

Actividad 05 (Clases y Objetos)

Jose Eduardo Silva Canizales

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

Captura de pantalla de los datos antes de usar el método `agregar_inicio()`.

```
22 l01 = Particula(id=11, origen_x=10, origen_y=10, destino_x=20, destino_y=30, velocidad=200, red=1,  
23         green=3, blue=7, distancia=distancia_euclidiana)  
24  
25 particulas = Administrador()  
26 particulas.agregar_inicio(l01)  
27 particulas.mostrar()  
28
```

Captura de pantalla del método mostrar() después de haber utilizado el método agregar_inicio().

```
algoritmos.py | partícula.py | administrador.py > ...
22 l01 = Partícula(id=11, origen_x=10, origen_y=10, destino_x=20, destino_y=30, velocidad=200, red=1,
23         green=3, blue=7, distancia=distancia_euclidiana)
24
25 particulas = Adminisrador()
26 particulas.agregar_inicio(l01)
27 particulas.mostrar()
28
```

PROBLEMAS | **Salida** | CONSOLA DE DEPURACIÓN | TERMINAL | JUPYTER

[Running] python -u "c:\Users\juan\Desktop\Actividad5\administrador.py"

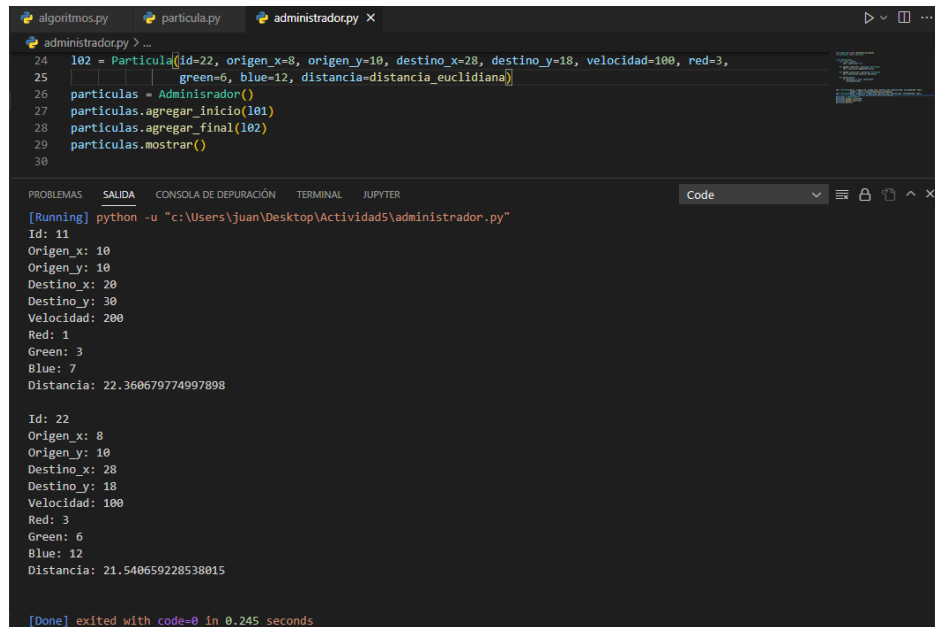
Id: 11
Origen_x: 10
Origen_y: 10
Destino_x: 20
Destino_y: 30
Velocidad: 200
Red: 1
Green: 3
Blue: 7
Distancia: 22.360679774997898

[Done] exited with code=0 in 2.69 seconds

Captura de pantalla de los datos antes de usar el método `agregar_final()`.

```
algoritmos.py | partícula.py | administrador.py x
administrador.py > ...
24 l02 = Partícula(id=22, origen_x=8, origen_y=10, destino_x=28, destino_y=18, velocidad=100, red=3,
25         green=6, blue=17, distancia=distancia_euclidiana)
26 partículas = Administrador()
27 partículas.agregar_inicio(l01)
28 partículas.agregar_final(l02)
29 partículas.mostrar()
30
```

Captura de pantalla del método `mostrar()` después de haber utilizado el método `agregar_final()`.



The screenshot shows a Jupyter Notebook with three tabs: 'algoritmos.py', 'particula.py', and 'administrador.py'. The 'administrador.py' tab is active, displaying the following Python code:

```
24  l02 = Particula(id=22, origen_x=8, origen_y=10, destino_x=28, destino_y=18, velocidad=100, red=3,  
25      green=6, blue=12, distancia=distancia_euclidiana)  
26  particulas = Administrador()  
27  particulas.agregar_inicio(l02)  
28  particulas.agregar_final(l02)  
29  particulas.mostrar()  
30
```

Below the code editor, the 'Code' tab is selected in the output area, showing the execution results:

```
[Running] python -u "c:\Users\juan\Desktop\Actividad5\administrador.py"  
Id: 11  
Origen_x: 10  
Origen_y: 10  
Destino_x: 20  
Destino_y: 30  
Velocidad: 200  
Red: 1  
Green: 3  
Blue: 7  
Distancia: 22.360679774997898  
  
Id: 22  
Origen_x: 8  
Origen_y: 10  
Destino_x: 28  
Destino_y: 18  
Velocidad: 100  
Red: 3  
Green: 6  
Blue: 12  
Distancia: 21.540659228538015  
  
[Done] exited with code=0 in 0.245 seconds
```

Conclusiones

Que aprendiste

Durante esta actividad aprendí la implementación e la programación orientada a objetos (POO) en conjunto con el uso de listas.

Que no entendiste desde el inicio y como hiciste para entenderlo

El primer problema fue como se puede usar listas en poo mediante python ya que desconocía el cómo hacerlo en pyhton, la solución fue ver el video de referencia así como varios que están en internet.

Referencias

Primera referencia

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

Título: PySide2 - Clases y Objetos (Qt for Python)(II)

Autor: MICHEL DAVALOS BOITES

Código

Algoritmos.py

```
from cmath import sqrt
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)
```

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, 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, origen_y,
destino_x, 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'
        )
```

Administrador.py

```
from algoritmos import distancia_euclidiana
from particula import Particula
```

```

class Adminisrador:
    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)

101 = Particula(id=11, origen_x=10, origen_y=10, destino_x=20, destino_y=30,
                velocidad=200, red=1,
                green=3, blue=7, distancia=distancia_euclidiana)
102 = Particula(id=22, origen_x=8, origen_y=10, destino_x=28, destino_y=18,
                velocidad=100, red=3,
                green=6, blue=12, distancia=distancia_euclidiana)
particulas = Adminisrador()
particulas.agregar_inicio(101)
particulas.agregar_final(102)
particulas.mostrar()

```