

Actividad 05 - UI: Clases y Objetos

**JAIRO CAIN SANCHEZ
ESTRADA// Luis Angel
Elisea Graciano**

**SEMINARIO DE SOLUCION DE PROBLEMAS DE
ALGORITMIA**

Lineamientos de evaluación

- El programa funciona sin problema alguno, al momento de crear los objetos estos funcionan de manera correcta
- Al momento de importar las clases y funciones, estas trabajan sin ningún problema y se realiza las funciones esperadas
- Se muestran en pantalla los elementos de cada partícula sin problemas
- Se calcula la distancia de cada partícula sin problema alguno
- Se almacenan partículas y se insertan en el inicio o final de la lista

Desarrollo

Para comenzar se crea el archivo `particula.py` en el cual se crea el objeto partícula con sus atributos y una función la cual nos permite imprimir los atributos en pantalla, pero para uno de estos atributos se necesita llamar una función la cual es para calcular la distancia y esta en un archivo aparte el cual se llama `algoritmos.py`

```
self.__distancia=distancia_euclidiana(origen_x, destino_x, origen_y, destino_y)
```

```
def distancia_euclidiana(x_1, y_1, x_2, y_2):  
    """ Calcula la distancia euclidiana  
  
    Devuelve el resultado de la fórmula  
  
    También se le conoce a la fórmula como:  
    distancia entre dos puntos  
  
    Parámetros:  
    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)))
```

Posterior mente se crea un archivo llamado `particulas.py` el cual es el que administrara en una lista las partículas creadas, por lo cual el objeto `Partículas` crea una lista y permite almacenar los objetos partícula, `Partículas` tiene las funciones de insertar al final, inicio y mostrar. Las cuales reciben como parámetro un objeto tipo partícula.

```
p= Particula(1,2,1,3,4,5,6,7,8,9)  
p1= Particula(20,69,59,41,55,66,99,80,52,63)  
p2= Particula(2,69,50,200,55,66,100,80,52,63)  
particulas=Particulas()  
particulas.agregar_inicio(p2)  
particulas.mostrar()  
particulas.agregar_final(p)  
particulas.mostrar()  
particulas.agregar_inicio(p1)  
particulas.mostrar()
```

Se insertan las partículas y se muestran, primero se inserta la partícula con id 2 al inicio y

se muestra, posteriormente se inserta la partícula con el id 1 y se inserta al final, y por ultimo se inserta la partícula con el id 20 al inicio y se vuelve a mostrar

```
$ C:/Users/Angel/AppData/Local/Programs/Python/Python39-64/Scripts/python.exe
Id: 2
Origen x: 69
Origen y: 50
Destino x: 200
Destino y: 55
Velocidad: 66
Red: 100
Green: 80
Blue: 52
Distancia: 146.23952953972466

Id: 2
Origen x: 69
Origen y: 50
Destino x: 200
Destino y: 55
Velocidad: 66
Red: 100
Green: 80
Blue: 52
Distancia: 146.23952953972466

Id: 1
Origen x: 2
Origen y: 1
Destino x: 3
Destino y: 4
Velocidad: 5
```

```
Id: 20
Origen x: 69
Origen y: 59
Destino x: 41
Destino y: 55
Velocidad: 66
Red: 99
Green: 80
Blue: 52
Distancia: 17.204650534085253

Id: 2
Origen x: 69
Origen y: 50
Destino x: 200
Destino y: 55
Velocidad: 66
Red: 100
Green: 80
Blue: 52
Distancia: 146.23952953972466

Id: 1
Origen x: 2
Origen y: 1
Destino x: 3
Destino y: 4
Velocidad: 5
Red: 6
Green: 7
Blue: 8
Distancia: 1.4142135623730951
```

Conclusiones

Esta practica me gusto bastante ya que pude reforzar muchas cosas de orientado a objetos en Python que no recordaba o incluso aprender varias cosas que no sabia, como el uso del `__str__` para poder mostrar los datos de nuestras clases, los buenos métodos de programación estructurada.

Referencias

Boites, M. D. [UC3jaAJ6X3vMZJKpi9KAcy0w]. (2020, October 8). PySide2 - Clases y Objetos (Qt for Python)(II). Youtube. <https://www.youtube.com/watch?v=KfQDtrrL2OU>

Código

```
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,distancia=0.0):
    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'
        )
```

```

from math import sqrt
def distancia_euclidiana(x_1, y_1, x_2, y_2):
    """ Calcula la distancia euclidiana
    Devuelve el resultado de la fórmula
    También se le conoce a la fórmula como:
    distancia entre dos puntos
    Parámetros:
    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)))

```

```

from particula import Particula
class Particulas:
    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)

p= Particula(1,2,1,3,4,5,6,7,8,9)
p1= Particula(20,69,59,41,55,66,99,80,52,63)
p2= Particula(2,69,50,200,55,66,100,80,52,63)
particulas=Particulas()
particulas.agregar_inicio(p2)
particulas.mostrar()
particulas.agregar_final(p)
particulas.mostrar()
particulas.agregar_inicio(p1)
particulas.mostrar()

```

