

Actividad 05 - Clases y Objetos

Gabriel Eduardo Sevilla Chavez

Seminario 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

Resultado del código ejecutado con éxito

Datos con el método `mostrar()` antes del método `agregar_inicio()`

```
[Running] python -u "c:\Users\Eduard\Documents\Actividad 05 - Clases y Objetos\main.py"
[Done] exited with code=0 in 0.543 seconds
```

Datos con el método `mostrar()` después del método `agregar_inicio()`

```
prueba = Particula(1,2,3,4,5,6,7,8,9)
```

```
Id: 1
origen_x: 2
origen_y: 3
destino_x: 4
destino_y: 5
velocidad: 6
red: 7
green: 8
blue: 9
distancia: 2.8284271247461903
```

Datos con el método `mostrar()` después del método `agregar_final()`

```
prueba3 = Particula(65,7,4,8,3,9,1,78,76)
```

```

Id: agregar_final() y la captura de par
origen_x: 1
origen_y: 3
destino_x: 4
destino_y: 5
velocidad: 6
red: 7
green: 8
blue: 9
distancia: 2.8284271247461903

Id: 65
origen_x: 7
origen_y: 4
destino_x: 8
destino_y: 3
velocidad: 9
red: 1
green: 78
blue: 70
distancia: 1.4142135623730951

```

Metodo para la formula de distancia euclidiana para el atributo distancia

```

def distancia_euclidiana(x1,y1,x2,y2):
    #Formula distancia entre dos puntos
    """Parámetros:
    x1 -- origen_x
    y1 -- origen_y
    x2 -- destino_x
    y2 -- destino_y"""
    result = math.sqrt((x2-x1)**2 + (y2-y1)**2)
    return result

```

Conclusiones

No hubo contratiempos ni dificultades nuevamente en el desarrollo de este programa, aunque, se pudo recatar un nuevo aprendizaje en la materia de POO gracias al desarrollo de dicho programa

Referencias

MICHEL DAVALOS BOITES. (2020, 8 octubre). *PySide2 - Clases y Objetos*

(Qt for Python)(II). YouTube. Recuperado 15 de octubre de 2022, de

<https://www.youtube.com/watch?v=KfQDtrrL2OU>

Código

partícula.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, vel
ocidad=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) -> str:
    return(
        "Id: \t\t" + str(self.__id) + '\n' +
        "origen_x: \t" + str(self.__origen_x) + '\n' +
        "origen_y: \t" + str(self.__origen_y) + '\n' +
        "destino_x: \t" + str(self.__destino_x) + '\n' +
        "destino_y: \t" + str(self.__destino_y) + '\n' +
        "velocidad: \t" + str(self.__velocidad) + '\n' +
        "red: \t\t" + str(self.__red) + '\n' +
        "green: \t\t" + str(self.__green) + '\n' +
        "blue: \t\t" + str(self.__blue) + '\n' +
        "distancia: \t" + str(self.__distancia) + '\n')

```

particulas.py

```
from particula import Particula
```

```

class Particulas:
    def __init__(self):
        self.__particulas = []

```

```

def agregar_final(self,particula:Particula):
    self.__particulas.append(particula)

def agregar_inicio(self,particula:Particula):
    self.__particulas.insert(0,particula)

def mostrar(self):
    for particula in self.__particulas:
        print(particula)

preuba = Particula(1,2,3,4,5,6,7,8,9)
prueba2 = Particula(10,20,30,40,50,60,70,80,90)
prueba3 = Particula(65,7,4,8,3,9,1,78,70)
particulas = Particulas()
particulas.agregar_inicio(preuba)
particulas.agregar_final(prueba3)
particulas.agregar_inicio(prueba2)
particulas.mostrar()

```

algoritmos.py

```

import math

def distancia_euclidiana(x1,y1,x2,y2):
    #Formula distancia entre dos puntos
    """Parámetros:
    x1 -- origen_x
    y1 -- origen_y
    x2 -- destino_x
    y2 -- destino_y"""
    result = math.sqrt((x2-x1)**2 + (y2-y1)**2)
    return result

```