

```

1  #!/usr/bin/env python2
2  # -*- coding: utf-8 -*-
3
4  from grafo import *
5  from archivos import *
6  from texto import *
7  from interaccion import *
8  from sexykml import *
9  from constantes import *
10
11
12 def main():
13     mapa_ok = False
14
15
16     #~ Inicializo el kml
17     kml = SexyKML("Pizzeria Gerli, hoy Gerli, mañana... el mundo!")
18
19
20     #Consigo los grafos
21     while not mapa_ok:
22         ruta_mapa = obtener_texto(msj_mapa)
23         grafo_nodos, grafo_calles, info_nodos = obtener_datos(ruta_mapa)
24         if grafo_nodos and grafo_calles and info_nodos:
25             mapa_ok = True
26     #Busco la pizzeria
27     dir_pizzeria, vertice_pizzeria = ingresar_calles(grafo_calles, msj_pizzeria
)
28
29     #Debo obtener el vertice de grafo_nodos, por que los algoritmos se manejan
30     #con la posicion de memoria ( lo mismo con el cliente)
31     vertice_pizzeria = grafo_nodos.obtener_vertice(vertice_pizzeria.clave)
32     #Obtengo la info de la pizzeria
33     info_pizzeria = info_nodos[vertice_pizzeria.clave]
34     #Agrego el marcador de la pizzeria
35     #~ kml.agregar_marcador("Pizzeria", info_pizzeria["latitud"], info_pizzeria
["longitud"])
36
37     pizzeria = (vertice_pizzeria, dir_pizzeria, info_pizzeria)
38
39
40     while True:
41         #Bucle principal
42         opcion = menu()
43         if opcion == "3":
44             #SALIR
45             kml.finalizar()
46             print msj_kml_finalizado
47             return 0
48
49         elif opcion == "1":
50             #1)Generar ruta
51             #~ #Busco al cliente
52             dir_cliente, vertice_cliente = ingresar_calles(grafo_calles, msj
_cliente)
53             vertice_cliente = grafo_nodos.obtener_vertice(vertice_cliente.cl
ave)
54             info_cliente = info_nodos[vertice_cliente.clave]
55             cliente = (vertice_cliente, dir_cliente, info_cliente)
56             #Verifico que sean distintos el inicio del final
57             if pizzeria != cliente:
58                 print "Ida..."
59                 viaje(grafo_nodos, cliente, "Cliente", pizzeria, "Pizzeria", i
nfo_nodos, kml)

```

```

60         print "Vuelta..."
61         viaje(grafo_nodos, pizzeria, "Pizzeria", cliente, "Cliente", i
nfo_nodos, kml)
62
63         print msj_kml_ok
64     else:
65         print msj_err_lugar
66
67     elif opcion == "2":
68         #~ 2) Ruta e/puntos cualesquiera
69         #~ #VerticeA
70         dir_A, verticeA = ingresar_calles(grafo_calles, "Ingrese interse
ccion del vertice_A:")
71         verticeA = grafo_nodos.obtener_vertice(verticeA.clave)
72         #~ #VerticeB
73         dir_B, verticeB = ingresar_calles(grafo_calles, "Ingrese interse
ccion del vertice_B:")
74         verticeB = grafo_nodos.obtener_vertice(verticeB.clave)
75
76         info_A = info_nodos[verticeA.clave]
77         info_B = info_nodos[verticeB.clave]
78
79         A = (verticeA, dir_A, info_A)
80         B = (verticeB, dir_B, info_B)
81         #Verifico que sean distitnos el inicio del final
82         if A != B:
83             print "Ida..."
84             viaje(grafo_nodos, A, "VerticeA", B, "VerticeB", info_nodos,
kml)
85             #el regreso
86             print "Vuelta..."
87             viaje(grafo_nodos, B, "VerticeB", A, "VerticeA", info_nodos,
kml)
88
89             print msj_kml_ok
90         else:
91             print msj_err_lugar
92
93
94 if __name__ == '__main__':
95     main()
96
97

```