

```

1  #!/usr/bin/env python2
2  # -*- coding: utf8 -*-
3
4  """
5      Todo lo que trate de operaciones con archivos
6  """
7
8  ##Importaciones
9  import os, csv, sys
10 from grafo import Grafo
11 from texto import texto
12 from constantes import *
13
14
15
16 ##Funciones
17 def obtener_datos(ruta):
18     """wrapper a la subfuncion de validacion"""
19     return validar_archivo(ruta)
20
21 def validar_archivo(ruta):
22     """Corroborar que el archivo exista, no este vacio, y se pueda abrir"""
23
24     #Valido la existencia
25     if not os.path.isfile(ruta):
26         print msj_err_no_arc
27         return None, None, None
28     #valido que no este vacio
29     if not os.path.getsize(ruta):
30         print msj_err_vac_arc
31         return None, None, None
32
33     #Intento abrirlo
34     try:
35         handler = open(ruta, "r")
36
37         #Devuelvo los datos extraidos
38         retorno = _extraer_datos(handler)
39         handler.close()
40         return retorno
41
42     except:
43         print "Ocurrio un error al intentar abrir el archivo", sys.exc_info(
44 ) [0]
45         return None, None, None
46
47
48
49
50 def _extraer_datos(handler):
51     """Recibe: archivo, Devuelve: 2 grafos
52     Extrae los datos del archivo y los devuelve"""
53     #variables
54     fila = []
55     nodos_cant = 0
56     calles_cant = 0
57     i = 0 #indice generico
58     info_nodo = {} #donde guardo la informacion de los nodos
59
60     grafo_nodos = Grafo()
61     grafo_calles = Grafo()
62
63     #Cargo el manejador de csv
64     contenido_csv = csv.reader(handler, delimiter=",")

```

```

65 #obtengo la primer linea, que me dice cuantos -NODOS- son
66 fila = contenido_csv.next()
67 nodos_cant = int(fila[0])
68 #avanzo para obtener la primer fila de -NODOS-
69 fila = contenido_csv.next()
70
71 #obtengo los datos de los -NODOS-
72 for i in range(nodos_cant-1):
73     #guardo la info del nodo
74     info_nodo[ int(fila[0]) ] = {'x': float(fila[1]), 'y': float(fila[2
75 ]), 'lon': float(fila[4]), 'lat': float(fila[3])}
76     #avanzo una posicion si no estoy en el ultimo -NODO-
77     if i != nodos_cant:
78         fila = contenido_csv.next()
79 #obtengo la cantidad de -CALLES-
80 fila = contenido_csv.next()
81 calles_cant = int(fila[0])
82
83 #avanzo para obtener la primer fila de -CALLES-
84 fila = contenido_csv.next()
85 #obtengo los datos de las -CALLES-
86 for i in range(calles_cant-1):
87     if int(int(fila[4])) not in grafo_nodos:
88         grafo_nodos.agregar_vertice(int(fila[4]))
89     if texto(fila[1]) not in grafo_calles:
90         grafo_calles.agregar_vertice(texto(fila[1]))
91     #Agrego las aristas
92     grafo_nodos.agregar_arista(int(fila[4]),int(fila[5]),int(fila[2]))
93     grafo_calles.agregar_arista(texto(fila[1]),int(fila[4]),int(fila[5]
94 ))
95     #Cuando son mano unica
96     if int(fila[3]) == 0:
97         if int(fila[5]) not in grafo_nodos:
98             grafo_nodos.agregar_vertice(int(fila[5]))
99             grafo_nodos.agregar_arista(int(fila[5]),int(fila[4]),int(fila[2
100 ]))
101     #devuelvo los grafos e info_nodo
102     if i != calles_cant:
103         fila = contenido_csv.next()
104
105 return grafo_nodos, grafo_calles ,info_nodo
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121 #####<EOF
122

```