

In [30]:

```
1 from IPython.display import Image
2 Image("imagem.jpg", width=500)
```

Out[30]:



In [37]:

```
1 from IPython.display import Image
2 Image('imagem1.png')
```

Out[37]:



CONHECIMENTO E COMPORTAMENTO DOS ESTUDANTES DA FEUP PERANTE NOTÍCIAS DA ATUALIDADE

O presente questionário, elaborado na unidade curricular de Análise e Visualização de Dados do Mestrado em Ciência da Informação da Faculdade e Engenharia da Universidade do Porto, tem como objetivos comparar os estudantes da FEUP com a generalidade das pessoas no que diz respeito ao conhecimento e importância atribuída ao tema das "Fake News" e avaliar o comportamento dos estudantes perante a informação.

Com este estudo, pretende-se que os estudantes reflitam e adotem uma postura mais criteriosa e crítica durante a seleção da informação, de forma a construir uma comunidade estudantil mais culta e consciente capaz de responder aos constantes desafios eminentes nas suas vidas pessoais, académicas e profissionais.

Os dados são recolhidos de forma anónima e para o uso exclusivamente académico.

O tempo de preenchimento do questionário é de 3 minutos.

Agradecemos a sua participação neste inquérito.

Curso a frequentar

Ciência da Informação

Unidade curricular de estudo

Análise e Visualização de Dados

Autores de trabalho

Simão Machado up201704685@fe.up.pt

A. ÍNDICE

B. Introdução

C. Enquadramento Teórico

D. Metodologia CRISP-DM

Data Analytics

1. Entendimento do Negócio
 - 1.1 Identificação do Problema Descritivo e Preditivo
2. Entendimento dos Dados
 - 2.1 Identificação de Grupos Sociais
 - 2.2 Conhecimento e Comportamento dos Estudantes da FEUP
 - 2.3 Caracterização da Amostra
 - 2.4 Bónus: Caracterização da População
3. Preparação, Modelação e Avaliação dos Dados
 - 3.1 Problema Descritivo
 - 3.1.1 Seleção e Transformação dos Atributos
 - 3.1.2 Abordagem Clustering
 - 3.1.3 Validação
 - 3.2 Problema Preditivo
 - 3.2.1 Seleção dos Atributos
 - 3.2.2 Abordagem Classification
 - 3.2.3 Validação

E. Conclusão

F. Bibliografia

G. Anexos

B. INTRODUÇÃO

- 1 O presente relatório, realizado no âmbito da unidade curricular Análise e Visualização de Dados, do Mestrado em Ciência da Informação, se refere ao tema "Data Analytics".
- 2 Para começar, teve-se a ideia de se desenvolver um questionário como fonte de dados. De todas as temáticas possíveis de serem abordadas, selecionou-se o tema do conhecimento e comportamento dos estudantes da FEUP perante as notícias da actualidade (e sua relação com as "Fake News"). A razão da sua escolha deveu-se, sobretudo, à actualidade do mesmo, bem como pelo facto de este estar a impactar, direta ou indiretamente, toda a noção e consciência de que uma pessoa tem no que diz respeito a informações digitais fidedignas.
- 3 Tendo como base o questionário referido, este trabalho irá debater esta questão minuciosamente. Assim, este trabalho tem como objetivos comparar os estudantes com a generalidade das pessoas relativamente ao conhecimento e importância atribuída ao tema das "Fake News" e avaliar o comportamento dos estudantes perante a informação.
- 4 No seu longo processo de desenvolvimento, e numa perspetiva mais universitária, o trabalho tem como objetivo primordial aplicar os conhecimentos adquiridos do curso em um problema real. Assim sendo, irá ser feita uma descrição do conhecimento e comportamento dos estudantes da FEUP perante as notícias da actualidade (e sua relação com as "Fake News") aplicando-se técnicas descriptivas e preditivas.
- 5 Com este estudo pretende-se que os estudantes reflitam e adotem uma postura mais criteriosa e crítica durante a seleção de informação, de forma a construir uma comunidade estudantil mais culta capaz de responder aos constantes desafios eminentes nas suas vidas pessoais, académicas e profissionais.

C. ENQUADRAMENTO TEÓRICO

- 1 Antes de começar a desmistificar as respostas dos estudantes, é preciso, em primeiro lugar, definir o conceito de "Fake News", pois só com a informação correta da sua definição é que é possível, posteriormente, avaliar corretamente as respostas dadas pelos estudantes. Caso contrário, a interpretação dos seus pareceres resultará em conclusões não verossímeis.
- 2 "Fake News", tal como o próprio nome indica, são notícias falsas que são divulgadas propositalmente por pessoas de forma a que estas obtenham um qualquer tipo de vantagem financeira, política, social, entre outras, permitindo o aumento exponencial da atenção de um particular assunto (Rush, 2018). Existe uma relação forte entre o termo "Fake News" e "Clickbait" - "conteúdo da internet que é destinado à geração de receita de publicidade online, normalmente às custas da qualidade e da precisão da informação, por meio de manchetes sensacionalistas e/ou imagens em miniatura chamativas para atrair cliques e incentivar o compartilhamento do material pelas redes sociais" (Contribuidores do Wikipédia, 2018). Estas notícias são maioritariamente divulgadas na internet, especialmente nas redes sociais. Não obstante, também são divulgadas via rádio, televisão e jornal, embora com menor frequência. Nestes últimos, existe um maior controlo da informação a ser distribuída. De reforçar a ideia de que sátira é diferente de "Fake News" (Musgrove, 2018) (Contribuidores do Wikipédia, 2019).

D. Metodologia CRISP-DM

Descriptive Data Analytics

1. Entendimento do Negócio

Identificação do Problema Descritivo e Preditivo

- 1 O problema aqui a ser discutido é o conhecimento e comportamento dos estudantes da FEUP perante as notícias da atualidade (e sua relação com as "Fake News").
- 2 Na fase "Entendimento dos Dados" será feito uma análise exploratória dos dados.
- 3 A fase "Preparação, Modelação e Avaliação dos Dados" será dividida em duas partes: problema descritivo e problema preditivo.
- 4 Relativamente ao problema descritivo, será feito a seleção e transformação dos atributos escolhidos, tendo em conta a abordagem clustering, e verificar se existe partição entre grupos sociais e o conhecimento e comportamento dos estudantes perante as notícias da atualidade (e sua relação com as "Fake News").
- 5 Relativamente ao problema preditivo, será feito a seleção e transformação dos atributos escolhidos, tendo em conta a abordagem classification, e será feito a qualificação sobre o conhecimento e comportamento dos estudantes perante as notícias da atualidade (e sua relação com as "Fake News").

2. Entendimento dos Dados

- 1 Com o intuito de se explorar todos os resultados possíveis de serem analisados, ou seja, de modo a se encontrar padrões entre as respostas das diferentes perguntas, procedeu-se à utilização da linguagem de programação Python. Com a sua utilização, foi possível entender os dados mais dinamicamente.

A partir daqui, todo o processo de análise de dados será feito utilizando Python, sendo que todo o código será explicado à medida que o mesmo se vai construindo. Os comentários imediatamente acima servirão como um guia da construção do raciocínio e estarão seguidos do símbolo "#".

In [1]:

```
1 # Faz o upgrade do Python Package Index
2 !python -m pip install --upgrade pip
```

Requirement already satisfied: pip in c:\users\simao\anaconda3\lib\site-packages (22.3.1)

In [5]:

```
1 # Importando os pacotes de análise de dados - PYTHON OPEN DATA SCIENCE STACK
2 ## Pacotes típicos
3 import os
4 import numpy as np
5 import pandas as pd
6 import matplotlib.pyplot as plt
7
8 from scipy import stats
9
10 # Pacotes para machine Learning
11 import sklearn.preprocessing
12 import sklearn.cluster
13 import sklearn.metrics
14 import sklearn.model_selection
15 import sklearn.ensemble
16
17 from sklearn.neighbors import NearestNeighbors
18 from scipy.spatial.distance import cdist
19
20 from imblearn.pipeline import Pipeline
21 from imblearn.over_sampling import SMOTE
22 from imblearn.under_sampling import RandomUnderSampler
23
24 # Pacotes de complemento ao matplotlib.pyplot
25 from pylab import *
26 import seaborn as sns
27
28 # Pacotes Tweepy, Datetime e Json
29 from tweepy.streaming import StreamListener
30 from tweepy import OAuthHandler
31 from tweepy import Stream
32 from datetime import datetime
33 import json
```

In [45]:

```

1 # Lendo o Microsoft Excel em formato Comma Separated Values (.csv) com as respostas do questionário do Google Forms
2 # e importando para o formato tabular do pacote Pandas
3 df = pd.read_csv("Quiz.csv")
4 df.drop("Carimbo de data/hora", axis=1).head()

```

Out[45]:

	Pontuação total	1.1 - Idade < 18	1.1 - Idade [Pontuação]	1.1 - Idade [Feedback]	1.2 - Sexo	1.2 - Sexo [Pontuação]	1.2 - Sexo [Feedback]	1.3 - Habilidades Académicas	1.3 - Habilidades Académicas [Pontuação]	1.3 - Habilidades Académicas [Feedback]	...	3 - Recomendaria este questionário a alguém?	Recomendação [Pont]
0	69.00 / 100	< 18	-- / 0	NaN	Masculino	-- / 0	NaN	Ensino Secundário	-- / 0	NaN	...	Sim	
1	58.00 / 100	18-25	-- / 0	NaN	Feminino	-- / 0	NaN	Ensino Secundário	-- / 0	NaN	...	Sim	
2	63.00 / 100	18-25	-- / 0	NaN	Feminino	-- / 0	NaN	Ensino Secundário	-- / 0	NaN	...	Sim	
3	64.00 / 100	18-25	-- / 0	NaN	Masculino	-- / 0	NaN	Licenciatura	-- / 0	NaN	...	Sim	
4	57.00 / 100	18-25	-- / 0	NaN	Masculino	-- / 0	NaN	Ensino Secundário	-- / 0	NaN	...	Não	

5 rows × 84 columns



In [3]:

```

1 # Organizando os dados para uma melhor análise, excluindo aqueles que não foram considerados relevantes
2 ## Eliminando colunas desnecessárias
3 for x in df:
4     if ("[Pontuação]" in x):
5         df = df.drop(x, 1)
6     elif ("[Feedback]" in x):
7         df = df.drop(x, 1)
8     elif ("Unnamed" in x):
9         df = df.drop(x, axis=1)
10    elif ("Carimbo de data/hora" == x):
11        df = df.drop(x, axis=1)
12
13 ## Eliminando Linhas de teste que foram feitas para testar o questionário
14 for y in range(0, 4):
15     df = df.drop(y)
16 df = df.drop(7)
17
18 ## Renomeando Linhas
19 df.index = list(range(1,201))
20 df.head()

```

Out[3]:

Pontuação total	1.1 - Idade	1.2 - Sexo	1.3 - Habilidades Académicas	1.4 - Curso a frequentar	2.1 - Tem acesso à Internet em casa?	2.2 - Costuma utilizar a Internet noutra lugar?	2.3 - Qual é o meio que mais utiliza para estar a par das notícias?	2.4 - Sabe o que significa "Fake News"?	2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?	2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?	2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Fiabilidade]
1 57.00 / 100	18-25	Masculino	Ensino Secundário	MIEIC	Sim	Sim	Redes Sociais	Sim	"Fake News" são informações fabricadas que imi...	Frequentemente	4º
2 75.00 / 100	18-25	Masculino	Mestrado	MIEEC	Sim	Sim	Redes Sociais	Sim	"Fake News" são informações fabricadas proposi...	Raramente	5º
3 80.00 / 100	18-25	Masculino	Licenciatura	Mestrado em Multimedia	Sim	Sim	Jornais	Sim	"Fake News" são informações fabricadas proposi...	Nunca	1º
4 77.00 / 100	18-25	Masculino	Ensino Secundário	Mestrado Integrado em Bioengenharia	Sim	Sim	Redes Sociais	Sim	"Fake News" são informações fabricadas proposi...	Nunca	1º
5 38.00 / 100	18-25	Masculino	Mestrado	Mestrado em Engenharia Civil	Sim	Sim	Televisão	Sim	"Fake News" são informações fabricadas que imi...	Raramente	5º

5 rows × 31 columns



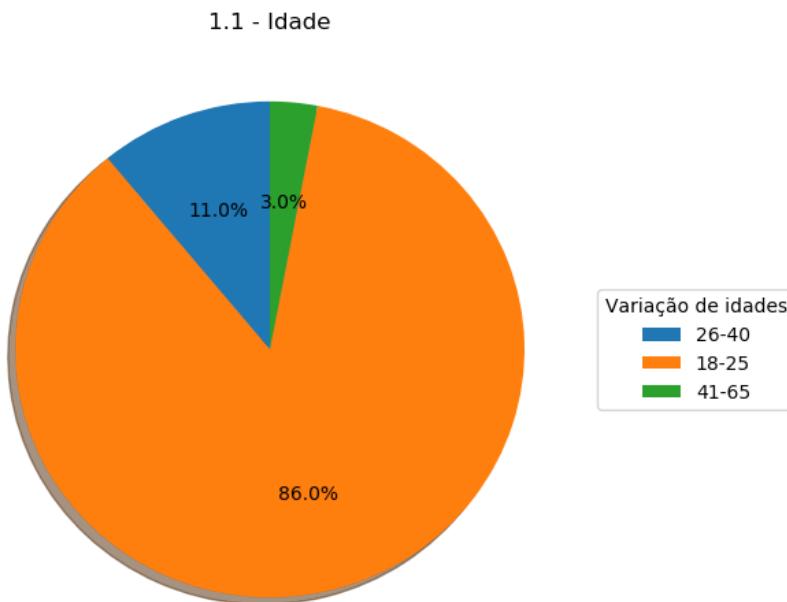
Identificação de Grupos Sociais

In [5]:

```

1 # Pergunta 1.1
2 ## Calcular as categorias e suas frequências
3 idades = ["26-40", "18-25", "41-65"]
4 frequencia = []
5 for x in idades:
6     mask = df["1.1 - Idade"] == x
7     df1 = df[mask]
8     df2 = df1["1.1 - Idade"]
9     df3 = df2.count()
10    frequencia.append(df3)
11
12 ## Visualizar a pergunta 1.1 num gráfico circular
13 plt.figure(figsize = (6, 6), dpi = 100)
14 plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")
15 plt.title("1.1 - Idade")
16 plt.legend(idades, title="Variação de idades", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
17 plt.show()
18 resultado = list(zip(idades, frequencia))
19 resultado

```



Out[5]:

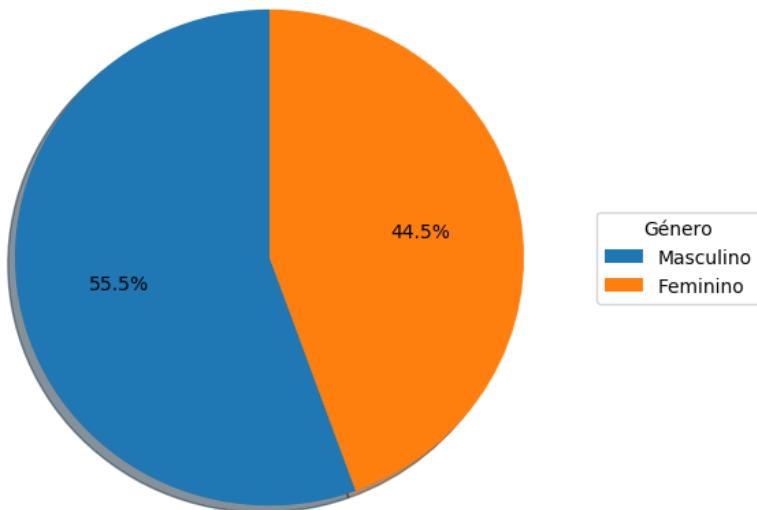
[('26-40', 22), ('18-25', 172), ('41-65', 6)]

Note-se que não existem as restantes categorias, pelo que a sua frequência é 0

In [6]:

```
1 # Pergunta 1.2
2 ## Calcular as categorias e suas frequências
3 genero = ["Masculino", "Feminino"]
4 frequencia = []
5 for x in genero:
6     mask = df["1.2 - Sexo"] == x
7     df1 = df[mask]
8     df2 = df1["1.2 - Sexo"]
9     df3 = df2.count()
10    frequencia.append(df3)
11
12 ## Visualizar a pergunta 1.2 num gráfico circular
13 plt.figure(figsize = (6, 6), dpi = 100)
14 plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%")
15 plt.title("1.2 - Sexo")
16 plt.legend(genero, title="Género", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
17 plt.show()
18 resultado = list(zip(genero, frequencia))
19 resultado
```

1.2 - Sexo



Out[6]:

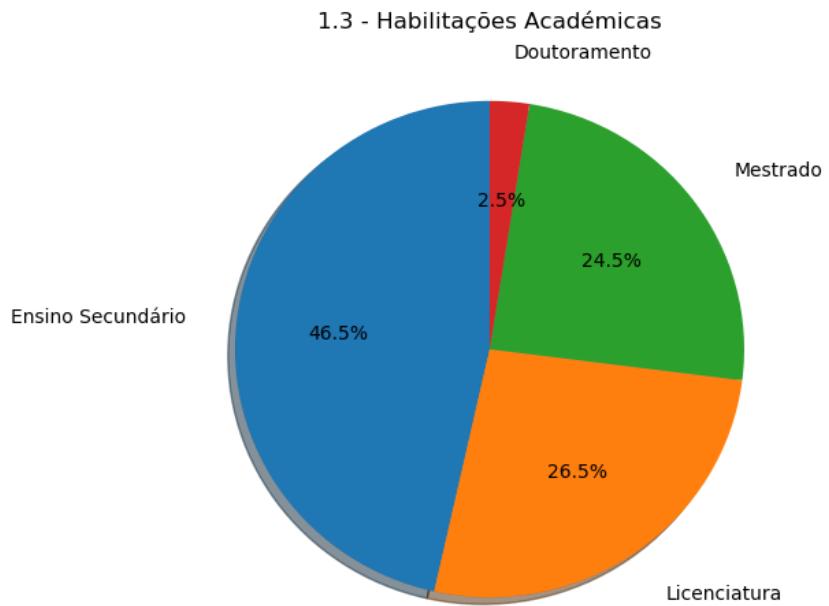
[('Masculino', 111), ('Feminino', 89)]

In [7]:

```

1 # Pergunta 1.3
2 ## Calcular as categorias e suas frequências
3 habilitacoes = ["Ensino Secundário", "Licenciatura", "Mestrado", "Doutoramento"]
4 frequencia = []
5 for x in habilitacoes:
6     mask = df["1.3 - Habilidades Académicas"] == x
7     df1 = df[mask]
8     df2 = df1["1.3 - Habilidades Académicas"]
9     df3 = df2.count()
10    frequencia.append(df3)
11
12 ## Visualizar a pergunta 1.3 num gráfico circular
13 plt.figure(figsize = (6, 6), dpi = 100)
14 plt.pie(frequencia, labels = habilitacoes, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%")
15 plt.title("1.3 - Habilidades Académicas")
16 plt.show()
17 resultado = list(zip(habilitacoes, frequencia))
18 resultado

```



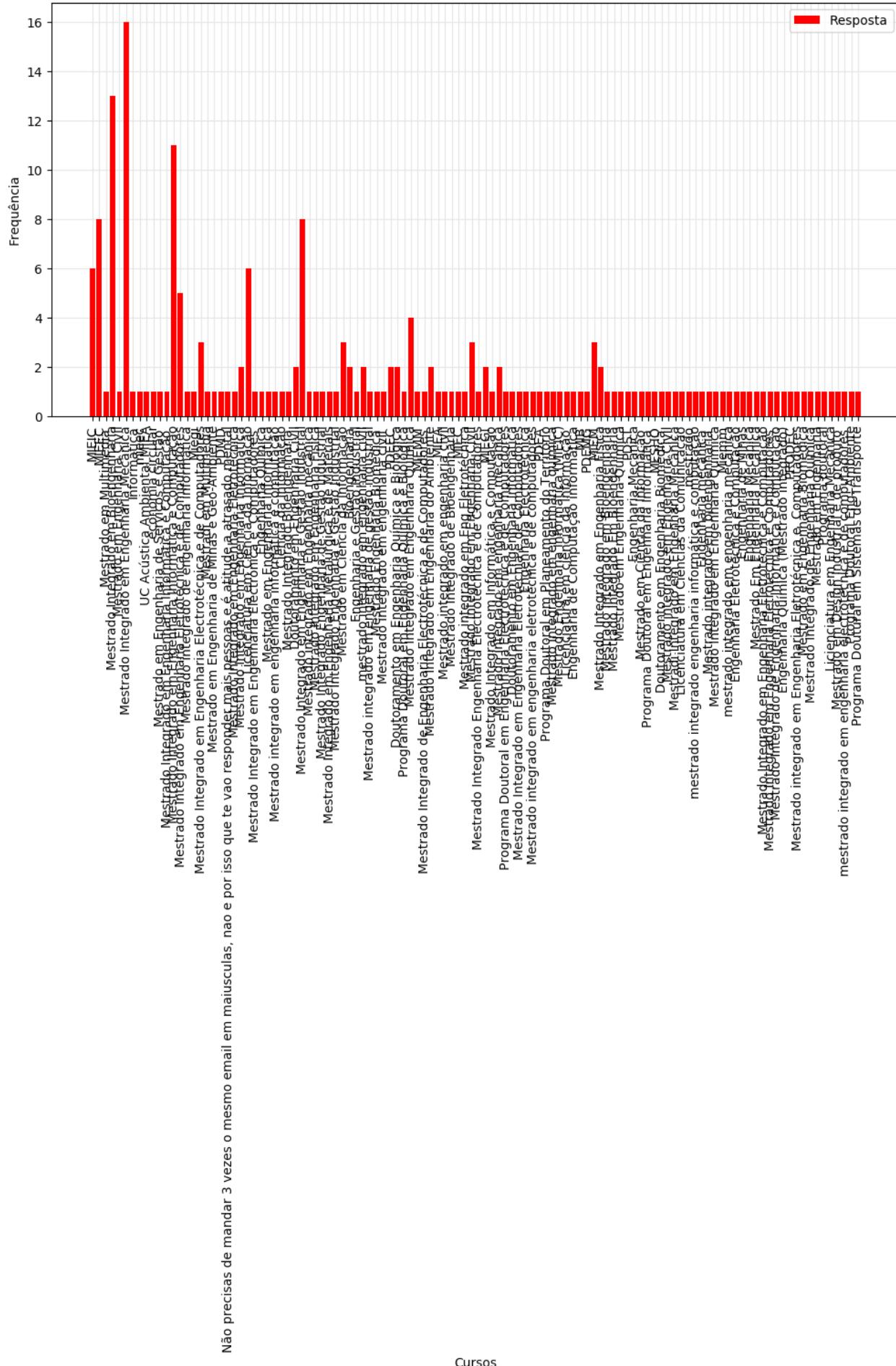
Out[7]:

```
[('Ensino Secundário', 93),
 ('Licenciatura', 53),
 ('Mestrado', 49),
 ('Doutoramento', 5)]
```

In [34]:

```
1 # Pergunta 1.4
2 ## 1º Round
3 ### Calcular todas as categorias (considerando todas as possibilidades para um curso) e suas frequências
4 frequencia = []
5 curso = []
6 for x in df["1.4 - Curso a frequentar"]:
7     if (x not in curso):
8         curso.append(x)
9         mask = df["1.4 - Curso a frequentar"] == x
10        df1 = df[mask]
11        df2 = df1["1.4 - Curso a frequentar"]
12        df3 = df2.count()
13        frequencia.append(df3)
14    else:
15        pass
16
17 ### Visualizar a pergunta 1.4 num gráfico de barras com todas as categorias e frequências
18 plt.figure(figsize = (12, 6), dpi = 100)
19 plt.grid(color = '0.90')
20 plt.bar(curso, frequencia, zorder = 100, color="r", label="Resposta")
21 plt.xlabel("Cursos")
22 plt.title("1.4 - Curso a frequentar")
23 plt.ylabel("Frequência")
24 plt.xticks(rotation=90)
25 plt.legend()
26 plt.show()
27 resultado = list(zip(curso, frequencia))
28 resultado
```

1.4 - Curso a frequentar



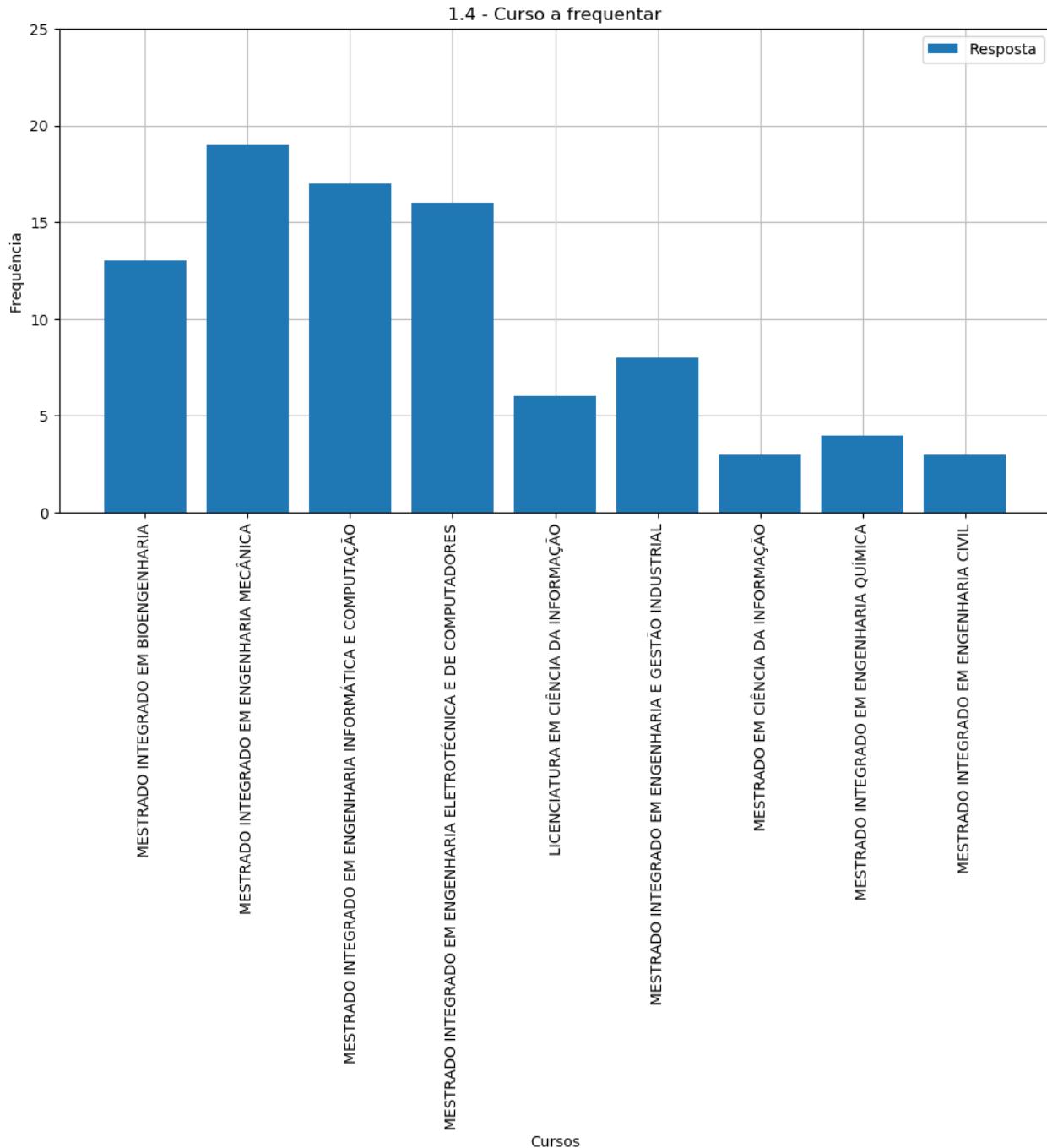
Out[34]:

[('MIEIC', 6),
 ('MIEEC', 8),
 ('Mestrado em Multimédia ', 1),
 ('Mestrado Integrado em Bioengenharia', 13),
 ('Mestrado em Engenharia Civil', 1),
 ('Mestrado Integrado em Engenharia Mecânica', 16),
 ('Informática', 1),
 ('mieec', 1),
 ('UC Acústica Ambiental MIEA', 1),
 ('economia circular', 1),
 ('Mestrado em Engenharia de Serviços e Gestão ', 1),
 ('Mestrado Integrado em Engenharia Informática e Computação ', 1),
 ('Mestrado Integrado em Engenharia Informática e Computação', 11),
 ('Mestrado Integrado em Engenharia Eletrotécnica e de Computadores', 5),
 ('Mestrado integrado de engenharia informática', 1),
 ('Miegi', 1),
 ('Mestrado Integrado em Engenharia Electrotécnica e de Computadores', 3),
 ('Mestradi em Multimédia', 1),
 ('Mestrado em Engenharia de Minas e Geo-Ambiente', 1),
 ('PDMD', 1),
 ('Não precisas de mandar 3 vezes o mesmo email em maiusculas, nao e por isso que te vao responder mais pessoas e é atitude a atrasado mental',
 1),
 ('Mestrado Integrado em Engenharia Eletrotécnica', 1),
 Note: estes integrados em engenharia, sendo muitos deles foram escritos com letras maiúsculas/minúsculas ou com siglas
 ('Licenciatura em Ciência da Informação', 6),
 ('Mestrado Integrado em Engenharia Electrónica e Computadores ', 1),
 ('Engenharia Química', 1),
 ('Mestrado em Engenharia Informática', 1),
 ('Mestrado integrado em engenharia informática e computação', 1),
 ('Ciência da Informação' por 1 letra maiúscula, solucionando o problema dos cursos com letras maiúsculas/minúsculas
 ('Mestrado Integrado Bioengenharia', 1),
 ('Doutoramento em Engenharia Civil', 2),
 ('Mestrado Integrado em Engenharia e Gestão Industrial', 8),
 ('Mestrado integrado em Engenharia mecânica', 1),
 ('Mestrado Integrado em Engenharia Física', 1),
 ('Mestrado Integrado Engenharia de Gestão Industrial', 1),
 ('Mestrado Integrado em Engenharia Metalúrgica e de Materiais', 1),
 ('Mestrado Integrado Engenharia e Gestão Industrial', 1),
 ('Mestrado em Ciência da Informação', 3),
 ('Bioengenharia', 2),
 ('Engenharia e Gestão Industrial', 1),
 ('mestrado integrado em engenharia civil', 2),
 ('Mestrado integrado em engenharia de gestão industrial', 1),
 ('Mestrado Engenharia Biomédica ', 1),
 ('Mestrado integrado em engenharia civil ', 1),
 ('PDEEC', 2),
 ('Doutoramento em Engenharia Química e Biológica', 2),
 ('Programa Doutoral em Engenharia Química e Biológica', 1),
 ('Mestrado Integrado em Engenharia Química', 4),
 ('MIEMM', 1),
 ('Mestrado Integrado de Engenharia Electrotécnica e de Computadores ', 1),
 ('Mestrado Integrado em Engenharia do Ambiente', 2),
 ('MIEA', 1),
 ('Mestrado integrado em engenharia civil', 1),
 ('Mestrado Integrado de Bioengenharia', 1),
 ('MIEC', 1),
 ('Mestrado integrado em Eng Electrotécnica', 1),
 ('Mestrado Integrado em Engenharia Civil', 3),
 ('Mestrado Integrado Engenharia Electrotécnica e de Computadores', 1),
 ('MIEGI', 2),
 ('Mestrado Integrado em Informática e Computação', 1),
 ('Mestrado integrado em engenharia mecânica', 2),
 ('Programa Doutoral em Engenharia Electrotécnica e de Computadores', 1),
 ('Doutoramento em Engenharia Informática', 1),
 ('Mestrado Integrado em Engenharia Eletrotécnica e de Computadores', 1),
 ('Engenharia Electrotécnica', 1),
 ('Mestrado integrado em engenharia eletrotécnica e de computadores', 1),
 ('PDEA', 1),
 ('Programa Doutoral em Planeamento do Território', 1),
 ('Mestrado integrado em engenharia química', 1),
 ('Mestrado Engenharia Informatica (MIEIC)', 1),
 ('Licenciatura em ciência da informação ', 1),
 ('Engenharia de Computação informática', 1),
 ('MIB', 1),
 ('PDEMG', 1),
 ('MIEM', 3),
 ('Mestrado Integrado em Engenharia Física', 2),
 ('Mestrado Integrado em Bioengenharia', 1),
 ('Mestrado Integrado Em Bioengenharia', 1),
 ('Mestrado em Engenharia Química', 1),
 ('PDST', 1),
 ('Engenharia Mecanica', 1),
 ('Mestrado em Ciência da Informação ', 1),
 ('Programa Doutoral em Engenharia Informática ', 1),
 ('MESHO', 1),
 ('Doutoramento em Engenharia Biomédica ', 1),
 ('Mestrado integrado em Engenharia Civil', 1),
 ('Mestradi Integrado em Engenharia Mecânica', 1),

```

('Licenciatura em Ciências da Comunicação', 1),
In['Mestrado', 1),
('mestrado integrado engenharia informática e computação', 1),
Engenharia mecânica ', 1),
Mestrado integrado em bioengenharia ', 1)
Mestrado Integrado em Engenharia Química', 1),
frequencia []
Mitemm = 1)
mestrado integrado em engenharia mecânica', 1),
for x in df1[1:4] - Curso a frequentar]:
Engenharia Eletrotécnica e Computação', 1),
Engenharia de Minas', 1),
Engenharia Mecânica', 1),
Mestrado Em Engenharia Mecânica', 1),
mask = df1[1:4] - Curso a frequentar] == x
Mestrado integrado em Engenharia Eletrotécnica e Computação', 1),
Mestrado Integrado mas
df2 = df1[1:4] - Curso a frequentar]
Mestrado Integrado de Engenharia Informática e Computação', 1),
df3 = df2.count()
Engenharia Química (Mestrado integrado)', 1),
PRODEC', 1), frequencia.append(df3)
Mestrado integrado em Engenharia Eletrotécnica e Computadores', 1),
Mestrado em Engenharia Biomedica', 1),
Mestrado Integrado de Engenharia Química', 1),
Mestrado em Engenharia', 1),
else:
Programa doutoral', 1),
Licenciatura em Engenharia Mecânica', 1),
Mestrado em Design Industrial e de produto ', 1),
##### Somar os valores dos respetivos cursos
mestrado MIEC; MIEEC; MIEEC1; array[frequencia[0]], np.array(frequencia[1]), \
Programa Doutoral Eng. Ambiente', 1),
np.array(frequencia[6]), np.array(frequencia[-1])
Programa Doutoral em Sistemas de Transporte', 1)
array = np.array(frequencia)
array[4] += MIEC; array[5] += MIEEC; array[5] += MIEEC1; array[3] += MIE
frequencia = list(array)
del(frequencia[0], frequencia[0], frequencia[4], frequencia[-1]); del(curso[0], curso[0], curso[4], curso[-1])
curso, frequencia
## Visualizar a pergunta 1.4 num gráfico de barras dos cursos únicos e com frequências superiores a 2
plt.figure(figsize = (12, 6), dpi = 100)
plt.grid(color = '0.75')
plt.bar(curso, frequencia, zorder = 100, label="Resposta")
plt.xlabel("Cursos")
plt.ylabel("Frequência")
plt.xticks(rotation=90)
plt.ylim(0, 25)
plt.title("1.4 - Curso a frequentar")
plt.legend()
plt.show()
resultado = list(zip(curso, frequencia))
resultado

```



Out[54]:

```
[('MESTRADO INTEGRADO EM BIOENGENHARIA', 13),  
 ('MESTRADO INTEGRADO EM ENGENHARIA MECÂNICA', 19),  
 ('MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO', 17),  
 ('MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES', 16),  
 ('LICENCIATURA EM CIÊNCIA DA INFORMAÇÃO', 6),  
 ('MESTRADO INTEGRADO EM ENGENHARIA E GESTÃO INDUSTRIAL', 8),  
 ('MESTRADO EM CIÊNCIA DA INFORMAÇÃO', 3),  
 ('MESTRADO INTEGRADO EM ENGENHARIA QUÍMICA', 4),  
 ('MESTRADO INTEGRADO EM ENGENHARIA CIVIL', 3)]
```

Conhecimento e Comportamento dos Estudantes da FEUP

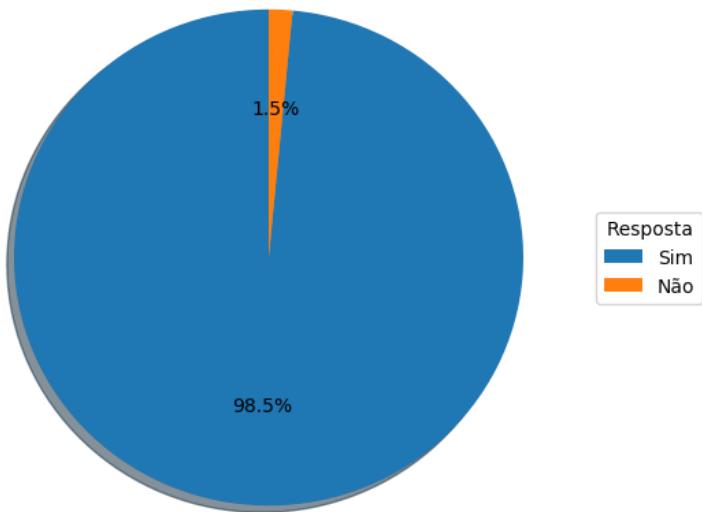
In [10]:

```

1 # Pergunta 2.1
2 ## Calcular as categorias e suas frequências
3 acesso = ["Sim", "Não"]
4 frequencia = []
5 for x in acesso:
6     mask = df["2.1 - Tem acesso à Internet em casa?"] == x
7     df1 = df[mask]
8     df2 = df1["2.1 - Tem acesso à Internet em casa?"]
9     df3 = df2.count()
10    frequencia.append(df3)
11
12 ## Visualizar a pergunta 2.1 num gráfico circular
13 plt.figure(figsize = (6, 6), dpi = 100)
14 plt.pie(frequencia, startangle = 90, shadow = True, autopct="%1.1f%%")
15 plt.title("2.1 - Tem acesso à Internet em casa?")
16 plt.legend(acesso, title="Resposta", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
17 plt.show()
18 resultado = list(zip(acesso, frequencia))
19 resultado

```

2.1 - Tem acesso à Internet em casa?



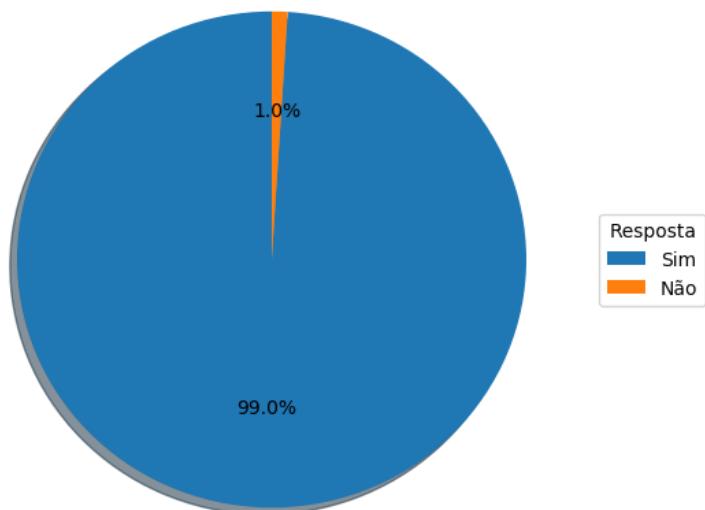
Out[10]:

[('Sim', 197), ('Não', 3)]

In [11]:

```
1 # Pergunta 2.2
2 ## Calcular as categorias e suas frequências
3 utilizacao = ["Sim", "Não"]
4 frequencia = []
5 for x in utilizacao:
6     mask = df["2.2 - Costuma utilizar a Internet noutro lugar?"] == x
7     df1 = df[mask]
8     df2 = df1["2.2 - Costuma utilizar a Internet noutro lugar?"]
9     df3 = df2.count()
10    frequencia.append(df3)
11
12 ## Visualizar a pergunta 2.2 num gráfico circular
13 plt.figure(figsize = (6, 6), dpi = 100)
14 plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%")
15 plt.title("2.2 - Costuma utilizar a Internet noutro lugar?")
16 plt.legend(utilizacao, title="Resposta", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
17 plt.show()
18 resultado = list(zip(utilizacao, frequencia))
19 resultado
```

2.2 - Costuma utilizar a Internet noutro lugar?



Out[11]:

[('Sim', 198), ('Não', 2)]

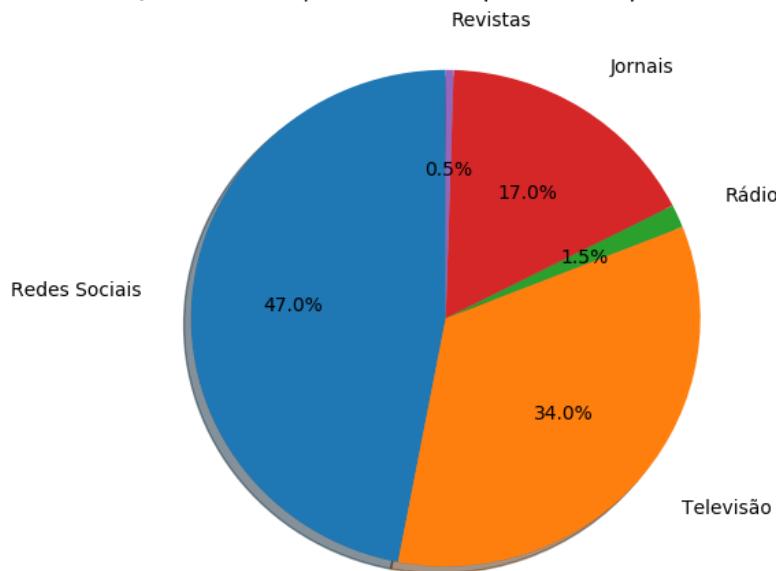
In [12]:

```

1 # Pergunta 2.3
2 ## Calcular as categorias e suas frequências
3 meio = ["Redes Sociais", "Televisão", "Rádio", "Jornais", "Revistas"]
4 frequencia = []
5 for x in meio:
6     mask = df["2.3 - Qual é o meio que mais utiliza para estar a par das notícias?"] == x
7     df1 = df[mask]
8     df2 = df1["2.3 - Qual é o meio que mais utiliza para estar a par das notícias?"]
9     df3 = df2.count()
10    frequencia.append(df3)
11
12 ## Visualizar a pergunta 2.3 num gráfico circular
13 plt.figure(figsize = (6, 6), dpi = 100)
14 plt.pie(frequencia, labels = meio, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%")
15 plt.title("2.3 - Qual é o meio que mais utiliza para estar a par das notícias?")
16 plt.show()
17 resultado = list(zip(meio, frequencia))
18 resultado

```

2.3 - Qual é o meio que mais utiliza para estar a par das notícias?



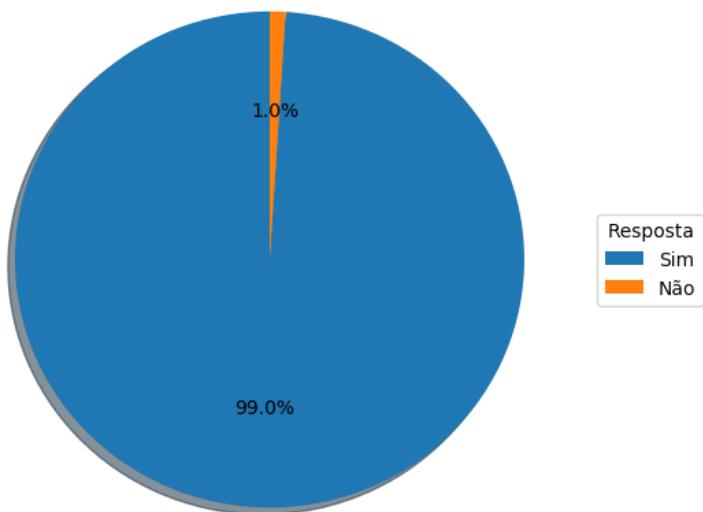
Out[12]:

```
[('Redes Sociais', 94),
 ('Televisão', 68),
 ('Rádio', 3),
 ('Jornais', 34),
 ('Revistas', 1)]
```

In [13]:

```
1 # Pergunta 2.4 (Pontuada)
2 ## Calcular as categorias e suas frequências
3 conhecimento = ["Sim", "Não"]
4 frequencia = []
5 for x in conhecimento:
6     mask = df['2.4 - Sabe o que significa "Fake News"?'] == x
7     df1 = df[mask]
8     df2 = df1['2.4 - Sabe o que significa "Fake News"?']
9     df3 = df2.count()
10    frequencia.append(df3)
11
12 ## Visualizar a pergunta 2.4 num gráfico circular
13 plt.figure(figsize = (6, 6), dpi = 100)
14 plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%")
15 plt.title('2.4 - Sabe o que significa "Fake News"?')
16 plt.legend(conhecimento, title="Resposta", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
17 plt.show()
18 resultado = list(zip(conhecimento, frequencia))
19 resultado
```

2.4 - Sabe o que significa "Fake News"?



Out[13]:

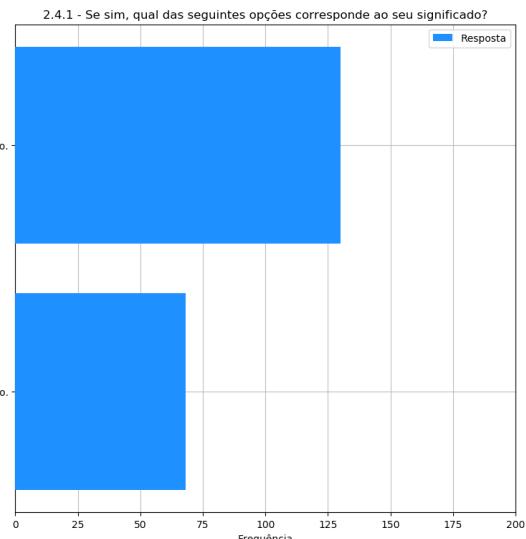
[('Sim', 198), ('Não', 2)]

In [14]:

```

1 # Pergunta 2.4.1 (Pontuada)
2 ## Calcular todas as categorias e frequências
3 frequencia = []
4 resposta = []
5 for x in df["2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?"]:
6     if (type(x) != str):
7         pass
8     else:
9         if (x not in resposta):
10             resposta.append(x)
11             mask = df["2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?"] == x
12             df1 = df[mask]
13             df2 = df1["2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?"]
14             df3 = df2.count()
15             frequencia.append(df3)
16     else:
17         pass
18
19 frequencia = tuple(frequencia)
20 y_pos = np.arange(len(resposta))
21
22 ## Visualizar a pergunta 2.4.1 num gráfico de barras
23 plt.figure(figsize = (9, 9), dpi = 100)
24 plt.grid(color = '0.75')
25 plt.barh(y_pos, frequencia, zorder = 100, label = "Resposta", color = "dodgerblue")
26 plt.xlabel("Frequência")
27 plt.xlim(0, 200)
28 plt.yticks(y_pos, resposta)
29 plt.title("2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?")
30 plt.legend()
31 plt.show()
32 resultado = list(zip(resposta, frequencia))
33 resultado

```



Out[14]:

```

[("“Fake News” são informações fabricadas que imitam propositalmente o conteúdo dos media na forma e no processo ou na intenção.",  
 68),  
 ("“Fake News” são informações fabricadas propositalmente que imitam o conteúdo dos media na forma, mas não no seu proc  
 esso ou na intenção.",  
 130)]

```

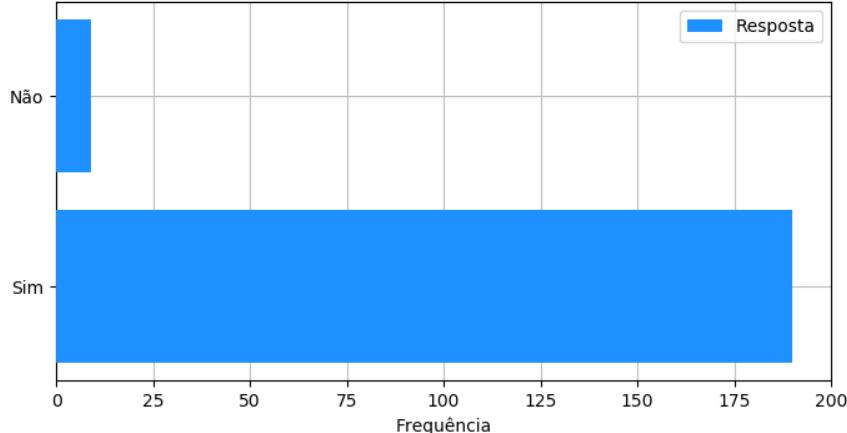
In [58]:

```

1 # Pergunta 2.4.2 (Pontuada)
2 ## Calcular todas as categorias e frequências
3 frequencia = []
4 resposta = []
5 for x in df['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a \
6 divulgação do conhecimento.']:
7     if (type(x) != str):
8         pass
9     else:
10        if (x not in resposta):
11            resposta.append(x)
12            mask = df['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que \
13 prejudica a divulgação do conhecimento.] == x
14            df1 = df[mask]
15            df2 = df1['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que \
16 prejudica a divulgação do conhecimento.']
17            df3 = df2.count()
18            frequencia.append(df3)
19    else:
20        pass
21
22 frequencia = tuple(frequencia)
23 y_pos = np.arange(len(resposta))
24
25 ## Visualizar a pergunta 2.4.2 num gráfico de barras
26 plt.figure(figsize = (8, 4), dpi = 100)
27 plt.grid(color = '0.75')
28 plt.barh(y_pos, frequencia, zorder = 100, label = "Resposta", color = "dodgerblue")
29 plt.xlabel("Frequência")
30 plt.xlim(0, 200)
31 plt.yticks(y_pos, resposta)
32 plt.title('2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a \
33 divulgação do conhecimento.')
34 plt.legend()
35 plt.show()
36 resultado = list(zip(resposta, frequencia))
37 resultado

```

2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.



Out[58]:

```
[('Sim', 190), ('Não', 9)]
```

Para confirmar a importância que se dá ao tema das "Fake News", utilizar-se-á agora a rede Twitter e ver a quantidade de pessoas que estão a falar sobre esta questão neste preciso momento

Preparando a Conexão com o Twitter

In [73]:

```

1 # Adicione aqui sua Consumer Key
2 consumer_key = "k4XWQWP20STZzX0PojQBnTa15"
3
4 # Adicione aqui sua Consumer Secret
5 consumer_secret = "1sT159sofqbJABDgqTkkEJH4L6eWSjcbcYtifikMHsHGJC9ouQA"
6
7 # Adicione aqui seu Access Token
8 access_token = "930524504803115011-gCeP39s6K9grVnqN1FuxxFACu9rd1IO"
9
10 # Adicione aqui seu Access Token Secret
11 access_token_secret = "honuPSnY9HtYhygwQjIQfBagLAxhLrPfpKWhDqONMdF3H"
12
13 # Criando as chaves de autenticação
14 auth = OAuthHandler(consumer_key, consumer_secret)
15 auth.set_access_token(access_token, access_token_secret)
16
17 # Criando uma classe para capturar os stream de dados do Twitter e
18 # armazenar no MongoDB
19 class MyListener(StreamListener):
20     def on_data(self, dados):
21         tweet = json.loads(dados)
22         created_at = tweet["created_at"]
23         id_str = tweet["id_str"]
24         text = tweet["text"]
25         obj = {"created_at":created_at,"id_str":id_str,"text":text}
26         tweetind = col.insert_one(obj).inserted_id
27         print (obj)
28         return True
29
30 # Criando o objeto mylistener
31 mylistener = MyListener()
32
33 # Criando o objeto mystream
34 mystream = Stream(auth, listener = mylistener)
```

Preparando a Conexão com o MongoDB

In [74]:

```

1 # Importando do PyMongo o módulo MongoClient
2 from pymongo import MongoClient
3
4 # Criando a conexão ao MongoDB
5 client = MongoClient('localhost', 27017)
6
7 # Criando o banco de dados twitterdb
8 db = client.twitterdb
9
10 # Criando a collection "col"
11 col = db.tweets
12
13 # Criando uma lista de palavras chave para buscar nos Tweets
14 keywords = ["fake news", "Fake News", "FAKE NEWS"]
```

Coletando os Tweets

In [75]:

```
1 # Iniciando o filtro e gravando os tweets no MongoDB  
2 mystream.filter(track=keywords)
```

```
{'created_at': 'Thu May 23 18:17:39 +0000 2019', 'id_str': '1131625254093316096', 'text': 'Pete Buttigieg: Donald Trump Used \'Status To Fake A Disability,\'' Dodge Vietnam Draft | NBC\xat0News... https://t.co/3GH5LwPW0C', (https://t.co/3GH5LwPW0C)}, '_id': ObjectId('5ce6e3c95ed83f4e8fa76360')}, {'created_at': 'Thu May 23 18:17:40 +0000 2019', 'id_str': '113162525573872646', 'text': 'RT @realDonaldTrump: I was extremely calm yesterday with my meeting with Pelosi and Schumer, knowing that they would say I was raging, whic...', '_id': ObjectId('5ce6e3c95ed83f4e8fa76361')}, {'created_at': 'Thu May 23 18:17:40 +0000 2019', 'id_str': '1131625257293344768', 'text': 'RT @ChatByCC: Ugly hateful people like Joy Behar and all of the fake news ugly haters continue talking ugly hate on President Trump.\n\nWhat...', '_id': ObjectId('5ce6e3c95ed83f4e8fa76362')}, {'created_at': 'Thu May 23 18:17:43 +0000 2019', 'id_str': '1131625268521648128', 'text': '@realDonaldTrump Why don\'t you grow up and be a man. Quit all of your lies and if you have nothing to hide then you... https://t.co/CdxfsSXWZ', (https://t.co/CdxfsSXWZ)}, '_id': ObjectId('5ce6e3cc5ed83f4e8fa76363')}, {'created_at': 'Thu May 23 18:17:43 +0000 2019', 'id_str': '1131625268622299137', 'text': 'Pete Buttigieg: Donald Trump Used \'Status To Fake A Disability,\'' Dodge Vietnam Draft | NBC\xat0News... https://t.co/NlWPCTKK2L', (https://t.co/NlWPCTKK2L)}, '_id': ObjectId('5ce6e3cc5ed83f4e8fa76364')}, {'created_at': 'Thu May 23 18:17:44 +0000 2019', 'id_str': '1131625275123539970', 'text': 'RT @jeffphillips1: Until a popular publisher is docked and jailed for fake news and misinformation, you media people will not stop this irre...', '_id': ObjectId('5ce6e3cd5ed83f4e8fa76365')}, {'created_at': 'Thu May 23 18:17:45 +0000 2019', 'id_str': '1131625279154216960', 'text': '@ambrosia_omG @7thletterja
```

Assim, deduz-se que o tema das "Fake News" é, de facto, um tema de extrema importância, visto que em apenas alguns segundos foram coletados inúmeros tweets por todo mundo

Encerrar a captura dos Tweets

In [76]:

```
1 mystream.disconnect()
```

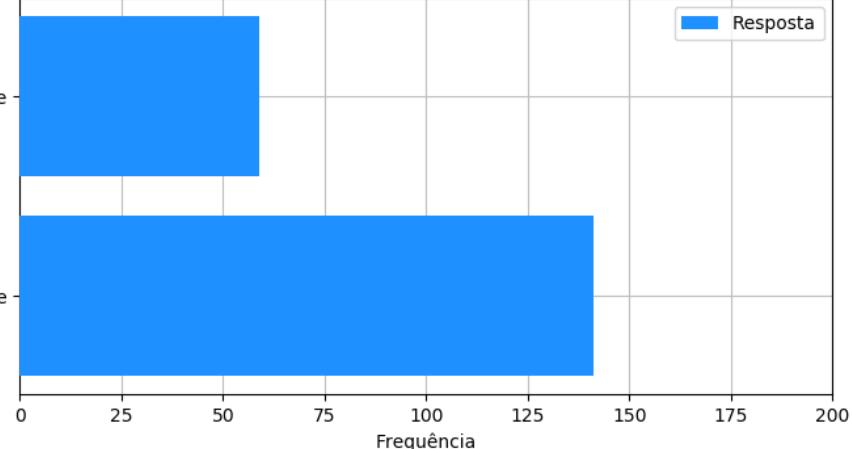
In [16]:

```

1 # Pergunta 2.5 (Pontuada)
2 ## Calcular todas as categorias e frequências
3 frequencia = []
4 resposta = []
5 for x in df["2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?"]:
6     if (x not in resposta):
7         resposta.append(x)
8         mask = df["2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?"] == x
9         df1 = df[mask]
10        df2 = df1["2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?"]
11        df3 = df2.count()
12        frequencia.append(df3)
13 else:
14     pass
15
16 frequencia = tuple(frequencia)
17 y_pos = np.arange(len(resposta))
18
19 ## Visualizar a pergunta 2.5 num gráfico de barras
20 plt.figure(figsize = (8, 4), dpi = 100)
21 plt.grid(color = '0.75')
22 plt.barh(y_pos, frequencia, zorder = 100, label = "Resposta", color = "dodgerblue")
23 plt.xlabel("Frequência")
24 plt.xlim(0, 200)
25 plt.yticks(y_pos, resposta)
26 plt.title("2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?")
27 plt.legend()
28 plt.show()
29 resultado = list(zip(resposta, frequencia))
30 resultado

```

2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?



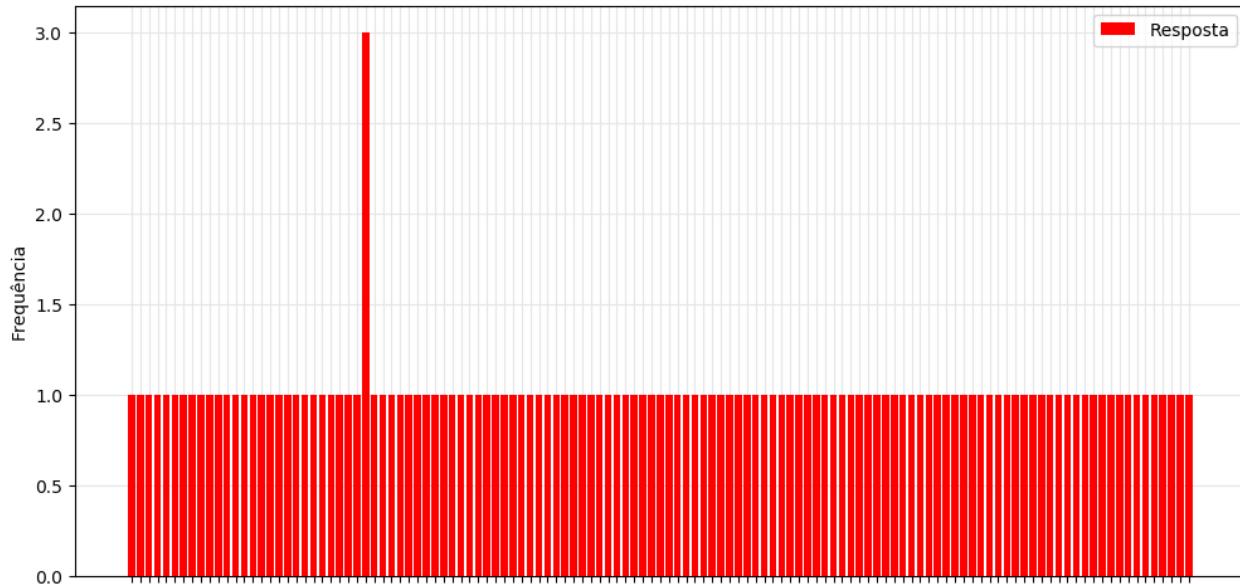
Out[16]:

```
[('Duvido, de imediato, da sua veracidade', 141),
 ('Aceito, de imediato, a sua veracidade', 59)]
```

In [60]:

```
1 # Pergunta 2.5.1
2 ## 1º Round
3 ### Calcular todas as categorias e suas frequências
4 frequencia = []
5 resposta = []
6 for x in df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma \
7 notícia é uma "Fake New"?']:
8     if (type(x) != str):
9         pass
10    else:
11        if (x not in curso):
12            resposta.append(x)
13            mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é \
14 uma "Fake New"?'] == x
15            df1 = df[mask]
16            df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é \
17 uma "Fake New"?']
18            df3 = df2.count()
19            frequencia.append(df3)
20        else:
21            pass
22
23 ### Visualizar a pergunta 2.5.1 num gráfico de barras com todas as categorias e frequências
24 plt.figure(figsize = (12, 6), dpi = 100)
25 plt.grid(color = '0.90')
26 plt.bar(resposta, frequencia, zorder = 100, color="r", label="Resposta")
27 plt.xlabel("Vários tipos de resposta")
28 plt.title('2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "Fake New"?')
29 plt.ylabel("Frequência")
30 plt.xticks(rotation=90)
31 plt.legend()
32 plt.show()
33 resultado = list(zip(resposta, frequencia))
34 resultado
```


2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "Fake News"?



Já as fontes. Aceitam tudo o que Ihes é
Difficilmente uma notícia será 'falsa'
fonte pelo qual estou a ler. Há determinadas
proveniente de um site duvidoso, tipo
o existe um bia óbvio no texto da notícia.
Primeiro, a atenção continua
a escuro saber quem escreve, por norma
as, destacando certas valências que n
C
e sentido. Certamente também com um
Irregularidades na
O que me leva a pensar ser uma falso
A
penso, dentro dos meus dorminhos de sono
de notícia a notícia, mas avalo a notícias (cêrveis) publicarem o falso

A	Hoje em dia os jornais não verificam as suas fontes de informação. O resultado é demasiado "bombástico" ou quando menos é excessivamente simplista.
Pro	Podem apresentar o facto de várias forma diferentes.
Coesão na linguagem	Depende da estrutura do texto, hora da publicação. Se diversas fontes
fonte (redes sociais) e	Depende da estrutura do texto, hora da publicação. Se diversas fontes

Out[60]:

Quanto o t

Os jornalista

Vários tipos de resposta

Se for apenas uma fonte de publicação, con

[('Algo fora do comum', 1),
 ('Por vezes o conteúdo retratado ', 1),
 ('Fontes pouco seguras ou títulos demasiado "clickbait"', 1),
 ('Título muito "atrautivo", chocante.', 1),
 ('O extremo e desnecessário pormenor das notícias', 1),
 ('Conteúdo', 1)]

Nota: que são demasiadas respostas, sendo que muitas delas representam o mesmo tema, mas que foram escritas de diferentes formas

(‘O singular de fake news não é fake new’, 1),
 (‘Erros ortográficos, qualidade das imagens, numero anuncios na pagina’, 1),
 (‘Hoje em dia os jornais não verificam as suas fontes. Aceitam tudo o que lhes é fornecido, porque mesmo que seja mentira ao menos tiveram clicks e deram a notícia.’, 1),
 (‘Procedimentos’),
 (‘Depende da origem da notícia. Do site/canal em questão.’, 1),
 (‘A entidade que a concedeu’, 1),
 (‘A fonte da notícia’, 1),
 (‘Dificilmente uma notícia será ‘fake new’, porque o inglês não admite o singular da palavra ‘news’ na aceção em causa.’, 1),
 (‘A forma como está escrita’, 1),
 (‘Depende do grau de conhecimento que tenho do assunto envolvido’, 1),
 (‘Se não fizer sentido, a.k.a X pessoa não tem razão para fazer Y’, 1),
 (‘Estrutura’, 1),
 (‘A fonte pelo qual estou a ler. Há determinadas fontes em que acredito de imediato, como por exemplo, o JN ou o Cidade Hoje.’, 1),
 (‘Se não tiver uma fonte fidedigna’, 1),
 (‘Título mal estruturado ou com palavras que uma verdadeira notícia não teria.’, 1),
 (‘Website de origem da notícia’, 1),
 (‘A forma de apresentação da notícia e o conteúdo é por vezes claramente duvidoso’, 1),
 (‘Normalmente, há sempre um facto que não ‘bate certo’ com o geral’, 1),
 (‘conteúdo chocante’, 1),
 (‘proveniente de um site duvidoso, tipo de escrita utilizada, não existem fontes, não existem outros artigos sobre o assunto’, 1),
 (‘Quando a notícia não vai ao encontro do meu pensamento crítico.’, 1),
 (‘Quanto o título é demasiado “bombástico” ou quando existe um bias óbvio no texto da notícia (especialmente em notícias relacionadas com assuntos ou personalidades políticas’, 1),
 (‘Fonte’, 3),
 (‘Primeiramente, pelo conteúdo exagerado ou duvidoso da notícia e pela fiabilidade da fonte.’, 1),
 (‘Confiabilidade da fonte, teor da notícia’, 1),
 (‘Redireciona para sites duvidosos, fraco conteúdo de notícia’, 1),
 (‘O site em que estou, vejo as notícias pelo celular’, 1),
 (‘Se parecer pouco plausível ou demasiado sensacionalista’, 1),
 (‘A quantidade de desinformação que existe atualmente’, 1),
 (‘O título muito sensacionalista, a fonte da notícia’, 1),
 (‘A atenção continua que dou aos jornais /0 histórico dos caos, informações, eventos informados nos jornais’, 1),
 (‘Citações de pessoas são geralmente descontextualizadas, propositadamente.’, 1),
 (‘Demasiado sensacionalismo ou a proveniência da fonte’, 1),
 (‘Depende da notícia’, 1),
 (‘Procuro saber quem escreve, por norma pode dizer “redação” o que é duvidoso nos dias de hoje. Depois procuro outras fontes.’, 1),
 (‘A falta de fontes, por exemplo.’, 1),
 (‘Título demasiado extravagante’, 1),
 (‘A improbabilidade de tal situação ocorrer’, 1),
 (‘Os jornalistas podem apresentar o facto de várias formas, destacando certas valências que nem sempre são as que realmente importam. Além disso, os sites são fáceis de hakear.’, 1),
 (‘Se parecer algo demasiado exagerado ou ridículo para ser verdade.’, 1),
 (‘Clickbait’, 1),
 (‘Outras notícias, pesquisas, estado da situação..’, 1),
 (‘fonte’, 1),
 (‘título não muito específico’, 1),
 (‘Fonte da notícia’, 1),
 (‘Inconsistências factuais’, 1),
 (‘Dependendo do quanto credível é a notícia mas principalmente a fonte da mesma’, 1),
 (‘Títulos sensacionalistas, conteúdo pouco coerente, falta de investigação e de factos’, 1),
 (‘Ser polémica, chamativa e ser divulgada só por uma fonte não confiável’, 1),
 (‘O local onde encontrei a notícia, a probabilidade do acontecimento, e quem a divulgou’, 1),
 (‘Fonte, título sensacionalista, forma como está escrita (falta de clareza)’, 1),
 (‘Procedência’, 1),
 (‘A fonte.’, 1),
 (‘A fonte e a forma como é escrita’, 1),
 (‘Quando é corroborada por uma só fonte.’, 1),
 (‘Outra notícia disse algo contrário’, 1),
 (‘forma como está escrita, método de abordagem ao tema’, 1),
 (‘Incongruência com a realidade’, 1),
 (‘conteúdo’, 1),
 (‘Drama excessivo’, 1),
 (‘A fonte’, 1),

('Se for demasiado sensacionalista', 1),
('Escrita, imagens,...', 1),
('.', 1),
('O remetente da notícia e a probabilidade da ocorrência do sucedido.', 1),
('Não estar em outros meios de divulgação', 1),
('Fonte', 3),
('Fonte', 3),
('Quando surge descontextualizada da realidade.', 1),
('A fonte (quem a divulga)', 1),
('Quando é publicada por certas Imprensas.', 1),
('O conteúdo e a fonte de informação ', 1),
('Tento pesquisar melhor sobre o assunto para perceber se de facto é ou não verdade', 1),
('falta de fundamentação', 1),
('Ausência de fontes credíveis.', 1),
('O conteúdo e a forma como este é apresentado', 1),
('O formato, a informação exposta, a fonte', 1),
('Defesa de alguma ideologia ', 1),
('Origem da notícia', 1),
('A fonte que publicou a notícia', 1),
('Se me parecer irrealista ou conveniente para quem a partilha', 1),
('Forma como está escrito, sensacionalismo demasiado', 1),
('Quem a publica, parcialidade excessiva.', 1),
('pouca clareza, factos imprecisos, títulos ou expressões não rigorosas', 1),
('Muito sensacionalista, fonte, erros ortográficos, entre outros.', 1),
('Lógica', 1),
('ser do buzzfeed', 1),
('O nome do site', 1),
('A legitimidade do autor ', 1),
('o local onde esta está publicada', 1),
('Coesão na linguagem e sentido. Certamente também com um pouco de investigação autónoma no caso de o tema incitar um certo interesse nele.', 1),
('Se o seu conteúdo for algo anormal', 1),
('Os títulos nem sempre correspondem à verdade', 1),
('Irregularidades na forma como é apresentada e o facto de ser uma novidade incomum no dado contexto', 1),
('se for uma notícia proveniente de uma fonte não fiável', 1),
('O que me leva a pensar ser uma fake news, é o facto da notícia em si parecer irrealista e/ou acompanhada por "clickbait".', 1),
('congência e coerência do conteúdo', 1),
('A importância do tópico em termos de influência.', 1),
('A falta de fontes (de confiança)', 1),
('A ausência da notícia noutras fontes e meios ou a baixa probabilidade de ela acontecer.', 1),
('Se ela for "bizarra" é um bom indicador', 1),
('Credibilidade da fonte', 1),
('A fonte da notícia ', 1),
('Fontes desconhecidas, etc', 1),
('A fonte, o estilo, e mesmo o conteúdo....', 1),
('Sensacionalismo', 1),
('Informação vaga ou que impinge um tipo de pensamento', 1),
('Quando o meio de divulgação não tem grande reputação ou é conhecido por "inventar" notícias.', 1),
('fonte (redes sociais) e penso, dentro dos meus domínios de conhecimento, o quanto aquilo parece real. E então busco outras fontes que confio mais.', 1),
('Se a acho inverosímil ', 1),
('A minha intuição', 1),
('Principalmente, a fonte', 1),
('Quando o tema é exageradamente absurdo', 1),
('a fonte, o conteúdo ', 1),
('Depende de notícia a notícia, mas avalio a notícia contra o meu conhecimento básico do mundo, estatísticas básicas e senso comum.', 1),
('A forma como está escrita, se faz sentido.', 1),
('Source duvidosa, subjetividade da notícia ou dados que contradizem algo que sei que é um facto', 1),
('Se for apenas uma fonte de publicação, contexto, hora da publicação. Se diversas fontes (credíveis) publicarem o mesmo tipo de notícia é menos provável, ainda que não seja impossível, que não seja fake news', 1),
('Se não vier de um site fidedigno ou de uma fonte verificável', 1),
(['credibilidade do site ou canal onde a vi', 1])

In [61]:

```

1 # Pergunta 2.5.1
2 ## 2º Round
3 ### Calcular apenas as categorias mais utilizadas e suas frequências
4 frequencia = []
5 resposta = []
6
7 frequencia_f = []
8 for x in df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "Fake New"?']:
9     count_font = []
10    if (type(x) == str):
11        ##### Filtrar as respostas por palavras-chave respetivos ao tema da "Fonte"
12        if ("Fontes" in x) or ("fonte" in x) or ("Fontes" in x) or ("fontes" in x):
13            if ("Fontes" not in resposta):
14                resposta.append("Fontes")
15                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
16                df1 = df[mask]
17                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
18                df3 = df2.count()
19                count_font.append(df3)
20            else:
21                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
22                df1 = df[mask]
23                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
24                df3 = df2.count()
25                count_font.append(df3)
26        else:
27            pass
28    else:
29        pass
30
31    count_font = np.array(count_font).sum()
32    frequencia_f.append(count_font)
33
34
35    frequencia_c = []
36 for x in df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "Fake New"?']:
37     count_cont = []
38     if (type(x) == str):
39         ##### Filtrar as respostas por palavras-chave respetivos ao tema do "Conteúdo"
40         if ("Conteúdo" in x) or ("conteúdo" in x) or ("Conteudo" in x) or ("conteudo" in x) or ("Conteúdos" in x) or \
41             ("conteúdos" in x) or ("Conteudos" in x) or ("conteudos" in x):
42             if ("Conteúdo" not in resposta):
43                 resposta.append("Conteúdo")
44                 mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
45                 df1 = df[mask]
46                 df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
47                 df3 = df2.count()
48                 count_cont.append(df3)
49             else:
50                 mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
51                 df1 = df[mask]
52                 df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
53                 df3 = df2.count()
54                 count_cont.append(df3)
55         else:
56             pass
57     else:
58         pass
59
60     count_cont = np.array(count_cont).sum()
61     frequencia_c.append(count_cont)
62
63
64    frequencia_t = []
65 for x in df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é \
uma "Fake New"?']:
66     count_tit = []
67     if (type(x) == str):
68         ##### Filtrar as respostas por palavras-chave respetivos ao tema do "Título"
69         if ("Título" in x) or ("titulo" in x) or ("Titulo" in x) or ("titulo" in x) or ("Títulos" in x) or \
70             ("títulos" in x) or ("Titulos" in x) or ("titulos" in x):
71             if ("Título" not in resposta):
72                 resposta.append("Título")
73                 mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
74                 df1 = df[mask]
75                 df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
76                 df3 = df2.count()
77                 count_tit.append(df3)
78             else:
79                 mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
80
81
82
83
84
85
86
87
88

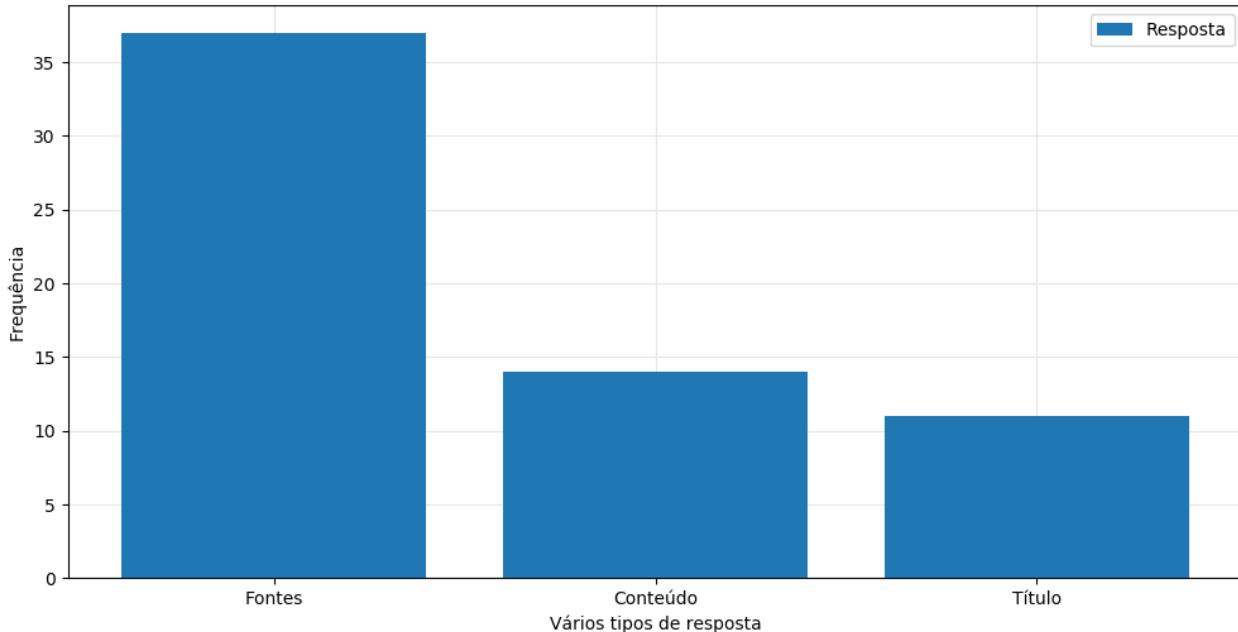
```

```

89     uma notícia é uma "Fake New"?] == x
90         df1 = df[mask]
91         df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
92         df3 = df2.count()
93         count_tit.append(df3)
94
95     else:
96         pass
97 else:
98     pass
99
100    count_tit = np.array(count_tit).sum()
101    frequencia_t.append(count_tit)
102
103
104    frequencia_f = np.array(frequencia_f).sum()
105    frequencia_c = np.array(frequencia_c).sum()
106    frequencia_t = np.array(frequencia_t).sum()
107    frequencia.append(frequencia_f); frequencia.append(frequencia_c); frequencia.append(frequencia_t)
108
109
110    ### Visualizar a pergunta 2.5.1 num gráfico de barras com os temas e suas respetivas frequências
111    plt.figure(figsize = (12, 6), dpi = 100)
112    plt.grid(color = '0.90')
113    plt.bar(resposta, frequencia, zorder = 100, label="Resposta")
114    plt.xlabel("Vários tipos de resposta")
115    plt.title('2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "Fake New"?')
116    plt.ylabel("Frequência")
117    plt.legend()
118    plt.show()
119    resultado = list(zip(resposta, frequencia))
120    resultado

```

2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "Fake New"?



Out[61]:

[('Fontes', 37.0), ('Conteúdo', 14.0), ('Título', 11.0)]

In [62]:

```

1 # Pergunta 2.6 (Pontuada)
2 ## Calcular as categorias e suas frequências
3 resposta = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
4 titulo = ["Verifico a fonte", "Verifico outras fontes para confirmar",
5           "Verifico o autor da notícia", "Verifico o aspetto da página",
6           "Verifico a data", "Consulto uma pessoa"]
7
8 ### Calcular as frequências das respostas para cada sub-título dentro do título
9 frequencia_1 = []
10 for x in resposta:
11     mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
12 de analisar uma notícia? [Verifico a fonte da notícia]"] == x
13     df1 = df[mask]
14     df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
15 de analisar uma notícia? [Verifico a fonte da notícia]"]
16     df3 = df2.count()
17     frequencia_1.append(df3)
18
19
20 frequencia_2 = []
21 for x in resposta:
22     mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
23 de analisar uma notícia? [Verifico outras fontes para confirmar a notícia]"] == x
24     df1 = df[mask]
25     df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
26 de analisar uma notícia? [Verifico outras fontes para confirmar a notícia]"]
27     df3 = df2.count()
28     frequencia_2.append(df3)
29
30
31 frequencia_3 = []
32 for x in resposta:
33     mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
34 de analisar uma notícia? [Verifico o autor da notícia]"] == x
35     df1 = df[mask]
36     df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
37 de analisar uma notícia? [Verifico o autor da notícia]"]
38     df3 = df2.count()
39     frequencia_3.append(df3)
40
41
42 frequencia_4 = []
43 for x in resposta:
44     mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
45 de analisar uma notícia? [Verifico o aspetto da página (caso se trate de uma notícia online)]"] == x
46     df1 = df[mask]
47     df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
48 de analisar uma notícia? [Verifico o aspetto da página (caso se trate de uma notícia online)]"]
49     df3 = df2.count()
50     frequencia_4.append(df3)
51
52
53 frequencia_5 = []
54 for x in resposta:
55     mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
56 de analisar uma notícia? [Verifico a data da notícia]"] == x
57     df1 = df[mask]
58     df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
59 de analisar uma notícia? [Verifico a data da notícia]"]
60     df3 = df2.count()
61     frequencia_5.append(df3)
62
63
64 frequencia_6 = []
65 for x in resposta:
66     mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
67 de analisar uma notícia? [Consulto uma pessoa que esteja mais à vontade com o assunto da notícia]"] == x
68     df1 = df[mask]
69     df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
70 de analisar uma notícia? [Consulto uma pessoa que esteja mais à vontade com o assunto da notícia]"]
71     df3 = df2.count()
72     frequencia_6.append(df3)
73
74
75 ## Visualizar a pergunta 2.6 num gráfico de barras empilhadas
76 N = 6
77 ind = np.arange(N)
78 width = 0.70
79
80 frequencia_a = [frequencia_1[0], frequencia_2[0], frequencia_3[0], frequencia_4[0], frequencia_5[0], frequencia_6[0]]
81 frequencia_b = [frequencia_1[1], frequencia_2[1], frequencia_3[1], frequencia_4[1], frequencia_5[1], frequencia_6[1]]
82 frequencia_c = [frequencia_1[2], frequencia_2[2], frequencia_3[2], frequencia_4[2], frequencia_5[2], frequencia_6[2]]
83 frequencia_d = [frequencia_1[3], frequencia_2[3], frequencia_3[3], frequencia_4[3], frequencia_5[3], frequencia_6[3]]
84 frequencia_e = [frequencia_1[4], frequencia_2[4], frequencia_3[4], frequencia_4[4], frequencia_5[4], frequencia_6[4]]
85
86 plt.figure(figsize = (12, 6), dpi = 100)
87 p1 = plt.bar(ind, frequencia_a, width, zorder = 100) # Apagar e inserir para ver a evolução dos gráficos
88 p2 = plt.bar(ind, frequencia_b, width, zorder = 100)
89 p3 = plt.bar(ind, frequencia_c, width, zorder = 100)
90 p4 = plt.bar(ind, frequencia_d, width, zorder = 100)
91 p5 = plt.bar(ind, frequencia_e, width, zorder = 100)

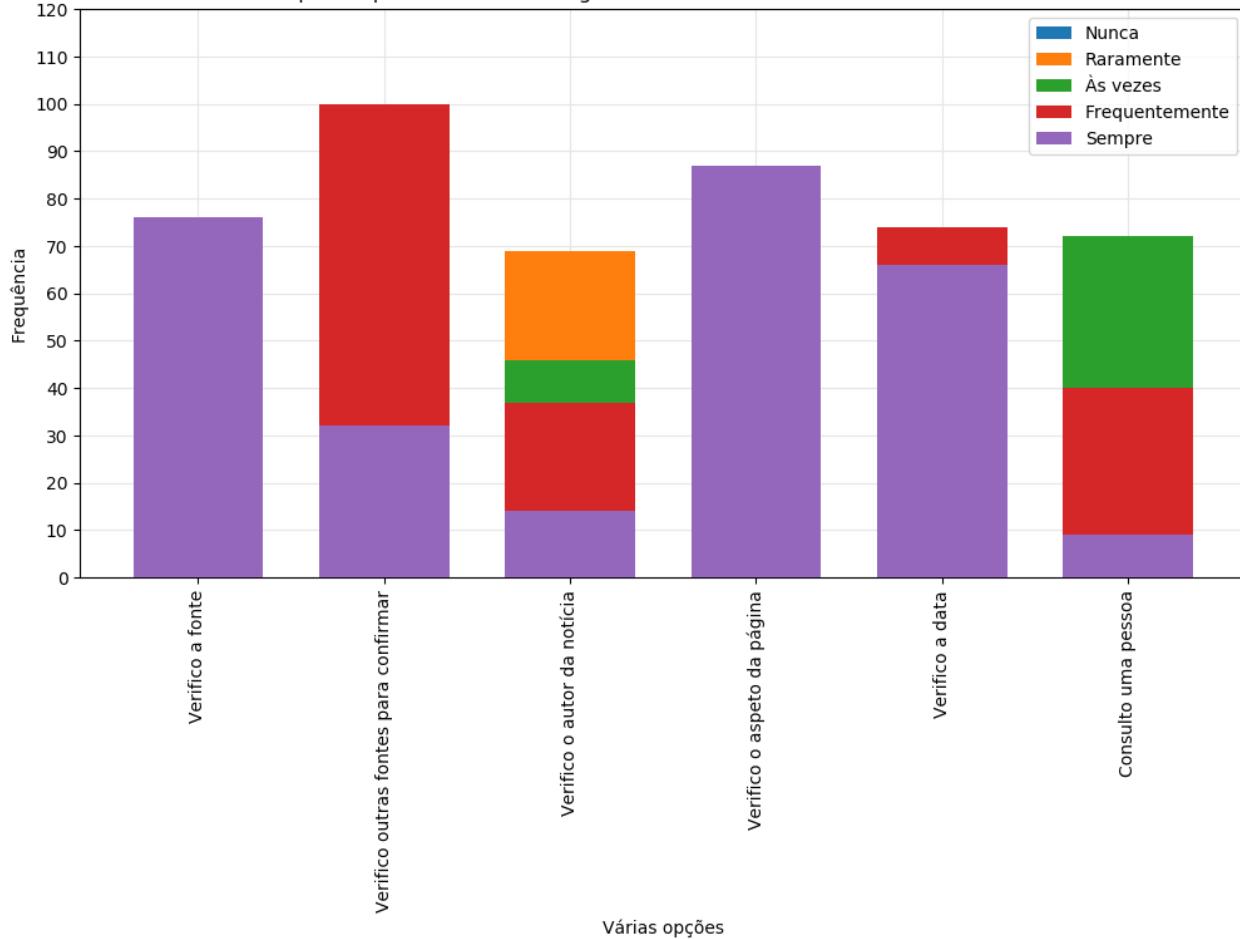
```

```

89 plt.grid(color = '0.90')
90 plt.xlabel('Várias opções')
91 plt.title('2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia?')
92 plt.xticks(ind, (titulo[0], titulo[1], titulo[2], titulo[3], titulo[4], titulo[5]), rotation=90)
93 plt.yticks(np.arange(0, 130, 10))
94 plt.ylabel('Frequência')
95 plt.legend((p1[0], p2[0], p3[0], p4[0], p5[0]), (resposta[0], resposta[1], resposta[2], resposta[3], resposta[4])))
96 plt.show()
97 print(frequencia_1, frequencia_2, frequencia_3, frequencia_4, frequencia_5, frequencia_6)
98 print(titulo)

```

2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia?



[4, 22, 29, 69, 76] [4, 13, 51, 100, 32] [34, 69, 46, 37, 14] [5, 11, 31, 66, 87] [2, 17, 41, 74, 66] [26, 53, 72, 40, 9]
['Verifico a fonte', 'Verifico outras fontes para confirmar', 'Verifico o autor da notícia', 'Verifico o aspeto da página', 'Verifico a data', 'Consulto uma pessoa']

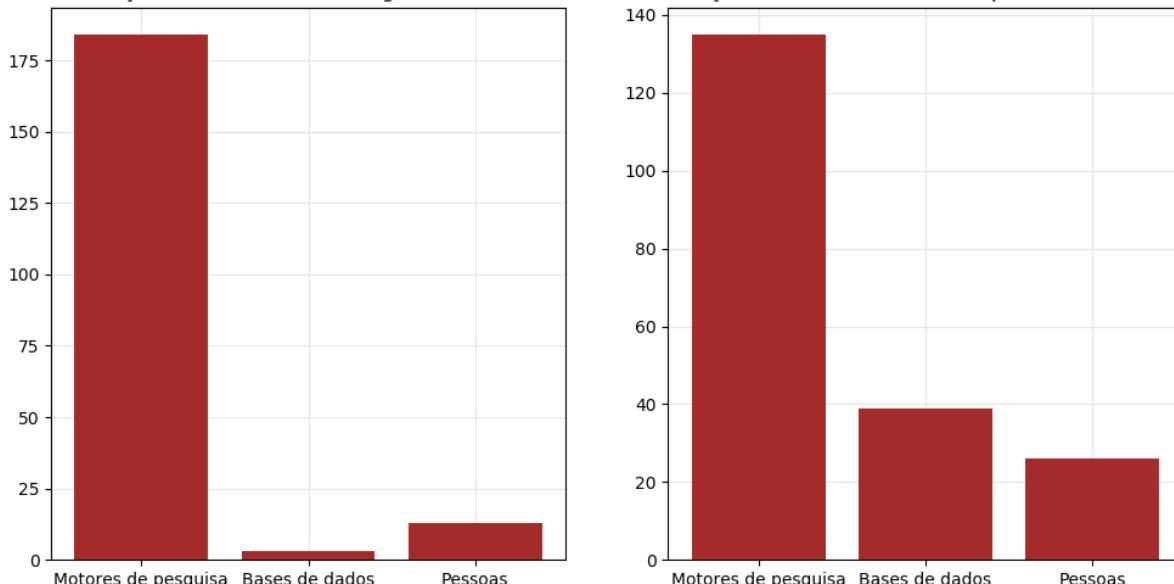
In [63]:

```

1 # Perguntas 2.7 e 2.8 (Pontuada)
2 ## Calcular todas as categorias e suas frequências
3 fonte = ["Motores de pesquisa (Google, entre outros)", "Bases de dados", "Pessoas"]
4
5 ### Calcular a frequência de cada fonte para cada pergunta
6 frequencia_1 = []
7 for x in fonte:
8     mask = df["2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma \
9 necessidade de informação geral de fácil acesso?"] == x
10    df1 = df[mask]
11    df2 = df1["2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma \
12 necessidade de informação geral de fácil acesso?"]
13    df3 = df2.count()
14    frequencia_1.append(df3)
15
16
17 frequencia_2 = []
18 for x in fonte:
19     mask = df["2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma \
20 necessidade de informação específica de difícil acesso?"] == x
21     df1 = df[mask]
22     df2 = df1["2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma \
23 necessidade de informação específica de difícil acesso?"]
24     df3 = df2.count()
25     frequencia_2.append(df3)
26
27 fonte[0] = "Motores de pesquisa"
28
29 ## Visualizar a pergunta 2.7 e 2.8 numa figura com dois gráficos de barras em paralelo
30 fig = plt.figure(figsize = (12, 6), dpi = 100)
31
32 ax1 = fig.add_subplot(1,2,1)
33 ax1.bar(fonte, frequencia_1, color = "brown", zorder = 100)
34 ax1.set_title("2.7 Satisfação de uma necessidade geral de fácil acesso?")
35 ax1.grid(color = '0.90')
36
37
38 ax2 = fig.add_subplot(1,2,2)
39 ax2.bar(fonte, frequencia_2, color = "brown", zorder = 100)
40 ax2.set_title("2.8 Satisfação de uma necessidade específica de difícil acesso?")
41 ax2.grid(color = '0.90')
42
43 plt.show()
44
45 resultado1 = list(zip(fonte, frequencia_1)); resultado2 = list(zip(fonte, frequencia_2))
46 print(resultado1)
47 print(resultado2)
48

```

2.7 Satisfação de uma necessidade geral de fácil acesso? 2.8 Satisfação de uma necessidade específica de difícil acesso?



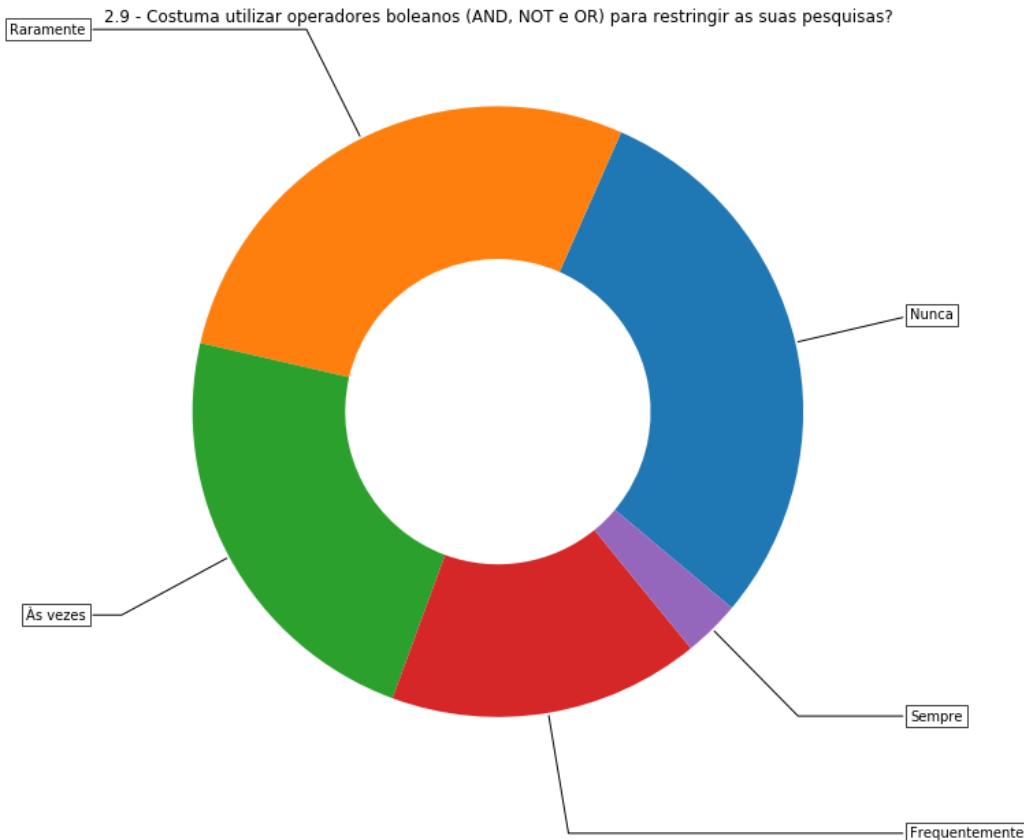
```
[('Motores de pesquisa', 184), ('Bases de dados', 3), ('Pessoas', 13)]
[('Motores de pesquisa', 135), ('Bases de dados', 39), ('Pessoas', 26)]
```

In [64]:

```

1 # Pergunta 2.9 (Pontuada)
2 ## Calcular as categorias e suas frequências
3 resposta = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
4 frequencia = []
5 for x in resposta:
6     mask = df["2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir \
7 as suas pesquisas?"] == x
8     df1 = df[mask]
9     df2 = df1["2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir \
10 as suas pesquisas?"]
11     df3 = df2.count()
12     frequencia.append(df3)
13
14 ## Visualizar a pergunta 1.1 num gráfico circular (donut)
15 fig, ax = plt.subplots(figsize=(10, 10), subplot_kw=dict(aspect="equal"))
16
17 wedges, texts = ax.pie(frequencia, wedgeprops=dict(width=0.5), startangle=-40)
18
19 bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
20 kw = dict(xycoords='data', textcoords='data', arrowprops=dict(arrowstyle="-"),
21           bbox=bbox_props, zorder=0, va="center")
22
23 for i, p in enumerate(wedges):
24     ang = (p.theta2 - p.theta1)/2. + p.theta1
25     y = np.sin(np.deg2rad(ang))
26     x = np.cos(np.deg2rad(ang))
27     horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
28     connectionstyle = "angle,angleA=0,angleB={}".format(ang)
29     kw["arrowprops"].update({"connectionstyle": connectionstyle})
30     ax.annotate(resposta[i], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
31                 horizontalalignment=horizontalalignment, **kw)
32
33 ax.set_title("2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?")
34
35 plt.show()
36 resultado = list(zip(resposta, frequencia))
37 print(resultado)

```



```
[('Nunca', 59), ('Raramente', 56), ('Às vezes', 46), ('Frequentemente', 33), ('Sempre', 6)]
```


In [22]:

```

1 # Pergunta 2.10 e 2.10.1 (Pontuada)
2 ## Calcular as categorias e suas frequências
3 posicao = ["1º", "2º", "3º", "4º", "5º"]
4 criterio = ["Fiabilidade", "Relevância", "Atualização", "Completa", "Explicita"]
5 ordem = "Ordem"
6
7 ### Calcular a frequência de cada posição para cada pergunta (2.10)
8 frequencia_1 = []
9 for x in posicao:
10     for y in df:
11         if ("[Fiabilidade]" in y):
12             mask = df[y] == x
13             df1 = df[mask]
14             df2 = df1[y]
15             df3 = df2.count()
16             frequencia_1.append(df3)
17
18
19 frequencia_2 = []
20 for x in posicao:
21     for y in df:
22         if ("[Relevância]" in y):
23             mask = df[y] == x
24             df1 = df[mask]
25             df2 = df1[y]
26             df3 = df2.count()
27             frequencia_2.append(df3)
28
29
30 frequencia_3 = []
31 for x in posicao:
32     for y in df:
33         if ("[Atualização]" in y):
34             mask = df[y] == x
35             df1 = df[mask]
36             df2 = df1[y]
37             df3 = df2.count()
38             frequencia_3.append(df3)
39
40
41 frequencia_4 = []
42 for x in posicao:
43     for y in df:
44         if ("[Completa]" in y):
45             mask = df[y] == x
46             df1 = df[mask]
47             df2 = df1[y]
48             df3 = df2.count()
49             frequencia_4.append(df3)
50
51
52 frequencia_5 = []
53 for x in posicao:
54     for y in df:
55         if ("[Explicita]" in y):
56             mask = df[y] == x
57             df1 = df[mask]
58             df2 = df1[y]
59             df3 = df2.count()
60             frequencia_5.append(df3)
61
62
63
64 frequencia = []
65 mask = df["2.10.1 - Ordem"] == "[2º, 1º, 3º, 5º, 4º]"
66 df1 = df[mask]
67 df2 = df1["2.10.1 - Ordem"]
68 df3 = df2.count()
69 frequencia.append(df3)
70
71
72 ## Visualizar as perguntas 2.10 numa figura com múltiplos gráficos respetivos às opções
73 print("2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação")
74 fig, ax = plt.subplots(2,3, figsize = (16, 8))
75
76 ax[0,0].bar(posicao, frequencia_1, color = "darkmagenta")
77 ax[0,0].set_title("Fiabilidade")
78
79 ax[0,1].bar(posicao, frequencia_2, color = "goldenrod")
80 ax[0,1].set_title("Relevância")
81
82 ax[0,2].bar(posicao, frequencia_3, color = "crimson")
83 ax[0,2].set_title("Atualização")
84
85 ax[1,0].bar(posicao, frequencia_4, color = "orangered")
86 ax[1,0].set_title("Completa")
87
88 ax[1,2].bar(posicao, frequencia_5, color = "mediumspringgreen")
89 ax[1,2].set_title("Explicita")

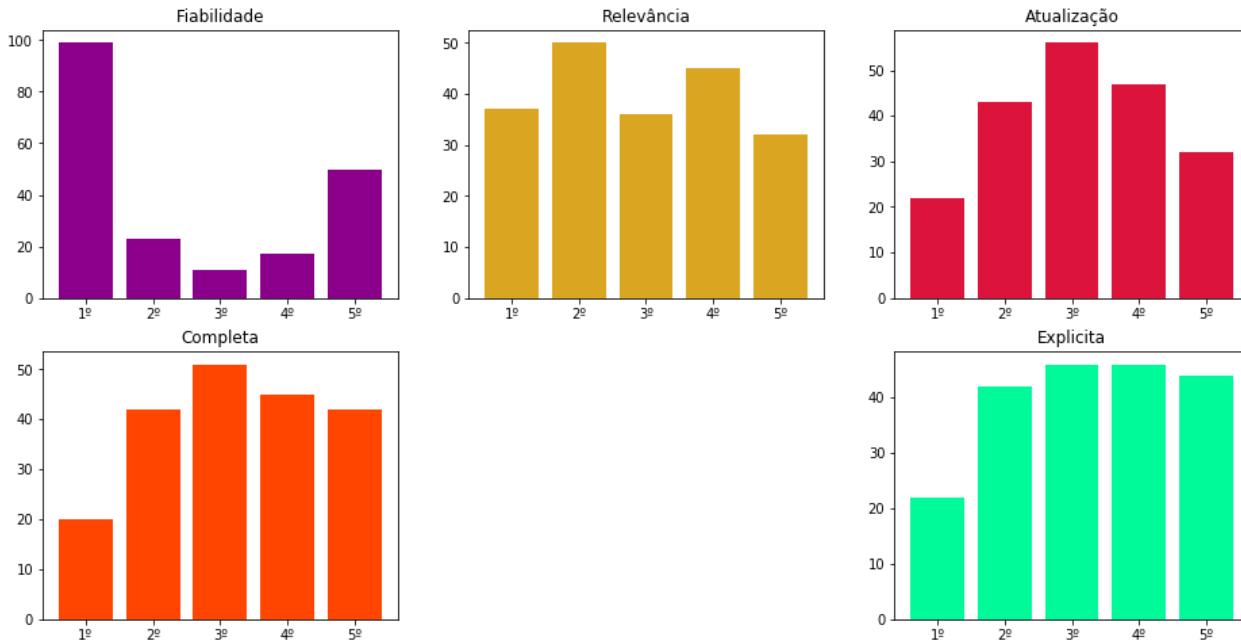
```

```

89
90 ax[1,1].remove()
91
92 plt.show()
93 print(frequencia_1, frequencia_2, frequencia_3, frequencia_4, frequencia_5, frequencia_6)
94 print(criterio)
95 print("-----")
96 cursos = ["LCI", "MIEM"]
97
98 print("Frequência das pessoas que acertaram na ordem correta:", str(frequencia))
99 resultado = list(zip(cursos, df2))
100 print(resultado)
101

```

2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação



[99, 23, 11, 17, 50] [37, 50, 36, 45, 32] [22, 43, 56, 47, 32] [20, 42, 51, 45, 42] [22, 42, 46, 46, 44] [26, 53, 72, 40, 9]

['Fiabilidade', 'Relevância', 'Atualização', 'Completa', 'Explicita']

Frequência das pessoas que acertaram na ordem correta: [2]
[('LCI', "[2º', '1º', '3º', '5º', '4º']"), ('MIEM', "[2º', '1º', '3º', '5º', '4º']")]

Caracterização da Amostra

In [67]:

```

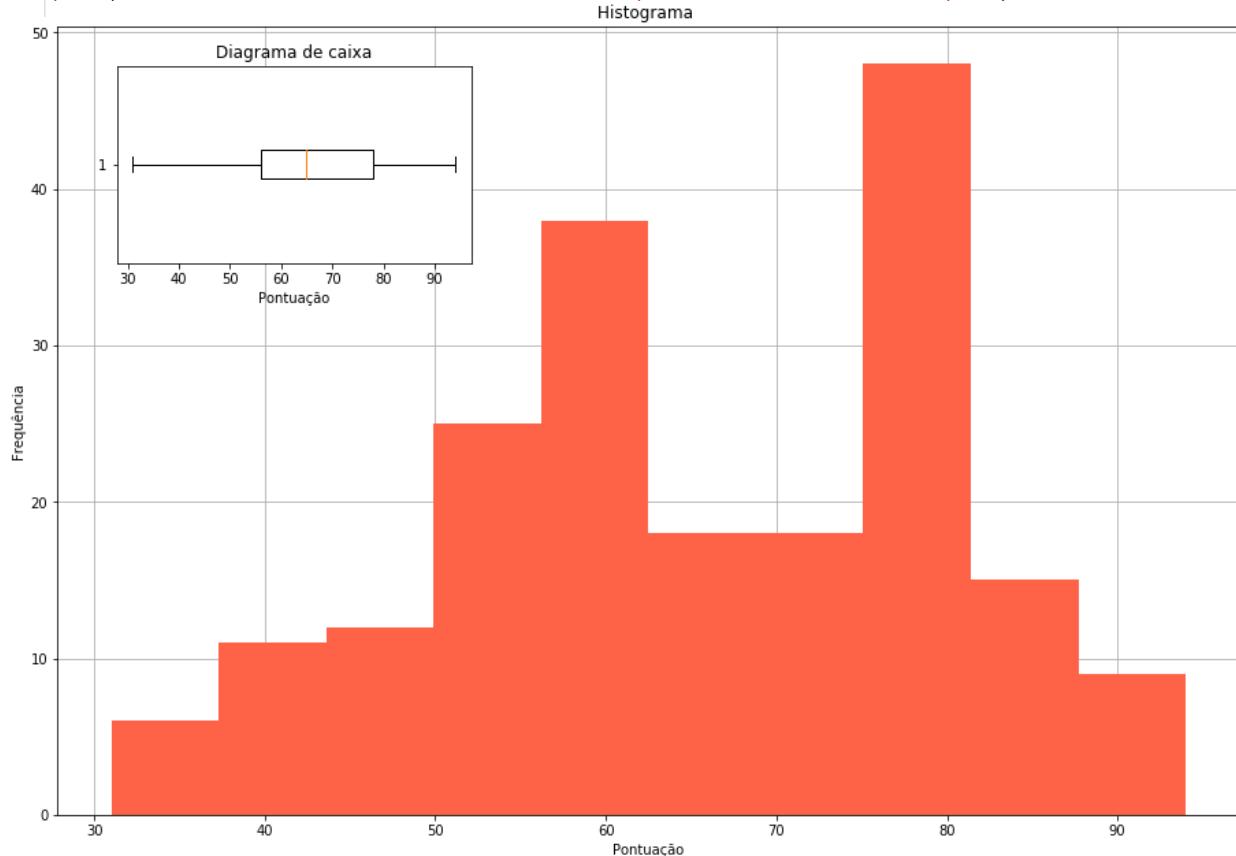
1 # Pontuação Total
2 ## Dados agrupados de uma amostra unidimensional quantitativa discreta (absoluta)
3 ### Calcular a pontuação dos alunos
4 frequencia = []
5 for x in df["Pontuação total"]:
6     num = int(x[0:2])
7     frequencia.append(num)
8
9
10 ### Calcular estatísticas gerais
11 frequencia_a = np.array(frequencia)
12 maxx = frequencia_a.max()
13 minn = frequencia_a.min()
14 gerais_nom = ["Máximo", "Mínimo"]
15 gerais_num = [maxx, minn]
16
17 mask1 = df["Pontuação total"] == "94.00 / 100"
18 df1 = df[mask1]
19 df2 = df1["1.4 - Curso a frequentar"].values[0]
20
21 mask2 = df["Pontuação total"] == "31.00 / 100"
22 df3 = df[mask2]
23 df4 = df3["1.4 - Curso a frequentar"].values[0]
24
25 ### Calcular as estatísticas de localização
26 media = frequencia_a.mean()
27 moda = int(stats.mode(frequencia)[0])
28 mediana, q2 = np.median(frequencia), np.median(frequencia)
29 q1 = np.percentile(frequencia, 25)
30 q3 = np.percentile(frequencia, 75)
31 localizacao_nom = ["Média", "Moda", "Mediana", "Q1", "Q2", "Q3"]
32 localizacao_num = [media, moda, mediana, q1, q2, q3]
33
34 ##### Verificar se existem Outliers
35 amplit_var = maxx - minn
36 amplit_quart = q3 - q1
37 BII = q1 - 1.5 * amplit_quart
38 BIS = q3 + 1.5 * amplit_quart
39 BEI = q1 - 3 * amplit_quart
40 BES = q3 + 3 * amplit_quart
41 barreira_nom = ["Barreira Interna Inferior", "Barreira Interna Superior",
42                 "Barreira Externa Inferior", "Barreira Externa Superior"]
43 barreira_num = [BII, BIS, BEI, BES]
44
45 ### Calcular as estatísticas de dispersão
46 coef_var = stats.variation(frequencia)
47 desvio = frequencia_a.std()
48 variancia = frequencia_a.var()
49 dispersao_nom = ["Amplitude intervalo-variação", "Amplitude inter-quartis",
50                   "Coeficiente de dispersão", "Desvio", "Variância"]
51 dispersao_num = [amplit_var, amplit_quart, coef_var, desvio, variancia]
52
53 ### Calcular as estatísticas de assimetria
54 assimetria = stats.skew(frequencia)
55 assimetria_nom = ["Coeficiente"]
56 assimetria_num = [assimetria]
57
58 ### Calcular as estatísticas de achatamento
59 curtose = stats.kurtosis(frequencia)
60 achatamento_nom = ["Curtose"]
61 achatamento_num = [curtose]
62
63 ## Visualizar a frequência de cada pontuação dos alunos através de um histograma e de um diagrama de caixa
64 fig = plt.figure()
65
66 axes1 = fig.add_axes([0.1, 0.1, 2, 2]) # eixos da figura principal
67 axes2 = fig.add_axes([0.2, 1.5, 0.6, 0.5]) # eixos da figura secundária
68
69 axes1.hist(frequencia, color="tomato", zorder=100)
70 axes1.grid()
71 axes1.set_xlabel('Pontuação')
72 axes1.set_ylabel('Frequência')
73 axes1.set_title('Histograma ')
74
75 axes2.boxplot(frequencia, vert=False)
76 axes2.set_xlabel('Pontuação')
77 axes2.set_title('Diagrama de caixa')
78
79 plt.show()
80
81 print("-----Estatísticas Gerais-----\
82 -----")
83 print("\n")
84 resultado = list(zip(gerais_nom, gerais_num))
85 print(resultado)
86 print(df2, df4)
87 print("\n")
88 print("-----Estatísticas de Localização-----\

```

```

89 -----")
90 print("\n")
91 resultado1 = list(zip(localizacao_nom, localizacao_num))
92 print(resultado1)
93 print("\n")
94 print("-----Estatísticas de Dispersão-----\
95 -----")
96 print("\n")
97 resultado2 = list(zip(dispersao_nom, dispersao_num))
98 print(resultado2)
99 print("\n")
100 print("Nota: Dispersão baixa no intervalo dos quartis em relação ao intervalo da amostra e dispersão \
101 baixa no intervalo da amostra")
102 print("\n")
103 resultado2a = list(zip(barreira_nom, barreira_num))
104 print(resultado2a)
105 print("\n")
106 print("Nota: Não existem outliers moderados, muito menos severos, visto que nenhum dos valores dos \
107 dados assume um valor <23(BII) e >111(BIS)")
108 print("\n")
109 print("-----Estatísticas de Assimetria-----\
110 -----")
111 print("\n")
112 resultado3 = list(zip(assimetria_nom, assimetria_num))
113 print(resultado3)
114 print("\n")
115 print("Nota: Os dados seguem uma distribuição assimétrica negativa ou alongada à esquerda \
116 (Média < Mediana < Moda)")
117 print("\n")
118 print("Nota: Como a distribuição não tem outliers e é uma distribuição com pouca dispersão, é \
119 compreensível a aproximação da Média à Mediana e vice-versa")
120 print("\n")
121 print("-----Estatísticas de Achatamento-----\
122 -----")
123 print("\n")
124 resultado4 = list(zip(achatamento_nom, achatamento_num))
125 print(resultado4)
126 print("\n")
127 print("Nota: Distribuição dos dados é mais achatada do que uma distribuição normal típica")

```



-----Estatísticas Gerais-----

[('Máximo', 94), ('Mínimo', 31)]
 Mestrado integrado em engenharia civil Licienciatura em Engenharia Mecânica

-----Estatísticas de Localização-----

[('Média', 65.735), ('Moda', 78), ('Mediana', 65.0), ('Q1', 56.0), ('Q2', 65.0), ('Q3', 78.0)]

-----Estatísticas de Dispersão-----

[('Amplitude intervalo-variação', 63), ('Amplitude inter-quartis', 22.0), ('Coeficiente de dispersão', 0.22189268775824555), ('Desvio', 14.586115829788271), ('Variância', 212.754775)]

Nota: Dispersão baixa no intervalo dos quartis em relação ao intervalo da amostra e dispersão baixa no intervalo da amostra

[('Barreira Interna Inferior', 23.0), ('Barreira Interna Superior', 111.0), ('Barreira Externa Inferior', -10.0), ('Barreira Externa Superior', 144.0)]

Nota: Não existem outliers moderados, muito menos severos, visto que nenhum dos valores dos dados assume um valor $<23(BI)$ e $>111(BIS)$

-----Estatísticas de Assimetria-----

[('Coeficiente', -0.2393627855811923)]

Nota: Os dados seguem uma distribuição assimétrica negativa ou alongada à esquerda (Média < Mediana < Moda)

Nota: Como a distribuição não tem outliers e é uma distribuição com pouca dispersão, é compreensível a aproximação da Média à Mediana e vice-versa
Note-se que as estatísticas amostrais são variáveis, pelo que a escolha de outra amostra iria fazer com que houvesse alteração nas estatísticas, não representando a população em estudo. Para além disso, esta não segue uma distribuição normal. Logo, é preciso inferir sobre os parâmetros da população (pois estes são fixos) e garantir que a distribuição siga uma distribuição normal

-----Estatísticas de Achatamento-----

Utilizar-se-á a estatística inferencial

Procedimentos:

[('Curtose', -0.7898222065455163)]
 1. Caracterizar a distribuição das estatísticas amostrais associadas à população
 1.1 Utilização do Teorema do Limite Central para garantir que a distribuição segue uma distribuição normal

Nota: Distribuídos dos parâmetros populacionais, a amostra pode seguir uma distribuição normal da população

BÓNUS: Caracterização da População

In [68]:

```

1 # Pontuação Total
2 ## Caracterizar a distribuição das estatísticas amostrais associadas à população
3 ### Utilização do Teorema do Limite Central para garantir que a distribuição segue uma Distribuição normal
4 amostras_quant = 100000
5 amostras_a_complt = 0
6 amostras_dimens = 50
7 medias_amostrais = []
8 while amostras_a_complt < amostras_quant:
9     fre = np.random.choice(frequencia, amostras_dimens).mean()
10    medias_amostrais.append(fre)
11    amostras_a_complt += 1
12
13 ### Calcular parâmetros gerais
14 frequencia_a = np.array(medias_amostrais)
15 maxx = frequencia_a.max()
16 minn = frequencia_a.min()
17 gerais_nom = ["Máximo", "Mínimo"]
18 gerais_num = [maxx, minn]
19
20 ### Calcular as parâmetros de localização
21 media_medias = frequencia_a.mean()
22 moda_medias = int(stats.mode(medias_amostrais)[0])
23 mediana_medias, q2 = np.median(medias_amostrais), np.median(medias_amostrais)
24 q1 = np.percentile(medias_amostrais, 25)
25 q3 = np.percentile(medias_amostrais, 75)
26 localizacao_nom = ["Média das médias amostrais", "Moda das médias amostrais",
27                      "Mediana das médias amostrais", "Q1", "Q2", "Q3"]
28 localizacao_num = [media_medias, moda_medias, mediana_medias, q1, q2, q3]
29
30 ##### Verificar se existem outliers
31 amplit_var = maxx - minn
32 amplit_quart = q3 - q1
33 BII = q1 - 1.5 * amplit_quart
34 BIS = q3 + 1.5 * amplit_quart
35 BEI = q1 - 3 * amplit_quart
36 BES = q3 + 3 * amplit_quart
37 barreira_nom = ["Barreira Interna Inferior", "Barreira Interna Superior",
38                  "Barreira Externa Inferior", "Barreira Externa Superior"]
39 barreira_num = [BII, BIS, BEI, BES]
40
41 ### Calcular os parâmetros de dispersão
42 coef_var = stats.variation(medias_amostrais)
43 desvio_medias = frequencia_a.std()
44 variância_medias = frequencia_a.var()
45 dispersao_nom = ["Amplitude intervalo-variação", "Amplitude inter-quartis", "Coeficiente de dispersão",
46                   "Desvio padrão das médias amostrais", "Variância das médias amostrais"]
47 dispersao_num = [amplit_var, amplit_quart, coef_var, desvio_medias, variância_medias]
48
49 ### Calcular as parâmetros de assimetria
50 assimetria = stats.skew(medias_amostrais, bias=False)
51 assimetria_nom = ["Coeficiente"]
52 assimetria_num = [assimetria]
53
54 ### Calcular as parâmetros de achatamento
55 curtose = stats.kurtosis(medias_amostrais)
56 achatamento_nom = ["Curtose"]
57 achatamento_num = [curtose]
58
59 ## Visualizar a frequência de cada média através de um histograma e de um diagrama de caixa
60 fig = plt.figure()
61
62 axes1 = fig.add_axes([0.1, 0.1, 2, 2]) # eixos da figura principal
63 axes2 = fig.add_axes([0.2, 1.5, 0.6, 0.5]) # eixos da figura secundária
64 axes3 = fig.add_axes([1.5, 1.6, 0.4, 0.4]) # eixos da figura terciária
65
66 axes1.hist(medias_amostrais, color="magenta", zorder=100)
67 axes1.grid()
68 axes1.set_xlabel('Pontuação')
69 axes1.set_ylabel('Frequência')
70 axes1.set_title('Histograma')
71
72 axes2.boxplot(medias_amostrais, vert=False)
73 axes2.set_xlabel('Pontuação')
74 axes2.set_title('Diagrama de caixa')
75
76 axes3.hist(medias_amostrais, cumulative=True, color="mediumvioletred")
77 axes3.set_xlabel('Pontuação')
78 axes3.set_title('Diagrama de caixa')
79 axes3.set_ylabel('Frequência')
80 axes3.set_title('Polígono de Frequências Acumuladas')
81
82 plt.show()
83
84 print("-----Parâmetros Gerais-----\n-----")
85 resultado = list(zip(gerais_nom, gerais_num))
86 print(resultado)
87

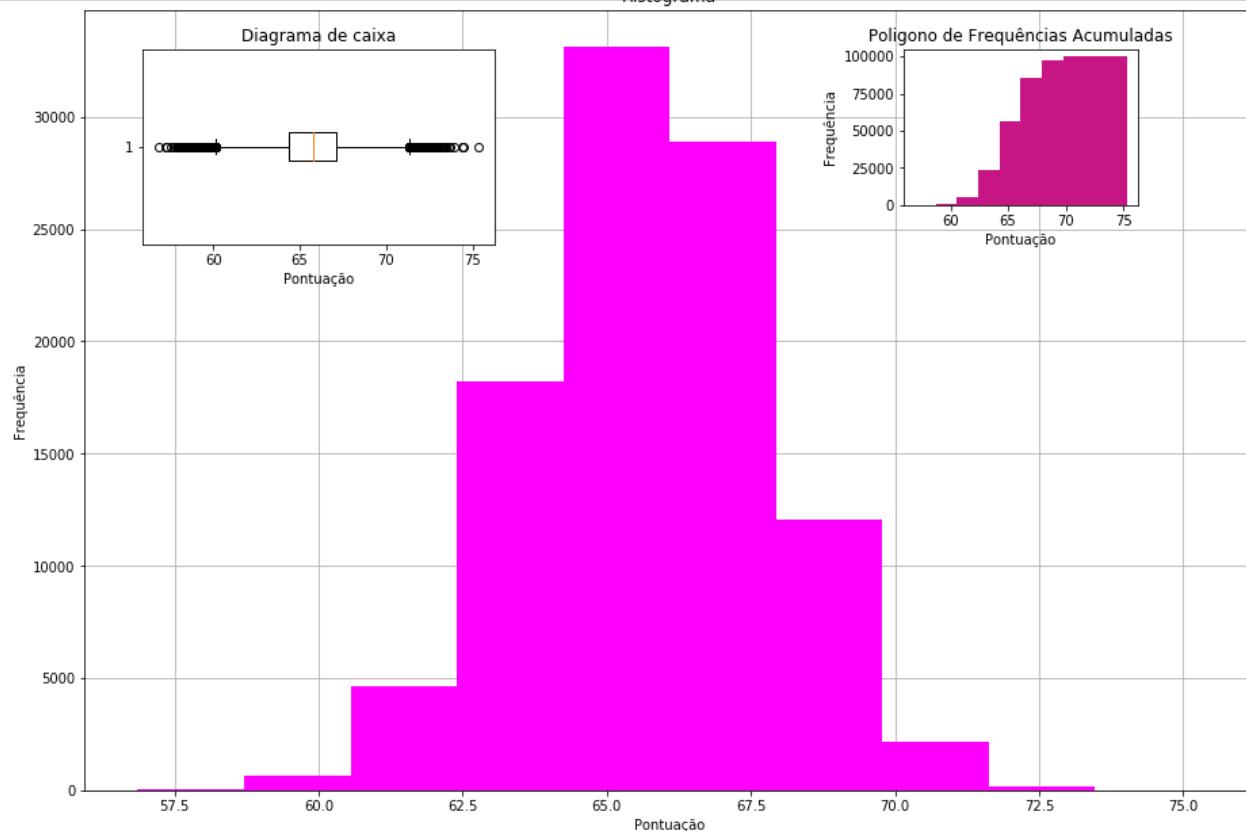
```

```

89 print("\n")
90 print("-----Parâmetros de Localização-----\n")
91 print("\n")
92 resultado1 = list(zip(localizacao_nom, localizacao_num))
93 print(resultado1)
94 print("\n")
95 print("Nota: Como a Média das médias amostrais é igual ao Valor médio da população (média das médias \
96 amostrais = valor medio populacional) e como os dados seguem uma distribuição simétrica \
97 (média = mediana = moda), logo a Mediana das médias amostrais e a Moda das médias amostrais \
98 é igual à Mediana populacional e Moda populacional, respectivamente")
99 print("\n")
100 print("-----Parâmetros de Dispersão-----\n")
101 print("\n")
102 print("\n")
103 print("\n")
104 resultado2 = list(zip(dispersao_nom, dispersao_num))
105 print(resultado2)
106 print("\n")
107 print("Nota: Dispersão baixa no intervalo dos quartis em relação ao intervalo da distribuição das \
108 médias amostrais e dispersão baixa no intervalo da distribuição das médias amostrais")
109 print("\n")
110 print("Nota A Variância das médias amostrais é inferior à Variância populacional \
111 (variância das médias amostrais < variância populacional). \
112 O mesmo se aplica ao Desvio padrão (desvio padrão das médias amostrais < desvio padrão populacional)")
113 print("\n")
114 resultado2a = list(zip(barreira_nom, barreira_num))
115 print(resultado2a)
116 print("\n")
117 print("Nota: Existem outliers moderados, visto que existe valores em que se encontram entre as \
118 Barreiras Externas e Internas (BEI < 60.17 < BII e BES > 71.29 > BIS). No entanto, não nos interessa")
119 print("\n")
120 print("-----Parâmetros de Assimetria-----\n")
121 print("\n")
122 print("\n")
123 resultado3 = list(zip(assimetria_nom, assimetria_num))
124 print(resultado3)
125 print("\n")
126 print("Nota: Os dados seguem uma distribuição simétrica (Média = Mediana = Moda)")
127 print("\n")
128 print("-----Parâmetros de Achatamento-----\n")
129 print("\n")
130 print("\n")
131 resultado4 = list(zip(achatamento_nom, achatamento_num))
132 print(resultado4)
133 print("\n")
134 print("Nota: Distribuição dos dados segue uma distribuição normal típica")
135
136 fig = plt.figure(figsize = (12, 8), dpi = 100)
137
138 sns.distplot(medias_amostrais, color="magenta")

```

Histograma



-----Parâmetros Gerais-----

```
[('Máximo', 75.32), ('Mínimo', 56.86)]
```

-----Parâmetros de Localização-----

```
[('Média das médias amostrais', 65.73201599999999), ('Moda das médias amostrais', 65), ('Mediana das médias amostrais', 65.74), ('Q1', 64.34), ('Q2', 65.74), ('Q3', 67.14)]
```

Nota: Como a Média das médias amostrais é igual ao Valor médio da população (média das médias amostrais = valor medio populacional) e como os dados seguem uma distribuição simétrica (média = mediana = moda), logo a Mediana das médias amostrais e a Moda das médias amostrais é igual à Mediana populacional e Moda populacional, respectivamente

-----Parâmetros de Dispersão-----

```
[('Amplitude intervalo-variação', 18.45999999999994), ('Amplitude inter-quartis', 2.79999999999997), ('Coeficiente de dispersão', 0.03135007310586076), ('Desvio padrão das médias amostrais', 2.060703506995609), ('Variância das médias amostrais', 4.246498943744)]
```

Nota: Dispersão baixa no intervalo dos quartis em relação ao intervalo da distribuição das médias amostrais e dispersão baixa no intervalo da distribuição das médias amostrais

Nota A Variância das médias amostrais é inferior à Variância populacional (variância das médias amostrais < variância populacional). O mesmo se aplica ao Desvio padrão (desvio padrão das médias amostrais < desvio padrão populacional)

```
[('Barreira Interna Inferior', 60.14000000000001), ('Barreira Interna Superior', 71.34), ('Barreira Externa Inferior', 5.94000000000001), ('Barreira Externa Superior', 75.53999999999999)]
```

Nota: Existem outliers moderados, visto que existe valores em que se encontram entre as Barreiras Externas e Internas (BEI < 60.17 < BII e BES > 71.29 > BIS). No entanto, não nos interessa

-----Parâmetros de Assimetria-----

```
[('Coeficiente', -0.032338840296450184)]
```

Nota: Os dados seguem uma distribuição simétrica (Média = Mediana = Moda)

-----Parâmetros de Achatamento-----

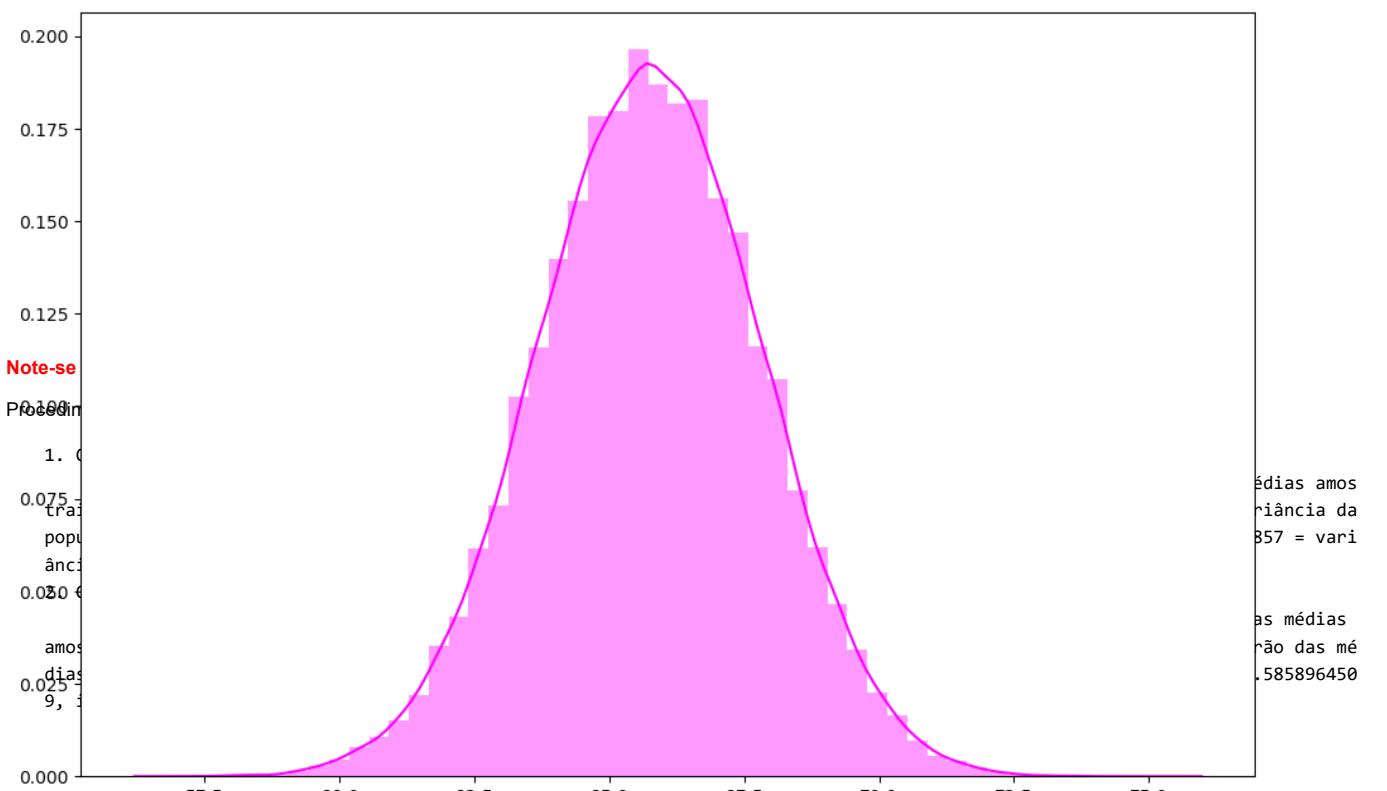
```
[('Curtose', -0.022921232379195633)]
```

Nota: Distribuição dos dados segue uma distribuição normal típica

```
C:\Users\simao\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[68]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ed84299860>
```



Para se confirmar, de uma forma geral, que os parâmetros da população se encontram certos, a estatística dispõe de várias formas de avaliação, sendo que a utilização de uma delas é suficiente: Estimação pontual, Intervalo de confiança e o Teste de hipóteses

Utilizar-se-á a estimação por intervalo para este caso

Procedimentos:

1. Calcular o intervalo de confiança para o valor médio populacional
2. Calcular o intervalo de confiança para a variância populacional
3. Calcular o intervalo de confiança para o desvio padrão populacional

In [69]:

```

1 # Calcular o intervalo de confiança para o valor médio populacional
2 ## Nível de confiança - 95%; Nível de significância - 5%; Dimensão da mostra - Grande; desvio populacional - Conhecido
3 ### intervalo = [média da amostra +- z * desvio padrão populacional/raiz quadrada da dimensão da amostra]
4 amostras_a_complt = 100
5 amostras_quant = 0
6 amostras_dimens = 50
7 z = 1.96 # Ir à tabela e encontrar o número em que a probabilidade é 0.0025 (nível de significância/2 = 0.05/2),
8         #visto que a distribuição é simétrica
9 desvio_pop = 14.5858964509
10 intervalo_amostrais_media_final = []
11 while amostras_quant < amostras_a_complt:
12     intervalo_amostrais_media = []
13     x1 = np.random.choice(frequencia, amostras_dimens).mean()
14     int1 = x1 - z * desvio_pop/np.sqrt(amostras_dimens)
15     intervalo_amostrais_media.append(int1)
16     x2 = np.random.choice(frequencia, amostras_dimens).mean()
17     int2 = x2 + z * desvio_pop/np.sqrt(amostras_dimens)
18     intervalo_amostrais_media.append(int2)
19     intervalo_amostrais_media_final.append(intervalo_amostrais_media)
20     amostras_quant += 1
21
22 # Calcular o intervalo de confiança para a variância populacional
23 ## Nível de confiança - 95%; Nível de significância - 5%; Dimensão da mostra - Grande
24 amostras_a_complt = 100
25 amostras_quant = 0
26 amostras_dimens = 50
27 xn2_a = 71.42 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1 = 50-1)
28         #e com uma probabilidade a 0.025 (nível de significância/2 = 0.05/2)
29 xn2_b = 32.357 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1 = 50-1)
30         #e com uma probabilidade a 0.975 (1 - (nível de significância/2) = 1 - (0.05/2))
31 intervalo_amostrais_var_final = []
32 while amostras_quant < amostras_a_complt:
33     intervalo_amostrais_var = []
34     var1 = np.random.choice(frequencia, amostras_dimens).var()
35     int1 = ((amostras_dimens - 1) * var1) / (xn2_a)
36     intervalo_amostrais_var.append(int1)
37     var2 = np.random.choice(frequencia, amostras_dimens).var()
38     int2 = ((amostras_dimens - 1) * var1) / (xn2_b)
39     intervalo_amostrais_var.append(int2)
40     intervalo_amostrais_var_final.append(intervalo_amostrais_var)
41     amostras_quant += 1
42
43 # Calcular o intervalo de confiança para o desvio padrão populacional
44 ## Nível de confiança - 95%; Nível de significância - 5%; Dimensão da mostra - Grande
45 amostras_a_complt = 100
46 amostras_quant = 0
47 amostras_dimens = 50
48 xn2_a = 71.42 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1 = 50-1)
49         #e com uma probabilidade a 0.025 (nível de significância/2 = 0.05/2)
50 xn2_b = 32.357 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1 = 50-1)
51         #e com uma probabilidade a 0.975 (1 - (nível de significância/2) = 1 - (0.05/2))
52 intervalo_amostrais_desv_final = []
53 while amostras_quant < amostras_a_complt:
54     intervalo_amostrais_desv = []
55     var1 = np.random.choice(frequencia, amostras_dimens).var()
56     int1 = ((amostras_dimens - 1) * var1) / (xn2_a)
57     intervalo_amostrais_desv.append(np.sqrt(int1))
58     var2 = np.random.choice(frequencia, amostras_dimens).var()
59     int2 = ((amostras_dimens - 1) * var1) / (xn2_b)
60     intervalo_amostrais_desv.append(np.sqrt(int2))
61     intervalo_amostrais_desv_final.append(intervalo_amostrais_desv)
62     amostras_quant += 1
63
64 print("-----Intervalos de confiança para o valor médio populacional---\
65 -----")
66 print("\n")
67 for x in intervalo_amostrais_media_final:
68     print(x)
69 print("\n")
70 print("Intervalos que não cumprem os requisitos")
71 print("\n")
72 for x in intervalo_amostrais_media_final:
73     if (x[0] > 65.73343383999999) or (x[1] < 65.73343383999999):
74         print(x)
75 print("\n")
76 print("-----Intervalos de confiança para a variância populacional-----\
77 -----")
78 print("\n")
79 for y in intervalo_amostrais_var_final:
80     print(y)
81 print("\n")
82 print("Intervalos que não cumprem os requisitos")
83 print("\n")
84 for x in intervalo_amostrais_var_final:
85     if (x[0] > 212.748375277) or (x[1] < 212.748375277):
86         print(x)
87 print("\n")
88 print("-----Intervalos de confiança para o desvio padrão populacional-----\

```

```
89 | -----")
90 | print("\n")
91 | for w in intervalo_amostrais_desv_final:
92 |     print(w)
93 | print("\n")
94 | print("Intervalos que não cumprem os requisitos")
95 | print("\n")
96 | for x in intervalo_amostrais_desv_final:
97 |     if (x[0] > 14.5858964509) or (x[1] < 14.5858964509):
98 |         print(x)
99 | print("\n")
100 | print("Nota: Se formos a tirar amostras aleatórias repetidamente das 100 geradas, o valor real de cada \
101 | parâmetro estará dentro das 100 amostras 95% das vezes. Se sair outro valor, é normal pois existe 5% de erro")
```

-----Intervalos de confiança para o valor médio populacional-----

[59.99699577427446, 68.26300422572554]
[63.11699577427445, 70.46300422572554]
[57.99699577427445, 70.56300422572554]
[62.87699577427445, 71.58300422572555]
[61.75699577427445, 67.58300422572555]
[62.17699577427445, 72.00300422572553]
[63.09699577427445, 70.40300422572554]
[60.57699577427446, 67.42300422572555]
[60.33699577427445, 72.64300422572553]
[61.05699577427445, 67.46300422572554]
[63.19699577427445, 66.94300422572555]
[60.61699577427445, 70.80300422572554]
[61.23699577427445, 71.76300422572554]
[62.99699577427446, 70.18300422572554]
[61.05699577427445, 69.80300422572554]
[59.83699577427445, 68.82300422572554]
[59.67699577427445, 70.08300422572555]
[65.27699577427445, 70.82300422572554]
[65.29699577427445, 68.16300422572554]
[58.71699577427445, 71.90300422572554]
[59.33699577427445, 72.20300422572554]
[62.01699577427445, 72.28300422572553]
[63.83699577427445, 69.72300422572555]
[63.09699577427445, 71.20300422572554]
[64.15699577427446, 70.72300422572555]
[64.75699577427446, 70.66300422572554]
[62.67699577427445, 71.76300422572554]
[62.31699577427445, 70.44300422572555]
[59.59699577427445, 62.78300422572555]
[61.63699577427445, 68.94300422572555]
[59.91699577427445, 67.12300422572554]
[61.75699577427445, 67.78300422572555]
[59.83699577427445, 70.70300422572554]
[58.57699577427445, 67.04300422572554]
[62.99699577427446, 71.84300422572554]
[61.31699577427445, 72.08300422572555]
[61.31699577427445, 72.40300422572554]
[60.21699577427446, 68.28300422572553]
[61.91699577427445, 62.80300422572554]
[60.15699577427445, 71.08300422572555]
[61.27699577427445, 70.08300422572555]
[62.61699577427445, 68.28300422572553]
[61.67699577427445, 66.64300422572555]
[58.85699577427445, 69.40300422572554]
[62.33699577427445, 75.90300422572554]
[61.61699577427445, 71.38300422572554]
[60.95699577427445, 69.88300422572554]
[61.69699577427445, 72.08300422572555]
[63.21699577427446, 69.64300422572553]
[59.47699577427446, 71.38300422572554]
[62.35699577427445, 71.94300422572555]
[62.99699577427446, 66.46300422572554]
[62.03699577427445, 69.78300422572553]
[58.79699577427446, 68.52300422572554]
[62.53699577427445, 72.56300422572554]
[57.77699577427445, 69.72300422572555]
[62.81699577427445, 68.20300422572554]
[59.83699577427445, 68.16300422572554]
[62.39699577427445, 67.46300422572554]
[62.49699577427446, 71.62300422572554]
[62.73699577427445, 67.78300422572555]
[60.01699577427445, 68.94300422572555]
[62.43699577427446, 66.24300422572554]
[62.55699577427445, 70.38300422572554]
[62.33699577427445, 73.78300422572553]
[62.33699577427445, 63.98300422572554]
[63.45699577427445, 68.24300422572554]
[63.47699577427445, 69.92300422572553]
[63.03699577427445, 66.54300422572554]
[63.11699577427445, 70.64300422572553]
[66.07699577427447, 69.82300422572554]
[64.99699577427447, 68.80300422572554]
[62.59699577427445, 66.98300422572554]
[59.73699577427445, 70.86300422572553]
[64.17699577427446, 69.08300422572555]
[60.91699577427445, 67.70300422572554]
[64.65699577427446, 70.64300422572553]
[62.87699577427445, 68.58300422572555]
[61.29699577427446, 67.08300422572555]
[61.07699577427446, 67.62300422572554]
[65.17699577427446, 70.70300422572554]
[60.89699577427445, 66.86300422572555]
[63.41699577427445, 69.04300422572554]
[61.49699577427446, 66.40300422572554]
[62.73699577427445, 68.18300422572554]

```
[62.49699577427446, 73.20300422572554]
[62.096995774274454, 69.82300422572554]
[61.57699577427446, 73.98300422572554]
[60.91699577427445, 68.14300422572553]
[60.236995774274455, 66.68300422572554]
[61.13699577427446, 69.22300422572555]
[57.916995774274454, 69.76300422572554]
[62.27699577427445, 70.22300422572555]
[62.29699577427446, 68.34300422572554]
[62.85699577427446, 68.82300422572554]
[59.81699577427445, 71.64300422572553]
[62.55699577427445, 69.52300422572554]
[63.93699577427446, 71.28300422572553]
[60.57699577427446, 68.88300422572554]
[61.97699577427445, 73.80300422572554]
```

Intervalos que não cumprem os requisitos

```
[59.596995774274454, 62.78300422572555]
[61.91699577427445, 62.80300422572554]
[62.33699577427445, 63.98300422572554]
[66.07699577427447, 69.82300422572554]
```

-----Intervalos de confiança para a variância populacional-----

```
[134.48750490058805, 296.8475940291127]
[133.86838420610474, 295.4810396513892]
[158.95045645477458, 350.8434527304757]
[156.72206104732572, 345.924826158173]
[134.43481377765332, 296.73129152888094]
[129.758751050126, 286.4100503754983]
[166.1090394847382, 366.6442377229039]
[135.02896107532982, 298.0427233674321]
[175.4735144217306, 387.31397842816085]
[153.38742649117896, 338.56445282319135]
[149.57280873704843, 330.1446364001607]
[158.84617194063287, 350.6132706987669]
[161.05618034164098, 355.4913125444262]
[156.69818538224584, 345.8721265877553]
[151.2347745729488, 333.81301109497184]
[158.15130775693086, 349.07953147696026]
[129.86962195463454, 286.65477021973606]
[116.28299635956311, 256.6656859412182]
[147.59689162699522, 325.78329264146856]
[161.28999719966396, 356.0074048892048]
[184.39148697843743, 406.9981765923912]
[145.3385830299636, 320.7986401705968]
[141.1268608232988, 311.5023148005069]
[152.9494315317838, 337.59768829001456]
[149.06346121534582, 329.02037889792007]
[161.76064967796137, 357.04625274283774]
[157.13727807336883, 346.841314089687]
[158.75835340240832, 350.41943319838066]
[131.11746849621952, 289.4090799517878]
[120.03175021002518, 264.9401242389591]
[137.76094091290955, 304.0728868560126]
[124.82966115933914, 275.53031492412777]
[123.34141136936431, 272.24537503476836]
[136.0366788014562, 300.26700868436507]
[156.72343321198545, 345.92785486911646]
[131.15040044805377, 289.48176901443276]
[141.0157154858583, 311.2569892140804]
[149.87358723046765, 330.8085298389838]
[131.11005880705687, 289.3927249126928]
[139.39244469336322, 307.6740241678771]
[133.6312741528983, 294.9576784003461]
[113.05237188462615, 249.53488889575672]
[153.50927471296552, 338.83340235497724]
[116.57252310277232, 257.30474395030444]
[143.63819658359006, 317.0454615693668]
[116.887846541585, 258.00074172512905]
[159.1461271352562, 351.2753469110238]
[137.00268272192662, 302.3992211886145]
[118.88873704844579, 262.41720802299346]
[157.1208120974517, 346.8049695583645]
[157.97045645477456, 348.6803473746021]
[153.8879921590591, 339.6693265753933]
[92.18888266591989, 203.48394474147787]
[147.30489498739848, 325.13878295268415]
[161.25569308316997, 355.93168711561646]
[161.96373004760568, 357.494501962481]
[123.55656678801455, 272.72027691071486]
[139.5000224026883, 307.91147510585034]
[171.3353402408289, 378.1799919646445]
[192.62447493699244, 425.1704422536082]
```

```
[108.68669280313635, 239.898742157802]
[138.93688602632318, 306.6684921346232]
[139.79503780453652, 308.56264795871056]
[119.09181741809016, 262.8654572426368]
[134.37032203864467, 296.5889421145348]
[172.76239148697846, 381.3298513459221]
[151.1258246989639, 333.572531446055]
[122.09521142537105, 269.49469975584884]
[120.04492299075888, 264.969199864017]
[131.3666535984318, 289.95909385913404]
[203.61386726407167, 449.4267824582007]
[189.60269392327078, 418.50061501375285]
[117.28275553066368, 258.87240473467875]
[131.78269392327078, 290.8773990172142]
[119.83415849901988, 264.5039898630899]
[134.5613273592831, 297.01053867787493]
[129.42915709885185, 285.6825540068609]
[131.33372164659758, 289.88640479648916]
[133.14278353402412, 293.8794573044473]
[175.3352002240269, 387.00868436505243]
[108.38344441332957, 239.22939703928049]
[94.34345561467376, 208.2396266650184]
[177.2894371324559, 391.3221744908367]
[114.55269672360683, 252.84648144141914]
[125.60822738728646, 277.24880551349014]
[186.9736264351722, 412.6976048459375]
[123.20831139736768, 271.95159007324537]
[94.98370764491737, 209.65282319127235]
[127.81027723326798, 282.10928083567694]
[127.8717502100252, 282.2449670859474]
[155.6103332399888, 343.4709645517199]
[153.7672416690003, 339.40280001236215]
[148.55493699243908, 327.89793862224553]
[134.812707924951, 297.56539852273085]
[168.2847437692523, 371.4465617949748]
[139.02799775973114, 306.8695985412739]
[145.75133015961916, 321.7096764224125]
[105.95196863623633, 233.8625212473344]
[153.63194623354804, 339.1041691133295]
[157.79454494539343, 348.29206663164075]
```

Intervalos que não cumprem os requisitos

```
[92.18888266591989, 203.48394474147787]
[94.34345561467376, 208.2396266650184]
[94.98370764491737, 209.65282319127235]
```

-----Intervalos de confiança para o desvio padrão populacional-----

```
[12.226753355642469, 18.165064737226356]
[12.75055052868911, 18.94326065570192]
[11.394433833003067, 16.928502784021923]
[11.205977122535657, 16.64851608221891]
[10.862116656959282, 16.137648852294376]
[11.673765785594382, 17.343501177659714]
[11.655744350540802, 17.316727059880122]
[11.821324750054941, 17.562726843216073]
[12.390495087938291, 18.40833284616932]
[12.7204578378969, 18.898552493162143]
[13.215944648157134, 19.634688221349162]
[11.756376273578958, 17.466234074820935]
[12.365885630042431, 18.37177102286076]
[12.872296528675474, 19.12413647014943]
[11.170705177315307, 16.59611319572085]
[11.663570390961414, 17.328354065573414]
[11.95758957624056, 17.76517301327203]
[10.829526260953635, 16.089229894617336]
[12.937932308181603, 19.2216503521365]
[10.671768738019006, 15.854852416513587]
[11.513191392013054, 17.104938726148443]
[11.052662564376579, 16.420739435948896]
[11.952137584178812, 17.757073087979695]
[11.337693221926202, 16.844204291735302]
[14.00906315433175, 20.81300993847986]
[10.475710728243284, 15.563572602793931]
[11.34210983964949, 16.850765980233426]
[11.722795625296582, 17.416343917374856]
[12.589693659750962, 18.704278535698613]
[12.163383219604718, 18.07991688046916]
[11.043459374643216, 16.40706642460788]
[12.87890392618723, 19.133952960271905]
[11.590341579183534, 17.219559353855324]
[12.650364508845774, 18.79441611101505]
[11.156643889823952, 16.575222597036177]
[12.30465962787378, 18.28080866673522]
```

```
[10.84674457264201, 16.114810826642984]
[12.45874648765618, 18.509732707458713]
[9.818898048715392, 14.587757969357389]
[12.341360520685724, 18.33533451464032]
[10.399131859593473, 15.449800772606041]
[10.973020298028036, 16.30241637160303]
[13.249924892661065, 19.685172051622537]
[11.979006185435537, 17.79699127942681]
[13.387515134160488, 19.889587367066373]
[11.605970180267526, 17.2427784817943]
[10.875688192994527, 16.157811836162043]
[12.840362712546353, 19.07669298123465]
[13.084521778239349, 19.439435657518967]
[12.582421852757985, 18.693474944512808]
[13.20432135674905, 19.617419709034724]
[11.953147825119185, 17.7585739845422]
[12.749259069954611, 18.94134195898536]
[12.259121498997848, 18.213153498189456]
[13.008279895570398, 19.326164481264282]
[11.650999094223089, 17.30967711900876]
[10.546049492544016, 15.668073623620499]
[9.697970698877223, 14.408098408522472]
[11.8417363932194, 17.593052049645824]
[10.505728858135493, 15.60816999156278]
[12.537706556235987, 18.62704224300652]
[11.318712435173127, 16.816004882611377]
[12.633953443540689, 18.770034490236206]
[12.379194114869998, 18.39154319634094]
[12.029985871446733, 17.87273087028471]
[14.621225503637763, 21.722488389658775]
[12.223666739707348, 18.16047900815752]
[10.895465980924037, 16.18719533541558]
[12.00550606434828, 17.83636166680413]
[12.148844460713441, 18.04931691122327]
[11.829064443643606, 17.57422556499404]
[11.166048722681147, 16.58919518589313]
[11.42604947270868, 16.97547356402022]
[10.556648272000382, 15.683820037190838]
[13.638823340019183, 20.26295139066571]
[11.997228280664254, 17.82406351437336]
[12.4877557505439, 18.552831240896577]
[10.968955454283282, 16.296377307296883]
[12.22052320248528, 18.155808712169552]
[10.548222134503229, 15.671301478202329]
[11.806120706579982, 17.540138472783063]
[10.48079172770019, 15.571121351131275]
[11.017136419024004, 16.36795888894491]
[11.09231874515952, 16.479655901358814]
[12.472175921653227, 18.52968457291677]
[13.92926262667478, 20.694451748190957]
[10.582820017818669, 15.72270292320702]
[9.885471577309639, 14.6866658048082585]
[13.969544021879221, 20.754297083277955]
[12.114743350181751, 17.99865351249415]
[12.488360080350805, 18.553729082682185]
[11.837042513611932, 17.586078438216994]
[11.911001059253921, 17.69595730224412]
[10.057632858553054, 14.942441927525994]
[11.2741941838226, 16.74986492752696]
[11.706021925227958, 17.391423536733598]
[12.128101141583379, 18.018498939853245]
[12.685989599548567, 18.84734365932346]
[11.140249468435897, 16.550865703821678]
[12.131494845239049, 18.02354090355457]
```

Intervalos que não cumprem os requisitos

```
[9.697970698877223, 14.408098408522472]
[14.621225503637763, 21.722488389658775]
```

Nota: Se formos a tirar amostras aleatórias repetidamente das 100 geradas, o valor real de cada parâmetro estará dentro das 100 amostras 95% das vezes. Se sair outro valor, é normal pois existe 5% de erro

3. Preparação, Modelação e Avaliação dos Dados

Os algoritmos selecionados foram escolhidos tendo sempre em conta os critérios de melhores algoritmos disponíveis no mercado em termos de desempenho geral, configuração mínima possível de hiperparâmetros, préprocessamento mínimo dos dados, menor custo de processamento, melhor interpretabilidade e, para o caso preditivo, ter um desempenho de, pelo menos, 70%.

Problema Descritivo

Seleção e Transformação dos Atributos

De modo a aproveitar ao máximo a abordagem de clustering que se irá utilizar de seguida, será feito, em primeiro lugar, uma seleção dos devidos atributos e sua posterior transformação.

In [31]:

```

1 # Selecionando os atributos essenciais relativos à identificação dos grupos sociais ("1.3 - Habilidades Académicas")
2 # e ao conhecimento e comportamento dos estudantes perante as notícias da actualidade ("Pontuação total")
3 df = pd.read_csv("Quiz.csv")[["Pontuação total", "1.3 - Habilidades Académicas"]]
4
5 ## Eliminando Linhas de teste que foram feitas para testar o questionário
6 for y in range(0, 4):
7     df = df.drop(y)
8 df = df.drop(7)
9
10 ## Renomeando Linhas
11 df.index = list(range(1,201))
12 df.head()

```

Out[31]:

	Pontuação total	1.3 - Habilidades Académicas
1	57.00 / 100	Ensino Secundário
2	75.00 / 100	Mestrado
3	80.00 / 100	Licenciatura
4	77.00 / 100	Ensino Secundário
5	38.00 / 100	Mestrado

In [32]:

```

1 # Transformando os atributos
2 ## Convertendo para a escala correta
3 ### Convertendo a escala do tipo qualitativa ordinal do atributo "1.3 - Habilidades Académicas"
4 ### para uma escala do tipo quantitativa relativa ou absoluta
5 habilidades_academicas_ordinal = ["Ensino Secundário", "Licenciatura", "Mestrado", "Doutoramento"]
6 habilidades_academicas_relativa_ou_absoluta = [0.0, 1.0, 2.0, 3.0]
7 df["1.3 - Habilidades Académicas"] = df["1.3 - Habilidades Académicas"].replace(habilidades_academicas_ordinal,
8                                         habilidades_academicas_relativa_ou_absoluta)
9 #### Ajustando os dados do atributo "Pontuação total"
10 def ajustar(valor):
11     return float(valor[0:2])
12
13 pontuacao_total_ajustada = list(map(ajustar, list(df["Pontuação total"])))
14 df["Pontuação total"] = pontuacao_total_ajustada
15
16 ## Normalizando os dados
17 ### Utilizando o método standardization
18 scaler = sklearn.preprocessing.StandardScaler()
19 df.iloc[:,0:] = scaler.fit_transform(df.iloc[:,0:].to_numpy())
20 df.head()

```

Out[32]:

	Pontuação total	1.3 - Habilidades Académicas
1	-0.598857	-0.939128
2	0.635193	1.323831
3	0.977985	0.192352
4	0.772310	-0.939128
5	-1.901466	1.323831

Abordagem Clustering

Na procura do melhor resultado possível, irão ser utilizados os algoritmos de Clustering K-means e DBSCAN.

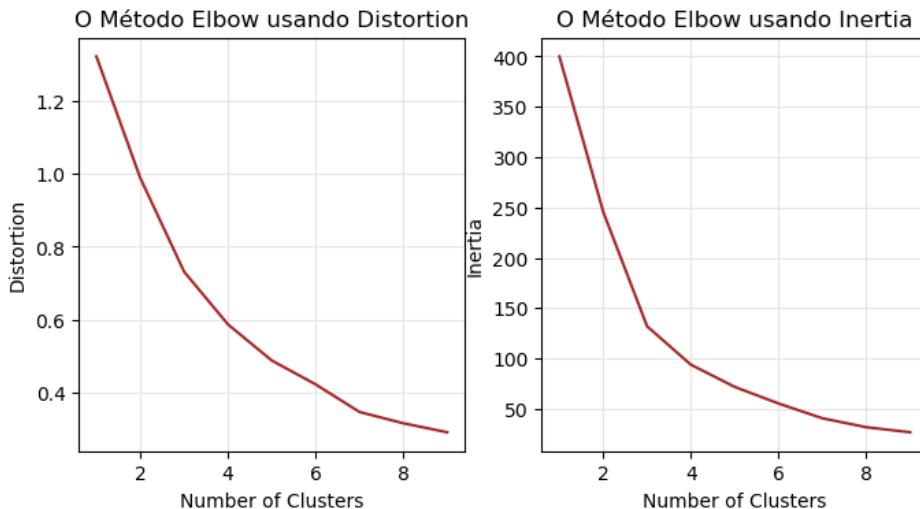
K-means

In [33]:

```

1 # Encontrando o número ideal para o hiperparâmetro de clusters com o método Elbow
2 distortions = []
3 inertias = []
4 clusters = range(1, 10)
5 for cluster in clusters:
6     modelo = sklearn.cluster.KMeans(n_clusters = cluster).fit(df)
7
8     # Utilizando Distortion
9     distortions.append(sum(np.min(cdist(df, modelo.cluster_centers_,
10                             'euclidean'), axis=1)) / df.shape[0])
11    # Utilizando Inertia
12    inertias.append(modelo.inertia_)
13
14 fig = plt.figure(figsize = (8, 4), dpi = 100)
15
16 ax1 = fig.add_subplot(1,2,1)
17 ax1.plot(clusters, distortions, color = "brown", zorder = 100)
18 ax1.set_title('O Método Elbow usando Distortion')
19 ax1.set_xlabel('Number of Clusters')
20 ax1.set_ylabel('Distortion')
21 ax1.grid(color = '0.90')
22
23
24 ax2 = fig.add_subplot(1,2,2)
25 ax2.plot(clusters, inertias, color = "brown", zorder = 100)
26 ax2.set_title('O Método Elbow usando Inertia')
27 ax2.set_xlabel('Number of Clusters')
28 ax2.set_ylabel('Inertia')
29 ax2.grid(color = '0.90')
30
31 plt.show()

```



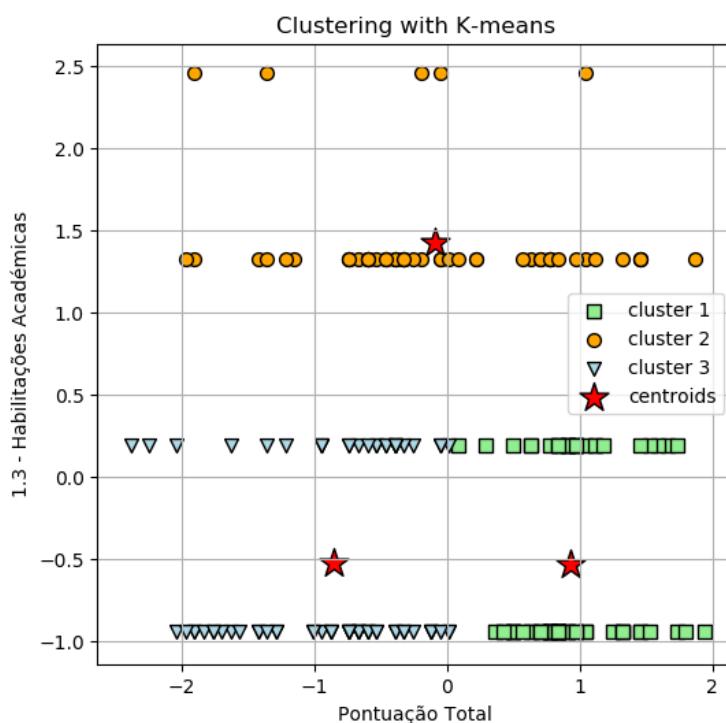
De acordo com as figuras, será utilizado o valor 3 como hiperparâmetro dos clusters

In [34]:

```

1 # Visualizando os resultados
2 km = sklearn.cluster.KMeans(
3     n_clusters = 3, init='random',
4     n_init=10000, max_iter=10000,
5     tol=1e-04, random_state=0
6 )
7 y_km = km.fit_predict(df)
8
9 plt.figure(figsize = (6, 6), dpi = 100)
10 plt.scatter(
11     np.array(df)[y_km == 0, 0], np.array(df)[y_km == 0, 1],
12     s=50, c='lightgreen',
13     marker='s', edgecolor='black',
14     label='cluster 1'
15 )
16
17 plt.scatter(
18     np.array(df)[y_km == 1, 0], np.array(df)[y_km == 1, 1],
19     s=50, c='orange',
20     marker='o', edgecolor='black',
21     label='cluster 2'
22 )
23
24 plt.scatter(
25     np.array(df)[y_km == 2, 0], np.array(df)[y_km == 2, 1],
26     s=50, c='lightblue',
27     marker='v', edgecolor='black',
28     label='cluster 3'
29 )
30
31 plt.scatter(
32     km.cluster_centers_[:, 0], km.cluster_centers_[:, 1],
33     s=250, marker='*',
34     c='red', edgecolor='black',
35     label='centroids'
36 )
37 plt.legend(scatterpoints=1)
38 plt.title("Clustering with K-means")
39 plt.xlabel("Pontuação Total")
40 plt.ylabel("1.3 - Habilidades Académicas")
41 plt.grid()
42 plt.show()

```



Como se pode ver, parece não existir clusters no conjunto de dados

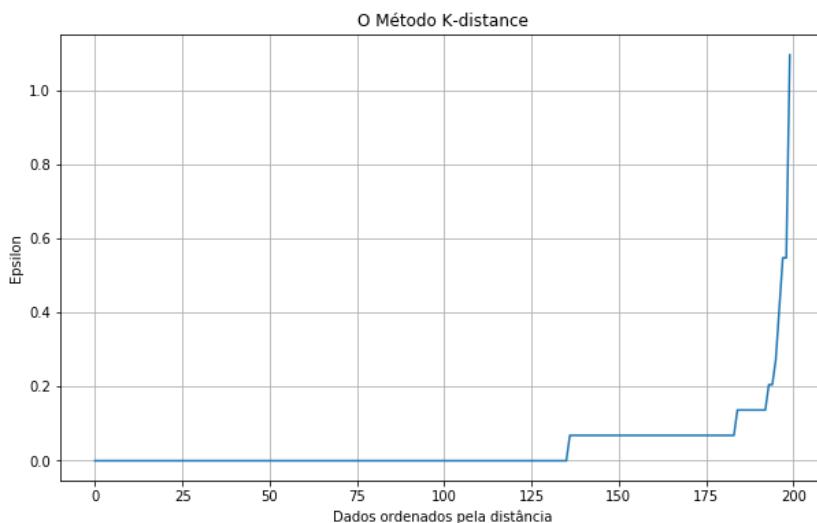
DBSCAN

In [35]:

```

1 # Encontrando o número ideal para os hiperparâmetros Delta e Epsilon
2 ## Delta
3 ### Será utilizado o valor 4, visto que o conjunto de dados é bidimensional (Delta = 2 x dimensão do conjunto de dados)
4
5 ## Epsilon com o método K-distance
6 vizinho = NearestNeighbors(n_neighbors=2)
7 modelo = vizinho.fit(df)
8 distances,indices = modelo.kneighbors(df)
9
10 distances = np.sort(distances, axis = 0)
11 distances = distances[:, 1]
12 plt.rcParams['figure.figsize'] = (10,6)
13 plt.plot(distances)
14 plt.title("Método K-distance")
15 plt.xlabel("Dados ordenados pela distância")
16 plt.ylabel("Epsilon")
17 plt.grid()
18 plt.show()

```



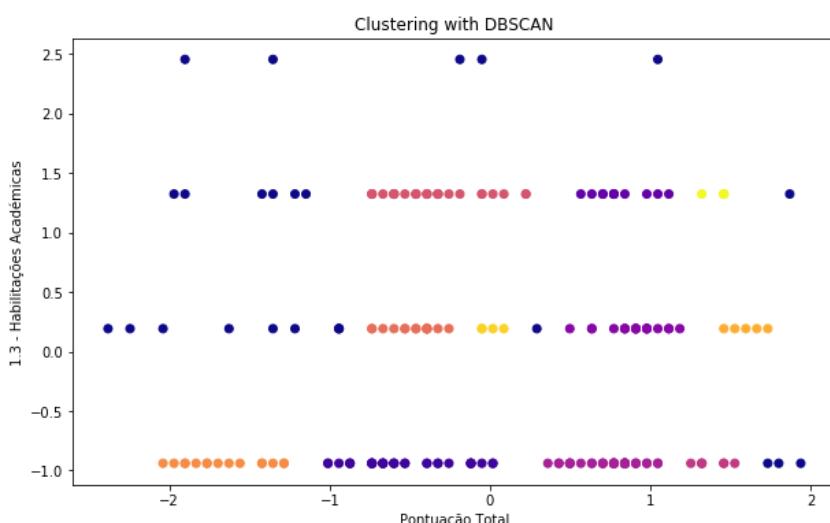
Assim, serão utilizados o valor 4 como hiperparâmetro do Delta e o valor 0.2 como hiperparâmetro do Epsilon

In [36]:

```

1 ## Visualizando os resultados
2 modelo = sklearn.cluster.DBSCAN(eps = 0.2, min_samples = 4).fit(df)
3 classificacao = modelo.labels_
4
5 plt.scatter(df["Pontuação total"], df["1.3 - Habilidades Académicas"], c = classificacao, cmap= "plasma")
6 plt.title("Clustering with DBSCAN")
7 plt.xlabel("Pontuação Total")
8 plt.ylabel("1.3 - Habilidades Académicas")
9 plt.show()

```



Como se pode ver, parece não existir clusters no conjunto de dados

Validação

Para verificar a qualidade das partições, iriam ser utilizadas as técnicas de índices internos e relativos para validar a compacidade dentro de cada partição e separação entre diferentes partições e para comparar as partições encontradas pelos dois algoritmos escolhidos ou por diferentes execuções do mesmo algoritmo, respetivamente. No entanto, decidiu-se não avançar com esta etapa devido a não existir clusters no conjunto de dados.

Problema Preditivo

Seleção dos Atributos

De modo a aproveitar ao máximo a abordagem de classification que se irá utilizar de seguida, será feito, em primeiro lugar, uma seleção dos devidos atributos e sua posterior transformação.

In [157]:

```

1 # Selecionando os atributos essenciais relativos ao conhecimento e comportamento dos estudantes
2 # perante as notícias da atualidade
3 df = pd.concat(
4     map(pd.read_csv, ['Quiz.csv', 'Grades.csv']), axis = 1)
5 df = df[['2.4 - Sabe o que significa "Fake News"?',
6          "2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?", 
7          "2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.", 
8          "2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?", 
9          "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a fonte da notícia]", 
10         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para confirmar a notícia]", 
11         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia]", 
12         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspetto da página]", 
13         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia]", 
14         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais à vontade com o assunto da notícia]", 
15         "2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso?", 
16         "2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil acesso?", 
17         "2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?", 
18         "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Fiabilidade]", 
19         "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Relevância]", 
20         "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Atualização]", 
21         "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Completa]", 
22         "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Explicita]", 
23         "4 - Terá mais atenção agora quando for ler uma notícia?", 
24         "Nota"]]

25
26 ## Eliminando Linhas de teste que foram feitas para testar o questionário
27 for y in range(0, 4):
28     df = df.drop(y)
29 df = df.drop(7)
30
31 ## Renomeando Linhas
32 df.index = list(range(1,201))
33 df.head()

```

Out[157]:

		2.4 - Sabe o que significa "Fake News"?	2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?	2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.	2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?	2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a fonte da notícia]	2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para confirmar a notícia]	2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia]	2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspetto da página]	2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia]	2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais à vontade com o assunto da notícia]	2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais à vontade com o assunto da notícia]	2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso?	2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil acesso?	2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?	2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Fiabilidade]	2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Relevância]	2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Atualização]	2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Completa]	2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Explicita]	4 - Terá mais atenção agora quando for ler uma notícia?	Nota
1	Sim	"Fake News" são informações fabricadas que im...	Sim	Duvido, de imediato, da sua veracidade	Frequentemente	Frequentemente	Raramente	Sempre	Sempre	Às vezes	Motores de pesquisa (Google, entre outros)	N										
2	Sim	"Fake News" são informações fabricadas proposi...	Sim	Duvido, de imediato, da sua veracidade	Sempre	Frequentemente	Raramente	Sempre	Sempre	Se...	Motores de pesquisa (Google, entre outros)	N										
3	Sim	"Fake News" são informações fabricadas proposi...	Sim	Duvido, de imediato, da sua veracidade	Sempre	Sempre	Sempre	Sempre	Sempre	Às vezes	Motores de pesquisa (Google, entre outros)	N										
4	Sim	"Fake News" são informações fabricadas proposi...	Sim	Duvido, de imediato, da sua veracidade	Frequentemente	Às vezes	Nunca	Às vezes	Às vezes	Nunca	Motores de pesquisa (Google, entre outros)	N										
5	Sim	"Fake News" são informações fabricadas que im...	Sim	Aceito, de imediato, a sua veracidade	Sempre	Às vezes	Às vezes	Sempre	Às vezes	Às vezes	Motores de pesquisa (Google, entre outros)	N										

In [158]:

```

1 # Transformando os atributos
2 ## Removendo instâncias com valores a nulo
3 df = df.dropna()
4
5 ## Convertendo para a escala correta
6 ### Convertendo a escala do tipo qualitativa dos atributos
7 ### para uma escala do tipo quantitativa relativa ou absoluta
8
9 ##### Atributo '2.4 - Sabe o que significa "Fake News"?'
10 x = ["Sim", "Não"]
11 y = [1.0, 0.0]
12 df['2.4 - Sabe o que significa "Fake News"?'] = df['2.4 - Sabe o que significa "Fake News"?'].replace(x,
13
14                                         y)
15 ##### Atributo '2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?'
16 x = ["Fake News" são informações fabricadas propositalmente que imitam o conteúdo dos media na forma, mas não no seu processo ou na intenção, "Fake News" são informações fabricadas que imitam propositalmente o conteúdo dos media na forma e no processo ou na intenção]
17 y = [1.0, 0.0]
18 df['2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?'] = df['2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?'].replace(x,
19
20                                         y)
21 ##### Atributo '2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.'
22 x = ["Sim", "Não"]
23 y = [1.0, 0.0]
24 df['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.'] = df['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.'].replace(x,
25
26                                         y)
27 ##### Atributo '2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?'
28 x = ["Duvido, de imediato, da sua veracidade", "Aceito, de imediato, a sua veracidade"]
29 y = [1.0, 0.0]
30 df['2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?'] = df['2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia.'].replace(x,
31
32                                         y)
33 ##### Atributo '2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a fonte da notícia]'
34 x = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
35 y = [1.0, 2.0, 3.0, 4.0, 5.0]
36 df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a fonte da notícia]'] = df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a fonte da notícia.].replace(x,
37
38                                         y)
39 ##### Atributo '2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para confirmar a veracidade da notícia]'
40 x = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
41 y = [1.0, 2.0, 3.0, 4.0, 5.0]
42 df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para confirmar a veracidade da notícia.]'] = df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para confirmar a veracidade da notícia.].replace(x,
43
44                                         y)
45 ##### Atributo '2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia]'
46 x = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
47 y = [1.0, 2.0, 3.0, 4.0, 5.0]
48 df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia.]'] = df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia.].replace(x,
49
50                                         y)
51 ##### Atributo '2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspeto da página (caso de uso)]'
52 x = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
53 y = [1.0, 2.0, 3.0, 4.0, 5.0]
54 df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspeto da página (caso de uso).]'] = df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspeto da página (caso de uso.].replace(x,
55
56                                         y)
57 ##### Atributo '2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia]'
58 x = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
59 y = [1.0, 2.0, 3.0, 4.0, 5.0]
60 df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia.]'] = df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia.].replace(x,
61
62                                         y)
63 ##### Atributo '2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais informada sobre o tema]'
64 x = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
65 y = [1.0, 2.0, 3.0, 4.0, 5.0]
66 df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais informada sobre o tema.]'] = df['2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais informada sobre o tema.].replace(x,
67
68                                         y)
69 ##### Atributo '2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso'
70 x = ["Bases de dados", "Pessoas", "Motores de pesquisa (Google, entre outros)"]
71 y = [1.0, 2.0, 3.0]
72 df['2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso?'] = df['2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso.].replace(x,
73
74                                         y)
75 ##### Atributo '2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil acesso'
76 x = ["Motores de pesquisa (Google, entre outros)", "Pessoas", "Bases de dados"]
77 y = [1.0, 2.0, 3.0]
78 df['2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil acesso?'] = df['2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil acesso.].replace(x,
79
80                                         y)
81 ##### Atributo '2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?'
82 x = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
83 y = [1.0, 2.0, 3.0, 4.0, 5.0]
84 df['2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?'] = df['2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas.'].replace(x,
85
86                                         y)
87 ##### Atributo '2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Fiabilidade]'
88

```

```

89 x = ["1º", "2º", "3º", "4º", "5º"]
90 y = [1.0, 2.0, 3.0, 4.0, 5.0]
91 df['2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Fiabilidade]'] = df['2.10 - Ordenar y]
92
93
94 ##### Atributo '2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Relevância]'
95 x = ["1º", "2º", "3º", "4º", "5º"]
96 y = [1.0, 2.0, 3.0, 4.0, 5.0]
97 df['2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Relevância]'] = df['2.10 - Ordenar y]
98
99
100 ##### Atributo '2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Atualização]'
101 x = ["1º", "2º", "3º", "4º", "5º"]
102 y = [1.0, 2.0, 3.0, 4.0, 5.0]
103 df['2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Atualização]'] = df['2.10 - Ordenar y]
104
105
106 ##### Atributo '2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Completa]'
107 x = ["1º", "2º", "3º", "4º", "5º"]
108 y = [1.0, 2.0, 3.0, 4.0, 5.0]
109 df['2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Completa]'] = df['2.10 - Ordenar y']
110
111
112 ##### Atributo '2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Explicita]'
113 x = ["1º", "2º", "3º", "4º", "5º"]
114 y = [1.0, 2.0, 3.0, 4.0, 5.0]
115 df['2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Explicita]'] = df['2.10 - Ordenar y']
116
117
118 ##### Atributo '4 - Terá mais atenção agora quando for ler uma notícia?'
119 x = ["Sim", "Não"]
120 y = [1.0, 0.0]
121 df['4 - Terá mais atenção agora quando for ler uma notícia?'] = df['4 - Terá mais atenção agora quando for ler uma notícia?'].replace(x, y)
122
123
124 ##### Atributo 'Nota'
125 x = ["Fraco", "Insuficiente", "Suficiente", "Bom", "Excelente"]
126 y = [1.0, 2.0, 3.0, 4.0, 5.0]
127 df['Nota'] = df['Nota'].replace(x, y)
128
129 df.head()

```

Out[158]:

2.4 - Sabe o que significa "Fake News"?	2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?	2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.	2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?	2.6 - Com frequência utiliza os seguintes critérios como forma de analisar uma notícia?	2.6 - Com frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para confirmar a notícia]	2.6 - Com frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspecto da página (caso se trate de uma notícia online)]	2.6 - Com frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia]	2.6 - Com frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia]	2.6 - Com frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais à vontade com o assunto da notícia]	2.6 - Com frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que esteja mais à vontade com o assunto da notícia]	2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso?	2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil acesso?	
1	1.0	0.0	1.0	1.0	4.0	4.0	2.0	5.0	5.0	5.0	3.0	3.0	1.0
2	1.0	1.0	1.0	1.0	5.0	4.0	2.0	5.0	5.0	5.0	5.0	3.0	1.0
3	1.0	1.0	1.0	1.0	5.0	5.0	5.0	5.0	5.0	5.0	3.0	3.0	3.0
4	1.0	1.0	1.0	1.0	4.0	3.0	1.0	3.0	3.0	3.0	1.0	3.0	1.0
5	1.0	0.0	1.0	0.0	5.0	3.0	3.0	5.0	3.0	3.0	3.0	3.0	2.0

Abordagem Classification

Na procura do melhor resultado possível, irá ser utilizado um algoritmo ensemble, mais especificamente, o algoritmo Random Forest.

In [159]:

```

1 # Definir o conjunto de dados
2 X = df[['2.4 - Sabe o que significa "Fake News?"',
3         "2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?", 
4         "2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.', 
5         "2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?", 
6         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a fonte da notícia]', 
7         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para', 
8         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia]', 
9         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspecto da página]', 
10        "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia]', 
11        "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que este', 
12        "2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso?", 
13        "2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil ac', 
14        "2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?", 
15        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Fiabilidade]', 
16        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Relevância]', 
17        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Atualização]', 
18        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Completa]', 
19        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Explicita]', 
20        "4 - Terá mais atenção agora quando for ler uma notícia?']].to_numpy()
21 y = df['Nota'].to_numpy()
22
23 # Valores a avaliar
24 desempenhos = []
25 k_values = [1, 2, 3, 4]
26 for k in k_values:
27     # Criando um modelo de classificação
28     modelo = sklearn.ensemble.RandomForestClassifier()
29
30     # Criando um conjunto de dados de treino mais balanceado
31     over = SMOTE(sampling_strategy = {2.0: 55, 3.0: 81, 4.0: 82, 5.0: 20}, k_neighbors = k)
32     under = RandomUnderSampler(sampling_strategy = {2.0: 55, 3.0: 60, 4.0: 61, 5.0: 20})
33
34     # Preparando o processo a ser executado
35     passos = [('over', over), ('under', under), ('modelo', modelo)]
36     pipeline = Pipeline(steps = passos)
37
38     # Treinando o modelo e fazendo previsões
39     cv = sklearn.model_selection.RepeatedStratifiedKFold(n_splits = 6, n_repeats = 3, random_state = 1)
40     scores = sklearn.model_selection.cross_val_score(pipeline, X, y, scoring = 'accuracy', cv = cv, n_jobs = -1)
41     print(scores)
42     desempenhos.append(np.mean(scores))

```

```
[0.87878788 0.87878788 0.90909091 0.87878788 0.87878788 0.84375
 0.81818182 0.90909091 0.84848485 0.84848485 0.87878788 0.90625
 0.84848485 0.90909091 0.90909091 0.84848485 0.93939394 0.9375      ]
[0.84848485 0.93939394 0.93939394 0.87878788 0.90909091 0.84375
 0.87878788 0.84848485 0.87878788 0.81818182 0.84848485 0.90625
 0.81818182 0.90909091 0.90909091 0.87878788 0.87878788 0.9375      ]
[0.90909091 0.90909091 0.90909091 0.87878788 0.87878788 0.84375
 0.84848485 0.81818182 0.87878788 0.84848485 0.87878788 0.875
 0.87878788 0.84848485 0.90909091 0.87878788 0.87878788 0.90625      ]
[0.87878788 0.90909091 0.90909091 0.87878788 0.90909091 0.84375
 0.84848485 0.93939394 0.87878788 0.81818182 0.87878788 0.90625
 0.87878788 0.90909091 0.90909091 0.87878788 0.90909091 0.84375
 0.87878788 0.90909091 0.90909091 0.87878788 0.90909091 0.90625
 0.87878788 0.90909091 0.90909091 0.87878788 0.90909091 0.90625      ]
```

Validation

Para verificar a qualidade do modelo, será utilizado o acerto

In [160]:

```

1 vizinho = 0
2 for desempenho in desempenhos:
3     print('> k=%d, Mean Accuracy: %.3f' % (vizinho + 1, desempenho))
4     vizinho += 1

```

```
> k=1, Mean Accuracy: 0.882
> k=2, Mean Accuracy: 0.882
> k=3, Mean Accuracy: 0.876
> k=4, Mean Accuracy: 0.888
```

Abordagem alternativa

In [161]:

```

1 # Dividindo os dados em treino (70%) e teste (30%)
2 X = df[['2.4 - Sabe o que significa "Fake News"?',
3         "2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?", 
4         "2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.", 
5         "2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?", 
6         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a fonte da notícia]", 
7         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico outras fontes para", 
8         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o autor da notícia]", 
9         "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico o aspecto da página]", 
10        "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Verifico a data da notícia]", 
11        "2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? [Consulto uma pessoa que este", 
12        "2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso?", 
13        "2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil ac", 
14        "2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?", 
15        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Fiabilidade]", 
16        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Relevância]", 
17        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Atualização]", 
18        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Completa]", 
19        "2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação [Explicita]", 
20        "4 - Terá mais atenção agora quando for ler uma notícia?"]]
21 y = df['Nota']
22
23 X_train, X_test, y_train, y_test = sklearn.model_selection.train_test_split(X, y, test_size = 0.3)
24
25
26
27
28 # Criando um modelo de classificação
29 clf = sklearn.ensemble.RandomForestClassifier(n_estimators = 100)
30
31 # Treinando o modelo e fazendo previsões
32 clf.fit(X_train, y_train)
33 y_pred = clf.predict(X_test)
34
35 ## Utilizando a medida de desempenho do acerto
36 print("Acerto:", sklearn.metrics.accuracy_score(y_test, y_pred))

```

Acerto: 0.9

G. CONCLUSÃO

- 1 Em síntese, desde o enquadramento teórico - com a explicação da definição do termo "Fake News" - dos procedimentos metodológicos - com a utilização do Google Forms para a recolha dos dados, bem como com a utilização da linguagem Python para uma melhor manipulação - e da apresentação e discussão dos resultados - como investigar o comportamento e conhecimento dos estudantes aplicando-se técnicas descritivas e preditivas - cumpriu-se o objetivo de esmiuçar o comportamento informacional dos estudantes da FEUP perante a informação e comparar os estudantes com a generalidade das pessoas relativamente ao conhecimento e importância atribuída ao tema das "Fake News".
- 2 Após de se ter utilizado a estatística para a caracterização dos estudantes da Faculdade de Engenharia, constatou-se que a média da pontuação do comportamento informacional dos estudantes é suficientemente razoável. Por outras palavras, considerando uma hipotética escala qualitativa ordinal em que a pontuação que vai de 0 a 49 corresponde a insuficiente, de 50 a 74 de suficiente, de 75 a 89 de bom, e de 90 a 100 de excelente, os estudantes encontram-se situados na correspondência de suficiente, com uma média de 65.7, aproximadamente.
- 3 Para além disto, com a aplicação da abordagem clustering concluiu-se que não existe partições entre grupos sociais e o conhecimento e comportamento dos estudantes perante as notícias da atualidade (e sua relação com as "Fake News").
- 4 Ademais, com a aplicação da abordagem classification chegou-se a um modelo com um acerto acima do que foi definido inicialmente (>70%).
- 5 Apesar de alguns problemas terem surgido ao longo do desenvolvimento do trabalho, com esforço, empenho, tempo e perseverança, conseguiu-se realizar o relatório e ultrapassar todos os obstáculos que se deparam no caminho. De facto, a experiência foi benéfica, uma vez que permitiu adquirir um maior conhecimento acerca da realidade do comportamento informacional dos estudantes. Assim sendo, e depois de se ter feito uma averiguação pela análise bibliográfica de artigos que abordam este assunto, detetou-se que a sua existência é quase nula. Sendo um tema que prejudica a divulgação do conhecimento, é preciso tomar muita atenção a notícias que não traduzem a verdade. Neste cenário, adotar uma postura mais crítica para uma melhor literacia informacional, aliada a uma inclusão digital, irá fazer com que haja uma maior atenção dos estudantes perante informações que não devem ser consumidas. Após a realização deste trabalho, sentiu-se que se pôde, pela primeira vez, olhar para a realidade académica de forma diferente, admitindo que esta capacidade de observação será essencial na preparação de futuros profissionais na área da gestão da informação. Assim, aprendeu-se com os erros e foi possível chegar a conclusões.

H. REFERÊNCIAS BIBLIOGRÁFICAS

1. Academy, D. S. (2016). Python fundamentos para análise de dados. Retrieved November 06, 2022, from <https://www.datascienceacademy.com.br/start>
2. Community. (n.d.). Python documentation. Retrieved November 06, 2022, from <https://docs.python.org/3/>
3. Overflow, S. (2008). Stack Overflow - Where Developers Learn, Share, & Build Careers. Retrieved November 06, 2022, from <https://stackoverflow.com/>
4. Rush, L. lrush@odu. ed. (2018). Examining student perceptions of their knowledge, roles, and power in the information cycle: Findings from a “fake news” event. *Journal of Information Literacy*, 12(2), 121-130. <https://doi.org/10.11645/12.2.2484>
5. Musgrove, A. T. 1. musgrove@fau. ed., Powers, J. R. ., Rebar, L. C. ., & Musgrove, G. J. . (2018). Real or fake? Resources for teaching college students how to identify fake news. *College & Undergraduate Libraries*, 25(3), 243-260. <https://doi.org/10.1080/10691316.2018.1480444>
6. Notícia falsa. (2019, maio 1). Wikipédia, a encyclopédia livre. Retrieved November 06, 2022 from https://pt.wikipedia.org/w/index.php?title=Not%C3%ADcia_falsa&oldid=55000722.
7. Clickbait. (2018, junho 26). Wikipédia, a encyclopédia livre. Retrieved November 06, 2022 from <https://pt.wikipedia.org/w/index.php?title=Clickbait&oldid=52464537>.

I. ANEXOS

In [4]:

```
1 from IPython.display import Image
2 Image("print1.png", width=5000)
```

Out[4]:



CONHECIMENTO E COMPORTAMENTO DOS ESTUDANTES DA FEUP PERANTE NOTÍCIAS DA ATUALIDADE

O presente questionário, elaborado na unidade curricular de Análise e Visualização de Dados do Mestrado em Ciência da Informação da Faculdade e Engenharia da Universidade do Porto, tem como objetivos comparar os estudantes da FEUP com a generalidade das pessoas no que diz respeito ao conhecimento e importância atribuída ao tema das “Fake News” e avaliar o comportamento dos estudantes perante a informação.

Com este estudo, pretende-se que os estudantes reflitam e adotem uma postura mais criteriosa e crítica durante a seleção da informação, de forma a construir uma comunidade estudantil mais culta e consciente capaz de responder aos constantes desafios eminentes nas suas vidas pessoais, académicas e profissionais.

Os dados são recolhidos de forma anónima e para o uso exclusivamente académico.

O tempo de preenchimento do questionário é de 3 minutos.

Agradecemos a sua participação neste inquérito.

In [8]:

```
1 from IPython.display import Image  
2 Image("print2.png", width=5000)
```

Out[8]:

Seção 2 de 3

A - QUESTIONÁRIO

Responda, honestamente, às seguintes perguntas.

1 - IDENTIFICAR GRUPOS SOCIAIS

Descrição (opcional)

1.1 - Idade *

< 18
 18-25
 26-40
 41-65
 > 65

In [9]:

```
1 from IPython.display import Image  
2 Image("print3.png", width=5000)
```

Out[9]:

1.2 - Sexo *

Masculino
 Feminino

1.3 - Habilidades Académicas *

Ensino Secundário
 Licenciatura
 Mestrado
 Doutoramento

1.4 - Curso a frequentar *

Ex: Licenciatura em Ciéncia da Informação

Texto de resposta curta

2 - CONHECIMENTO E COMPORTAMENTO DOS ESTUDANTES

Descrição (opcional)

In [10]:

```
1 from IPython.display import Image
2 Image("print4.png", width=5000)
```

Out[10]:

2.1 - Tem acesso à Internet em casa? *

Sim
 Não

2.2 - Costuma utilizar a Internet noutra lugar? *

Sim
 Não

2.3 - Qual é o meio que mais utiliza para estar a par das notícias? *

Redes Sociais
 Televisão
 Rádio
 Jornais
 Revistas

In [11]:

```
1 from IPython.display import Image
2 Image("print5.png", width=5000)
```

Out[11]:

2.4 - Sabe o que significa "Fake News"? *

Sim
 Não

2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?

"Fake News" são informações fabricadas propositalmente que imitam o conteúdo dos media na forma, mas não ...
 "Fake News" são informações fabricadas que imitam propositalmente o conteúdo dos media na forma e no proc...

2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a divulgação do conhecimento.

Sim
 Não

2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia? *

Aceito, de imediato, a sua veracidade
 Duvido, de imediato, da sua veracidade

In [12]:

```
1 from IPython.display import Image
2 Image("print6.png", width=5000)
```

Out[12]:

2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "Fake New"?

Texto de resposta curta

2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia? *

	Nunca	Raramente	Às vezes	Frequentemente	Sempre
Verifico a fonte d...	<input type="radio"/>				
Verifico outras fo...	<input type="radio"/>				
Verifico o autor d...	<input type="radio"/>				
Verifico o aspeto ...	<input type="radio"/>				
Verifico a data da ...	<input type="radio"/>				
Consulto uma pe...	<input type="radio"/>				

In [13]:

```
1 from IPython.display import Image
2 Image("print7.png", width=5000)
```

Out[13]:

2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação geral de fácil acesso? *

- Motores de pesquisa (Google, entre outros)
- Bases de dados
- Pessoas

2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma necessidade de informação específica de difícil acesso? *

- Motores de pesquisa (Google, entre outros)
- Bases de dados
- Pessoas

2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas pesquisas?

- Sempre
- Frequentemente
- Às vezes
- Raramente
- Nunca

In [14]:

```
1 from IPython.display import Image
2 Image("print8.png", width=5000)
```

Out[14]:

2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação

1 - Mais importante; 5 - Menos importante

	1º	2º	3º	4º	5º
Fiabilidade	<input type="radio"/>				
Relevância	<input type="radio"/>				
Atualização	<input type="radio"/>				
Completa	<input type="radio"/>				
Explícita	<input type="radio"/>				

Após a seção 2 [Continuar para a próxima seção](#)

Seção 3 de 3

B - FEEDBACK

Responda, honestamente, às seguintes perguntas.

1 - Achou este questionário útil? *

- Sim
- ...

In [15]:

```
1 from IPython.display import Image  
2 Image("print9.png", width=5000)
```

Out[15]:

2 - Achou o preenchimento deste questionário acessível em termos de clareza, precisão e tempo? *

Sim

Não

3 - Recomendaria este questionário a alguém? *

Sim

Não

4 - Terá mais atenção agora quando for ler uma notícia? *

Sim

Não