teniendo problemas al comprender dos de las condiciones más importantes de la actividad, las capturas de pantalla, cosa que me preocupa haber realizado de forma errónea aún con el proyecto realizado correctamente.

***Referencias***

Pyside2 – Clases y Objetos

<https://youtu.be/KfQDtrrL2OU>

Convert integrer to string in python

<https://www.geeksforgeeks.org/convert-integer-to-string-in-python/>

***Código***

*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)

                )

*algoritmos.py*

# algortimos.py

import math

def distancia\_euclidiana(x\_1, y\_1, x\_2, y\_2):

    return math.sqrt((x\_2 - x\_1)\*\*2 + (y\_2 - y\_1)\*\*2)

*manager.py*

# Archivo manager.py

from particula import Particula

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)

particula0 = Particula(87, 10, 12, 50, 60, 80, 255, 45, 30)

particula1 = Particula(78, 1, 21, 5, 6, 8, 0, 54, 3)

manager = Manager()

manager.agregarInicio(particula1)

manager.agregarFinal(particula0)

manager.imprimir()