

Actividad 08 – QTableWidget

Valencia frías Jonathan ricardo

Seminario de resolución de problemas de algoritmia

Lineamientos de evaluación

- El programa corre sin errores.
- El programa cuenta con todas las funciones
- Es capaz de realizar las tareas pedidas

Desarrollo

```
[  
  {  
    "id": 1,  
    "origen_x": 2,  
    "origen_y": 2,  
    "destino_x": 3,  
    "destino_y": 3,  
    "velocidad": 4,  
    "red": 1,  
    "green": 1,  
    "blue": 1  
  },  
  {  
    "id": 2,  
    "origen_x": 3,  
    "origen_y": 3,  
    "destino_x": 4,  
    "destino_y": 4,  
    "velocidad": 5,  
    "red": 2,  
    "green": 2,  
    "blue": 2  
  },  
  {  
    "id": 3,  
    "origen_x": 4,  
    "origen_y": 4,  
    "destino x": 5,
```

```
    "destino_x": 5,  
    "destino_y": 5,  
    "velocidad": 6,  
    "red": 3,  
    "green": 3,  
    "blue": 3  
  },  
  {  
    "id": 4,  
    "origen_x": 5,  
    "origen_y": 5,  
    "destino_x": 6,  
    "destino_y": 6,  
    "velocidad": 7,  
    "red": 4,  
    "green": 4,  
    "blue": 4  
  },  
  {  
    "id": 5,  
    "origen_x": 5,  
    "origen_y": 5,  
    "destino_x": 6,  
    "destino_y": 6,  
    "velocidad": 7,  
    "red": 5,  
    "green": 5,  
    "blue": 5  
  }  
]
```

Datos en el archivo json

	ld	origen_x	origen_y	destino_x	destino_y	vel
1	1	2	2	3	3	4
2	2	3	3	4	4	5
3	3	4	4	5	5	6
4	4	5	5	6	6	7
5	5	5	5	6	6	7


Datos mostrados en la tabla

	1	2	3	4	5	
1	1	2	2	3	3	4

Datos mostrados con el buscar

	ld	origen_x	origen_y	destino_x	destino_y	vel
1	1	2	2	3	3	4

Atencion



libro no encontrado

OK

6

buscar

mostrar

Buscando dato no existente

Conclusiones

En conclusión, se pudo conectar de manera correcta la interfaz con el proceso lógico de las clases

Referencias

<https://docs.python.org/3/>

Código

```
import math
def distancia_euclidiana(x1,x2,y1,y2):
    return math.sqrt(((x2-x1)**2)+((y2-y1)**2))

from _particula import Particula

class Lista_Partículas:
    lt_particula=list()

    def insertar_inicio(self,p):
        self.lt_particula.insert(0,p)
    def insertar_final(self,p):
        self.lt_particula.append(p)
    def mostrar(self):
        for i in range(len(self.lt_particula)):
            print(str(self.lt_particula[i])+"\n")

from algoritmos import distancia_euclidiana

class Particula:
    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

    def
__init__(self,id,origen_x,origen_y,destino_x,destino_y,velocidad,red,green,blue):
    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) -> str:
        return
        "("+str(self.id)+",""+str(self.origen_x)+",""+str(self.origen_y)+",""+str(self.destino_x)
        )+",""+str(self.origen_y)+",""+str(self.velocidad)+",""+str(self.blue)+",""+str(self.gree
        n)+",""+str(self.red)+",""+str(self.distancia)+")"
```