

Definição da melhor localização para instalação de um hospital de campanha COVID-19



Ano Letivo: 2019/2020

Docentes: António Coelho; Ricardo Batista.

Discentes: Bruno Costa; Joana Teixeira; Rui Andrade.

Índice

- ✓ Introdução;
- ✓ Descrição geral do Script desenvolvido;
- ✓ Funcionalidades implementadas e por implementar;
- ✓ Detalhes relevantes de implementação;
- ✓ Principais dificuldades encontradas;
- ✓ Conclusão;
- ✓ Anexos.

Introdução

Programar corresponde a um processo de escrita, teste e manutenção de um programa de computador, o programa é escrito numa linguagem de programação, existem diferentes linguagens de programação que funcionam de diferentes modos. *Python* é uma linguagem de programação interpretada, de script, imperativa, orientada a objetos.

Este trabalho surge no âmbito da unidade curricular de Programação em Sistemas de Informação Geográfica e Ordenamento do Território, onde o objetivo do trabalho é a elaboração de uma biblioteca *Python* que execute operações através de dados geográficos e tabulares.

O objetivo principal é definir a melhor localização para um hospital de campanha dedicado à pandemia Covid-19, tendo em conta as várias regiões/municípios de Portugal, os resultados deverão ser apresentados em formato raster.

Para isso iremos recorrer à biblioteca *ArcPy* integrada no software ArcGIS 10.7, usando o sistema de coordenadas ETRS 1989 Portugal TM06.

Descrição geral do Script desenvolvido

Inicialmente foram importadas as bibliotecas necessárias e definidos os 2 sistemas de coordenadas usados no script, WGS84 e ETRS 1989 Portugal TM06.

- ✓ De seguida, na construção do script foram definidas as variáveis `workspace_path`, `pasta_resultados` e `pasta_tabela` de forma a dar liberdade ao utilizador de definir os seus diretórios.

```
C:\Users\35191\Desktop> Trabalho Final Programação.py > ...
1  #_*__.coding:.utf-8-_*_
2  import arcpy
3  from arcpy import env, da
4  import os
5
6  etrs89 = arcpy.SpatialReference(3763)
7  wgs84 = arcpy.SpatialReference(4326)
8
9  workspace_path = arcpy.GetParameterAsText(0)
10 pasta_resultados = arcpy.GetParameterAsText(1)
11 pasta_tabelas = arcpy.GetParameterAsText(2)
12
13
14 workspace_process = pasta_resultados
15 workspace_tables = pasta_tabelas
16
17 arcpy.env.workspace = workspace_path
18 arcpy.env.overwriteOutput = True
```

- ✓ Neste ponto está a ser criada uma shapefile através das coordenadas dadas, estas encontram-se em WGS84, porém terão de ser convertidas. O código desenvolvido converte, as mesmas, numa Layer Temporária que irá, mais tarde, ser convertida para shapefile, sendo que no final esta shapefile é projetada para o sistemas de coordenadas ETRS 1989 Portugal TM06.
- ✓ É dada a oportunidade ao utilizador de escolher o ficheiro Excel, restringindo-o ao nome da folha “Centros Hospitalares”.
- ✓ Este processo repete-se para criação da shapefile dos pontos dos Hospitais.

```

19
20 #Centros
21 xls_centros_h = arcpy.GetParameterAsText(3)
22 arcpy.ExcelToTable_conversion(xls_centros_h, pasta_resultados + "\\centros_h.dbf", "Centro_Hospitalares")
23 arcpy.MakeXYEventLayer_management(pasta_resultados + "\\centros_h.dbf", "X", "Y", "centros_h", wgs84)
24 arcpy.FeatureClassToFeatureClass_conversion("centros_h", workspace_process, "centro_h_wgs84")
25 arcpy.Project_management(workspace_process + "\\centro_h_wgs84.shp", workspace_process + "\\centro_h_etrs.shp", etrs89)
26
27 #Hospitais
28 xls_Hospitais = arcpy.GetParameterAsText(4)
29 arcpy.ExcelToTable_conversion(xls_Hospitais, pasta_resultados + "\\hospitais_h.dbf", "Hospitais")
30 arcpy.MakeXYEventLayer_management(pasta_resultados + "\\hospitais_h.dbf", "X", "Y", "Hospitais_lyr", wgs84)
31 arcpy.FeatureClassToFeatureClass_conversion("Hospitais_lyr", workspace_process, "Hospitais_wgs84")
32 arcpy.Project_management(workspace_process + "\\Hospitais_wgs84.shp", workspace_process + "\\hospitais_etrs.shp", etrs89)

```

- ✓ Mais uma vez o ficheiro CAOP é definido como “arcpy.GetParameterAsText” o que significa que é um Input à escolha do utilizador.
- ✓ Em seguida foi necessário criar um campo com o nome de “Dico”, com o intuito de copiar para esse campo os 4 primeiros números à esquerda, do campo “Dicofre”.
- ✓ Dessa forma foi realizado um Dissolve para a CAOP, levando consigo os campos “Concelho” e “Dico”.
- ✓ Na CAOP dissolvida, foram adicionados 2 campos, DensPop, e AreaKm2, onde através do “arcpy.CalculateField_management” foi calculada a área.

```

33
34 #Dissolve
35 CAOP = arcpy.GetParameterAsText(5)
36
37 arcpy.AddField_management(CAOP, "DICO", "TEXT", "", "", 4)
38 arcpy.CalculateField_management(CAOP, "DICO", "LEFT([Dicofre], 4)")
39
40 arcpy.Dissolve_management(CAOP, workspace_process + "\\CAOP DISSOLVE", ["Concelho", "DICO"])
41 arcpy.AddField_management(workspace_process + "\\CAOP DISSOLVE.shp", "DensPop", "DOUBLE", "", "", "", "", "NULLABLE")
42 arcpy.AddField_management(workspace_process + "\\CAOP DISSOLVE.shp", "AreaKm2", "DOUBLE", "", "", "", "", "NULLABLE")
43 CAOP DISSOLVE = workspace_process + "\\CAOP DISSOLVE.shp"
44
45 arcpy.CalculateField_management(CAOP DISSOLVE, "Areakm2", "float(!SHAPE.AREA@SQUAREKILOMETERS!)", "PYTHON")

```

- ✓ Para esta parte denotou-se uma grande agilidade na selecção do ficheiro Excel, sendo que a folha foi restringida à “PopResidente”.
- ✓ De seguida a tabela Excel foi convertida para dbf, e depois um Join da CAOP_Dissolve com a tabela dbf, criada anteriormente
- ✓ A Densidade populacional foi calculada através da expressão que está presente na linha 50

```
46
47 idtsat = arcpy.GetParameterAsText(6)
48 arcpy.ExcelToTable_conversion(idtsat, pasta_resultados + "\\idsta_join.dbf", "PopResidente")
49 arcpy.JoinField_management(CAOP DISSOLVE, "DICO", pasta_resultados + "\\idsta_join.dbf", "Cod")
50 arcpy.CalculateField_management(CAOP DISSOLVE, "DensPop", "float(!PR_2011!/!AreaKm2!)", "PYTHON")
```

- ✓ Juntar os Hospitais e os Centros Hospitalares numa *shapefile* de pontos.

```
51
52 #RACIO
53 arcpy.Merge_management([workspace_process + "\\hospitais_etr.shp", workspace_process + "\\centro_h_etr.shp"], workspace_process + "\\merge_hch")
54 arcpy.SpatialJoin_analysis(CAOP DISSOLVE, workspace_process + "\\merge_hch.shp", workspace_process + "\\racio_pontos")
55 racio_pontos = workspace_process + "\\racio_pontos.shp"
56
```

- ✓ Como Get_parameter, foi inserida a tabela com os valores dos infetados por Covid.
- ✓ Essa tabela foi convertida para dbf para depois, se fazer um Join entre o Merge dos pontos dos Hospitais e Centros Hospitalares

```
57 #COVID
58 covid_excel = arcpy.GetParameterAsText(7)
59 arcpy.ExcelToTable_conversion(covid_excel, pasta_resultados + "\\covid.dbf", "Convid")
60 DATA = workspace_process + "\\racio_pontos.shp"
61 arcpy.JoinField_management(DATA, "DICO", pasta_resultados + "\\covid.dbf", "DICO")
```

- ✓ Conversão da tabela dos indicadores estatísticos , onde é usada a folha COD, onde estão os códigos das NUTS III.
- ✓ Depois dessa conversão, foi feito um Join com a shapefile dos pontos dos Hospitais e dos Centros Hospitalares, já com o Merge entre e as 2.
- ✓ Depois disso realizou-se um Dissolve desse Merge pelas NUTS III, onde os campos enunciados na linha 69 foram os recolhidos..

```
62
63 #NUT3
64
65 nut3_dbf = workspace_process + "\\nut3.dbf"
66
67 arcpy.ExcelToTable_conversion(idtsat, pasta_resultados + "\\nut3.dbf", "COD")
68 arcpy.JoinField_management(DATA, "DICO", nut3_dbf, "DICO")
69 arcpy.Dissolve_management(DATA, workspace_process + "\\nut", "NUTIII Cod", "DensPop SUM; AreaKm2 SUM; PR_2011 SUM; PR25_64_20 SUM; PR65_2011 SUM; NUTIII FIRST")
```

- ✓ Seleção da Cos e posterior conversão da mesma para o Layer Temporários
- ✓ Seleção desse Layer Temporário pelos campos da linha 75.
- ✓ Gravação dessa layer tempor selecionada como shape com o nome "Cos_Select"

```
70
71 #COS
72 COS = arcpy.GetParameterAsText(8)
73 arcpy.MakeFeatureLayer_management(COS, "layer_cos", "", workspace_process)
74
75 arcpy.SelectLayerByAttribute_management("layer_cos", "NEW_SELECTION", "COS2018_n4 LIKE '1.1.3%' OR COS2018_n4 LIKE '1.6.1%' OR COS2018_n4 LIKE '1.6.5%' OR COS2018_n4 LIKE '1.7.1%'")
76 arcpy.FeatureClassToFeatureClass_conversion("layer_cos", workspace_process, "COS_Select.shp")
```

- ✓ Foi selecionado o concelho com densidade superior a 100 hab/km2 através de uma nova seleção com a ferramenta "SelectLayerbyAttribute_management".


```
77
78 #Estradas
79 ESTRADAS = arcpy.GetParameterAsText(9)
80
81 #DensPop
82 arcpy.MakeFeatureLayer_management(CAOP DISSOLVE, "lyr_denspop", "", workspace_process)
83 arcpy.SelectLayerByAttribute_management("lyr_denspop", "NEW_SELECTION", "DensPop > 100")
```

- ✓ Através do “Get ParameterAsText” foi inserida a shapefile das estradas. Com ferramenta “SelectLayerbyAttribute_management” foram selecionadas vias rodoviárias de 500 m a 3000 m do tipo IC e IP.

```
84
85 #Estradas
86 arcpy.MakeFeatureLayer_management(ESTRADAS, "lyr_estradas", "", workspace_process)
87 arcpy.SelectLayerByAttribute_management("lyr_estradas", "NEW_SELECTION", "LENGTH > 500 AND LENGTH < 3000")
88 arcpy.SelectLayerByAttribute_management("lyr_estradas", "SUBSET_SELECTION", "ITINERARIO LIKE 'IC%' OR ITINERARIO LIKE 'IP%'")
89 arcpy.FeatureClassToFeatureClass_conversion("lyr_estradas", workspace_process, "vias_rod")
```

- ✓ Através do “Intersect” entre as 2 layer temporárias, “denspop e estradas”, já selecionadas por atributos, anteriormente, foi realizada uma nova “subselecção”, e através do parâmetro “Invert” foi possível obter tudo o que não se intersectava.
- ✓ Depois foi gravada essa seleção, tudo o que não se intersectava, como uma shapefile com nome “municipios_extract”.
- ✓ Em seguida foi realizada uma seleção por localização do layer temporário da densidade populacional, onde foram selecionadas os/o hospital/centro hospitalar a menos de 50 km.

```
90
91 #Municipios
92 arcpy.SelectLayerByLocation_management("lyr_denspop", "INTERSECT", "lyr_estradas", "", "SUBSET_SELECTION", "INVERT")
93 arcpy.FeatureClassToFeatureClass_conversion ("lyr_denspop", workspace_process, "municipios_extract")
94 arcpy.management.SelectLayerByLocation("lyr_denspop", "WITHIN_A_DISTANCE", workspace_process + "\\merge_hch.shp", "50 Kilometers", "SUBSET_SELECTION", "INVERT")
95 arcpy.FeatureClassToFeatureClass_conversion ("lyr_denspop", workspace_process, "selecao_final")
```

- ✓ Começamos por converter a tabela com os resultados dos contaminados para dbf, onde depois foi feito um Join com a CAOP dissolvida.
- ✓ De seguida foi criada uma lista vazia com intuito de obtermos, sempre, a última coluna da tabela. Para isso utilizamos a instrução Field_name [-1].
- ✓ Após isto, executamos um cálculo na coluna criada anteriormente, linha 110.
- ✓ Depois criamos um Layer Temporário “DenInfe”, sendo que depois foi realizada a seleção por atributos da linha 112.
- ✓ Por fim foi criamos uma shapefile “Ifetados_sup01” através da seleção feita anteriormente.

```
96
97 # municípios com igual ou superior a 0,1% de população do município infetada
98
99 arcpy.ExcelToTable_conversion (covid_excel, pasta_resultados + "\\covid_total.dbf", "Convid")
100 arcpy.JoinField_management (CAOP DISSOLVE, "DICO", pasta_resultados + "\\covid_total.dbf", "DICO")
101
102 field_name = []
103 fields = arcpy.ListFields(pasta_resultados + "\\covid_total.dbf")
104 for f in fields:
105     field_name.append(f.name)
106
107 ultimo_campo = field_name[-1]
108
109 arcpy.AddField_management(CAOP DISSOLVE, "DenInfe", "DOUBLE", "", "", "", "", "NULLABLE")
110 arcpy.CalculateField_management(CAOP DISSOLVE, "DenInfe", "(1 + ultimo_campo + !*1.0 / IPR_2011!*1.0) * 100", "PYTHON")
111 arcpy.MakeFeatureLayer_management(CAOP DISSOLVE, "DenInfe", "", workspace_process)
112 arcpy.SelectLayerByAttribute_management("DenInfe", "NEW_SELECTION", "DenInfe >= 0.1")
113 arcpy.FeatureClassToFeatureClass_conversion ("DenInfe", workspace_process, "Ifetados_sup01")
```

- ✓ Nesta etapa foi criado um Layer Temporário do campo “Join_Count” da shapefile “racio_pontos”, desse layer foi feita uma seleção, como podemos observar na linha 118. Essa seleção foi posteriormente guardada como shafile.

```
114
115 #Municípios onde o rácio de Hospitais e de Centros e Hospitalares é superior ou igual 1.0
116 racio_pontos = workspace_process + "\\racio_pontos.shp"
117 arcpy.MakeFeatureLayer_management(racio_pontos, "Join_Count", "", workspace_process)
118 arcpy.SelectLayerByAttribute_management("Join_Count", "NEW_SELECTION", "Join_Count >= 1")
119 arcpy.FeatureClassToFeatureClass_conversion ("Join_Count", workspace_process, "racio_H_CH_1")
```

- ✓ Foi adicionado um campo “Dens25_64”;
- ✓ Posteriormente esse campo foi calculado através da expressão patente na linha 124, com o intuito de obter a densidade populacional de pessoas entre 25 e 64 anos.
- ✓ De seguida utilizamos as ferramentas SearchCursor e um ciclo for para retirar o valor médio, como podemos observar na linha 127 à 130;
- ✓ Por fim exportamos uma shafile com a seleção da “Dens25_64” superior à “média_nacional”.

```
120
121 #Municípios com maior Densidade Populacional da faixa etária 25-64 anos, superior à média nacional
122
123 arcpy.AddField_management(CAOP DISSOLVE, "Dens25_64", "DOUBLE", "", "", "", "", "", "NULLABLE")
124 arcpy.CalculateField_management(CAOP DISSOLVE, "Dens25_64", "float(!PR25_64_20!/!AreaKm2!)", "PYTHON")
125
126 arcpy.Statistics_analysis(CAOP DISSOLVE, workspace_process + "\\25_65_mean.dbf", [{"Dens25_64", "MEAN"}])
127 cursor_nacional = arcpy.SearchCursor(workspace_process + "\\25_65_mean.dbf")
128
129 for row in cursor_nacional:
130     media_nacional = str(row.MEAN_Dens2)
131
132 arcpy.MakeFeatureLayer_management(CAOP DISSOLVE, "Denslyr", "", workspace_process)
133 arcpy.SelectLayerByAttribute_management("Denslyr", "NEW_SELECTION", "Dens25_64 >" + media_nacional)
134 arcpy.FeatureClassToFeatureClass_conversion ("Denslyr", workspace_process, "sup_media")
```

- ✓ Neste tópico foi feita a conversão da tabela, valores infetados, para dbf.
- ✓ De seguida foi feito um Join entre a “CAOP_Dissolve” e a tabela criada, pelo campo “Dico”.
- ✓ Posteriormente foi criado uma lista onde foram adicionados os nomes das colunas e desses nomes, foi selecionada a última coluna. Para isso utilizamos a instrução Field_name [-1], como já foi realizado anteriormente.
- ✓ Adicionamos um novo campo chamado “infetados” à “CAOP_Dissolve” e calculamos esse campo através da atribuição da variável “ultimo_campo”. Desse modo garantimos que a informação deste campo seja sempre a da última coluna da tabela.
- ✓ Com o intuito de extrair a média deste último campo voltamos a fazer o Statics_Analysis.
- ✓ Aplicamos novamente o SearchCursor para obtermos o valor médio extraído no dbf

- ✓ Por fim com todos os dados necessários obtidos foi executada uma seleção por atributos onde o valor dos infetados é superior ao da média.

```

135
136 #Municípios com maior número de contaminados COVID sem qual hospital/centro hospitalar no município
137
138 arcpy.ExcelToTable_conversion(covid_excel, pasta_resultados + "\\covid_total.dbf", "Convid")
139 arcpy.JoinField_management(CAOP DISSOLVE, "DICO", pasta_resultados + "\\covid_total.dbf", "DICO")
140
141 field_name = []
142 fields = arcpy.ListFields(pasta_resultados + "\\covid_total.dbf")
143 for f in fields:
144     field_name.append(f.name)
145
146 ultimo_campo = field_name[-1]
147
148 arcpy.AddField_management(CAOP DISSOLVE, "infetados", "DOUBLE", "", "", "", "", "NULLABLE")
149 arcpy.management.CalculateField(CAOP DISSOLVE, "infetados", "!" + ultimo_campo + "!", "PYTHON")
150
151 covid_infet = workspace_tables + "\\covid_infetados.dbf"
152
153 arcpy.ExcelToTable_conversion(covid_excel, covid_infet, "Convid")
154 arcpy.Statistics_analysis(CAOP DISSOLVE, covid_infet, [{"infetados", "MEAN"}])
155
156 media_contaminados = arcpy.SearchCursor(covid_infet)
157
158 for row in media_contaminados:
159     media_contaminados = str(row.MEAN_infet)
160
161 arcpy.MakeFeatureLayer_management(CAOP DISSOLVE, "inflyr", "", workspace_process)
162 arcpy.SelectLayerByAttribute_management("inflyr", "NEW_SELECTION", "infetados >" + media_contaminados)
163 arcpy.FeatureClassToFeatureClass_conversion("inflyr", workspace_process, "municipios_supmedia")
164
165 infmedia = workspace_process + "\\municipios_supmedia.shp"
166 pontos_eh = workspace_process + "\\merge_hch.shp"
167
168 arcpy.MakeFeatureLayer_management(infmedia, "lyr_municipios", "", workspace_process)
169 arcpy.MakeFeatureLayer_management(pontos_eh, "lyr_pontos_eh", "", workspace_process)
170 arcpy.SelectLayerByLocation_management("lyr_municipios", "INTERSECT", "lyr_pontos_eh", "", "NEW_SELECTION", "INVERT")
171 arcpy.FeatureClassToFeatureClass_conversion("lyr_municipios", workspace_process, "intersect_municipios")
  
```

- ✓ Com os resultados obtidos da questão anterior foi realizada uma nova seleção onde foram intersectados os municípios com número de infetados superior à média, e os Hospitais/Centros Hospitalares,
- ✓ Aplicamos a função Invert, pois queríamos identificar os municípios com maior número de contaminados COVID sem qual hospital/centro hospitalar no município.

```

172
173
174 #Município com maior população infetada da área de estudo
175
176 arcpy.Statistics_analysis(CAOP DISSOLVE, workspace_process + "\\MAX_MUN.dbf", [{"infetados", "MAX"}])
177 max_infect = arcpy.SearchCursor(workspace_process + "\\MAX_MUN.dbf")
178
179 for row in max_infect:
180     max_nacional = str(row.MAX_infeta)
181
182 arcpy.MakeFeatureLayer_management(CAOP DISSOLVE, "MAX_lyr", "", workspace_process)
183 arcpy.SelectLayerByAttribute_management("MAX_lyr", "NEW_SELECTION", "infetados =" + max_nacional)
184 arcpy.FeatureClassToFeatureClass_conversion("MAX_lyr", workspace_process, "MAX_MUNICIPIO")
  
```

- ✓ Criação de um dbf com o número máximo de infetados, pertencentes à tabela "infetados".

- ✓ Colocação desse valor num SearchCursor e realização de uma seleção por atributos, onde o número de infetados é igual ao valor máximo extraído anteriormente.

```
185
186
187 #Classe de uso de solo (COSn3 --1.1.3, 1.6.1, 1.6.5, 1.7.1)
188
189 arcpy.MakeFeatureLayer_management(COS, "layer_cos", "", workspace_process)
190 arcpy.SelectLayerByAttribute_management("layer_cos", "NEW_SELECTION", "COS2018_n4 LIKE '1.1.3%' OR COS2018_n4 LIKE '1.6.1%' OR COS2018_n4 LIKE '1.6.5%' OR COS2018_n4 LIKE '1.7.1%'")
191 arcpy.FeatureClassToFeatureClass_conversion("layer_cos", workspace_process, "COS_select.shp")
192 solo_adm = workspace_process + "\\COS_select.shp"
```

- ✓ Criação de uma nova seleção aos IP's, IC's e EN.
- ✓ Uma vez que era pedido raio de 500m e 3000m destas vias usamos a ferramenta Buffer. Para obter o resultado desejado usamos um Erase_analysis.
- ✓ Efetuamos um Intersect entre as 2 shapefiles.

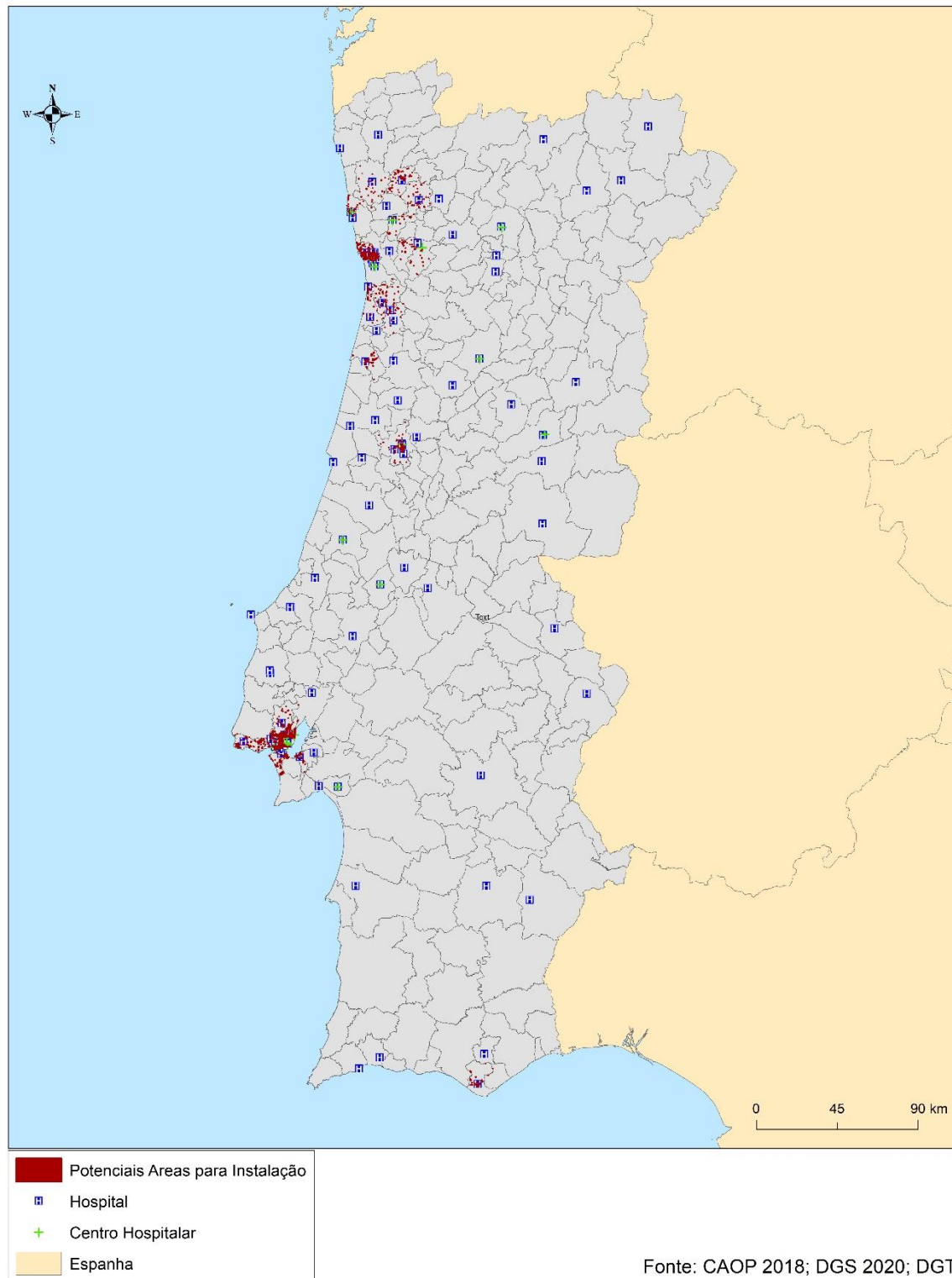
```
193
194 # raio de 500m e 3000m de qualquer vias rodoviárias do tipo IC, IP e EN
195
196 ESTRADAS = workspace_process + "\\ESTRADAS_ETRS89.shp"
197 arcpy.AddGeometryAttributes_management(ESTRADAS, "LENGTH", "METERS")
198 arcpy.MakeFeatureLayer_management(ESTRADAS, "lyr_estradas", "", workspace_process)
199 arcpy.SelectLayerByAttribute_management("lyr_estradas", "NEW_SELECTION", "ITINERARIO LIKE 'IC%' OR ITINERARIO LIKE 'IP%' OR VIA LIKE 'EN%'")
200 arcpy.FeatureClassToFeatureClass_conversion("lyr_estradas", workspace_process, "IC_IP_EN")
201
202 IC_IP_EN = workspace_process + "\\IC_IP_EN.shp"
203 arcpy.Buffer_analysis(IC_IP_EN, workspace_process + "\\buffer_500", 500, "", "", "ALL")
204 arcpy.Buffer_analysis(IC_IP_EN, workspace_process + "\\buffer_3000", 3000, "", "", "ALL")
205 arcpy.Erase_analysis(workspace_process + "\\buffer_3000.shp", workspace_process + "\\buffer_500.shp", workspace_process + "\\buffer3000_500")
206 estradas_adm = workspace_process + "\\buffer3000_500.shp"
207
208 arcpy.analysis.Intersect(estradas_adm + ";" + solo_adm, workspace_process + "\\intersect_adm", "ALL", None, "INPUT")
```

- ✓ Para a definição das áreas de grande potencial para a melhor localização de um hospital de campanha Covid-19 foi utilizada a ferramenta Intersect com as shapefiles obtidas anteriormente.

```
209
210 # Definição de áreas de grande potencial para a melhor localização
211
212 munic_sup_01 = workspace_process + "\\Infetados_sup01.shp"
213 Racio_H_CH = workspace_process + "\\Racio_H_CH_1.shp"
214 dens_pop_25_64 = workspace_process + "\\sup_media.shp"
215
216 arcpy.analysis.Intersect(dens_pop_25_64 + ";" + Racio_H_CH + ";" + solo_adm + ";" + munic_sup_01 + ";" + estradas_adm, workspace_process + "\\areas_potencial", "ALL", None, "INPUT")
217
218 print "DONE"
```

Com base no mapa criado podemos concluir que os concelhos com maior potencialidade de implementação ou instalação de um hospital de campanha são de Santo Tirso e Lisboa.

Áreas de potencial para instalação de hospital de campanha COVID-19



Funcionalidades implementadas e por implementar

Ao longo do trabalho foram implementadas algumas funcionalidades. Entre elas, a funcionalidade Field_name [-1] que nos permite obter, sempre, o último campo da tabela.

Utilizamos também a criação de listas de modo a podermos trabalhar com valores obtidos de tabelas.

Infelizmente tornou-se impossível a concretização da ToolBox devido a um erro constante e totalmente incontornável, pelo grupo.

ERROR 000735: Input Excel File: Value is required

Principais dificuldades encontradas

Ao longo da elaboração deste trabalho foram encontradas diversas dificuldades, porém as mesmas também foram colmatadas através de pesquisa e consulta na internet e, claro, várias tentativas realizadas.

Muitas das dificuldades encontradas basearam-se em alguma falta de conhecimento e à vontade para responder às problemática propostas, este processo de aprendizagem e resolução das problemáticas revelou-se moroso e bastante complexo.

Alguns exemplos dessas dificuldades foram:

- A realização do Dissolve por NUT III revelou-se problemática uma vez que não conseguimos de forma célere chegar ao resultado, pois ao realizar o dissolve este não agrupava os valores por NUT.
- Na seleção por atributos foram encontradas algumas dificuldades devido ao uso das “plicas” nos operadores lógicos.
- A elaboração do rácio de Hospitais e de Centros e Hospitalares não foi executada com a facilidade esperada pois não chegamos a uma conclusão em concreto sobre o que era pedido.
- Para responder à questão (Município com maior número de contaminados sem qualquer hospital), tornou-se difícil selecionar apenas última coluna da tabela.
- Durante a elaboração da Toolbox, a criação de parâmetros revelou-se problemática e bastante demorada.

Conclusão

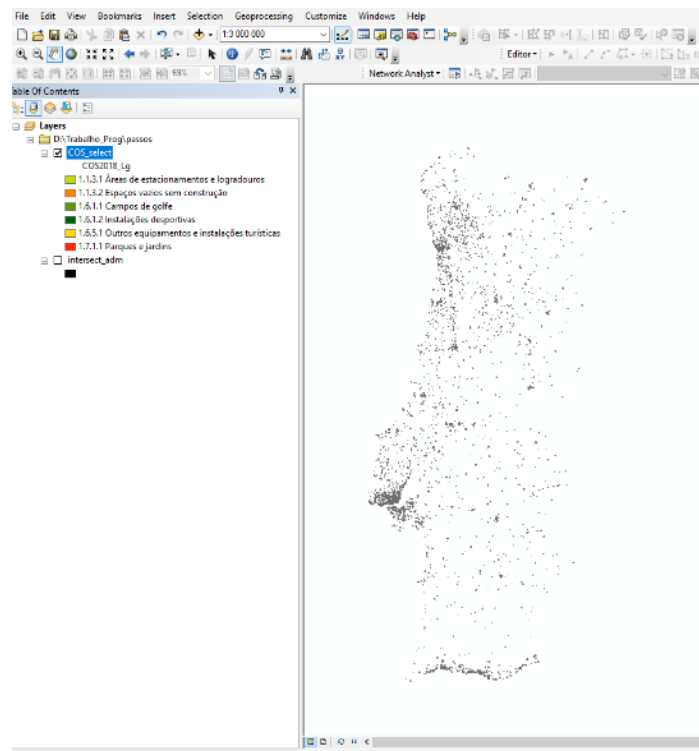
Através da elaboração deste trabalho foi possível desenvolver as nossas aptidões na construção de uma biblioteca Python, que no futuro nos irá facilitar em diversas ações durante o desenvolvimento de um projeto com os SIG, e, em especial, com o ArcGis.

Podemos concluir que este trabalho foi fulcral para nos tornar mais aptos a encarar a Programação e a sua linguagem que muitas vezes não é tao complicada quanto a fazemos.

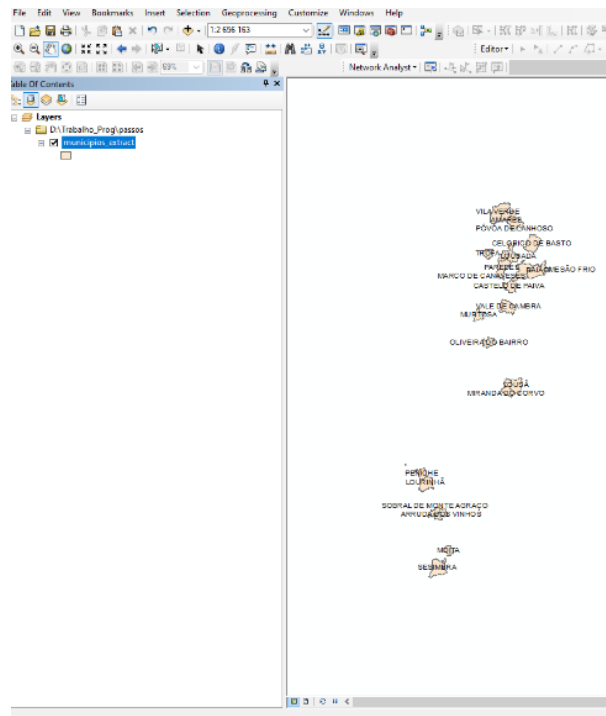
Com base no trabalho desenvolvido concluimos que a melhor localização para a instalação de um hospital de campanha contra o Covid19 deverá situar-se no concelho de Lisboa ou Santo Tirso.

Anexos

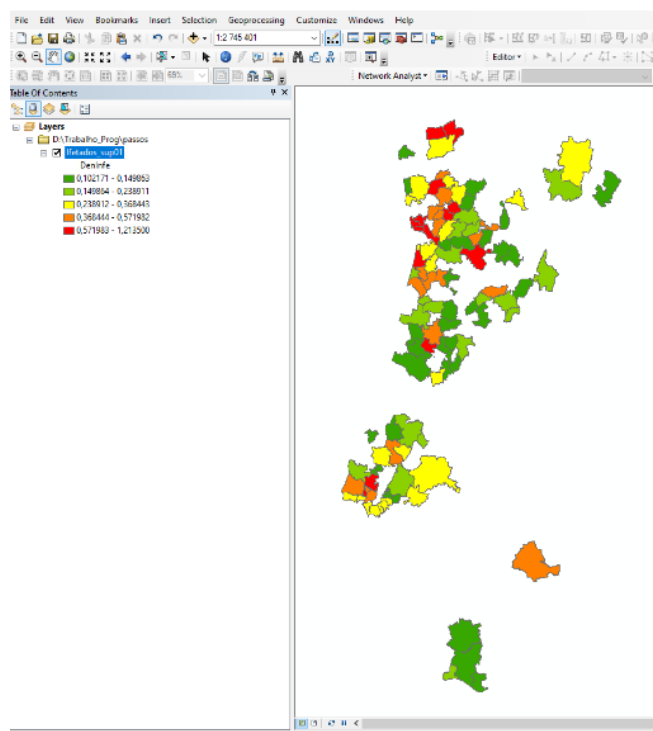
- ✓ Para identificar as áreas com classe de uso de solo admissível (Classe COSn3 – 1.1.3 espaços vazios em tecido edificado, 1.6.1 equipamentos desportivos, 1.6.5 outros equipamentos e instalações turísticas, 1.7.1 parque e jardins)



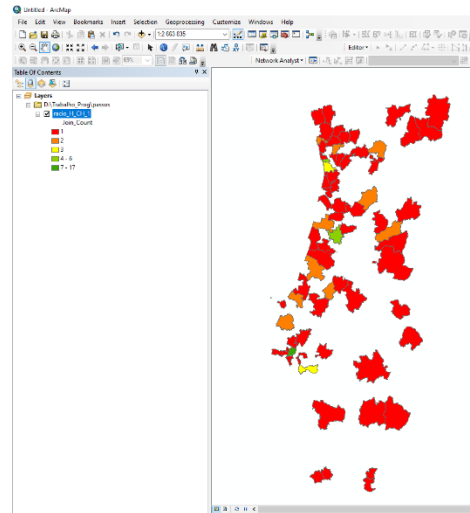
- ✓ Para identificar os municípios com densidade superior a 100 hab/km2 que não tenham vias rodoviárias de 500 m a 3000 m do tipo IC e IP e um hospital/centro hospitalar a menos de 50 km.



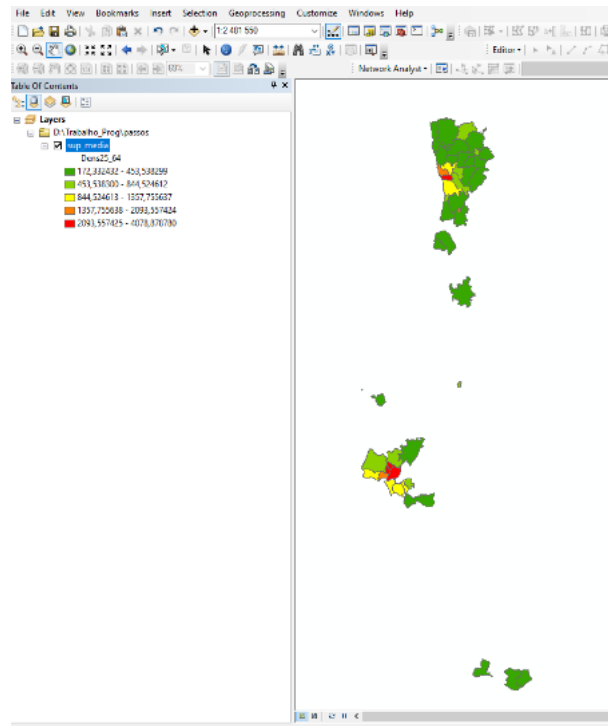
- ✓ Para identificar os municípios com igual ou superior a 0,1% de população do município encontra-se contaminados.



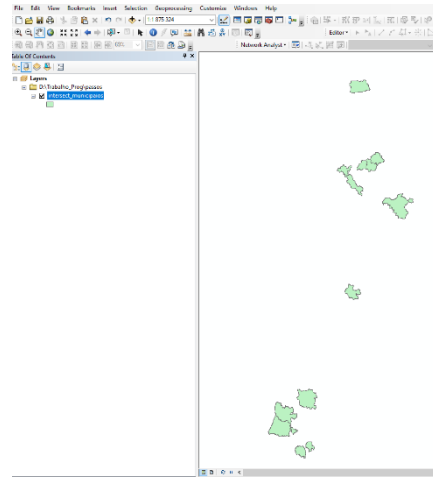
- ✓ Para identificar os municípios onde o rácio de Hospitais e de Centros e Hospitalares é superior ou igual 1.0.



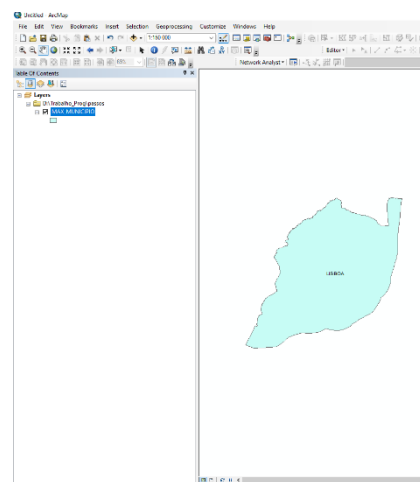
- ✓ Para identificar os municípios com maior Densidade Populacional da faixa etária 25-64 anos, superior à média nacional.



- ✓ Para identificar os municípios com maior número de contaminados COVID sem qual hospital/centro hospitalar no município.



- ✓ Para identificar o município com maior população infetada da área de estudo.



- ✓ Para identificar áreas com classe de uso de solo admissível (Classe COSn3 –1.1.3 espaços vazios em tecido edificado, 1.6.1 equipamentos desportivos, 1.6.5 outros equipamentos e instalações turísticas, 1.7.1 parque e jardins) num raio de 500 m e 3000 m de qualquer vias rodoviárias do tipo IC, IP e EN.

