

**Mestrado em Sistemas de Informação Geográfica e Ordenamento do Território**

Érica Amorim<sup>1</sup>  
João Gonçalves<sup>1</sup>  
Laura Fernandes<sup>1</sup>  
Lara Marques<sup>1</sup>

<sup>1</sup>Faculdade de Letras da Universidade do Porto, [up202308812@up.pt](mailto:up202308812@up.pt)

<sup>1</sup>Faculdade de Letras da Universidade do Porto, [up202005964@up.pt](mailto:up202005964@up.pt)

<sup>1</sup>Faculdade de Letras da Universidade do Porto, [up202007592@up.pt](mailto:up202007592@up.pt)

<sup>1</sup>Faculdade de Letras da Universidade do Porto, [up201907044@up.pt](mailto:up201907044@up.pt)

# **Análise Espacial aos Monumentos do Porto**

**Unidade Curricular de Programação SIG e OT**

**Docentes:** António Fernando Vasconcelos Cunha Castro Coelho  
Ricardo José Vieira Baptista

**Faculdade de Letras da Universidade do Porto**  
**Ano letivo: 2023/2024**

**Índice:**

Introdução	3
Metodologia	4
Enquadramento Geográfico da Área de Estudo	5
Análise da Espacial da Temática em Estudo	6
Conclusões Finais	20

## Introdução

A cidade do Porto, localizada no noroeste de Portugal, é uma das mais antigas e culturalmente ricas do país, possuindo uma vasta herança histórica que se reflete nos seus inúmeros monumentos e edifícios históricos. Estes monumentos não são apenas marcos arquitetónicos, mas também testemunhos vivos de diferentes épocas e eventos que moldaram a cidade ao longo dos séculos. A análise espacial dos monumentos do Porto permite explorar a distribuição, significância e acessibilidade destes, oferecendo uma perspectiva integrada que combina aspectos geográficos, históricos e culturais.

Através da utilização de ferramentas de Sistemas de Informação Geográfica (SIG) e técnicas de análise espacial, é possível mapear e examinar a disposição geográfica dos monumentos, identificar padrões de concentração e dispersão, bem como entender a relação desses marcos com os alojamentos locais.

No âmbito da Unidade Curricular de Programação em SIG e OT foi orientado aos alunos que formassem grupos, com o objetivo de definir temas de trabalho/estudo cuja problemática pretende solucionar o seguinte tema do turismo, associado aos monumentos.

O grupo 3 reuniu-se, pelo que se formou então o tema: “Análise Espacial aos Monumentos do Porto”. Esta temática tem como principal objetivo o estudo da distribuição espacial dos monumentos do município do Porto, através do uso da linguagem de programação Python que foi crucial no desenvolvimento do trabalho até se obter os dados para serem projetados no mapa.

Este estudo propõe uma abordagem detalhada para a análise espacial dos monumentos do Porto, começando pela coleta de dados geográficos e históricos, que posteriormente foram manipulados com recurso ao python e só depois é que se passou para a criação de mapas temáticos e interpretação dos mesmos com o devido carácter geográfico. Ao revelar os padrões espaciais e as interações entre os diferentes elementos do património cultural da cidade, espera-se contribuir para uma maior valorização e preservação dos monumentos do Porto, assim como para a melhoria da experiência dos turistas e dos respetivos residentes.

## Metodologia

No que diz respeito à metodologia, as fontes de dados utilizadas para esta pesquisa correspondem às secundárias de método tradicional, nomeadamente: o *website ForestGIS* na obtenção da CAOP 2011; o INE (Instituto Nacional de Estatística) para adquirir dados acerca da população residente em 2011 nas freguesias do município do Porto; o *Travel BI* para recolher a localização dos alojamentos locais e, por último ao Portal de Dados Abertos da Cidade do Porto para obter a localização dos monumentos. Recorreu-se ainda à utilização dos vários tutoriais fornecidos pelos docentes acerca do manuseamento de código em python, bem como outras fontes na *web*.

Face à cartografia produzida, a mesma foi elaborada com recurso ao *software ArcGIS Pro*, sendo que todas as shapefiles estão projetadas no sistema de coordenadas WGS84. Utilizou-se a CAOP2011 (Carta Administrativa Oficial de Portugal de 2011) para a cartografia sendo que optamos por esta versão porque, embora existam versões mais recentes, a CAOP2011 é a única disponível em formato JSON, online e gratuito, ao nível das freguesias, o que facilita a integração e manipulação dos dados em sistemas de informação geográfica (SIG).

Sendo a temática principal o estudo da análise espacial da distribuição dos monumentos no concelho do Porto, foi necessário criar uma estrutura de apoio, ou seja, um modelo que permitisse fazer a ligação entre o python e a estrutura geográfica, como o *ArcGIS Pro*. Para isto foram construídos 8 scripts, onde a partir daí foram desenvolvidas e construídas várias shapefiles, que posteriormente foram colocadas nos recursos cartográficos. Essas shapefiles foram, primeiramente, submetidas a uma primeira “limpeza” e só depois passaram por processos de seleção e de manipulação de dados. Todos estes processos foram executados com sucesso na medida em que todos estes passos foram realizados com o máximo rigor possível de forma a que isso se pudesse transmitir no resultado final cartográfico. Foram usadas várias ferramentas que ajudaram na obtenção dos seguintes resultados, como *read\_csv*; *clip*; *intersection*; *buffer*; *central feature*; *mean center*; *kernel density*, entre outras. Foi essencial o uso de bibliotecas que permitiram a execução de cada um dos scripts criados tais como *geopandas*, *pandas*, *numpy*, *fiona*, *matplotlib*, entre outras. Para além disso, foram criados plots que garantem uma boa visibilidade dos resultados, bem como outra forma de analisar espacialmente a distribuição dos monumentos no Porto.

A análise espacial concentrou-se em entender a distribuição geográfica dos monumentos, avaliando a sua localização e proximidade.

## Enquadramento Geográfico da Área de Estudo

No que concerne ao enquadramento geográfico da área de estudo, de acordo com a figura 1, observamos que o município do Porto faz fronteira: a norte, com os municípios de Matosinhos e Maia; a este, com Maia, Valongo e Gondomar; a sul por Vila Nova de Gaia; e a oeste, banhado pelo Oceano Atlântico. Embora não presente no mapa, é importante considerar que a área de estudo se situa na região Norte de Portugal (NUTSII), enquadrada na Área Metropolitana do Porto (NUTSIII), apresentando uma área de, aproximadamente, 41,42km<sup>2</sup>.

O município do Porto é composto por 15 freguesias, de acordo com a CAOP 2011, as quais são Paranhos, Campanhã, Bonfim, Santo Ildefonso, Cedofeita, Miragaia, Sé, São Nicolau, Vitória, Massarelos, Lordelo do Ouro, Foz do Douro, Nevogilde, Aldoar e Ramalde.

Relativamente à população, considera-se que existiam 237584 habitantes, de acordo com os dados dos censos 2011.

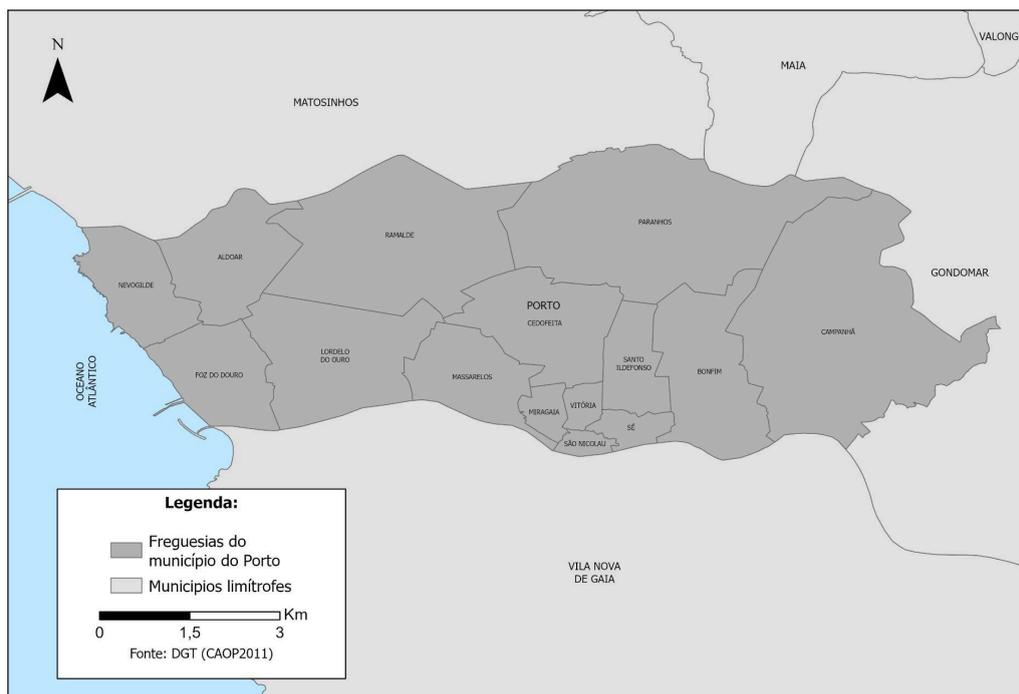


Figura 1: Enquadramento Geográfico das freguesias do concelho do Porto

## Análise da Espacial da Temática em Estudo

### a) Monumentos

Em primeiro lugar, começou-se por fazer download do CSV dos pontos principais do trabalho, os Monumentos. Após a leitura do seguinte CSV, dado que continha vasta informação que não foi considerada necessária ou importante para o decorrer do projeto e tal como indica a figura 2, fez-se uma série de procedimentos desde a limpeza, onde se eliminaram colunas para o efeito, até à manipulação desse ficheiro. Preferiu-se optar por guardar essas alterações num novo ficheiro CSV, e posteriormente guardá-lo em formato de *shapefile* do que salvar essas alterações no CSV original.

A *shapefile* é um formato padrão da indústria para dados geoespaciais que permite a representação e armazenamento de informações geográficas juntamente com atributos tabulares. Este código não apenas limpa e processa dados tabulares de um arquivo CSV, como também cria e armazena informações geoespaciais com base nas coordenadas de longitude e latitude fornecidas no arquivo original dos monumentos. Este tipo de formato foi o que se pretendia, visto que é o mais indicado para tarefas que envolvem análise, visualização e manipulação de dados geográficos juntamente com os seus atributos associados.

O código utiliza as bibliotecas *Pandas* e *GeoPandas* para realizar as operações de processamento de dados tabulares e a criação de dados geoespaciais em python.

```
1 import pandas as pd
2 import geopandas as gpd
3 from shapely.geometry import Point
4
5 def read_csv(file_path):
6     df = pd.read_csv(file_path)
7     print(df.to_string()) # mostra a tabela completa
8
9 usage
10 def clean_csv(file_path, output_path, fill_value=None):
11     df = pd.read_csv(file_path) # lê o ficheiro
12
13     if fill_value is not None:
14         df.fillna(fill_value, inplace=True) # Fill missing values
15
16     # elimina as colunas que indiquei na tabela
17     columns_to_remove = ['updated', 'created', 'others', 'lang', 'time']
18     df.drop(columns=[col for col in columns_to_remove if col in df.columns], inplace=True)
19
20     df.drop_duplicates(inplace=True) # remove valores duplicados
21
22     df.to_csv(output_path, index=False)
23     print(f"Cleaned data saved to {output_path}") # guarda o csv mas duplicado
24
25     # cria a geometria dos pontos através da latitude e da longitude
26     geometry = [Point(xy) for xy in zip(df['longitude'], df['latitude'])]
27
28     # GeoDataFrame
29     gdf = gpd.GeoDataFrame(df, geometry=geometry)
30
31     # coloca as coordenadas para o sistema WGS 84 (EPSG:4326)
32     gdf.set_crs(eps=4326, inplace=True)
33
34     # guarda o GeoDataFrame para formato shapefile
35     shapefile_path = output_path.replace('.csv', '.shp')
36     gdf.to_file(shapefile_path, driver='ESRI Shapefile')
37
38     print(f"Shapefile guardada em {shapefile_path}")
```

Figura 2: Script 1 - Leitura, Pré-processamento e Manipulação do CSV Monumentos

A *shapefile* resultante do código acima ilustrado é visível na figura 3, que mostra a distribuição de monumentos das 15 freguesias da cidade do Porto. Verifica-se que existem freguesias onde não existem monumentos, mas por outro lado as freguesias de Miragaia, São Nicolau, Sé e Vitória apresentam um maior número de monumentos localizados. Estas zonas são conhecidas por se localizarem na baixa do Porto, onde é característico o maior número de monumentos e por sua vez também de turistas. Relativamente à *shapefile* das freguesias, ou seja, a CAOP 2011, de seguida irá ser explicado como se procedeu à sua formulação desde a construção do script até ao seu formato final, como se verifica.

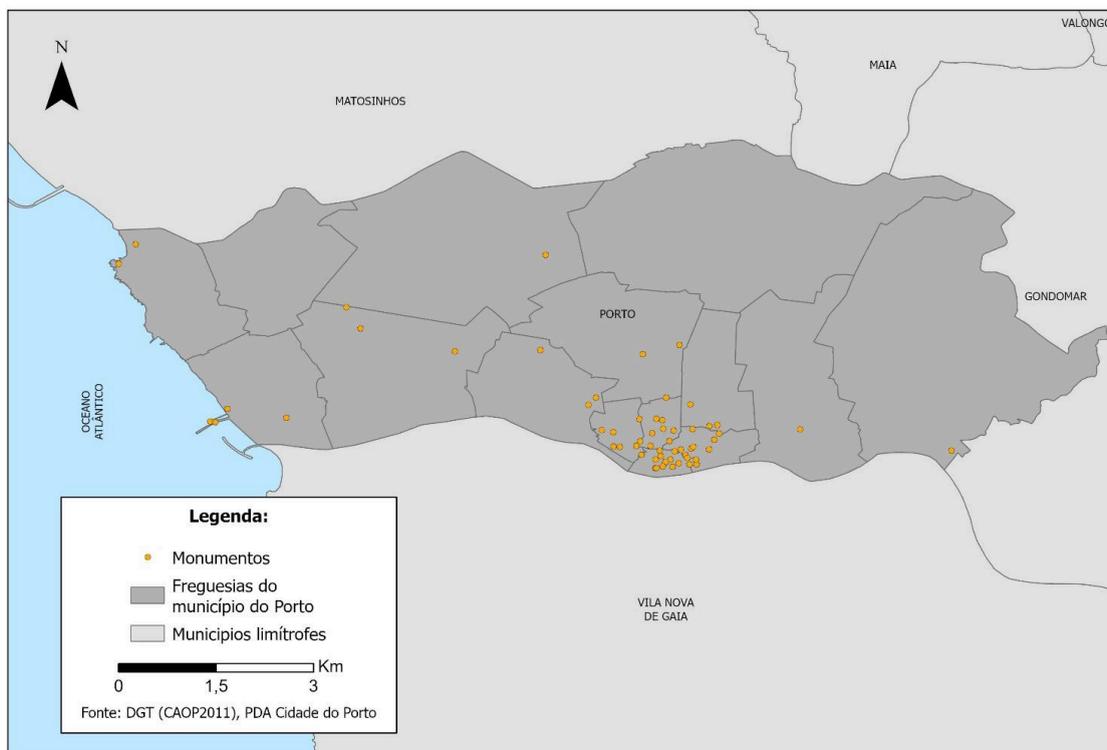


Figura 3: Distribuição dos monumentos na cidade do Porto

## b) Freguesias

Relativamente às freguesias, ou seja, a localização onde decorre o seguinte trabalho, começou-se por fazer *download* do ficheiro JSON onde continha a CAOP nível 3 - freguesias, relativa ao ano de 2011. Visto que era essencial mostrar com maior amplitude e detalhe os monumentos, ponto principal deste projeto, foi imprescindível conter informações do território distribuído em freguesias, e não nouro tipo de divisão administrativa.

Abriu-se esta fase com a transformação do ficheiro JSON para formato SHP, e logo após isto começou-se uma seleção por atributos, onde apenas se incluiu as freguesias pertencentes ao município do Porto, excluindo as restantes. Procurou-se neste tópico fazer várias manipulações na seguinte *shapefile* pelo que então foram criados vários campos na tabela de atributos: população (Censos 2011), área (km<sup>2</sup> - 2011) e *count* (nº total de monumentos), para no final se fazer um cálculo das densidades - densidade populacional e densidade de monumentos, através de duas funções, tal como indica a figura 4.

```
11 # Função para calcular a densidade populacional
12 1 usage
13 def calcular_densidade(populacao, area):
14     return populacao / area
15
16 # Função para calcular a área de uma geometria em m^2
17 def calcular_area(geometry):
18     transformer = Transformer.from_crs( crs_from: 'epsg:4326', crs_to: 'epsg:3857', always_xy=True)
19     projected_area = transformer.transform(geometry.centroid.x, geometry.centroid.y)
20     return projected_area
21
22
23 # Função para calcular a densidade dos monumentos
24 1 usage
25 def calcular_densMonu(monumentos, area):
26     return monumentos / area
27
28 # Função para calcular a área de uma geometria em m^2
29 def calcular_areakm2(geometry):
30     transformer = Transformer.from_crs( crs_from: 'epsg:4326', crs_to: 'epsg:3857', always_xy=True)
31     projected_area = transformer.transform(geometry.centroid.x, geometry.centroid.y)
32     return projected_area
```

Figura 4: Script 2 - Leitura e Manipulação do JSON CAOP 2011

Através das funções criadas para a densidade obtiveram-se os seguintes mapas que correspondem à densidade populacional e dos monumentos, respetivamente - figura 5 e 6.

A análise da densidade populacional, obtida através dos Censos de 2011, por freguesias no concelho do Porto, oferece uma visão detalhada da distribuição da população dentro de um dos mais importantes centros urbanos de Portugal. O concelho do Porto, área vibrante, compreende várias freguesias, cada uma com suas características únicas de densidade populacional. Através do mapa da Figura 5 que representa a densidade populacional, podemos perceber que as freguesias Cedofeita e de São Nicolau são as que apresentam

maior densidade populacional, com 7957 a 90076 habitantes por km<sup>2</sup>. Seguida das freguesias da Sé, Santo Ildefonso, Bonfim com 6693 a 7956 habitantes por km<sup>2</sup> e as freguesias de Vitória, Lordelo do Ouro, Aldoar, Ramalde e Paranhos com 4219 a 6692 habitantes por km<sup>2</sup>. Já as freguesias com menor densidade populacional correspondem a Campanhã, MiraGaia, Massarelos e Foz do Douro com 2510 a 4218 habitantes por km<sup>2</sup>, sendo que Nevogilde é a única freguesia com menos de 2508 habitantes por km<sup>2</sup>.

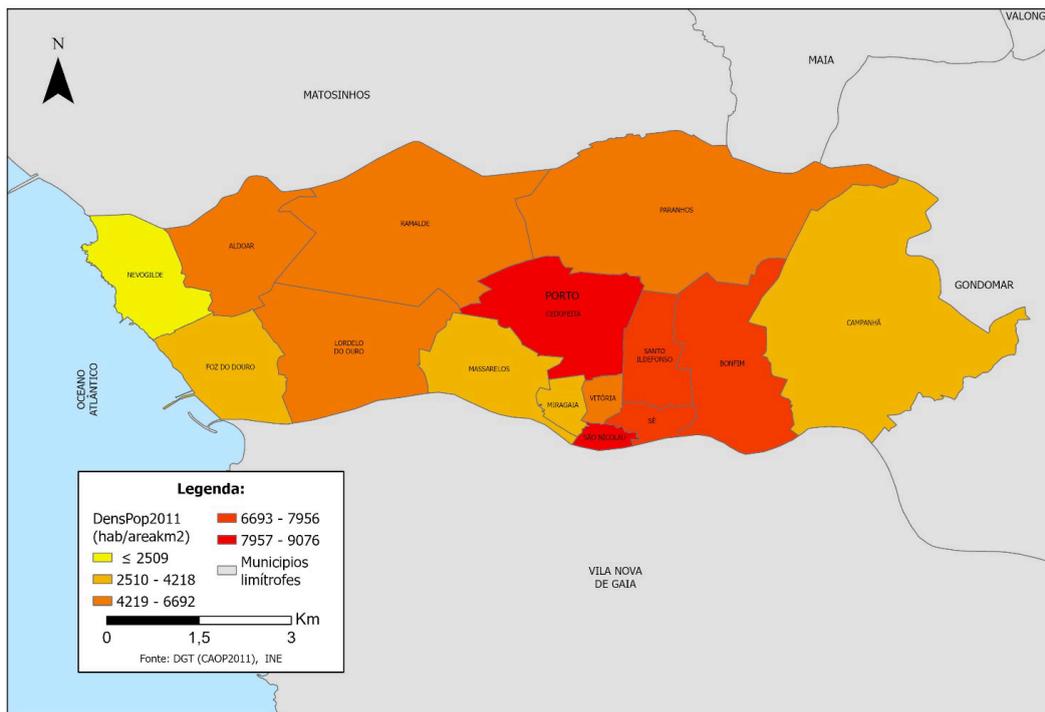


Figura 5: Densidade Populacional na cidade do Porto

A análise da densidade dos monumentos por freguesia permite alargar o conhecimento da distribuição destes elementos. Observando os resultados é claro que as freguesias da baixa do Porto, São Nicolau, Sé, Vitória e ainda Miragaia são aquelas onde os monumentos mais estão concentrados. Se estas freguesias se caracterizam por uma grande quantidade de monumentos na sua área, as restantes freguesias do Porto são caracterizadas pelo contrário, com muitos poucos monumentos pela sua área, principalmente nas freguesias a oeste e norte do concelho.

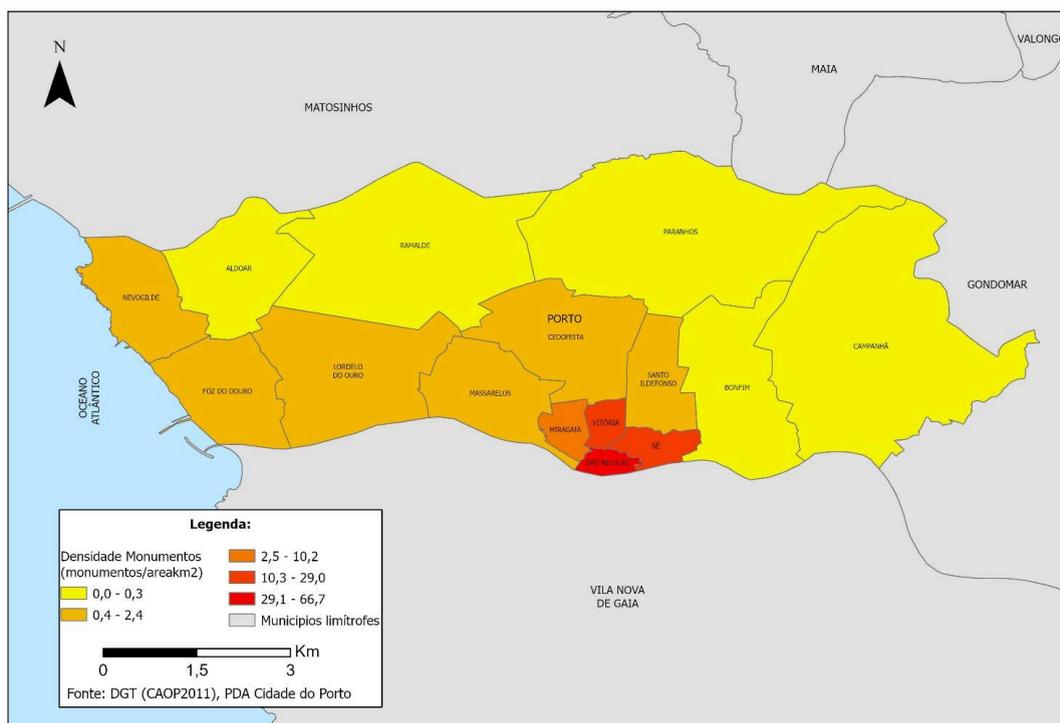


Figura 6: Densidade dos Monumentos na cidade do Porto

Construiu-se ainda um *plot*/gráfico que mostra o número de monumentos por freguesia, onde se identifica a localização de cada um dos 60 monumentos no concelho do Porto - figura 7. O seguinte gráfico mostra que, tal como já tinha sido possível admitir na figura 3, as freguesias com maior número de monumentos são São Nicolau, Sé, Vitória e MiraGaia, que se localizam na baixa do Porto. Aldoar e Paranhos são as que não admitem nenhum monumento identificado.

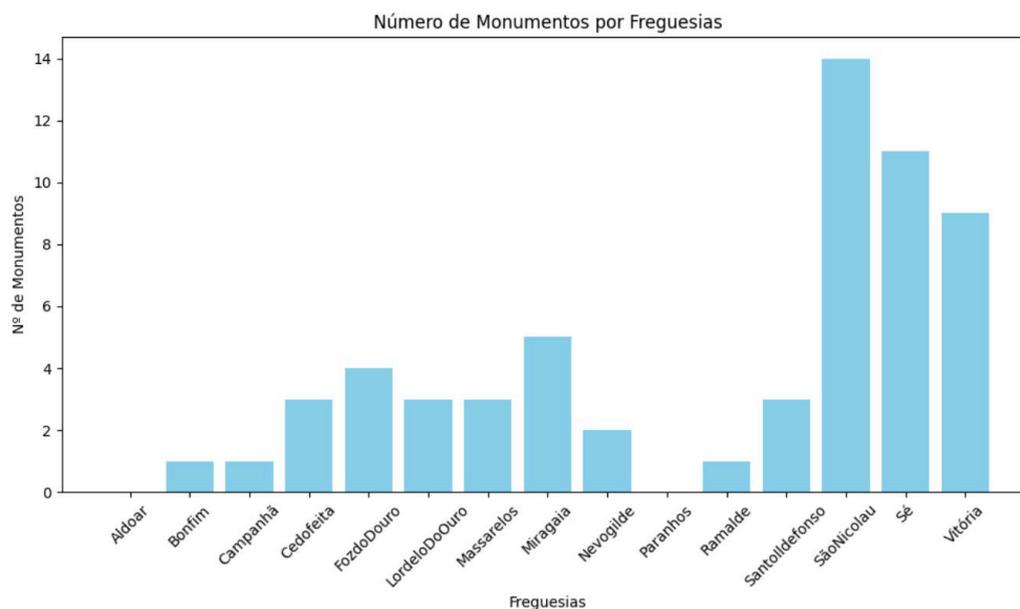


Figura 7: Número de Monumentos por Freguesia

### c) Alojamentos Locais

Os alojamentos locais constituem-se um dos grandes movimentadores de capital no setor do turismo. Sendo o Porto uma das cidades europeias com maior número de turistas e onde esse setor tem cada vez mais crescido na cidade, é essencial perceber como é o comportamento entre os AL's em relação aos monumentos que a cidade do Porto tem para oferecer aos seus visitantes.

Para isso fez-se *download* da *shapefile* dos AL's existentes em Portugal Continental, e em seguida aplicou-se a ferramenta *clip* entre esta nuvem de pontos em relação à CAOP anteriormente analisada. A figura 8 mostra a aplicação dessa ferramenta no script criada para o efeito.

```
1 import geopandas as gpd
2
3 # Caminho para as shapefiles input, output e do clip
4 input_shapefile = '/Users/laramarques/Desktop/MSIGOT/PSOT/Trabalho Prático/scripts/pass01/Estabelecimentos_de_Alojamento_Local.shp'
5 clip_shapefile = '/Users/laramarques/Desktop/MSIGOT/PSOT/Trabalho Prático/scripts/pass01/caop_Porto.shp'
6 output_shapefile = '/Users/laramarques/Desktop/MSIGOT/PSOT/Trabalho Prático/scripts/pass01/ALs_Porto.shp'
7
8 # Carregar as shapefiles através do geopandas
9 gdf_input = gpd.read_file(input_shapefile)
10 gdf_clip = gpd.read_file(clip_shapefile)
11
12 # Run da ferramenta clip
13 gdf_clipped = gpd.clip(gdf_input, gdf_clip)
14
15 # Guardar o output em shape
16 gdf_clipped.to_file(output_shapefile, driver='ESRI Shapefile')
17
18 print(f'Clip entre as shapefiles concluído em {output_shapefile}')
```

Figura 8: Script 3 - Leitura da *Shapefile* Alojamentos Locais e execução da ferramenta *Clip*

Após este processo passou-se o seguinte output, ou seja, os AL's pertencentes ao concelho do Porto, para reproduzir de forma cartográfica - figura 9. Observa-se que, tal como era esperado, existe uma ligação direta na localização de AL's em relação à localização de monumentos, pois também é nas mesmas freguesias - São Nicolau, Sé, Vitória e MiraGaia que o número de AL's é superior face às restantes freguesias do Porto. Visto que o turismo em "massas" é maior nestas áreas, leva por si só a fazer com que este tipo de alojamentos esteja perto, não só dos turistas mas também perto dos monumentos e das várias acessibilidade que rodeiam estas zonas das baixa do Porto.

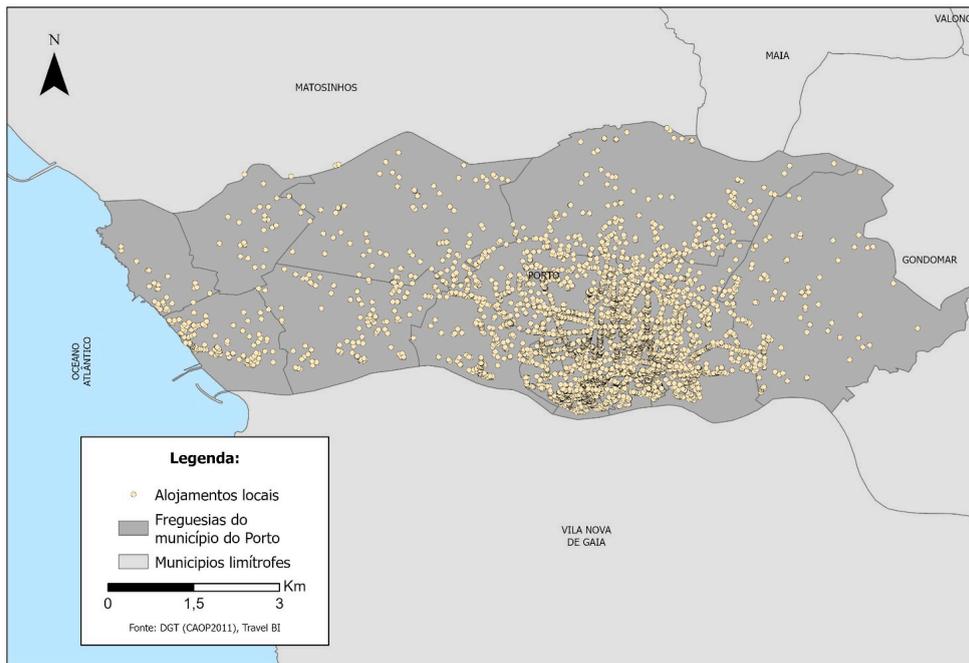


Figura 9: Distribuição de AL's na cidade do Porto

#### d) **Buffers e Interseção**

O conceito de *buffer* e interseção são fundamentais na análise espacial em SIG. O *buffer* cria uma área ao redor de um objeto espacial, sob forma de ponto, linha ou polígono, a uma distância especificada pelo utilizador, sendo essencial para determinar uma proximidade ou áreas de influência. No caso da interseção, esta é utilizada para identificar áreas compartilhadas entre diferentes camadas ou objetos espaciais e usada para variadas aplicações, como a análise do risco, a gestão de recursos, planeamento e ordenamento do território, entre outros.

Com o objetivo de perceber quantos alojamentos locais estão a 500m e a 1000m dos monumentos, procedeu-se primeiramente à criação desses dois *buffers* em redor dos pontos, representativos dos monumentos, tal como confere a figura 10. A função “create\_buffers” mostra que através dos monumentos, conseguiram-se criar duas áreas de influência em torno dessa nuvem de pontos, para os raios já indicados, para se obterem as respectivas *shapefiles* como *output*.

```

39 def create_buffers(input_shapefile_path, buffer_distances, output_shapefile_paths):
40
41     gdf = gpd.read_file(input_shapefile_path)
42
43     # verifica a geometria dos pontos da shape
44     if not all(gdf.geom_type == 'Point'):
45         raise ValueError("The shapefile must contain only points.")
46
47     # volta a projetar as coordenadas do GeoDataFrame para '32633'
48     projected_gdf = gdf.to_crs(eps=32633)
49
50     # cria os buffers para cada distância
51     for buffer_distance, output_shapefile_path in zip(buffer_distances, output_shapefile_paths):
52         buffer_column = f'buffer_{buffer_distance}m'
53         projected_gdf[buffer_column] = projected_gdf.geometry.buffer(buffer_distance)
54
55         # cria a GeoDataFrame para o buffer
56         gdf_buffer = projected_gdf[[buffer_column]].copy()
57         gdf_buffer = gdf_buffer.rename(columns={buffer_column: 'geometry'})
58         gdf_buffer = gdf_buffer.set_geometry('geometry')
59
60         # guarda o buffer em shapefile
61         gdf_buffer.to_file(output_shapefile_path, driver='ESRI Shapefile')
62         print(f"Buffer of {buffer_distance}m created and saved to {output_shapefile_path}")
63
64
65 file_path = '/Users/laramarques/Desktop/MSIG01/PS01/Trabalho Prático/scripts/passo1/monumentos.csv'
66 output_path = '/Users/laramarques/Desktop/MSIG01/PS01/Trabalho Prático/scripts/passo1/monumentos_clean.csv'
67 clean_csv(file_path, output_path, fill_value=None)
68
69 # distância dos buffers a colocar
70 buffer_distances = [500, 1000]
71 # output path das shapes
72 output_shapefile_paths = [
73     '/Users/laramarques/Desktop/MSIG01/PS01/Trabalho Prático/scripts/passo1/monumentos_buffer500.shp',
74     '/Users/laramarques/Desktop/MSIG01/PS01/Trabalho Prático/scripts/passo1/monumentos_buffer1000.shp'
75 ]
76
77 input_shapefile_path = output_path.replace(_old: '.csv', _new: '.shp')
78 create_buffers(input_shapefile_path, buffer_distances, output_shapefile_paths)

```

Figura 10: Script 1 - Execução da ferramenta *Buffer*

De seguida foi realizada a interseção espacial entre os AL's do Porto e o *buffer* relativo aos monumentos, utilizando a função *overlay* do *geopandas* que contém apenas os pontos que se encontram dentro do *buffer*.

```

1 import geopandas as gpd
2 from shapely.geometry import Point
3 from pyproj import Transformer
4
5 # Caminho para a shapefile de pontos AL's
6 pontos_path = '/Users/laramarques/Desktop/MSIG01/PS01/Trabalho Prático/scripts/passo1/ALs_Porto.shp'
7
8 # Caminho para o primeiro buffer 500
9 buffer_path = '/Users/laramarques/Desktop/MSIG01/PS01/Trabalho Prático/scripts/passo1/monumentos_buffer500.shp'
10
11 # Carregar a shape AL's como um GeoDataFrame
12 gdf_pontos = gpd.read_file(pontos_path)
13
14 # Carregar o buffer 500 como um GeoDataFrame
15 gdf_buffer = gpd.read_file(buffer_path)
16
17 # Verificar as coordenadas dos dados de entrada
18 print("CRS dos pontos (antes):", gdf_pontos.crs)
19 print("CRS do buffer (antes):", gdf_buffer.crs)
20
21 # voltar a projetar os AL's para o sistema de coordenadas do buffer (EPSG:32633)
22 gdf_pontos = gdf_pontos.to_crs(gdf_buffer.crs)
23
24 # Verificar as coordenadas dos dados
25 print("CRS dos pontos (depois):", gdf_pontos.crs)
26 print("CRS do buffer (depois):", gdf_buffer.crs)
27
28 # executar a interseção entre os AL's e o buffer 500
29 intersecao = gpd.overlay(gdf_pontos, gdf_buffer, how='intersection')
30
31 # Caminho para guardar o output
32 output_intersecao_path = '/Users/laramarques/Desktop/MSIG01/PS01/Trabalho Prático/scripts/passo1/ALs_intersect500.shp'

```

Figura 11: Script 4 - Leitura e Manipulação da *shapefile* AL's do Porto e execução de *Buffers* (500 e 1000m) e da *Intersection*

A partir da análise dos mapas da figura 12 seguida da figura 13, percebe-se que a 500 metros dos monumentos, estão localizados 69 448 alojamentos locais, enquanto que a 1000m podem ser encontrados 188 071 alojamentos locais, o que representa um aumento de 118 623 alojamentos locais com a passagem dos 500 metros para os 1000m. A região central do Porto possui uma densidade maior de monumentos, o que sugere uma alta concentração de patrimônio histórico e cultural, sendo que as áreas limítrofes possuem menos monumentos e, conseqüentemente, menos sobreposição de buffers. Assim, subentende-se que as áreas com sobreposição de buffers podem indicar necessidade de infraestrutura turística, como sinalização, manutenção e serviços de apoio aos visitantes.

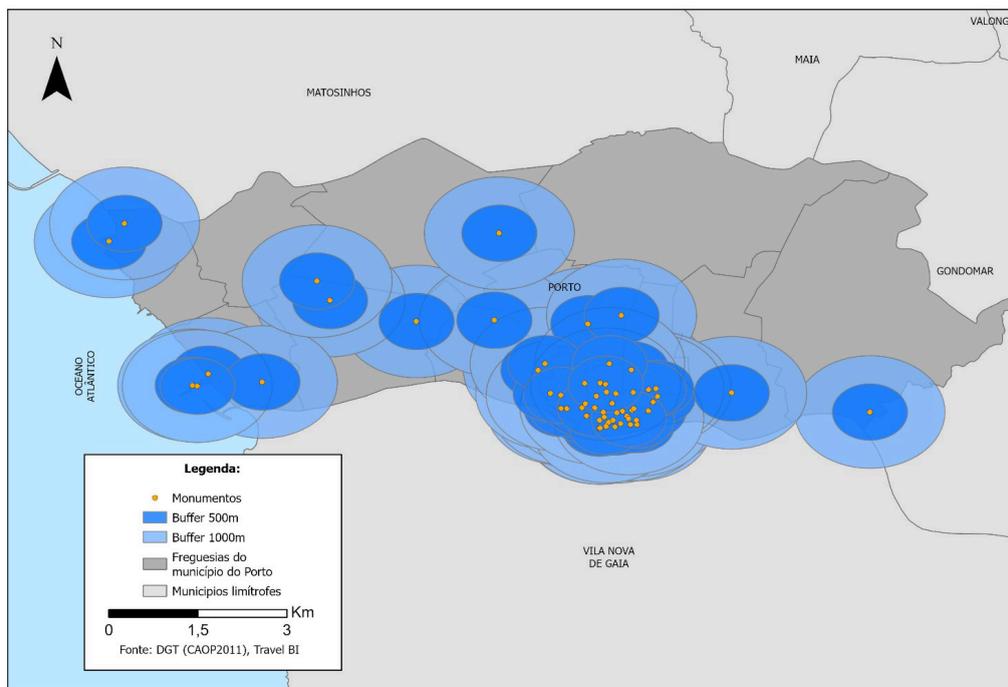


Figura 12: Buffers de 500 e 1000 metros em torno dos Monumentos da cidade do Porto

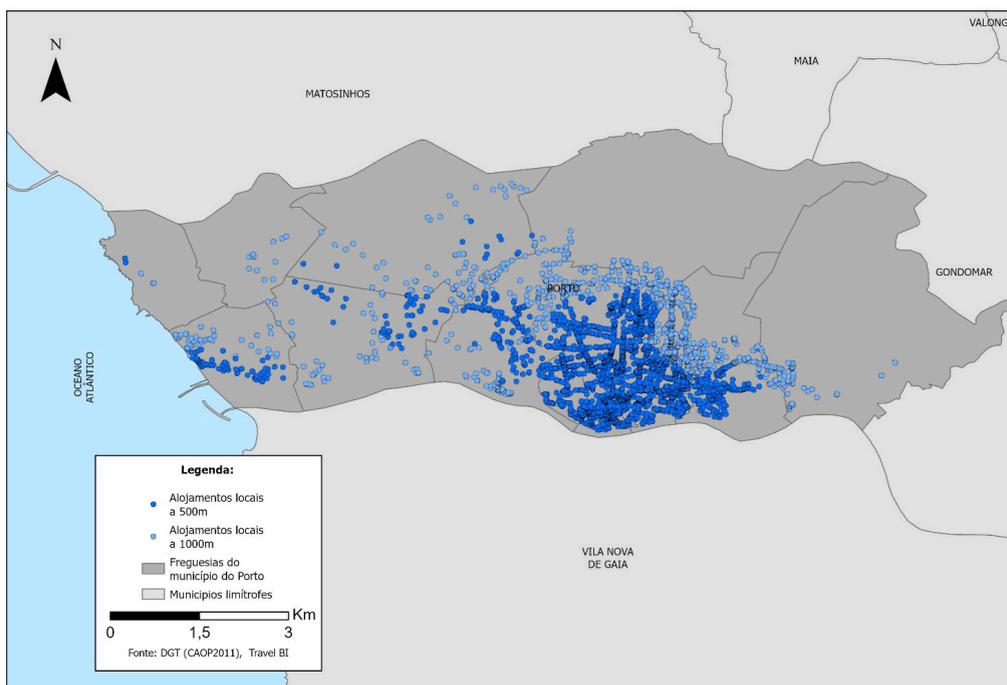


Figura 13: *Intersect* entre os *buffers* de 500 e 1000 metros com os AL's da cidade do Porto

### e) Ponto Central e Centro Médio

De forma a aprofundar a análise espacial dos monumentos do porto, realizou-se o estudo sobre a centralidade dos mesmos, e para tal procedeu-se à identificação do elemento central. Fez-se ainda um estudo relativamente ao centro médio, que representa média aritmética das coordenadas de um conjunto de pontos. No contexto da análise espacial, ambos os processos são essenciais para representar a localização central e média de um conjunto de pontos geográficos.

Para poder obter o elemento central foi criado o script representado pela figura 14, onde foram então extraídas as coordenadas x e y de cada ponto e calculada uma matriz de distâncias euclidianas entre todos os pares de pontos. É com base nessa matriz que foi determinado o ponto que possui a menor soma das distâncias para todos os outros pontos, identificando-o como o ponto central. Para poder encontrar o centro médio criou-se o script representado pela figura 15, onde por com o objetivo calcular o centro médio dos pontos, a partir da soma de todas as coordenadas e dividindo estas pelo número total de pontos. Ambos os resultados estão visíveis na figura 16.

```

1 import geopandas as gpd
2 import numpy as np
3 from shapely.geometry import Point
4
5 # Caminho para a shapefile de pontos monumentos
6 shapefile_path = '/Users/laramarques/Desktop/MSIGOT/PSOI/Trabalho Prático/scripts/passo1/monumentos.shp'
7
8 # Carregar a shape
9 gdf = gpd.read_file(shapefile_path)
10
11 # verificar se a geometria são pontos
12 if gdf.geom_type[0] != 'Point':
13     raise ValueError("A shapefile deve conter apenas pontos.")
14
15 # obter as coordenadas dos pontos
16 coords = np.array([[point.x, point.y] for point in gdf.geometry])
17
18 # calcular a matriz de distâncias
19 dist_matrix = np.sqrt(((coords[:, np.newaxis] - coords) ** 2).sum(axis=2))
20
21 # calcular a soma das distâncias para cada ponto
22 sum_distances = dist_matrix.sum(axis=1)
23
24 # encontrar o ponto central
25 central_index = np.argmin(sum_distances)
26 central_point = gdf.geometry.iloc[central_index]
27
28 # Criar um GeoDataFrame para o ponto central
29 central_gdf = gpd.GeoDataFrame(data={'geometry': [central_point]}, crs=gdf.crs)
30
31 # guardar o central feature em shape
32 output_shapefile_path = '/Users/laramarques/Desktop/MSIGOT/PSOI/Trabalho Prático/scripts/passo1/pCentral_monu.shp'
33 central_gdf.to_file(output_shapefile_path, driver='ESRI Shapefile')
34
35 print(f"Feature central calculada e guardada em {output_shapefile_path} com sucesso!")

```

Figura 14: Script 5 - Leitura e Manipulação da *shapefile* Monumentos e execução do *Central Feature*

```

1 import geopandas as gpd
2 from shapely.geometry import Point
3
4 # Caminho para a shapefile de pontos monumentos
5 shapefile_path = '/Users/laramarques/Desktop/MSIGOT/PSOI/Trabalho Prático/scripts/passo1/monumentos.shp'
6
7 # Carregar a shapefile
8 gdf = gpd.read_file(shapefile_path)
9
10 # verificar se a geometria são pontos
11 if gdf.geom_type[0] != 'Point':
12     raise ValueError("A shapefile deve conter apenas pontos.")
13
14 # obter as coordenadas dos pontos
15 coords = gdf.geometry.apply(lambda geom: (geom.x, geom.y))
16 x_coords, y_coords = zip(*coords)
17
18 # calcular as coordenadas médias (mean center)
19 mean_x = sum(x_coords) / len(x_coords)
20 mean_y = sum(y_coords) / len(y_coords)
21
22 mean_center = Point(*args=(mean_x, mean_y))
23
24 # Criar um GeoDataFrame para o ponto central
25 mean_center_gdf = gpd.GeoDataFrame(data={'geometry': [mean_center]}, crs=gdf.crs)
26
27 # guardar o mean center em shape
28 output_shapefile_path = '/Users/laramarques/Desktop/MSIGOT/PSOI/Trabalho Prático/scripts/passo1/pMedio_monu.shp'
29 mean_center_gdf.to_file(output_shapefile_path, driver='ESRI Shapefile')
30
31 print(f"Mean center calculado e guardado em {output_shapefile_path} com sucesso!")

```

Figura 15: Script 6 - Leitura e Manipulação da *shapefile* Monumentos e execução do *Mean Center*

Relativamente à figura 16, o elemento central está representado pela cor azul, que identifica o elemento mais central relativamente aos restantes monumentos existentes, localizado no limite norte da freguesia de São Nicolau, e que vai encontro com a freguesia que apresenta a maior densidade de monumentos e também de maior densidade populacional. Em relação ao centro médio representado pela cor vermelha, fica a noroeste do elemento central, localizando-se na freguesia de Miragaia. Esta localização mais para oeste do elemento central faz sentido uma vez que existem vários monumentos mais situados ao longo do limite oeste do concelho do Porto.

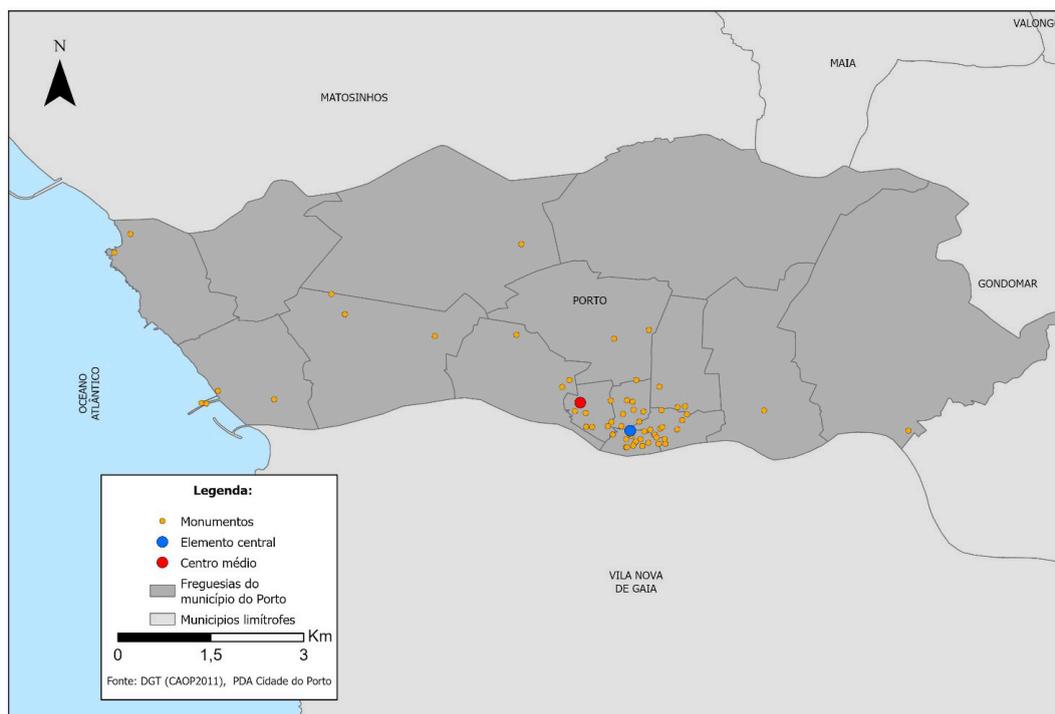


Figura 16: Ponto Central e Centro Médio dos Monumentos na cidade do Porto

## f) Estudo da Densidade de Kernel aos Monumentos e aos AL's

A densidade de kernel é uma técnica estatística fundamental para estimar a distribuição de probabilidade de uma variável contínua. O cálculo da estimativa da densidade de kernel para um ponto é determinada pela média ponderada pelos kernels, onde os pesos de cada um são baseados na proximidade espacial ou temporal dos dados, o que permite uma representação suave e contínua da distribuição dos dados sem a necessidade de assumir uma forma específica de distribuição. Esta técnica é amplamente utilizada na análise exploratória de dados, modelagem estatística e detecção de anomalias, proporcionando insights valiosos sobre a estrutura e padrões dos dados em diferentes contextos analíticos.

Para perceber a densidade de Kernel relativa aos monumentos e aos alojamentos locais foi criado um *script* - figura 17, que conseguiu-se estimar o modelo de densidade de kernel ajustado aos dados da matriz, onde o resultado foi convertido para corresponder ao formato da grade X e Y.

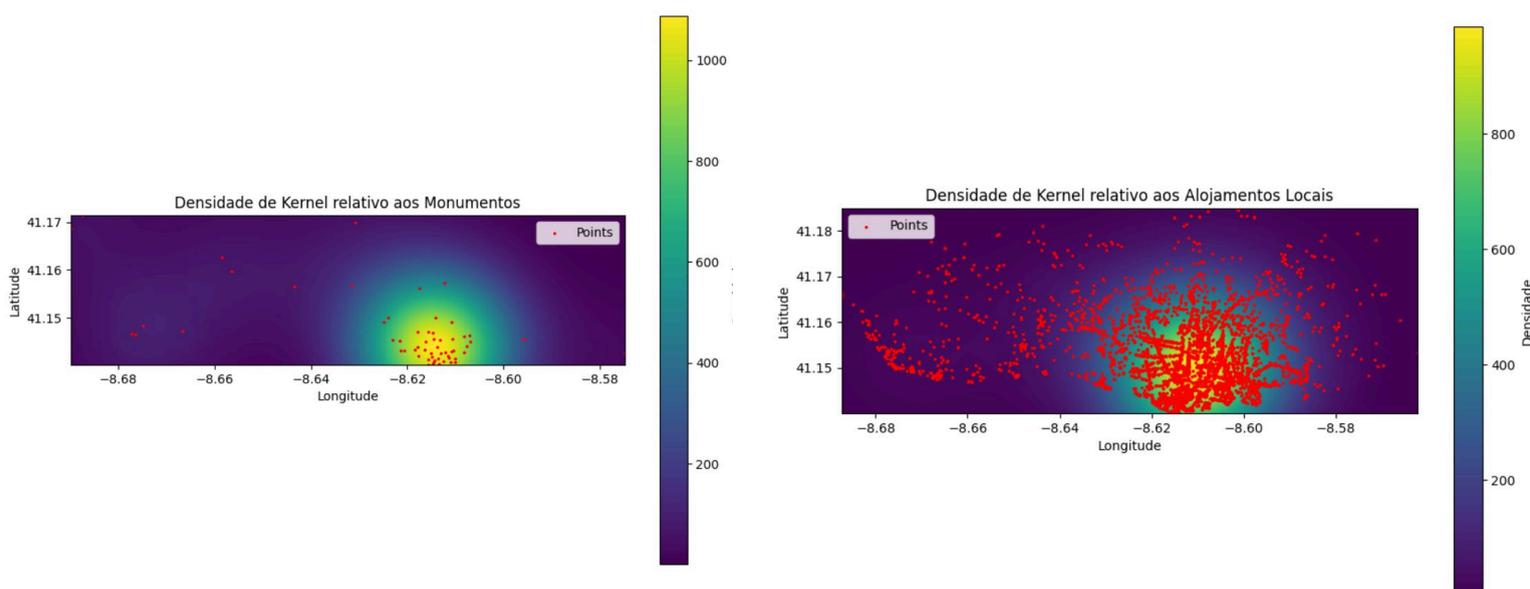
Por fim obtiveram-se os dois seguintes gráficos da densidade de Kernel - figura 18 e 19.

```
7 # Caminho para a shapefile de pontos monumentos
8 shapefile_path = '/Users/Laramarques/Desktop/MSIGOT/PSOT/Trabalho Prático/scripts/passo1/monumentos.shp'
9
10 # Carregar a shapefile
11 gdf = gpd.read_file(shapefile_path)
12
13 # verificar se a geometria são pontos, se não extrair centroides
14 if gdf.geom_type[0] != 'Point':
15     gdf['geometry'] = gdf['geometry'].centroid
16
17 # obter as coordenadas dos pontos
18 coords = np.array([[point.x, point.y] for point in gdf.geometry])
19
20 # Definir o espaço de amostragem para a kernel density
21 x_min, y_min = coords.min(axis=0)
22 x_max, y_max = coords.max(axis=0)
23 x_range = np.linspace(x_min, x_max, num=100)
24 y_range = np.linspace(y_min, y_max, num=100)
25 X, Y = np.meshgrid(*x_range, y_range)
26 sample_coords = np.vstack([X.ravel(), Y.ravel()]).T
27
28 # Calcular a kernel density
29 bandwidth = 0.01
30 kde = KernelDensity(bandwidth=bandwidth)
31 kde.fit(coords)
32 log_density = kde.score_samples(sample_coords)
33
34 # Converter para uma matriz de densidade
35 density = np.exp(log_density).reshape(X.shape)
36
37 # cria um plot da densidade
38 plt.figure(figsize=(10, 8))
39 plt.imshow(density, origin='lower', extent=[x_min, x_max, y_min, y_max], cmap='viridis')
40 plt.colorbar(Label='Densidade')
41 plt.scatter(coords[:, 0], coords[:, 1], c='red', s=2, label='Points')
42 plt.legend()
43 plt.title('Densidade de Kernel relativo aos Monumentos')
44 plt.xlabel('Longitude')
45 plt.ylabel('Latitude')
46 plt.show()
```

Figura 17: Script 7 e 8 - Leitura e Manipulação das *shapefiles* Monumentos e AL's do Porto e execução da *Kernel Density*

Os resultados das densidades dos monumentos e dos alojamentos locais revelam realidades que de certa forma já eram expectáveis. A densidade de kernel relativo aos monumentos apresenta grandes valores na área da baixa do município, onde existe uma maior concentração do fenómeno, representando uma redução significativa da densidade à medida que se afasta dessa concentração, sendo a restante área do Porto caracterizada por uma densidade muito baixa.

Relativamente aos alojamentos locais, a densidade de kernel representa uma realidade muito próxima daquela que é verificada em relação aos monumentos. Apesar de um número de alojamentos locais muito superior ao existente de monumentos, a densidade de kernel revela que é também nas freguesias da baixa portuense que os alojamentos locais se encontram mais concentrados, ainda que neste fenómeno a área se acabe por alargar mais, principalmente para este e norte do concelho. Assim, a densidade de kernel relativo aos alojamentos locais também forma uma circunferência na área da baixa onde a densidade é elevada mas que reduz significativamente ao afastar-se deste local.



Figuras 18 e 19: *Kernel Density* dos Monumentos e AL's do Porto

## Conclusões Finais

A análise espacial aos monumentos do Porto revelou padrões significativos na distribuição destes marcos históricos, destacando a concentração predominante nas freguesias centrais, como São Nicolau, Sé, Vitória e Miragaia. Estas áreas não só abrigam a maior densidade de monumentos, como também coincidem com as freguesias de alta densidade populacional, reforçando a relação entre a presença de património cultural e a densidade populacional.

A metodologia utilizada, com recurso a ferramentas dos SIG e à linguagem de programação Python, demonstrou-se eficaz quanto à manipulação, visualização e análise dos dados geográficos. A integração de *scripts* específicos permitiu a criação de *shapefiles* precisas assim como a aplicação de análises específicas como *buffers*, interseções e densidades de kernel, o que fornece uma compreensão detalhada da distribuição e acessibilidade dos monumentos. Os resultados indicam uma interdependência significativa entre os monumentos e os alojamentos locais, especialmente nas áreas centrais da cidade. A proximidade dos alojamentos aos monumentos sugere que a localização estratégica destes estabelecimentos é potencializada pela proximidade aos monumentos. A análise dos *buffers* a 500m e 1000m dos monumentos confirmou que a maioria dos alojamentos se encontra dentro destas zonas de influência, sublinhando a importância de uma infraestrutura turística bem distribuída. Os cálculos do elemento central e do centro médio dos monumentos indicam zonas de elevada importância histórica e cultural, cruciais para o planeamento urbano e turístico do Porto. A densidade de kernel aplicada aos monumentos e alojamentos locais reforçou a ideia que a concentração dos fenómenos se dá predominantemente na área central do Porto.

Ao longo deste estudo, foram identificadas algumas limitações que afetaram a profundidade e precisão da análise. Uma das principais limitações foi a disponibilidade de dados gratuitos online atualizados, como a CAOP em formato JSON. A falta de atualizações regulares nessa base de dados resulta em discrepâncias territoriais significativas, já que não reflete as alterações administrativas mais recentes. Além disso, a integração de diferentes fontes de dados com formatos e níveis de detalhe variados exigiu um esforço considerável de normalização e tratamento, o que pode ter introduzido erros ou inconsistências.

Desta forma, este estudo contribui para uma maior valorização e preservação dos monumentos do Porto, fornecendo uma base sólida para o planeamento estratégico que visa melhorar a infraestrutura turística e a conservação do património cultural. As ferramentas e metodologias aplicadas demonstraram a eficácia dos SIG na análise

espacial, assim como a linguagem Python, abrindo caminho para estudos futuros que possam explorar outras dimensões do município, como a relação entre espaços verdes, mobilidade urbana e qualidade de vida.

O reconhecimento da centralidade e acessibilidade dos monumentos históricos oferece *insights* valiosos para políticas públicas e iniciativas de desenvolvimento urbano, promovendo um ambiente que equilibra a preservação do património com as necessidades de crescimento e modernização. Além disso, a abordagem adotada neste estudo pode ser aplicada a outras áreas, contribuindo para uma compreensão mais ampla das interações entre o património cultural e o desenvolvimento urbano.

## Webgrafia

1. Shapefiles e dados GIS de Portugal. (n.d.). ForestGIS.com | O Seu Portal de Geotecnologia. Obtido a 13 de junho de 2024, do [https://forest-gis.com/shapefiles-de-portugal/#google\\_vignette](https://forest-gis.com/shapefiles-de-portugal/#google_vignette)
2. Estabelecimentos de Alojamento Local. (n.d.). Dadosabertos.turismodeportugal.pt. Obtido a 14 de junho, do 2024. <https://dadosabertos.turismodeportugal.pt/datasets/4e62eb1977564991bd01e61d7aa8266f6/explore>
3. Monumentos - Cultura e Património - Monumentos - Portal de Dados. (n.d.). Opendata.porto.digital. Obtido a 13 de junho de 20224, do <https://opendata.porto.digital/dataset/pontos-de-interesse-cultura-e-patrimonio-monumentos/resource/389a446c-8eaf-4406-b9ca-f1637c440de4>
4. Instituto Nacional de Estatística, Censos 2011. (2014). Censos.ine.pt. Obtido a 14 de junho de 2024, do [https://censos.ine.pt/xportal/xmain?xpid=CENSOS&xpgid=ine\\_censos\\_indicador&contexto=ind&indOcorrCod=0005981&selTab=tab10](https://censos.ine.pt/xportal/xmain?xpid=CENSOS&xpgid=ine_censos_indicador&contexto=ind&indOcorrCod=0005981&selTab=tab10)