

Clases y Objetos

Cruz Collazo Wendy Paola.

Seminario de solución de Problemas de Algoritmia.

Lineamientos 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 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 ().
- ➤ 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 ().

Desarrollo

Comencé creando como un nuevo proyecto y dentro del mismo cree la clase Partícula en la cual definí los atributos y una vez así los puse en privado, una vez declarando todos los atributos los cuales los 'inicialice' de cierta manera por si no llegara a recibir nada. Ya que los declare use un self para asignar cada atributo al que corresponde.

Para poder mostrar las partículas que iba agregando tuve que definir un método para que cuando agregara un objeto me lo convirtiera en String y así poder imprimirlo, entonces básicamente hice la estructura de todos los atributos que tenía.

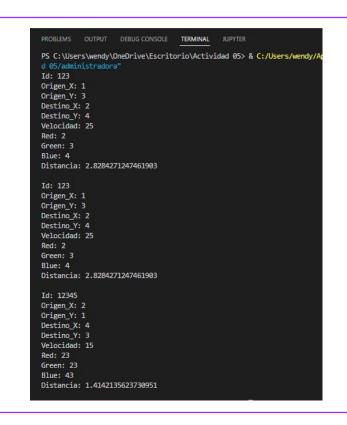
Para el atributo de Distancia tuve que crear otro archivo para poder crear esa función la cual era atreves de los datos obtenidos de algunos de los atributos sacar la distancia entre dos puntos. Teniendo la función ya hecha me regrese a la clase partícula y la importe para que se pudiera usar.

```
Id: 123
Origen_X: 1
Origen_Y: 3
Destino_X: 2
Destino_Y: 4
Velocidad: 25
Red: 2
Green: 3
Blue: 4
Distancia: 2.8284271247461903
```

Sin usar los métodos

Después de tener eso cree otra clase en la cual empecé a definir e implementar los métodos de agregar al inicio, agregar al final y mostrar.

Una vez teniendo los métodos y las clases correspondientes agregue partículas con los métodos, y compare con una calculadora web el resultado de la distancia.



Agregando al Inicio, final y inicio,

Conclusión:

El elaborar las clases y los métodos no se me complico ya que al ver el video pude entenderle, en lo que si me costo un poco de trabajo fue para sacar la distancia ya que intente hacerle de varias maneras y me lanzaba un error, vi varios videos que no me ayudaron mucho hasta que me di cuenta que al momento de definir mis atributos definí Distancia la cual me estaba dando error o mas bien no hacia lo que pedía así que la elimine y con eso pude solucionar ese error que me salía.

Referencias:

➤ Davalos Boites, Michel- Clase y Objetos (Qt for Python) (ll). https://www.youtube.com/watch?v=KfQDtrrL2OU

Código

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, destino_x, origen_y, destino_y)
  def __str__(self):
     return(
        '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.__velocidad) + '\n' +
        'Red: ' + str(self.__red) + '\n' +
        'Green: ' + str(self.__green) + '\n' +
        'Blue: ' + str(self.__blue) + '\n' +
        'Distancia: ' + str(self.__distancia) + '\n'
     )
```

Algoritmos.py

```
import math
from math import sqrt

def distancia_euclidiana(origen_x, origen_y, destino_x, destino_y):
    x_1=origen_x
```

```
y_1=origen_y
x_2=destino_x
y_2=destino_y
return sqrt((x_2 - x_1)**2 + (y_2 - y_1)**2)
```

administradora.py

```
from particula import Particula
class Administradora:
  def __init__(self):
     self.__particula = []
  def agregar_inicio(self, particula:Particula):
     self.__particula.insert(0, particula)
  def agregar_final(self, particula:Particula):
     self.__particula.append(particula)
  def mostrar(self):
     for particula in self._particula:
        print(particula)
p01 = Particula(123, 1, 3, 2, 4, 25, 2, 3, 4)
p02 = Particula(12345, 2, 1, 4, 3, 15, 23, 23, 43)
administradora = Administradora()
administradora.agregar_inicio(p01)
administradora.agregar_final(p02)
administradora.agregar_inicio(p01)
administradora.mostrar()
```