

In [30]:

```
from IPython.display import Image  
Image("imagem11.jpg", width=500)
```

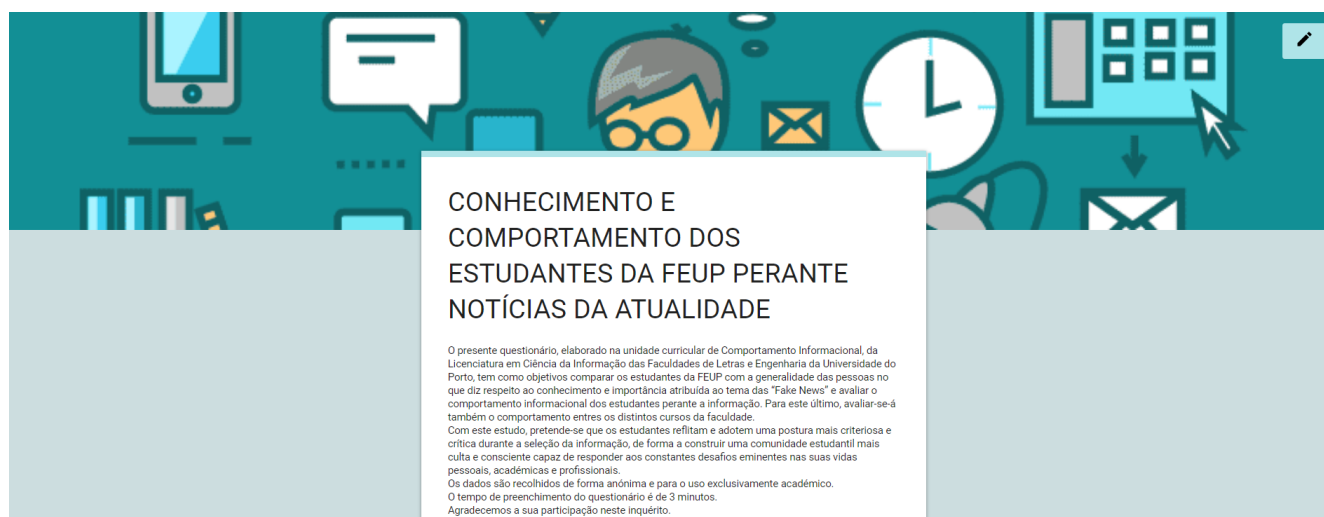
Out[30]:



In [28]:

```
from IPython.display import Image  
Image('imagem.png')
```

Out[28]:



Curso a frequentar

Ciência da Informação

Unidade curricular de estudo

Comportamento Informacional

Autores de trabalho

Simão Machado up201704685@letras.up.pt
Carina Borges up201705386@letras.up.pt
Cláudia Rocha up201705906@letras.up.pt

A. ÍNDICE

B. Introdução

C. Enquadramento Teórico

C. ENQUADRAMENTO TEÓRICO

D. Procedimentos Metodológicos

1. Coleta dos Dados com Google Analytics
2. Limpeza e Transformação com Python - Data Wrangling

E. Apresentação e Discussão dos Resultados

1. Visualização dos Dados - Pergunta 1.1 a 2.10
 - 1.1 Identificar Grupos Sociais
 - 1.2 Conhecimento e Comportamento dos estudantes da FEUP

F. Análise Exploratória dos Dados

1. Estatística Descritiva - Caracterização da Amostra
2. Estatística Inferencial - Caracterização da População

G. Conclusão

H. Bibliografia

I. Anexos

B. INTRODUÇÃO

No âmbito da unidade curricular de Comportamento Informacional, da licenciatura em Ciência da Informação, foi-nos proposto desenvolver um questionário. O público-alvo escolhido foram os estudantes da Faculdade de Engenharia do Porto. O questionário devia de estar relacionado com os conteúdos debruçados nas aulas da cadeira. De todas as temáticas possíveis de serem abordadas, selecionou-se o tema das "Fake News". A razão da sua escolha deveu-se, sobretudo, à atualidade do mesmo, bem como pelo facto de este estar a impactar, indiretamente e eficazmente, toda a noção e consciência de que uma pessoa tem no que diz respeito a informações digitais fidedignas. Tendo como base o questionário referido, este artigo irá debater esta questão minuciosamente. De uma forma geral, este artigo 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 informacional dos estudantes perante a informação. Primeiramente, utilizar-se-á este inquérito e a rede social Twitter para comparar se existe conhecimento do impacto das "Fake News" pelos estudantes e pelas pessoas, respetivamente. De seguida, irá se utilizar somente este inquérito na avaliação do comportamento informacional dos estudantes do Porto. No seu longo processo de desenvolvimento, os membros do grupo reuniram-se em várias ocasiões. Numa perspetiva mais universitária, o trabalho de grupo tem como objetivo primordial a demonstração das qualidades de pesquisa trabalhadas nas aulas da cadeira complementar de Fontes de Informação e Serviços de Referência (acesso e recuperação da informação em várias tipologias de fontes) e de qualidades interpretativas de amostras ou de populações trabalhadas nas aulas da cadeira de Estatística. Em ambos os casos, os conteúdos foram arraigados durante o semestre e irão ser aplicados, neste caso, a uma situação problemática precisa do mundo em rede. 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. Portanto, este artigo pode ser dividido da seguinte forma: enquadramento teórico, procedimentos metodológicos, apresentação e discussão dos resultados e análise exploratória de dados. Por último, será feita uma conclusão final sobre quais foram as considerações finais percebidas.

C. ENQUADRAMENTO TEÓRICO

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 verosímeis. "Fake News", tal como o próprio nome indica, são notícias falsas que são divulgadas propositadamente 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 on-line, 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).

2. Limpeza e Transformação com Python - Data Wrangling/Munging

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 manipular 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]:  
  
# Faz o upgrade do Pyhton Package Index  
!python -m pip install --upgrade pip
```

Requirement already up-to-date: pip in c:\users\simao\anaconda3\lib\site-packages (19.1.1)

```
In [1]:  
  
## Importando os pacotes de análise de dados - PYTHON OPEN DATA SCIENCE STACK  
import os  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from scipy import stats  
  
# Pacotes de complemento ao matplotlib.pyplot  
from pylab import *  
import seaborn as sns  
  
# Importando os módulos Tweepy, Datetime e Json  
from tweepy.streaming import StreamListener  
from tweepy import OAuthHandler  
from tweepy import Stream  
from datetime import datetime  
import json
```

```
In [4]:  
  
# Lendo o Microsoft Excel em formato Comma Separated Values (.csv) com as respostas do  
questionário  
# e importando para o formato tabular do pacote Pandas  
df = pd.read_csv("Questionario.csv")  
df.head()
```

Out[4]:

	Carimbo de data/hora	Pontuação total	1.1 - Idade	1.1 - Idade [Pontuação]	1.1 - Idade [Feedback]	1.2 - Sexo	1.2 - Sexo [Pontuação]	1.2 - Sexo [Feedback]	1.3 - Habilitações Académicas	1.3 - Habilitações Académicas [Pontuação]	
0	2019/04/18 3:23:12 PM GMT+1	69.00 / 100	< 18	-- / 0	NaN	Masculino	-- / 0	NaN	Ensino Secundário	-- / 0	.
1	2019/04/18 4:16:38 PM GMT+1	58.00 / 100	18-25	-- / 0	NaN	Feminino	-- / 0	NaN	Ensino Secundário	-- / 0	.
2	2019/04/18 4:24:54 PM GMT+1	63.00 / 100	18-25	-- / 0	NaN	Feminino	-- / 0	NaN	Ensino Secundário	-- / 0	.
3	2019/04/18 11:25:06 PM	64.00 / 100	18-25	-- / 0	NaN	Masculino	-- / 0	NaN	Licenciatura	-- / 0	.

	GMT+1										
4	2019/04/24 12:00 AM Carimbo de data/hora	Pontuação total	18-25 Idade	1.1 - Idade [Pontuação]	1.1 - Idade [Feedback]	Masculino Sexo	1.2 - Sexo [Pontuação]	1.2 - Sexo [Feedback]	Ensino Secundário 1.3 - Habilitações Académicas	1.3 - Habilitações Académicas [Pontuação]	
5 rows × 85 columns											

In [3]:

```
# Organizando os dados para uma melhor análise, excluindo aqueles que não foram considerados relevantes antes
## Eliminando colunas desnecessárias
for x in df:
    if "[Pontuação]" in x:
        df = df.drop(x, 1)
    elif "[Feedback]" in x:
        df = df.drop(x, 1)
    elif "Unnamed" in x:
        df = df.drop(x, axis=1)
    elif ("Carimbo de data/hora" == x):
        df = df.drop(x, axis=1)

## Eliminando linhas de teste que foram feitas para testar o questionário
for y in range(0, 4):
    df = df.drop(y)
df = df.drop(7)

## Renomeando linhas
df.index = list(range(1,201))
df.head()
```

Out[3]:

	Pontuação total	1.1 - Idade	1.2 - Sexo	1.3 - Habilitações Académicas	1.4 - Curso a frequentar	2.1 - Tem acesso à Internet em casa?	2.2 - Costuma utilizar a Internet noutro 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 - op (AN res pe
1	57.00 / 100	18-25	Masculino	Ensino Secundário	MIEIC	Sim	Sim	Redes Sociais	Sim	"Fake News" são informações fabricadas que imi...	Freque
2	75.00 / 100	18-25	Masculino	Mestrado	MIEEC	Sim	Sim	Redes Sociais	Sim	"Fake News" são informações fabricadas proposi...	Raram
3	80.00 / 100	18-25	Masculino	Licenciatura	Mestrado em Multimedia	Sim	Sim	Jornais	Sim	"Fake News" são informações fabricadas proposi...	Nunca
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

5	38.00 / 100	18-25	Masculino	Mestrado	Mestrado em Engenharia Civil	Sim	Sim	2.3 - Qual é o meio de comunicação que mais utiliza para obter informações?	Sim	"Fake News" são informações fabricadas que imitam notícias verdadeiras.	2.9 - Raramente ou nunca
	Pontuação	1.1 - Idade	1.2 - Sexo	1.3 - Habilitações Acadêmicas	1.4 - Curso a frequentar	2.1 - Tem acesso à Internet	2.2 - Costuma utilizar a Internet noutro		2.4 - Sabe o que significa "Fake News"?	2.4.1 - Se sim, qual das seguintes opções corresponde melhor à sua opinião?	
5 rows × 31 columns											

E. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

1. Visualização dos Dados - Pergunta 1.1 a 2.10

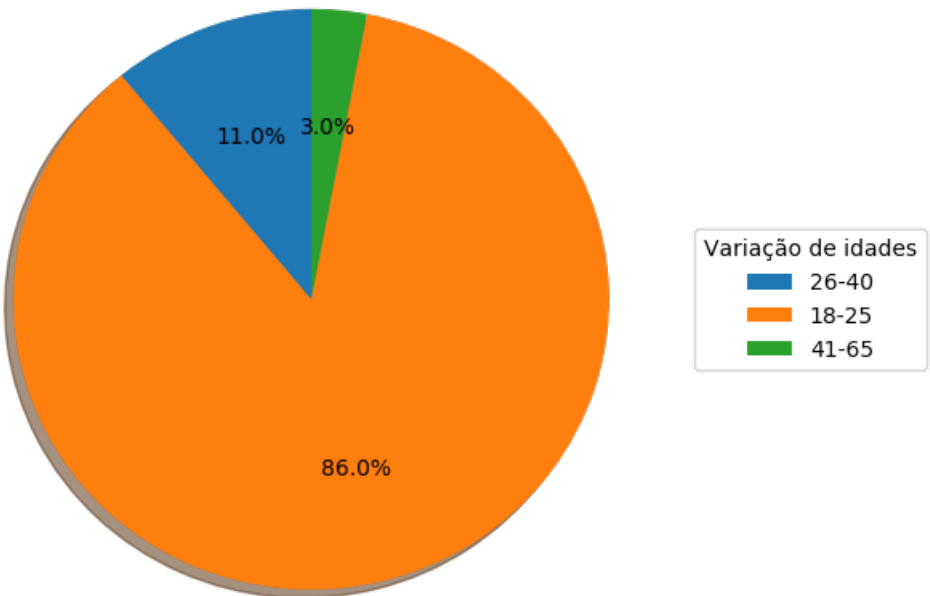
Identificar grupos sociais

In [5]:

```
# Pergunta 1.1
## Calcular as categorias e suas frequências
idades = ["26-40", "18-25", "41-65"]
frequencia = []
for x in idades:
    mask = df["1.1 - Idade"] == x
    df1 = df[mask]
    df2 = df1["1.1 - Idade"]
    df3 = df2.count()
    frequencia.append(df3)

## Visualizar a pergunta 1.1 num gráfico circular
plt.figure(figsize = (6, 6), dpi = 100)
plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")
plt.title("1.1 - Idade")
plt.legend(idades, title="Variação de idades", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
plt.show()
resultado = list(zip(idades, frequencia))
resultado
```

1.1 - Idade



Out[5]:

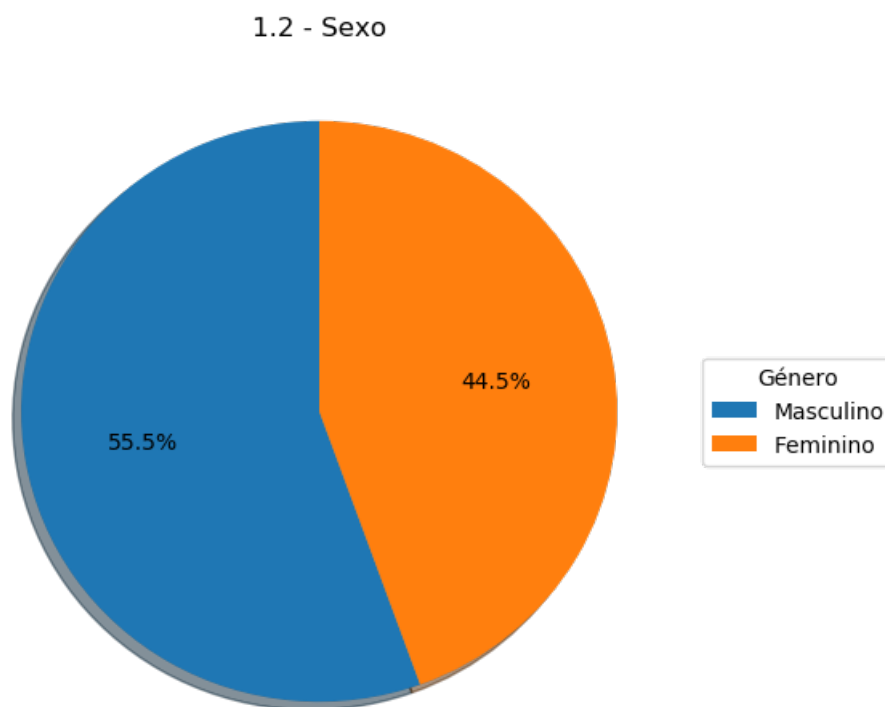
```
[('26-40', 22), ('18-25', 172), ('41-65', 6)]
```

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

In [6]:

```
# Pergunta 1.2
## Calcular as categorias e suas frequências
genero = ["Masculino", "Feminino"]
frequencia = []
for x in genero:
    mask = df["1.2 - Sexo"] == x
    df1 = df[mask]
    df2 = df1["1.2 - Sexo"]
    df3 = df2.count()
    frequencia.append(df3)

## Visualizar a pergunta 1.2 num gráfico circular
plt.figure(figsize = (6, 6), dpi = 100)
plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")
plt.title("1.2 - Sexo")
plt.legend(genero, title="Género", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
plt.show()
resultado = list(zip(genero, frequencia))
resultado
```



Out[6]:

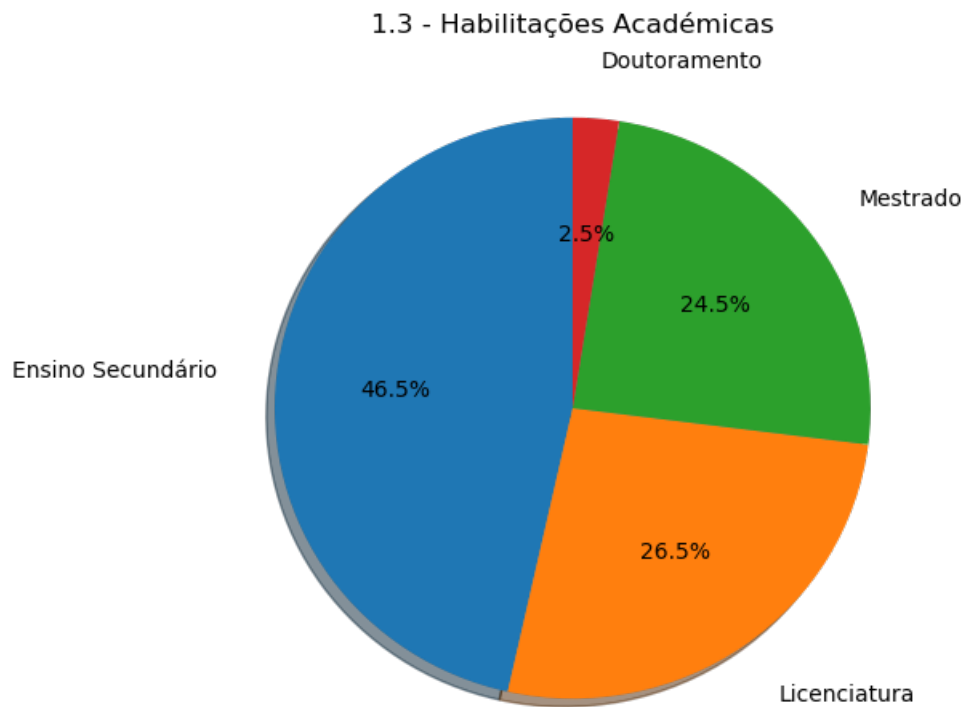
```
[('Masculino', 111), ('Feminino', 89)]
```

In [7]:

```
# Pergunta 1.3
## Calcular as categorias e suas frequências
habilitacoes = ["Ensino Secundário", "Licenciatura", "Mestrado", "Doutoramento"]
frequencia = []
for x in habilitacoes:
    mask = df["1.3 - Habilitações Académicas"] == x
    df1 = df[mask]
    df2 = df1["1.3 - Habilitações Académicas"]
    df3 = df2.count()
    frequencia.append(df3)

## Visualizar a pergunta 1.3 num gráfico circular
plt.figure(figsize = (6, 6), dpi = 100)
```

```
plt.pie(frequencia, labels = habilitacoes, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")
plt.title("1.3 - Habilitações Académicas")
plt.show()
resultado = list(zip(habilitacoes, frequencia))
resultado
```



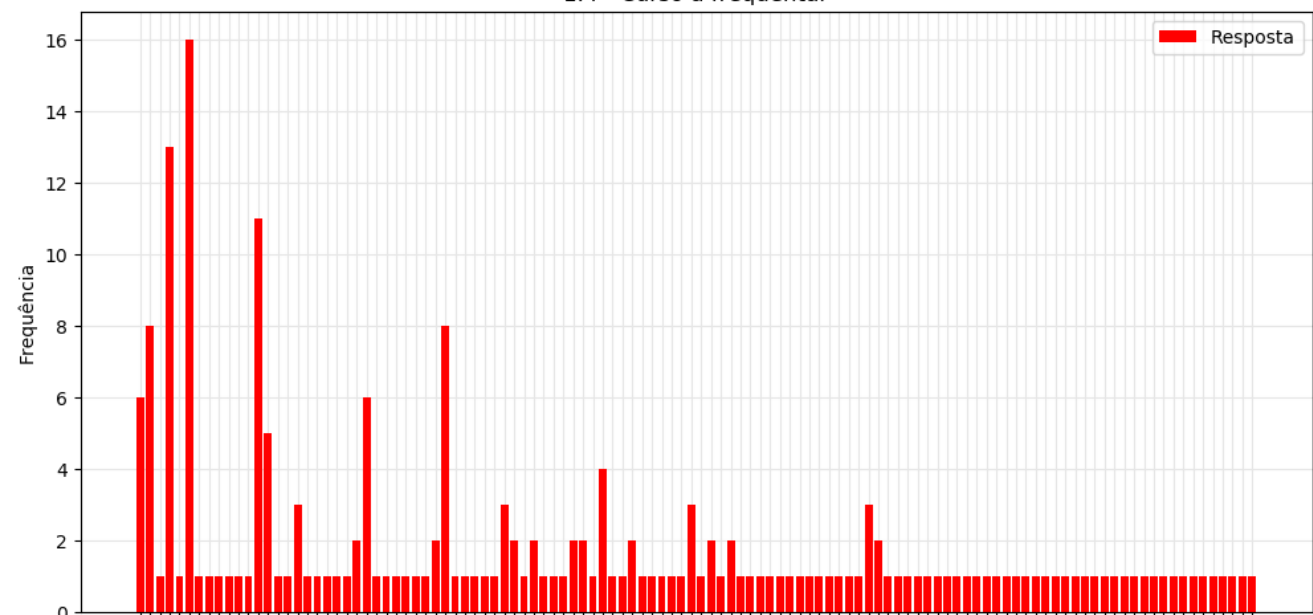
Out[7]:

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

In [34]:

```
# Pergunta 1.4
## 1º Round
### Calcular todas as categorias (considerando todas as possibilidades para um curso) e suas frequências
frequencia = []
curso = []
for x in df["1.4 - Curso a frequentar"]:
    if (x not in curso):
        curso.append(x)
        mask = df["1.4 - Curso a frequentar"] == x
        df1 = df[mask]
        df2 = df1["1.4 - Curso a frequentar"]
        df3 = df2.count()
        frequencia.append(df3)
    else:
        pass

### Visualizar a pergunta 1.4 num gráfico de barras com todas as categorias e frequências
plt.figure(figsize = (12, 6), dpi = 100)
plt.grid(color = '0.90')
plt.bar(curso, frequencia, zorder = 100, color="r", label="Resposta")
plt.xlabel("Cursos")
plt.title("1.4 - Curso a frequentar")
plt.ylabel("Frequência")
plt.xticks(rotation=90)
plt.legend()
plt.show()
resultado = list(zip(curso, frequencia))
resultado
```



Não precisas de mandar 3 vezes o mesmo email em maiúsculas, não

Out[34]:

```
[('MIEIC', 6),
 ('MIEEC', 8),
 ('Mestrado em Multimedia ', 1),
 ('Mestrado Integrado em Bioengenharia', 13),
 ('Mestrado em Engenharia Civil', 1),
 ('Mestrado Integrado em Engenharia Mecânica', 16),
 ('Informatica', 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),
 ('Mestrado Integrado em Engenharia Informática', 2),
 ('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', 1),
 ('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 Fisica', 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 Eletrotécnica', 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),
 ('MIEC', 1)]
```

```
( 'MIB', 1),
('PDEMG', 1),
('MIEM', 3),
('Mestrado Integrado em Engenharia Fisica', 2),
('Mestrado Integrado em Bioengenharia', 1),
('Mestrado Integrado Em Bioengenharia', 1),
('Mestrado em Engenharia Quimica', 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),
('mestrado', 1),
('mestrado integrado engenharia informática e computação', 1),
('Engenharia mecânica ', 1),
('Mestrado integrado em bioengenharia ', 1),
('Mestrado Integrado em Engenharia Quimica', 1),
('Miemmm', 1),
('mestrado integrado em engenharia mecânica', 1),
('Engenharia Eletrotécnica e Computação', 1),
('Engenharia de Minas', 1),
('Engenharia Mecânica', 1),
('Mestrado Em Engenharia Mecânica', 1),
('Mestrado Integrado em Engenharia Eletrotécnica e Computação', 1),
('Mestrado Integrado em Engenharia Eletrotécnica e Computadores ', 1),
('Mestrado Integrado de Engenharia Informática e Computação', 1),
('Engenharia Química (Mestrado integrado)', 1),
('PRODEC', 1),
('Mestrado integrado em Engenharia Eletrotécnica e Computadores', 1),
('Mestrado em Engenharia Biomédica', 1),
('Mestrado Integrado de Engenharia Química', 1),
('Mestrado em Engenharia', 1),
('Programa doutoral', 1),
('Licenciatura em Engenharia Mecânica', 1),
('Mestrado em Design Industrial e de produto ', 1),
('mestrado integrado em engenharia electrotécnica e de computadores ', 1),
('Programa Doutoral Eng. Ambiente', 1),
('Programa Doutural em Sistemas de Transporte', 1)]
```

Note-se que são demasiados cursos, sendo que muitos deles foram escritos com letras maiúsculas/minúsculas ou com siglas

Filtrar-se-á cada curso único somente uma vez

Procedimentos:

1. Filtrar os cursos por letra maiúscula, solucionando o problema dos cursos com letras maiúsculas/minúsculas
2. Somar as siglas aos respetivos cursos

In [54]:

```
# Pergunta 1.4
## 2° Round
### Calcular todas as categorias únicas e com frequências superiores a 2
frequencia = []
curso = []
for x in df["1.4 - Curso a frequentar"]:
    #### Filtrar os cursos por letra maiúscula, solucionando o problema dos cursos com letras maiúsculas/minúsculas
    y = x.upper()
    if (y not in curso):
        mask = df["1.4 - Curso a frequentar"] == x
        df1 = df[mask]
        df2 = df1["1.4 - Curso a frequentar"]
        df3 = df2.count()
        if (df3 > 2):
            frequencia.append(df3)
            curso.append(y)
    else:
```

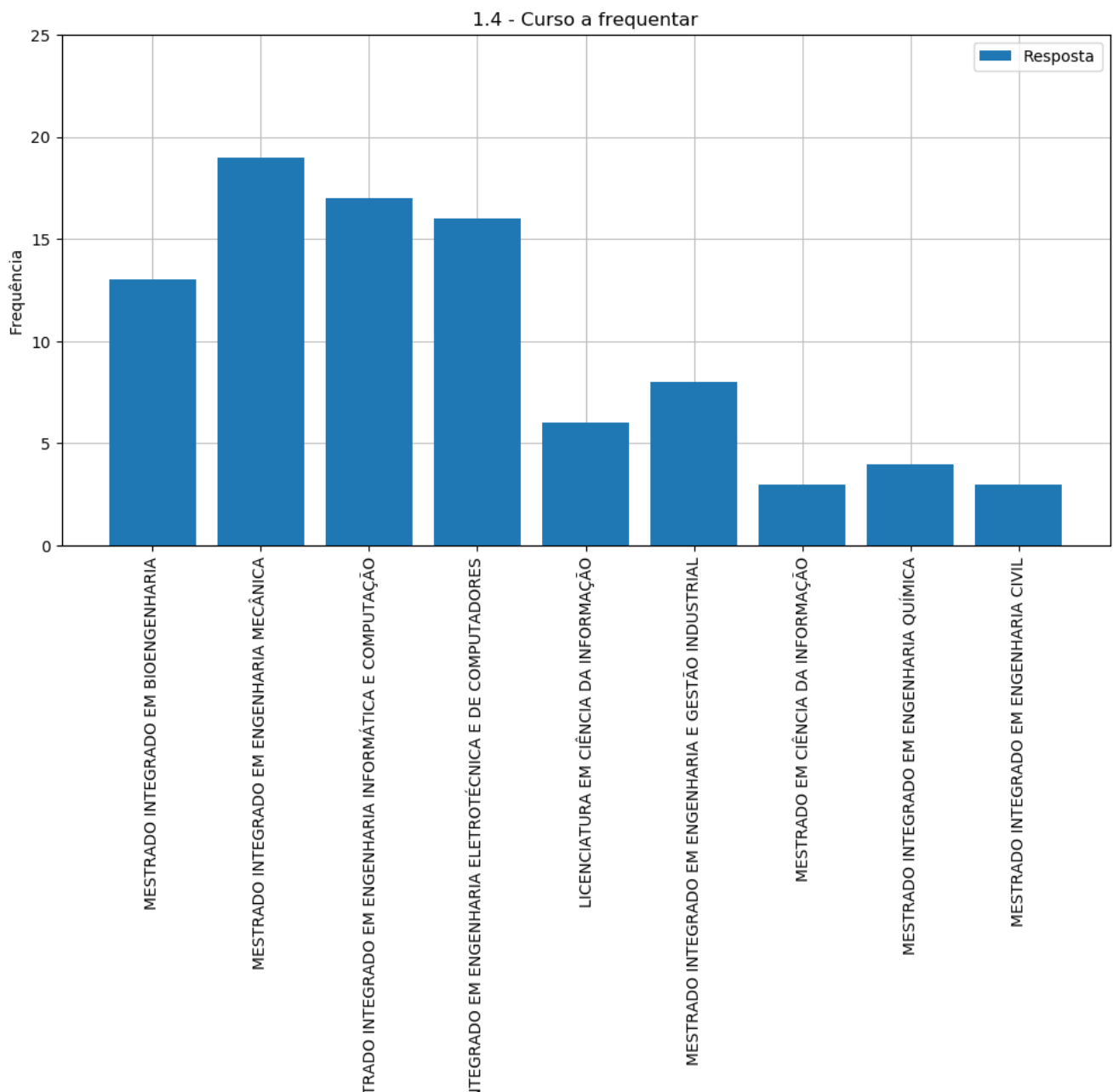
```

        pass
    else:
        pass

#### Somar as siglas aos respectivos cursos
MIEIC, MIEEC, MIEEC1, MIEM = np.array(frequencia[0]), np.array(frequencia[1]), \
np.array(frequencia[6]), np.array(frequencia[-1])
array = np.array(frequencia)
array[4] += MIEIC; array[5] += MIEEC; array[5] += MIEEC1; array[3] += MIEM
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]:

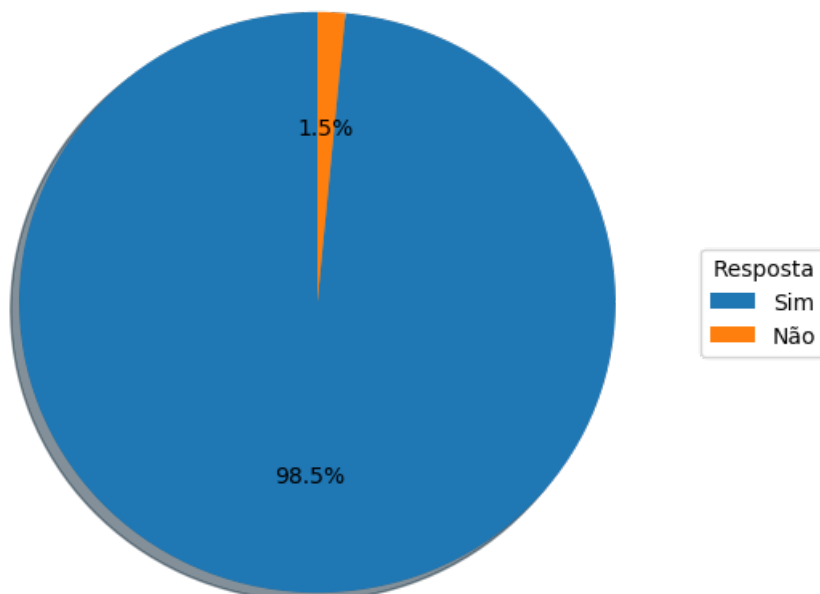
```
[('MESTRADO INTEGRADO EM BIOENGENHARIA', 13),  
 ('MESTRADO INTEGRADO EM ENGENHARIA MECÂNICA', 19),  
 ('MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO', 17),  
 ('MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES', 16),  
 ('LICENCIATURA EM CIÊNCIA DA INFORMAÇÃO', 6),  
 ('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]:

```
# Pergunta 2.1  
## Calcular as categorias e suas frequências  
acesso = ["Sim", "Não"]  
frequencia = []  
for x in acesso:  
    mask = df["2.1 - Tem acesso à Internet em casa?"] == x  
    df1 = df[mask]  
    df2 = df1["2.1 - Tem acesso à Internet em casa?"]  
    df3 = df2.count()  
    frequencia.append(df3)  
  
## Visualizar a pergunta 2.1 num gráfico circular  
plt.figure(figsize = (6, 6), dpi = 100)  
plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")  
plt.title("2.1 - Tem acesso à Internet em casa?")  
plt.legend(acesso, title="Resposta", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))  
plt.show()  
resultado = list(zip(acesso, frequencia))  
resultado
```

2.1 - Tem acesso à Internet em casa?



Out[10]:

```
Out[10]:
```

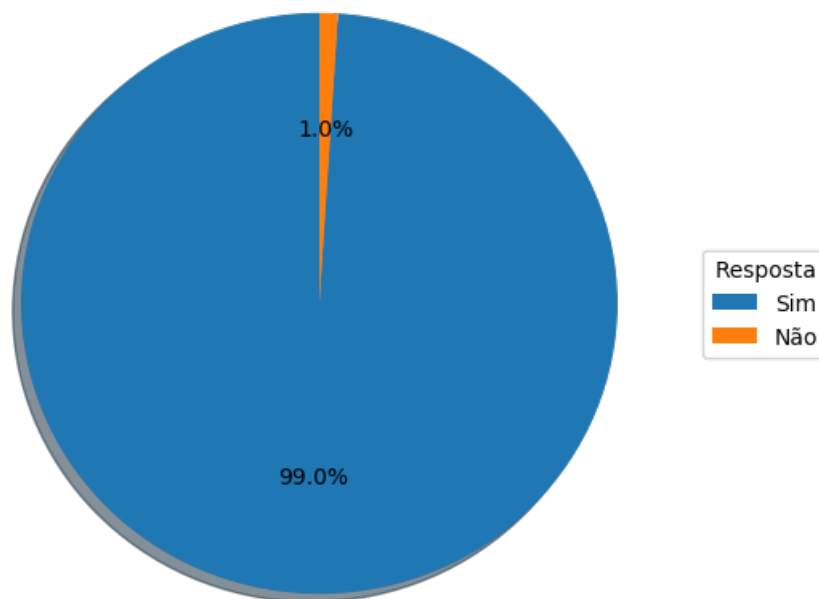
```
[('Sim', 197), ('Não', 3)]
```

```
In [11]:
```

```
# Pergunta 2.2
## Calcular as categorias e suas frequências
utilizacao = ["Sim", "Não"]
frequencia = []
for x in utilizacao:
    mask = df["2.2 - Costuma utilizar a Internet noutro lugar?"] == x
    df1 = df[mask]
    df2 = df1["2.2 - Costuma utilizar a Internet noutro lugar?"]
    df3 = df2.count()
    frequencia.append(df3)

## Visualizar a pergunta 2.2 num gráfico circular
plt.figure(figsize = (6, 6), dpi = 100)
plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")
plt.title("2.2 - Costuma utilizar a Internet noutro lugar?")
plt.legend(utilizacao, title="Resposta", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
plt.show()
resultado = list(zip(utilizacao, frequencia))
resultado
```

2.2 - Costuma utilizar a Internet noutro lugar?



```
Out[11]:
```

```
[('Sim', 198), ('Não', 2)]
```

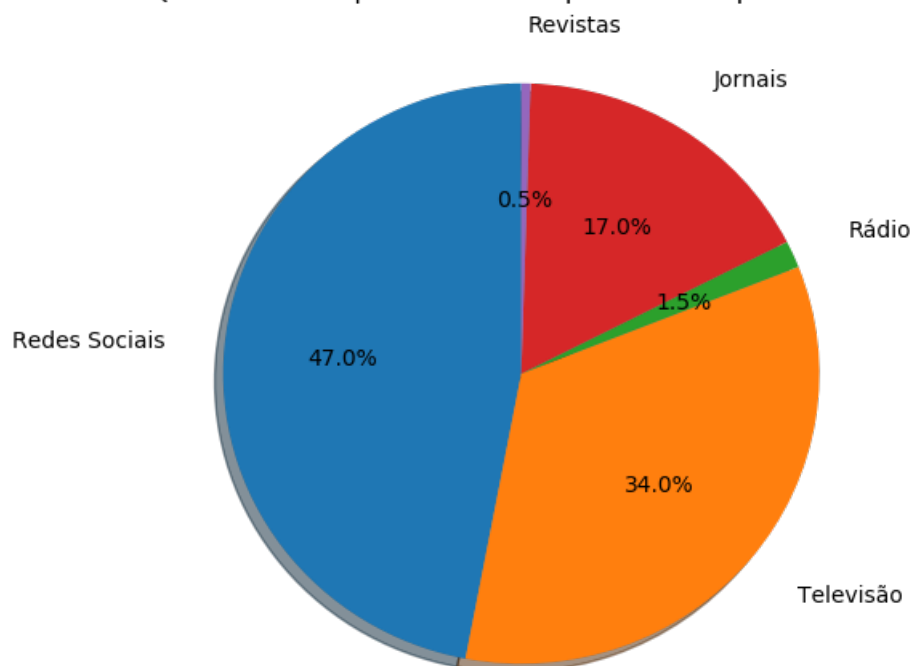
```
In [12]:
```

```
# Pergunta 2.3
## Calcular as categorias e suas frequências
meio = ["Redes Sociais", "Televisão", "Rádio", "Jornais", "Revistas"]
frequencia = []
for x in meio:
    mask = df["2.3 - Qual é o meio que mais utiliza para estar a par das notícias?"] == x
    df1 = df[mask]
    df2 = df1["2.3 - Qual é o meio que mais utiliza para estar a par das notícias?"]
    df3 = df2.count()
    frequencia.append(df3)

## Visualizar a pergunta 2.3 num gráfico circular
plt.figure(figsize = (6, 6), dpi = 100)
```

```
plt.pie(frequencia, labels = meio, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")
plt.title("2.3 - Qual é o meio que mais utiliza para estar a par das notícias?")
plt.show()
resultado = list(zip(meio, frequencia))
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)]
```

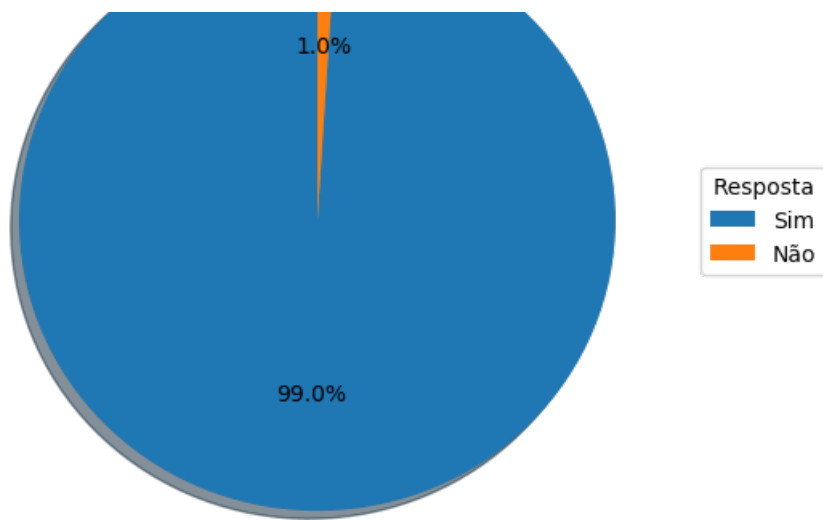
In [13]:

```
# Pergunta 2.4 (Pontuada)
## Calcular as categorias e suas frequências
conhecimento = ["Sim", "Não"]
frequencia = []
for x in conhecimento:
    mask = df['2.4 - Sabe o que significa "Fake News"?'] == x
    df1 = df[mask]
    df2 = df1['2.4 - Sabe o que significa "Fake News"?']
    df3 = df2.count()
    frequencia.append(df3)

## Visualizar a pergunta 2.4 num gráfico circular
plt.figure(figsize = (6, 6), dpi = 100)
plt.pie(frequencia, startangle = 90, shadow = True, labeldistance=1.2, autopct="%1.1f%%")
plt.title('2.4 - Sabe o que significa "Fake News"?')
plt.legend(conhecimento, title="Resposta", loc="center left", bbox_to_anchor=(1, 0, 0.5, 1))
plt.show()
resultado = list(zip(conhecimento, frequencia))
resultado
```

2.4 - Sabe o que significa "Fake News"?





Out[13]:

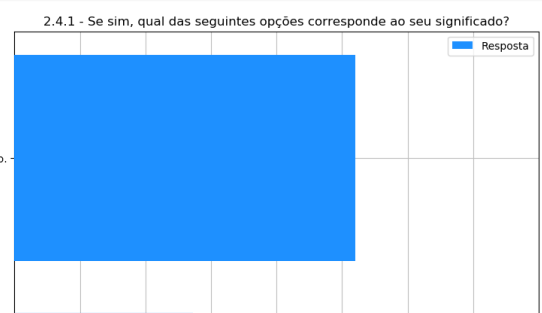
```
[('Sim', 198), ('Não', 2)]
```

In [14]:

```
# Pergunta 2.4.1 (Pontuada)
## Calcular todas as categorias e frequências
frequencia = []
resposta = []
for x in df["2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?"]:
    if (type(x) != str):
        pass
    else:
        if (x not in resposta):
            resposta.append(x)
            mask = df["2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?"]
            df1 = df[mask]
            df2 = df1["2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?"]
            df3 = df2.count()
            frequencia.append(df3)
        else:
            pass

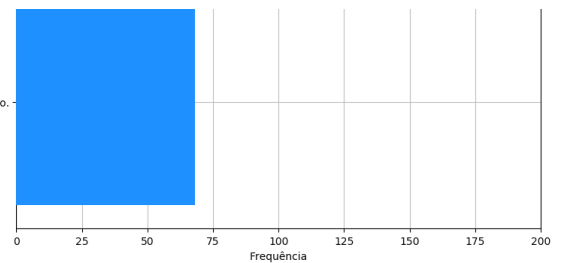
frequencia = tuple(frequencia)
y_pos = np.arange(len(resposta))

## Visualizar a pergunta 2.4.1 num gráfico de barras
plt.figure(figsize = (9, 9), dpi = 100)
plt.grid(color = '0.75')
plt.barh(y_pos, frequencia, zorder = 100, label = "Resposta", color = "dodgerblue")
plt.xlabel("Frequência")
plt.xlim(0, 200)
plt.yticks(y_pos, resposta)
plt.title("2.4.1 - Se sim, qual das seguintes opções corresponde ao seu significado?")
plt.legend()
plt.show()
resultado = list(zip(resposta, frequencia))
resultado
```



"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.



Out[14]:

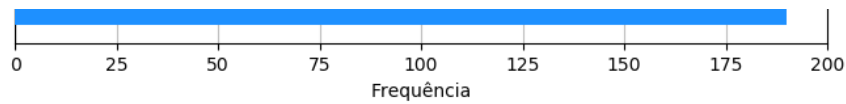
```
[("Fake News" são informações fabricadas que imitam propositalmente o conteúdo dos media na forma e no processo ou na intenção.',  
 68),  
 ("Fake News" são informações fabricadas propositalmente que imitam o conteúdo dos media na forma, mas não no seu processo ou na intenção.',  
 130)]
```

In [58]:

```
# Pergunta 2.4.2 (Pontuada)  
## Calcular todas as categorias e frequências  
frequencia = []  
resposta = []  
for x in df['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a \\  
divulgação do conhecimento.']:  
    if (type(x) != str):  
        pass  
    else:  
        if (x not in resposta):  
            resposta.append(x)  
            mask = df['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que \\  
prejudica a divulgação do conhecimento.'] == x  
            df1 = df[mask]  
            df2 = df1['2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que \\  
prejudica a divulgação do conhecimento.']  
            df3 = df2.count()  
            frequencia.append(df3)  
        else:  
            pass  
  
frequencia = tuple(frequencia)  
y_pos = np.arange(len(resposta))  
  
## Visualizar a pergunta 2.4.2 num gráfico de barras  
plt.figure(figsize = (8, 4), dpi = 100)  
plt.grid(color = '0.75')  
plt.barh(y_pos, frequencia, zorder = 100, label = "Resposta", color = "dodgerblue")  
plt.xlabel("Frequência")  
plt.xlim(0, 200)  
plt.yticks(y_pos, resposta)  
plt.title('2.4.2 - O tema das "Fake News" é um assunto de extrema importância e que prejudica a \\  
divulgação do conhecimento.')
```

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]:

```
# Adicione aqui sua Consumer Key
consumer_key = "k4XWQWP20STZzX0PojqBnTa15"

# Adicione aqui sua Consumer Secret
consumer_secret = "1sTl59sofqBJABDgqTkkeJH4L6eWSjcbcytifkMHsHGJC9ouQA"

# Adicione aqui seu Access Token
access_token = "930524504803115011-gCeP39s6K9grVnqNlFuxxFACu9rdlIO"

# Adicione aqui seu Access Token Secret
access_token_secret = "honuPSnY9HtYhygwQjIQfBagLAXhLrPfpKWnDqONMdF3H"

# Criando as chaves de autenticação
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

# Criando uma classe para capturar os stream de dados do Twitter e
# armazenar no MongoDB
class MyListener(StreamListener):
    def on_data(self, dados):
        tweet = json.loads(dados)
        created_at = tweet["created_at"]
        id_str = tweet["id_str"]
        text = tweet["text"]
        obj = {"created_at":created_at,"id_str":id_str,"text":text,}
        tweetind = col.insert_one(obj).inserted_id
        print (obj)
        return True

# Criando o objeto mylistener
mylistener = MyListener()

# Criando o objeto mystream
mystream = Stream(auth, listener = mylistener)
```

Preparando a Conexão com o MongoDB

In [74]:

```
# Importando do PyMongo o módulo MongoClient
from pymongo import MongoClient

# Criando a conexão ao MongoDB
client = MongoClient('localhost', 27017)

# Criando o banco de dados twitterdb
db = client.twitterdb

# Criando a collection "col"
col = db.tweets

# Criando uma lista de palavras chave para buscar nos Tweets
keywords = ["fake news", "Fake News", "FAKE NEWS"]
```

Coletando os Tweets

In [75]:

```
# Iniciando o filtro e gravando os tweets no MongoDB
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 News... https://t.co/3GH5LwPWOC', 'id': ObjectId('5ce6e3c95ed83f4e8fa76360')}\n{'created_at': 'Thu May 23 18:17:40 +0000 2019', 'id_str': '1131625255573872646', 'text': 'RT @realDonaldTrump: I was extremely calm yesterday with my meeting with Pelosi and Schumer, knowing that they would say I was raging, which...', 'id': ObjectId('5ce6e3c95ed83f4e8fa76361')}\n{'created_at': 'Thu May 23 18:17:40 +0000 2019', 'id_str': '1131625257293344768', 'text': 'RT @ChadByCC: 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')}\n{'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/CdxfsXWWZ', 'id': ObjectId('5ce6e3cc5ed83f4e8fa76363')}\n{'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 News... https://t.co/NlWPTCKK2L', 'id': ObjectId('5ce6e3cc5ed83f4e8fa76364')}\n{'created_at': 'Thu May 23 18:17:44 +0000 2019', 'id_str': '1131625275123539970', 'text': 'RT @jefphilips1: Until a popular publisher is docked and jailed for fake news and misinformation, you media people will not stop this irre...', 'id': ObjectId('5ce6e3cd5ed83f4e8fa76365')}\n{'created_at': 'Thu May 23 18:17:45 +0000 2019', 'id_str': '1131625279154216960', 'text': '@ambrosia_omG @7thletterja please tell me this is fake news', 'id': ObjectId('5ce6e3ce5ed83f4e8fa76366')}\n{'created_at': 'Thu May 23 18:17:47 +0000 2019', 'id_str': '1131625284447268864', 'text': 'RT @realDonaldTrump: I was extremely calm yesterday with my meeting with Pelosi and Schumer, knowing that they would say I was raging, which...', 'id': ObjectId('5ce6e3d05ed83f4e8fa76367')}\n{'created_at': 'Thu May 23 18:17:46 +0000 2019', 'id_str': '1131625283361148928', 'text': 'Pete Buttigieg: Donald Trump Used \'Status To Fake A Disability,\' Dodge Vietnam Draft | NBC News... https://t.co/D5l5tgXqv8', 'id': ObjectId('5ce6e3d05ed83f4e8fa76368')}\n{'created_at': 'Thu May 23 18:17:47 +0000 2019', 'id_str': '1131625286901161984', 'text': 'RT @realDonaldTrump: I was extremely calm yesterday with my meeting with Pelosi and Schumer, knowing that they would say I was raging, which...', 'id': ObjectId('5ce6e3d05ed83f4e8fa76369')}\n{'created_at': 'Thu May 23 18:17:49 +0000 2019', 'id_str': '1131625296212430849', 'text': '@bobjackk Tá vendo, @xicograziano, as fake news sobre as manifestações do dia 26/05 vêm de esquerdopatas parvos com... https://t.co/nptJ5FMGMs', 'id': ObjectId('5ce6e3d25ed83f4e8fa7636a')}\n{'created_at': 'Thu May 23 18:17:50 +0000 2019', 'id_str': '1131625298221568001', 'text': 'Pete Buttigieg: Donald Trump Used \'Status To Fake A Disability,\' Dodge Vietnam Draft | NBC News... https://t.co/bhW8hIJ9cV', 'id': ObjectId('5ce6e3d35ed83f4e8fa7636b')}\n{'created_at': 'Thu May 23 18:17:50 +0000 2019', 'id_str': '1131625300125732864', 'text': 'RT @o_antonista: Presidente da OAB bateu boca com advogados que questionaram evento sobre fake news com Toffoli e Lewandowski. https://t.co/...', 'id': ObjectId('5ce6e3d35ed83f4e8fa7636c')}\n{'created_at': 'Thu May 23 18:17:51 +0000 2019', 'id_str': '1131625302248054785', 'text': '@funder I just got a tweet on my phone to answer yes to support trump against FAKE NEWS and the liberal media...how... https://t.co/9f4B9nYu4Z', 'id': ObjectId('5ce6e3d45ed83f4e8fa7636d')}\n{'created_at': 'Thu May 23 18:17:51 +0000 2019', 'id_str': '1131625302764019712', 'text': 'RT @lunarcollins: FAKE NEWS... EU COMO LEITORA DIGO QUE SE FIZER ISSO COM UM LIVRO MEU O PRÓXIMO EVENTO QUE A PESSOA VAI PARTICIPAR VAI SER...', 'id': ObjectId('5ce6e3d45ed83f4e8fa7636e')}\n{'created_at': 'Thu May 23 18:17:52 +0000 2019', 'id_str': '1131625306249412608', 'text': 'RT @realDonaldTrump: I was extremely calm yesterday with my meeting with Pelosi and Schumer, knowing that they would say I was raging, which...', 'id': ObjectId('5ce6e3d55ed83f4e8fa7636f')}\n{'created_at': 'Thu May 23 18:17:52 +0000 2019', 'id_str': '1131625307109318656', 'text': 'RT @senacionalista: Harvard lança doutorado em fake news para aliados de Bolsonaro https://t.co/L5MPuneNFX', 'id': ObjectId('5ce6e3d55ed83f4e8fa76370')}
```

```
-----\nWantReadError                                Traceback (most recent call last)\n~\\Anaconda3\\lib\\site-packages\\urllib3\\contrib\\pyopenssl.py in recv_into(self, *args, **kwargs)\n    279         try:\n--> 280             return self.connection.recv_into(*args, **kwargs)\n    281         except OpenSSL.SSL.SysCallError as e:\n\n~\\Anaconda3\\lib\\site-packages\\OpenSSL\\SSL.py in recv_into(self, buffer, nbytes, flags)\n    1813         result = _lib.SSL_read(self._ssl, buf, nbytes)\n-> 1814         self._raise_ssl_error(self._ssl, result)\n    1815\n\n~\\Anaconda3\\lib\\site-packages\\OpenSSL\\SSL.py in _raise_ssl_error(self, ssl, result)\n    1613         if error == _lib.SSL_ERROR_WANT_READ:\n-> 1614             raise WantReadError()\n    1615         elif error == lib.SSL_ERROR_WANT_WRITE:
```

WantReadError:

During handling of the above exception, another exception occurred:

```
KeyboardInterrupt                                Traceback (most recent call last)
<ipython-input-75-6a9070444200> in <module>()
      1 # Iniciando o filtro e gravando os tweets no MongoDB
----> 2 mystream.filter(track=keywords)

~\Anaconda3\lib\site-packages\tweepy\streaming.py in filter(self, follow, track, is_async,
locations, stall_warnings, languages, encoding, filter_level)
    451     self.body['filter_level'] = filter_level.encode(encoding)
    452     self.session.params = {'delimited': 'length'}
--> 453     self._start(is_async)
    454
    455     def sitestream(self, follow, stall_warnings=False,

~\Anaconda3\lib\site-packages\tweepy\streaming.py in _start(self, is_async)
    366         self._thread.start()
    367     else:
--> 368         self._run()
    369
    370     def on_closed(self, resp):

~\Anaconda3\lib\site-packages\tweepy\streaming.py in _run(self)
    267         self.snooze_time = self.snooze_time_step
    268         self.listener.on_connect()
--> 269         self._read_loop(resp)
    270     except (Timeout, ssl.SSLError) as exc:
    271         # This is still necessary, as a SSLError can actually be

~\Anaconda3\lib\site-packages\tweepy\streaming.py in _read_loop(self, resp)
    317         length = 0
    318         while not resp.raw.closed:
--> 319             line = buf.read_line()
    320             stripped_line = line.strip() if line else line # line is sometimes None so
we need to check here
    321             if not stripped_line:

~\Anaconda3\lib\site-packages\tweepy\streaming.py in read_line(self, sep)
    179     else:
    180         start = len(self._buffer)
--> 181         self._buffer += self._stream.read(self._chunk_size)
    182         return six.b('')
    183

~\Anaconda3\lib\site-packages\urllib3\response.py in read(self, amt, decode_content,
cache_content)
    382     else:
    383         cache_content = False
--> 384         data = self._fp.read(amt)
    385         if amt != 0 and not data: # Platform-specific: Buggy versions of Python.
    386             # Close the connection when no data is returned

~\Anaconda3\lib\http\client.py in read(self, amt)
    447         # Amount is given, implement using readinto
    448         b = bytearray(amt)
--> 449         n = self.readinto(b)
    450         return memoryview(b)[:n].tobytes()
    451     else:

~\Anaconda3\lib\http\client.py in readinto(self, b)
    481
    482     if self.chunked:
--> 483         return self._readinto_chunked(b)
    484
    485     if self.length is not None:

~\Anaconda3\lib\http\client.py in _readinto_chunked(self, b)
    576     try:
    577         while True:
--> 578             chunk_left = self._get_chunk_left()
    579             if chunk_left is None:
    580                 return total_bytes

~\Anaconda3\lib\http\client.py in _get_chunk_left(self)
```

```

544         self._safe_read(2) # toss the CRLF at the end of the chunk
545     try:
--> 546         chunk_left = self._read_next_chunk_size()
547     except ValueError:
548         raise IncompleteRead(b'')

~\Anaconda3\lib\http\client.py in _read_next_chunk_size(self)
504     def _read_next_chunk_size(self):
505         # Read the next chunk size from the file
--> 506         line = self.fp.readline(_MAXLINE + 1)
507         if len(line) > _MAXLINE:
508             raise LineTooLong("chunk size")

~\Anaconda3\lib\socket.py in readinto(self, b)
584     while True:
585         try:
--> 586             return self._sock.recv_into(b)
587         except timeout:
588             self._timeout_occurred = True

~\Anaconda3\lib\site-packages\urllib3\contrib\pyopenssl.py in recv_into(self, *args, **kwargs)
290         raise
291     except OpenSSL.SSL.WantReadError:
--> 292         rd = util.wait_for_read(self.socket, self.socket.gettimeout())
293         if not rd:
294             raise timeout('The read operation timed out')

~\Anaconda3\lib\site-packages\urllib3\util\wait.py in wait_for_read(socks, timeout)
31     or optionally a single socket if passed in. Returns a list of
32     sockets that can be read from immediately. """
---> 33     return _wait_for_io_events(socks, EVENT_READ, timeout)
34
35

~\Anaconda3\lib\site-packages\urllib3\util\wait.py in _wait_for_io_events(socks, events, timeout)
24     selector.register(sock, events)
25     return [key[0].fileobj for key in
---> 26         selector.select(timeout) if key[1] & events]
27
28

~\Anaconda3\lib\site-packages\urllib3\util\selectors.py in select(self, timeout)
318     ready = []
319     r, w, _ = _syscall_wrapper(self._select, True, self._readers,
--> 320                               self._writers, timeout)
321     r = set(r)
322     w = set(w)

~\Anaconda3\lib\site-packages\urllib3\util\selectors.py in _syscall_wrapper(func, _, *args,
**kwargs)
62     and recalculate their timeouts. """
63     try:
---> 64         return func(*args, **kwargs)
65     except (OSError, IOError, select.error) as e:
66         errcode = None

~\Anaconda3\lib\site-packages\urllib3\util\selectors.py in _select(self, r, w, timeout)
308     def _select(self, r, w, timeout=None):
309         """ Wrapper for select.select because timeout is a positional arg """
--> 310         return select.select(r, w, [], timeout)
311
312     def select(self, timeout=None):

```

KeyboardInterrupt:

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]:

```
mystream.disconnect()
```

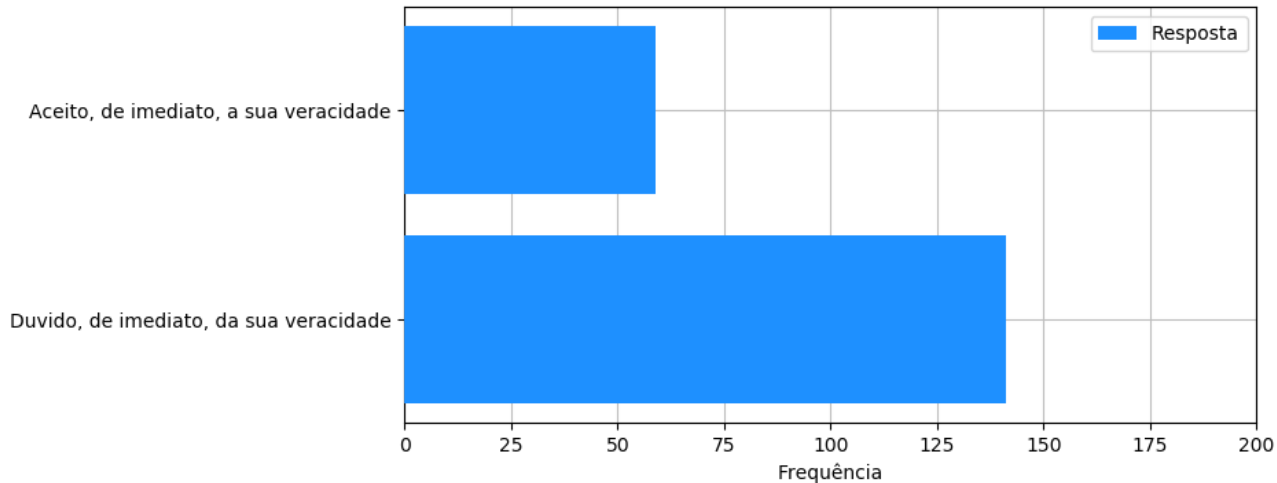
In [16]:

```
# Pergunta 2.5 (Pontuada)
## Calcular todas as categorias e frequências
frequencia = []
resposta = []
for x in df["2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?"]:
    if (x not in resposta):
        resposta.append(x)
        mask = df["2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?"
] == x
        df1 = df[mask]
        df2 = df1["2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?"
]
        df3 = df2.count()
        frequencia.append(df3)
    else:
        pass

frequencia = tuple(frequencia)
y_pos = np.arange(len(resposta))

## Visualizar a pergunta 2.5 num gráfico de barras
plt.figure(figsize = (8, 4), dpi = 100)
plt.grid(color = '0.75')
plt.barh(y_pos, frequencia, zorder = 100, label = "Resposta", color = "dodgerblue")
plt.xlabel("Frequência")
plt.xlim(0, 200)
plt.yticks(y_pos, resposta)
plt.title("2.5 - Na maioria dos casos, como se comporta quando se depara com uma notícia?")
plt.legend()
plt.show()
resultado = list(zip(resposta, frequencia))
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]:

```
# Pergunta 2.5.1
## 1º Round
### Calcular todas as categorias e suas frequências
frequencia = []
resposta = []
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\"?"]:
    if (type(x) != str):
        pass
    else:
        if (x not in curso):
```


Out[60]:

```
[('Algo fora do comum', 1),
 ('Por vezes o conteúdo retratado ', 1),
 ('Fontes pouco seguras ou títulos demasiado "clickbait"', 1),
 ('Título muito "atrativo", chocante.', 1),
 ('O extremo e desnecessário pormenor das notícias', 1),
 ('Conteúdo', 1),
 ('O singular de fake news não é fake new', 1),
 ('Erros ortográficos, qualidade das imagens, número anúncios na página', 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),
 ('Depende da origem da notícia. Do site/canal em questão. ', 1),
 ('A entidade que a concebeu', 1),
 ('A fonte da notícia', 1),
 ('"Difícilmente 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 Diário Hoje.', 1),
 ('Se não tiver uma fonte fidedigna', 1),
 ('Título mal estruturado ou com palavras que uma verdadeira notícia não teria.', 1).
```

Vários tipos de resposta

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.

Difícilmente uma notícia será 'fake new', porque o inglês não admite o singular da palavra 'news' na aceção em causa.

A fonte pelo qual estou a ler. Há determinadas fontes em que acredito de imediato, como por exemplo, o JN ou o Diário Hoje.

A forma como está escrita

Depende do grau de conhecimento que tenho do assunto envolvido

Se não fizer sentido, a.k.a X pessoa não tem razão para fazer Y

Estrutura

Quanto o título é demasiado "bombástico" ou quando existe um viés óbvio no texto da notícia (especialmente, no início).

A atenção continua que dou ao

Procuro saber quem escreve, por norma pode dizer-me se a fonte é credível ou não.

Os jornalistas podem apresentar o facto de várias formas, destacando certas valências que nem sempre

Título

O local onde

Tento perceber

Coesão na linguagem e sentido. Certamente também com um pouco de

Irregularidades na forma com

O que me leva a pensar ser uma fake new, é a

A ausência de

fonte (redes sociais) e penso, dentro dos meus domínios de conhecimento

Depende de notícia a notícia, mas avalio a notícia contra

Se for apenas uma fonte de publicação, contexto, hora da publicação. Se diversas fontes (creíveis) publicarem o mesmo conteúdo

```

-,,
('Website de origem da notícia', 1),
('A forma de apresentação da noticia e o conteudo é por vezes claramente duvidoso',
1),
('Normalmente, há sempre um facto que nao '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 titulo é demasiado "bombástico" ou quando existe um bias óbvio no texto da noticia (es
pecialmente em noticias 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 conteudo de noticia', 1),
('O site em que estou, vejo as noticias pelo celular', 1),
('Se parecer pouco plausível ou demasiado sensacionalista', 1),
('A quantidade de desinformação que existe atualmente', 1),
('O titulo muito sensacionalista, a fonte da notícia', 1),
('A atenção continua que dou aos jornais /O historico dos caos, informacoes, eventos informados n
os jornais',
1),
('Citações de pessoas são geralmente descontextualizadas, propositadamente.',
1),
('Demasiado sensacionalismo ou a proveniência da fonte', 1),
('Depende da notícia', 1),
('Procuo saber quem escreve, por norma pode dizer "redação" o que é duvidoso nos dias de hoje. D
epois procuro outras fontes.',
1),
('A falta de fontes, por exemplo.', 1),
('Titulo demasiado extravagante ', 1),
('A improbabilidade de tal situação ocorrer', 1),
('Os jornalistas podem apresentar o facto de várias formas, destacando certas valências que nem s
empre 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 noticias, pesquisas, estado da situação..', 1),
('fonte', 1),
('título não muito específico', 1),
('Fonte da noticia', 1),
('Inconsistências factuais', 1),
('Dependendo do quão credível é a noticia 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 divuga)', 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)

```



```

, 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 a verdade', 1),
('Irregularidades na forma como é apresentada e o facto de ser uma novidade incomum no dado conte
xto',
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 aco
mpanhada por "clickbait".',
1),
('congruência 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 quão aquilo parece r
eal. E então busco outras fontes que confio mais.',
1),
('Se a acho inverosímil ', 1),
('A minha intuição', 1),
('Principalmnte, 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, e
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 (credív
eis) publicarem o mesmo tipo de notícia é menos provável, ainda que não seja impossível, que não s
eja 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)]

```

Note-se que são demasiadas respostas, sendo que muitas delas representam o mesmo tema, mas que foram escritas de diferentes formas

Filtrar-se-á cada tema único somente uma vez

Procedimentos:

1. Filtrar as respostas por palavras-chave respetivos ao tema, tendo também em atenção as letras maiúsculas/minúsculas

In [61]:

```
# Pergunta 2.5.1
## 2° Round
### Calcular apenas as categorias mais utilizadas e suas frequências
frequencia = []
resposta = []

frequencia_f = []
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"?']:
    count_font = []
    if (type(x) == str):
        ### Filtrar as respostas por palavras-chave respetivos ao tema da "Fonte"
        if ("Fontes" in x) or ("fonte" in x) or ("Fontes" in x) or ("fontes" in x):
            if ("Fontes" not in resposta):
                resposta.append("Fontes")
                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
                df1 = df[mask]
                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
                df3 = df2.count()
                count_font.append(df3)
            else:
                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
                df1 = df[mask]
                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
                df3 = df2.count()
                count_font.append(df3)
        else:
            pass
    else:
        pass

    count_font = np.array(count_font).sum()
    frequencia_f.append(count_font)

frequencia_c = []
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"?']:
    count_cont = []
    if (type(x) == str):
        ### Filtrar as respostas por palavras-chave respetivos ao tema do "Conteúdo"
        if ("Conteúdo" in x) or ("conteúdo" in x) or ("Conteudo" in x) or ("conteudo" in x) or
("Conteúdos" in x) or \
        ("conteúdos" in x) or ("Conteudos" in x) or ("conteudos" in x):
            if ("Conteúdo" not in resposta):
                resposta.append("Conteúdo")
                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
                df1 = df[mask]
                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
                df3 = df2.count()
                count_cont.append(df3)
            else:
                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
                df1 = df[mask]
                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
                df3 = df2.count()
                count_cont.append(df3)
        else:
            pass
    else:
        pass

    count_cont = np.array(count_cont).sum()
    frequencia_c.append(count_cont)

frequencia_t = []
```

```

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"?']:
    count_tit = []
    if (type(x) == str):
        ### Filtrar as respostas por palavras-chave respetivos ao tema do "Título"
        if ("Título" in x) or ("título" in x) or ("Titulo" in x) or ("titulo" in x) or ("Títulos" in
x) or \
        ("títulos" in x) or ("Titulos" in x) or ("titulos" in x):
            if ("Título" not in resposta):
                resposta.append("Título")
                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
                df1 = df[mask]
                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
                df3 = df2.count()
                count_tit.append(df3)
            else:
                mask = df['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?'] == x
                df1 = df[mask]
                df2 = df1['2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que \
uma notícia é uma "Fake New"?']
                df3 = df2.count()
                count_tit.append(df3)
        else:
            pass
    else:
        pass

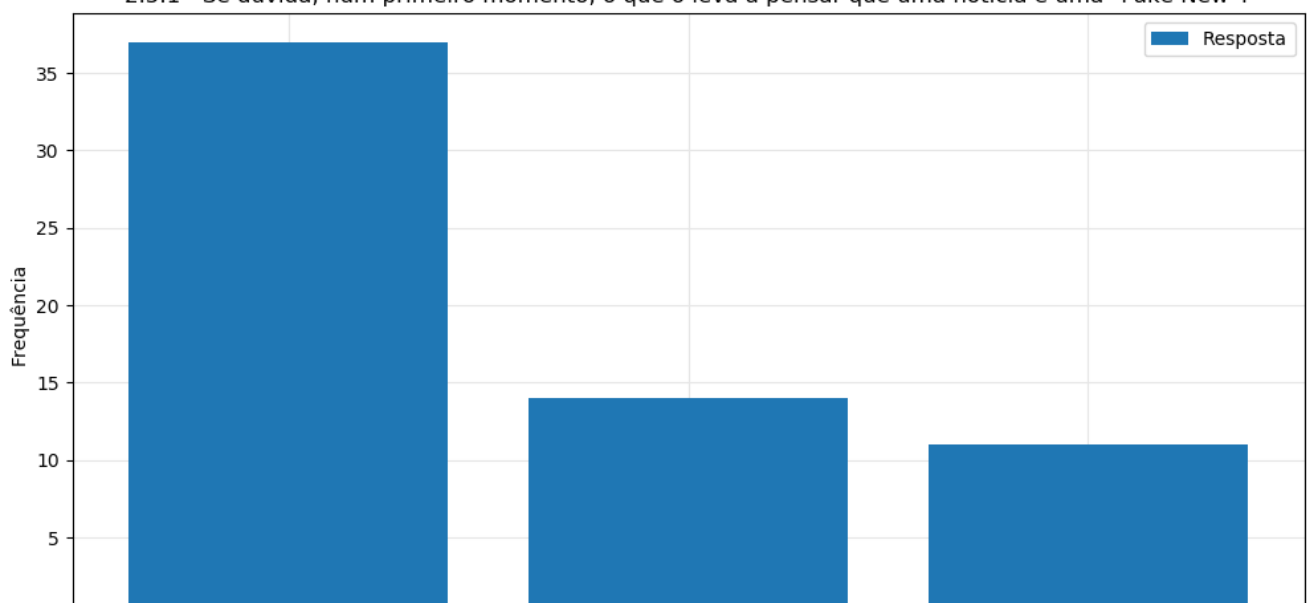
count_tit = np.array(count_tit).sum()
frequencia_t.append(count_tit)

frequencia_f = np.array(frequencia_f).sum()
frequencia_c = np.array(frequencia_c).sum()
frequencia_t = np.array(frequencia_t).sum()
frequencia.append(frequencia_f); frequencia.append(frequencia_c); frequencia.append(frequencia_t)

### Visualizar a pergunta 2.5.1 num gráfico de barras com os temas e suas respectivas frequências
plt.figure(figsize = (12, 6), dpi = 100)
plt.grid(color = '0.90')
plt.bar(resposta, frequencia, zorder = 100, label="Resposta")
plt.xlabel("Vários tipos de resposta")
plt.title('2.5.1 - Se duvida, num primeiro momento, o que o leva a pensar que uma notícia é uma "F
ake New"?')
plt.ylabel("Frequência")
plt.legend()
plt.show()
resultado = list(zip(resposta, frequencia))
resultado

```

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



0			
	Fontes	Conteúdo	Título
	Vários tipos de resposta		

Out[61]:

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

In [62]:

```
# Pergunta 2.6 (Pontuada)
## Calcular as categorias e suas frequências
resposta = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
titulo = ["Verifico a fonte", "Verifico outras fontes para confirmar",
          "Verifico o autor da notícia", "Verifico o aspeto da página",
          "Verifico a data", "Consulto uma pessoa"]

### Calcular as frequências das respostas para cada sub-título dentro do título
frequencia_1 = []
for x in resposta:
    mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico a fonte da notícia]"] == x
    df1 = df[mask]
    df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico a fonte da notícia]"]
    df3 = df2.count()
    frequencia_1.append(df3)

frequencia_2 = []
for x in resposta:
    mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico outras fontes para confirmar a notícia]"] == x
    df1 = df[mask]
    df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico outras fontes para confirmar a notícia]"]
    df3 = df2.count()
    frequencia_2.append(df3)

frequencia_3 = []
for x in resposta:
    mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico o autor da notícia]"] == x
    df1 = df[mask]
    df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico o autor da notícia]"]
    df3 = df2.count()
    frequencia_3.append(df3)

frequencia_4 = []
for x in resposta:
    mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico o aspeto da página (caso se trate de uma notícia online)]"] ==
x
    df1 = df[mask]
    df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico o aspeto da página (caso se trate de uma notícia online)]"]
    df3 = df2.count()
    frequencia_4.append(df3)

frequencia_5 = []
for x in resposta:
    mask = df["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico a data da notícia]"] == x
    df1 = df[mask]
    df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Verifico a data da notícia]"]
    df3 = df2.count()
    frequencia_5.append(df3)

frequencia_6 = []
for x in resposta:
    mask = df["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]"] == x
```

```

df1 = df[mask]
df2 = df1["2.6 - Com que frequência utiliza os seguintes critérios como forma \
de analisar uma notícia? [Consulta uma pessoa que esteja mais à vontade com o assunto da notícia]"
]

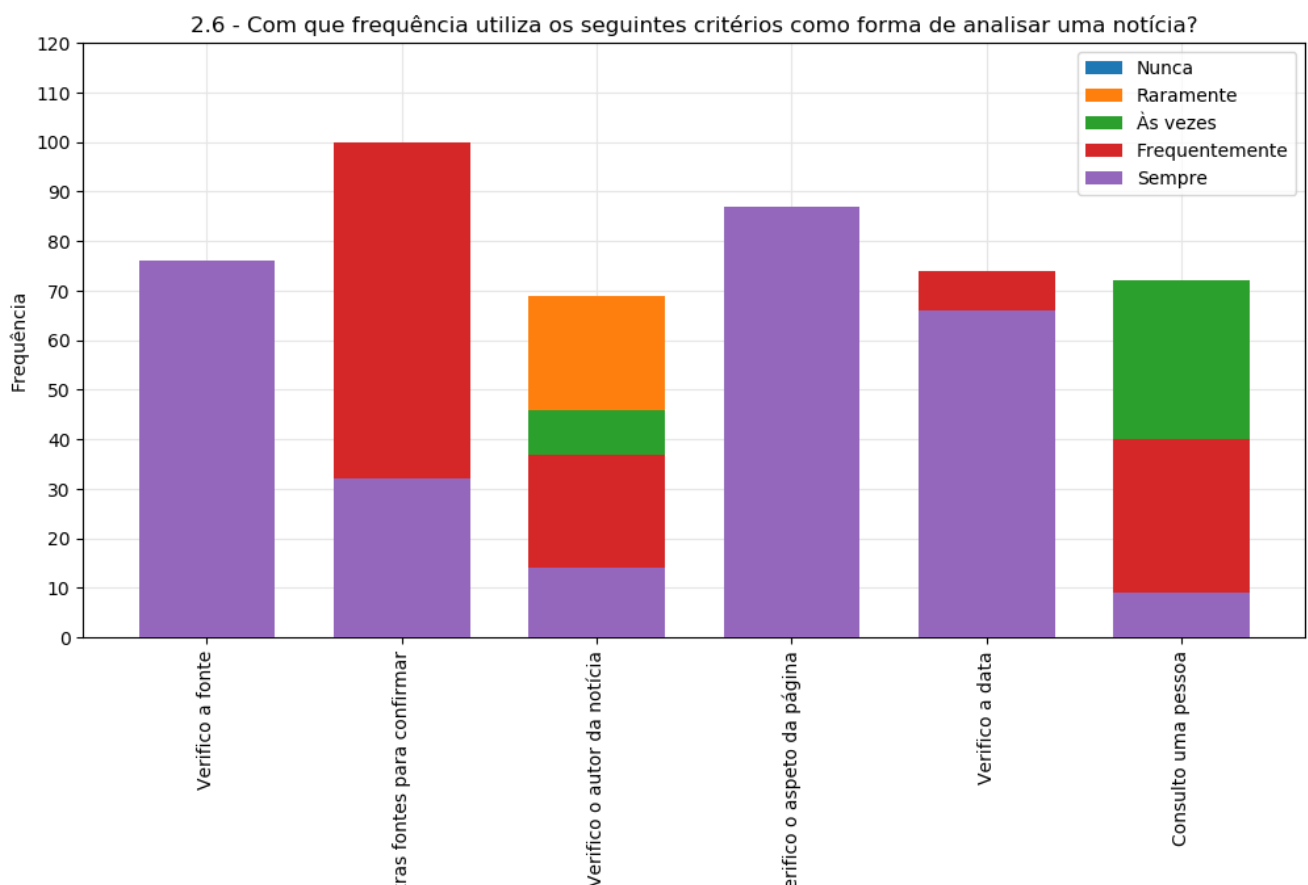
df3 = df2.count()
frequencia_6.append(df3)

## Visualizar a pergunta 2.6 num gráfico de barras empilhadas
N = 6
ind = np.arange(N)
width = 0.70

frequencia_a = [frequencia_1[0], frequencia_2[0], frequencia_3[0], frequencia_4[0], frequencia_5[0], frequencia_6[0]]
frequencia_b = [frequencia_1[1], frequencia_2[1], frequencia_3[1], frequencia_4[1], frequencia_5[1], frequencia_6[1]]
frequencia_c = [frequencia_1[2], frequencia_2[2], frequencia_3[2], frequencia_4[2], frequencia_5[2], frequencia_6[2]]
frequencia_d = [frequencia_1[3], frequencia_2[3], frequencia_3[3], frequencia_4[3], frequencia_5[3], frequencia_6[3]]
frequencia_e = [frequencia_1[4], frequencia_2[4], frequencia_3[4], frequencia_4[4], frequencia_5[4], frequencia_6[4]]

plt.figure(figsize = (12, 6), dpi = 100)
p1 = plt.bar(ind, frequencia_a, width, zorder = 100) # Apagar e inserir para ver a evolução dos gráficos
p2 = plt.bar(ind, frequencia_b, width, zorder = 100)
p3 = plt.bar(ind, frequencia_c, width, zorder = 100)
p4 = plt.bar(ind, frequencia_d, width, zorder = 100)
p5 = plt.bar(ind, frequencia_e, width, zorder = 100)
plt.grid(color = '0.90')
plt.xlabel('Várias opções')
plt.title('2.6 - Com que frequência utiliza os seguintes critérios como forma de analisar uma notícia?')
plt.xticks(ind, (titulo[0], titulo[1], titulo[2], titulo[3], titulo[4], titulo[5]), rotation=90)
plt.yticks(np.arange(0, 130, 10))
plt.ylabel('Frequência')
plt.legend((p1[0], p2[0], p3[0], p4[0], p5[0]), (resposta[0], resposta[1], resposta[2], resposta[3], resposta[4]))
plt.show()
print(frequencia_1, frequencia_2, frequencia_3, frequencia_4, frequencia_5, frequencia_6)
print(titulo)

```



Várias opções

```
[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]:

```
# Perguntas 2.7 e 2.8 (Pontuada)
## Calcular todas as categorias e suas frequências
fonte = ["Motores de pesquisa (Google, entre outros)", "Bases de dados", "Pessoas"]

### Calcular a frequência de cada fonte para cada pergunta
frequencia_1 = []
for x in fonte:
    mask = df["2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma \
necessidade de informação geral de fácil acesso?"] == x
    df1 = df[mask]
    df2 = df1["2.7 - Qual a fonte de informação que mais utiliza na satisfação de uma \
necessidade de informação geral de fácil acesso?"]
    df3 = df2.count()
    frequencia_1.append(df3)

frequencia_2 = []
for x in fonte:
    mask = df["2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma \
necessidade de informação específica de difícil acesso?"] == x
    df1 = df[mask]
    df2 = df1["2.8 - Qual a fonte de informação que mais utiliza na satisfação de uma \
necessidade de informação específica de difícil acesso?"]
    df3 = df2.count()
    frequencia_2.append(df3)

fonte[0] = "Motores de pesquisa"

## Visualizar a pergunta 2.7 e 2.8 numa figura com dois gráficos de barras em paralelo
fig = plt.figure(figsize = (12, 6), dpi = 100)

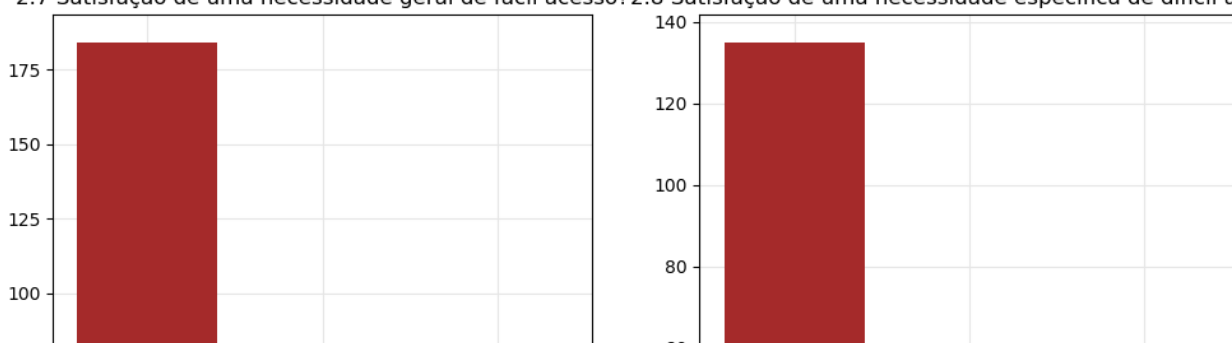
ax1 = fig.add_subplot(1,2,1)
ax1.bar(fonte, frequencia_1, color = "brown", zorder = 100)
ax1.set_title("2.7 Satisfação de uma necessidade geral de fácil acesso?")
ax1.grid(color = '0.90')

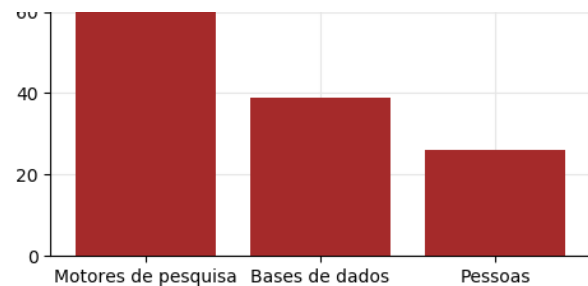
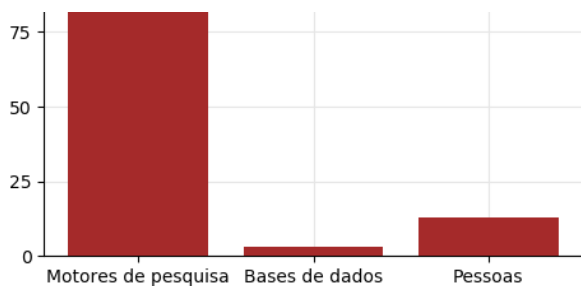
ax2 = fig.add_subplot(1,2,2)
ax2.bar(fonte, frequencia_2, color = "brown", zorder = 100)
ax2.set_title("2.8 Satisfação de uma necessidade específica de difícil acesso?")
ax2.grid(color = '0.90')

plt.show()

resultado1 = list(zip(fonte, frequencia_1)); resultado2 = list(zip(fonte, frequencia_2))
print(resultado1)
print(resultado2)
```

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]:

```
# Pergunta 2.9 (Pontuada)
## Calcular as categorias e suas frequências
resposta = ["Nunca", "Raramente", "Às vezes", "Frequentemente", "Sempre"]
frequencia = []
for x in resposta:
    mask = df["2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir \
as suas pesquisas?"] == x
    df1 = df[mask]
    df2 = df1["2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir \
as suas pesquisas?"]
    df3 = df2.count()
    frequencia.append(df3)

## Visualizar a pergunta 1.1 num gráfico circular (donut)
fig, ax = plt.subplots(figsize=(10, 20), subplot_kw=dict(aspect="equal"))

wedges, texts = ax.pie(frequencia, wedgeprops=dict(width=0.5), startangle=-40)

bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
kw = dict(xycoords='data', textcoords='data', arrowprops=dict(arrowstyle="-"),
          bbox=bbox_props, zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1)/2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = "angle,angleA=0,angleB={}".format(ang)
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(resposta[i], xy=(x, y), xytext=(1.35*np.sign(x), 1.4*y),
                horizontalalignment=horizontalalignment, **kw)

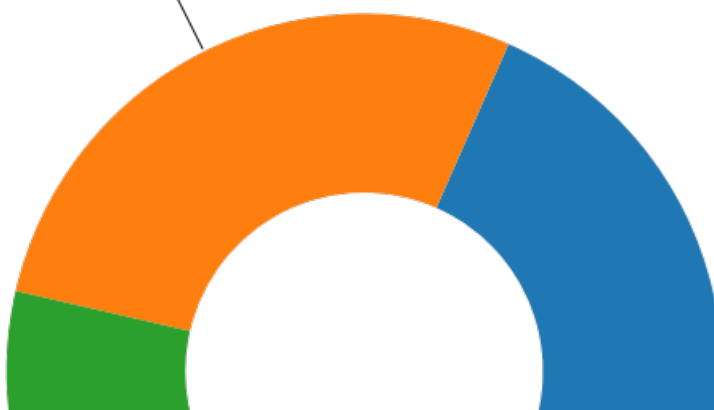
ax.set_title("2.9 - Costuma utilizar operadores booleanos (AND, NOT e OR) para restringir as suas p
esquisas?")

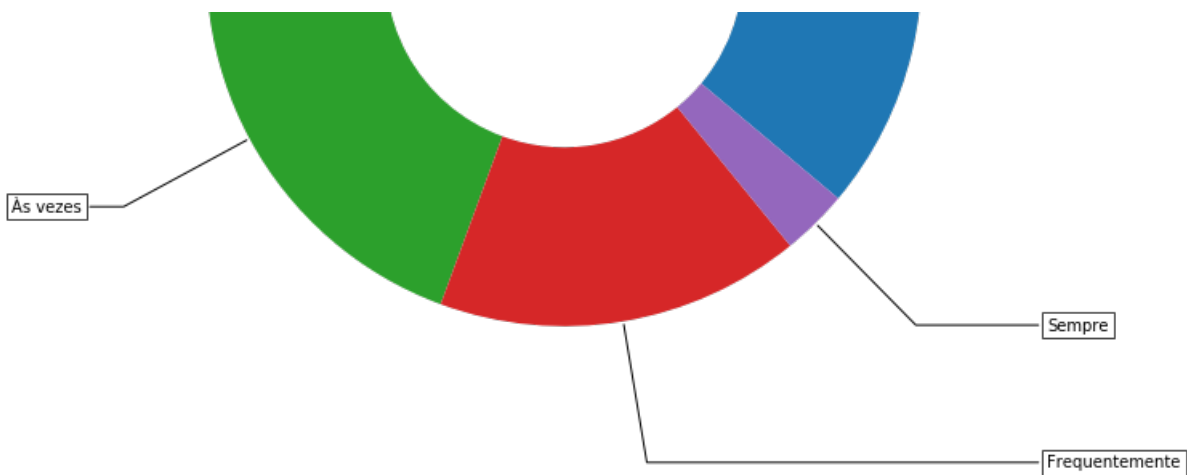
plt.show()
resultado = list(zip(resposta, frequencia))
print(resultado)
```

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

Raramente

Nunca





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

In [22]:

```
# Pergunta 2.10 e 2.10.1 (Pontuada)
## Calcular as categorias e suas frequências
posicao = ["1º", "2º", "3º", "4º", "5º"]
criterio = ["Fiabilidade", "Relevância", "Atualização", "Completa", "Explicita"]
ordem = "Ordem"
```

```
### Calcular a frequência de cada posição para cada pergunta (2.10)
```

```
frequencia_1 = []
for x in posicao:
    for y in df:
        if ("[Fiabilidade]" in y):
            mask = df[y] == x
            df1 = df[mask]
            df2 = df1[y]
            df3 = df2.count()
            frequencia_1.append(df3)
```

```
frequencia_2 = []
for x in posicao:
    for y in df:
        if ("[Relevância]" in y):
            mask = df[y] == x
            df1 = df[mask]
            df2 = df1[y]
            df3 = df2.count()
            frequencia_2.append(df3)
```

```
frequencia_3 = []
for x in posicao:
    for y in df:
        if ("[Atualização]" in y):
            mask = df[y] == x
            df1 = df[mask]
            df2 = df1[y]
            df3 = df2.count()
            frequencia_3.append(df3)
```

```
frequencia_4 = []
for x in posicao:
    for y in df:
        if ("[Completa]" in y):
            mask = df[y] == x
            df1 = df[mask]
            df2 = df1[y]
            df3 = df2.count()
            frequencia_4.append(df3)
```

```
frequencia_5 = []
for x in posicao:
    for y in df:
        if ("[Explicita]" in y):
            mask = df[y] == x
```



```

df1 = df[mask]
df2 = df1[y]
df3 = df2.count()
frequencia_5.append(df3)

### Calcular a frequência das pessoas que acertaram na ordem correta (2.10.1)
colunas_inicial = [x for x in df[list(df.columns[22:27])].values]
colunas_intermedia = [str(list(x)) for x in colunas_inicial]
df["2.10.1 - Ordem"] = colunas_intermedia

frequencia = []
mask = df["2.10.1 - Ordem"] == "['2º', '1º', '3º', '5º', '4º']"
df1 = df[mask]
df2 = df1["2.10.1 - Ordem"]
df3 = df2.count()
frequencia.append(df3)

## Visualizar as perguntas 2.10 numa figura com múltiplos gráficos respectivos às opções
print("2.10 - Ordene as seguintes opções em termos de relevância na busca de uma certa informação"
)
fig, ax = plt.subplots(2,3, figsize = (16, 8))

ax[0,0].bar(posicao, frequencia_1, color = "darkmagenta")
ax[0,0].set_title("Fiabilidade")

ax[0,1].bar(posicao, frequencia_2, color = "goldenrod")
ax[0,1].set_title("Relevância")

ax[0,2].bar(posicao, frequencia_3, color = "crimson")
ax[0,2].set_title("Atualização")

ax[1,0].bar(posicao, frequencia_4, color = "orangered")
ax[1,0].set_title("Completa")

ax[1,2].bar(posicao, frequencia_5, color = "mediumspringgreen")
ax[1,2].set_title("Explicita")

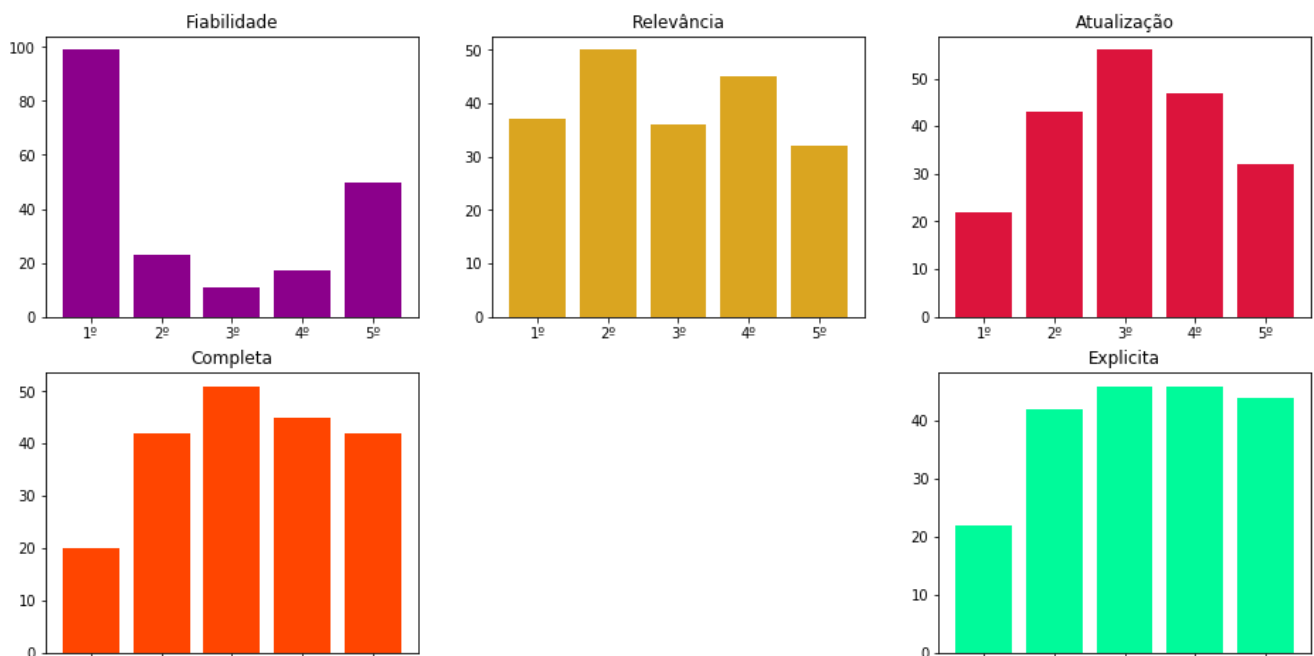
ax[1,1].remove()

plt.show()
print(frequencia_1, frequencia_2, frequencia_3, frequencia_4, frequencia_5, frequencia_6)
print(criterio)
print("-----")
-----")
cursos = ["LCI", "MIEM"]

print("Frequência das pessoas que acertaram na ordem correta:", str(frequencia))
resultado = list(zip(cursos, df2))
print(resultado)

```

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']")]
```

F. Análise exploratória de dados

1. Estatística Descritiva

Caracterização da amostra

In [67]:

```
# Pontuação Total
## Dados agrupados de uma amostra unidimensional quantitativa discreta (absoluta)
### Calcular a pontuação dos alunos
frequencia = []
for x in df["Pontuação total"]:
    num = int(x[0:2])
    frequencia.append(num)

### Calcular estatísticas gerais
frequencia_a = np.array(frequencia)
maxx = frequencia_a.max()
minn = frequencia_a.min()
gerais_nom = ["Máximo", "Mínimo"]
gerais_num = [maxx, minn]

mask1 = df["Pontuação total"] == "94.00 / 100"
df1 = df[mask1]
df2 = df1["1.4 - Curso a frequentar"].values[0]

mask2 = df["Pontuação total"] == "31.00 / 100"
df3 = df[mask2]
df4 = df3["1.4 - Curso a frequentar"].values[0]

### Calcular as estatísticas de localização
media = frequencia_a.mean()
moda = int(stats.mode(frequencia)[0])
mediana, q2 = np.median(frequencia), np.median(frequencia)
q1 = np.percentile(frequencia, 25)
q3 = np.percentile(frequencia, 75)
localizacao_nom = ["Média", "Moda", "Mediana", "Q1", "Q2", "Q3"]
localizacao_num = [media, moda, mediana, q1, q2, q3]

#### Verificar se existem Outliers
amplit_var = maxx - minn
amplit_quart = q3 - q1
BII = q1 - 1.5 * amplit_quart
BIS = q3 + 1.5 * amplit_quart
BEI = q1 - 3 * amplit_quart
BES = q3 + 3 * amplit_quart
barreira_nom = ["Barreira Interna Inferior", "Barreira Interna Superior",
                "Barreira Externa Inferior", "Barreira Externa Superior"]
barreira_num = [BII, BIS, BEI, BES]

### Calcular as estatísticas de dispersão
coef_var = stats.variation(frequencia)
desvio = frequencia_a.std()
variancia = frequencia_a.var()
dispersao_nom = ["Amplitude intervalo-variação", "Amplitude inter-quartis",
                "Coeficiente de dispersão", "Desvio", "Variância"]
dispersao_num = [amplit_var, amplit_quart, coef_var, desvio, variancia]

### Calcular as estatísticas de assimetria
assimetria = stats.skew(frequencia)
```

```

assimetria_nom = ["Coeficiente"]
assimetria_num = [assimetria]

### Calcular as estatísticas de achatamento
curtose = stats.kurtosis(frequencia)
achatamento_nom = ["Curtose"]
achatamento_num = [curtose]

## Visualizar a frequência de cada pontuação dos alunos através de um histograma e de um diagrama de caixa
fig = plt.figure()

axes1 = fig.add_axes([0.1, 0.1, 2, 2]) # eixos da figura principal
axes2 = fig.add_axes([0.2, 1.5, 0.6, 0.5]) # eixos da figura secundária

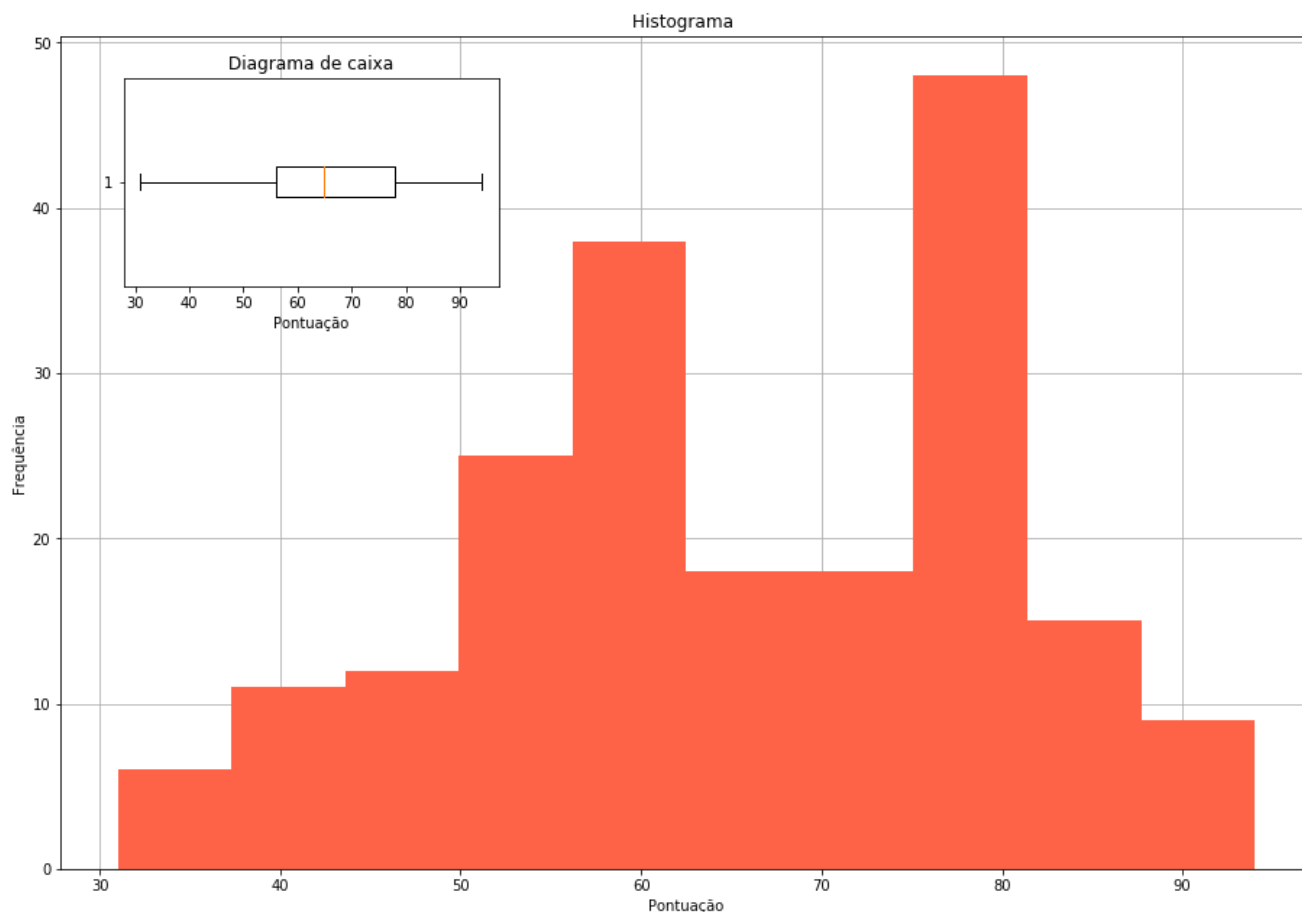
axes1.hist(frequencia, color="tomato", zorder=100)
axes1.grid()
axes1.set_xlabel('Pontuação')
axes1.set_ylabel('Frequência')
axes1.set_title('Histograma ')

axes2.boxplot(frequencia, vert=False)
axes2.set_xlabel('Pontuação')
axes2.set_title('Diagrama de caixa')

plt.show()

print("-----Estatísticas Gerais-----\n")
print("\n")
resultado = list(zip(gerais_nom, gerais_num))
print(resultado)
print(df2, df4)
print("\n")
print("-----Estatísticas de Localização-----\n")
print("\n")
resultado1 = list(zip(localizacao_nom, localizacao_num))
print(resultado1)
print("\n")
print("-----Estatísticas de Dispersão-----\n")
print("\n")
resultado2 = list(zip(dispersao_nom, dispersao_num))
print(resultado2)
print("\n")
print("Nota: Dispersão baixa no intervalo dos quartis em relação ao intervalo da amostra e dispersão \
baixa no intervalo da amostra")
print("\n")
resultado2a = list(zip(barreira_nom, barreira_num))
print(resultado2a)
print("\n")
print("Nota: Não existem outliers moderados, muito menos severos, visto que nenhum dos valores dos \
dados assume um valor <23(BII) e >111(BIS)")
print("\n")
print("-----Estatísticas de Assimetria-----\n")
print("\n")
resultado3 = list(zip(assimetria_nom, assimetria_num))
print(resultado3)
print("\n")
print("Nota: Os dados seguem uma distribuição assimétrica negativa ou alongada à esquerda \
(Média < Mediana < Moda)")
print("\n")
print("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")
print("\n")
print("-----Estatísticas de Achatamento-----\n")
print("\n")
resultado4 = list(zip(achatamento_nom, achatamento_num))
print(resultado4)
print("\n")
print("Nota: Distribuição dos dados é mais achatada do que uma distribuição normal típica")

```



-----Estadísticas Gerais-----

[('Máximo', 94), ('Mínimo', 31)]

Mestrado integrado em engenharia civil Licenciatura em Engenharia Mecânica

-----Estadísticas de Localização-----

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

-----Estadísticas 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(BII) e >111(BIS)

-----Estadísticas 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

-----Estatísticas de Achatamento-----

```
[('Curtose', -0.7898222065455163)]
```

Nota: Distribuição dos dados é mais achatada do que uma distribuição normal típica

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

Utilizar-se-á a estatística inferencial

Procedimentos:

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
 - 1.2 Calcular os parâmetros para que se possa entender mais acerca desta população

2. Estatística Inferencial

Caracterização da população

In [68]:

```
# Pontuação Total
## Caracterizar a distribuição das estatísticas amostrais associadas à população
### Utilização do Teorema do Limite Central para garantir que a distribuição segue uma
Distribuição normal
amostras_quant = 100000
amostras_a_complt = 0
amostras_dimens = 50
medias_amostrais = []
while amostras_a_complt < amostras_quant:
    fre = np.random.choice(frequencia, amostras_dimens).mean()
    medias_amostrais.append(fre)
    amostras_a_complt += 1

### Calcular parâmetros gerais
frequencia_a = np.array(medias_amostrais)
maxx = frequencia_a.max()
minn = frequencia_a.min()
gerais_nom = ["Máximo", "Mínimo"]
gerais_num = [maxx, minn]

### Calcular as parâmetros de localização
media_medias = frequencia_a.mean()
moda_medias = int(stats.mode(medias_amostrais)[0])
mediana_medias, q2 = np.median(medias_amostrais), np.median(medias_amostrais)
q1 = np.percentile(medias_amostrais, 25)
q3 = np.percentile(medias_amostrais, 75)
localizacao_nom = ["Média das médias amostrais", "Moda das médias amostrais",
                    "Mediana das médias amostrais", "Q1", "Q2", "Q3"]
localizacao_num = [media_medias, moda_medias, mediana_medias, q1, q2, q3]

#### Verificar se existem outliers
amplit_var = maxx - minn
amplit_quant = q3 - q1
```

```

amplitud_quart = q3 - q1
BII = q1 - 1.5 * amplitud_quart
BIS = q3 + 1.5 * amplitud_quart
BEI = q1 - 3 * amplitud_quart
BES = q3 + 3 * amplitud_quart
barreira_nom = ["Barreira Interna Inferior", "Barreira Interna Superior",
                "Barreira Externa Inferior", "Barreira Externa Superior"]
barreira_num = [BII, BIS, BEI, BES]

### Calcular os parâmetros de dispersão
coef_var = stats.variation(medias_amostrais)
desvio_medias = frequencia_a.std()
variancia_medias = frequencia_a.var()
dispersao_nom = ["Amplitude intervalo-variação", "Amplitude inter-quartis", "Coeficiente de
dispersão",
                "Desvio padrão das médias amostrais", "Variância das médias amostrais"]
dispersao_num = [amplitud_var, amplitud_quart, coef_var, desvio_medias, variancia_medias]

### Calcular as parâmetros de assimetria
assimetria = stats.skew(medias_amostrais, bias=False)
assimetria_nom = ["Coeficiente"]
assimetria_num = [assimetria]

### Calcular as parâmetros de achatamento
curtose = stats.kurtosis(medias_amostrais)
achatamento_nom = ["Curtose"]
achatamento_num = [curtose]

## Visualizar a frequência de cada média através de um histograma e de um diagrama de caixa
fig = plt.figure()

axes1 = fig.add_axes([0.1, 0.1, 2, 2]) # eixos da figura principal
axes2 = fig.add_axes([0.2, 1.5, 0.6, 0.5]) # eixos da figura secundária
axes3 = fig.add_axes([1.5, 1.6, 0.4, 0.4]) # eixos da figura terciária

axes1.hist(medias_amostrais, color="magenta", zorder=100)
axes1.grid()
axes1.set_xlabel('Pontuação')
axes1.set_ylabel('Frequência')
axes1.set_title('Histograma')

axes2.boxplot(medias_amostrais, vert=False)
axes2.set_xlabel('Pontuação')
axes2.set_title('Diagrama de caixa')

axes3.hist(medias_amostrais, cumulative=True, color="mediumvioletred")
axes3.set_xlabel('Pontuação')
axes3.set_title('Diagrama de caixa')
axes3.set_ylabel('Frequência')
axes3.set_title('Polígono de Frequências Acumuladas')

plt.show()

print("-----Parâmetros Gerais-----\n")
print("\n")
resultado = list(zip(gerais_nom, gerais_num))
print(resultado)
print("\n")
print("-----Parâmetros de Localização-----\n")
print("\n")
resultado1 = list(zip(localizacao_nom, localizacao_num))
print(resultado1)
print("\n")
print("Nota: Como a Média das médias amostrais é igual ao Valor médio da população (média das médi
as \
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, respetivamente")
print("\n")
print("-----Parâmetros de Dispersão-----\n")
print("\n")
resultado2 = list(zip(dispersao_nom, dispersao_num))
print(resultado2)
print("\n")
print("Nota: Dispersão indica o grau de dispersão dos dados em relação ao valor médio da distribuição de

```

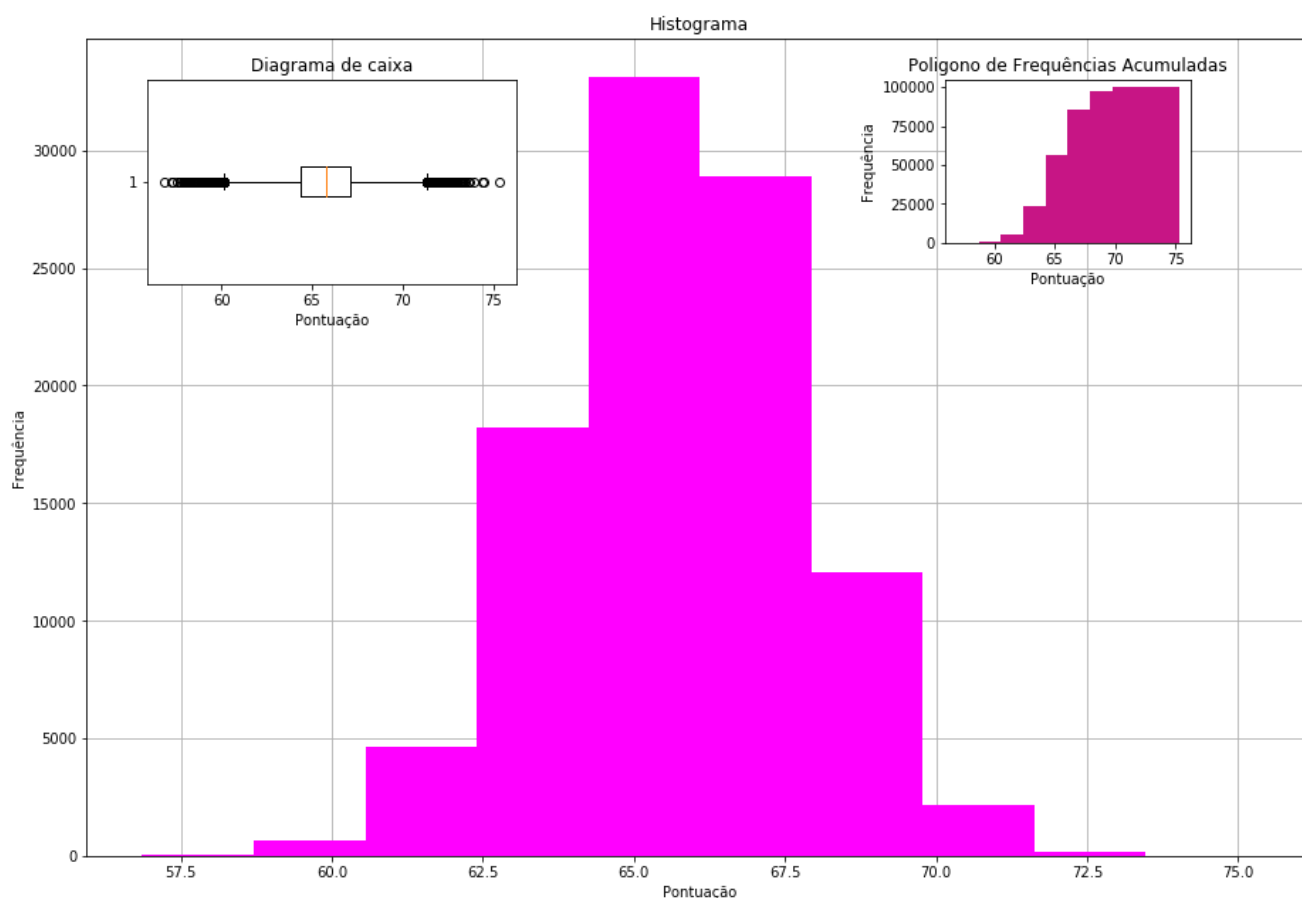
```

print("Nota: Dispersao baixa no intervalo dos quartis em relação ao intervalo da distribuição das \
médias amostrais e dispersão baixa no intervalo da distribuição das médias amostrais")
print("\n")
print("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 populaciona\
l)")
print("\n")
resultado2a = list(zip(barreira_nom, barreira_num))
print(resultado2a)
print("\n")
print("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 interes\
sa")
print("\n")
print("-----Parâmetros de Assimetria-----")
print("\n")
resultado3 = list(zip(assimetria_nom, assimetria_num))
print(resultado3)
print("\n")
print("Nota: Os dados seguem uma distribuição simétrica (Média = Mediana = Moda)")
print("\n")
print("-----Parâmetros de Achatamento-----")
print("\n")
resultado4 = list(zip(achatamento_nom, achatamento_num))
print(resultado4)
print("\n")
print("Nota: Distribuição dos dados segue uma distribuição normal típica")

fig = plt.figure(figsize = (12, 8), dpi = 100)

sns.distplot(medias_amostrais, color="magenta")

```



-----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, respetivamente

-----Parâmetros de Dispersão-----

```
[('Amplitude intervalo-variação', 18.459999999999994), ('Amplitude inter-quartis', 2.799999999999997), ('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', 55.94000000000001), ('Barreira Externa Superior', 75.53999999999999)]
```

Nota: Existem outliers moderados, visto que existem 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



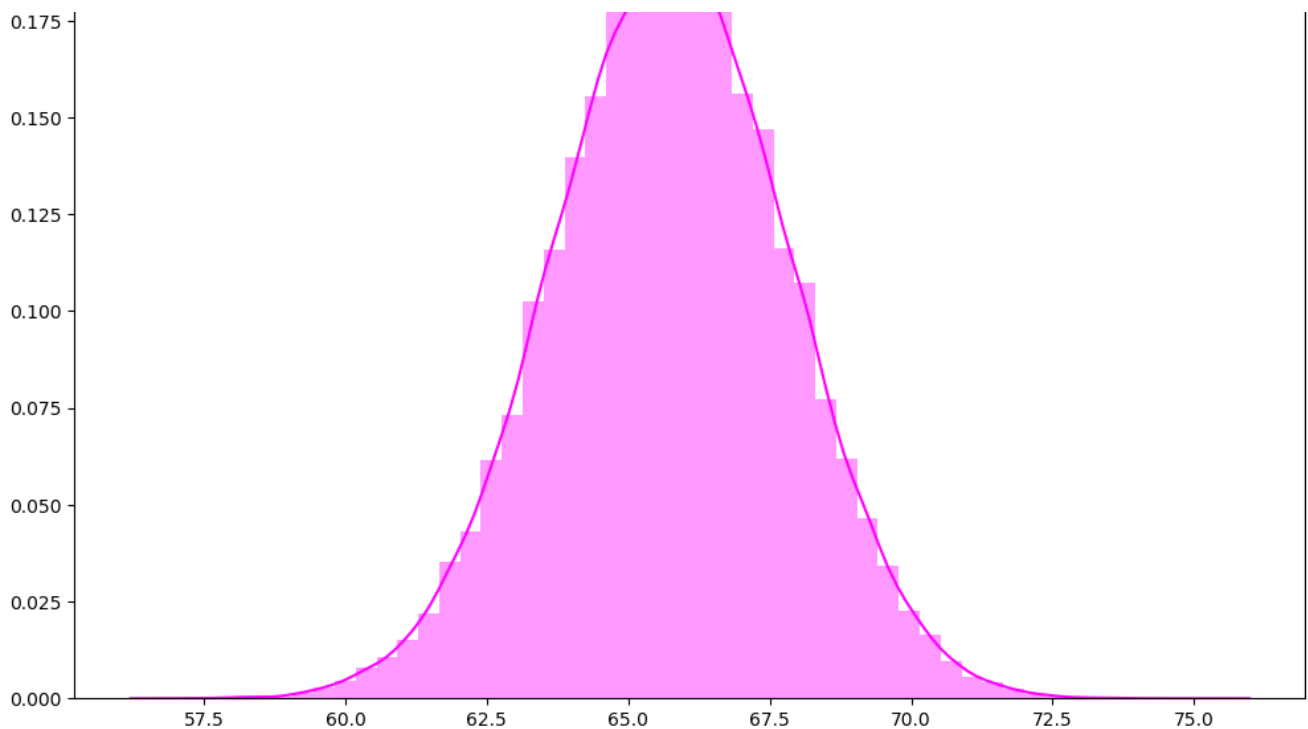
```
C:\Users\simao\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[68]:

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

0.200





Note-se que apesar de se saber o valor médio populacional, ainda falta saber a sua variância, assim como o seu desvio padrão.

Procedimentos:

1. Calcular a variância da população

R: Tratando-se de amostras aleatórias simples (infinitas e/ou com reposição) e sabendo-se que a variância das médias amostrais é igual à variância da população a dividir pela dimensão de cada amostra (variância das médias amostrais = variância da população/dimensão das amostras), o resultado da variância é de 212.748375277, isolando a variância ($4.254967505542857 = \text{variância populacional}/50$)

2. Calcular o desvio padrão da população

R: Tratando-se de amostras aleatórias simples (infinitas e/ou com reposição) e sabendo-se que o desvio padrão das médias amostrais é igual ao desvio padrão da população a dividir pela raiz quadrada da dimensão de cada amostra (desvio padrão das médias amostrais = desvio padrão da população/ raiz quadrada da dimensão das amostras), o resultado do desvio é de 14.5858964509, isolando o desvio padrão ($2.062757258026949 = \text{desvio padrão populacional}/\text{raiz quadrada de } 50$)

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]:

```
# Calcular o intervalo de confiança para o valor médio populacional
## Nível de confiança - 95%; Nível de significância - 5%; Dimensão da mostra - Grande; desvio populacional - Conhecido
### intervalo = [média da amostra +/- z * desvio padrão populacional/raiz quadrada da dimensão da amostra]
amostras_a_complt = 100
amostras_quant = 0
```

```

amostras_dimens = 50
z = 1.96 # Ir à tabela e encontrar o número em que a probabilidade é 0.0025 (nível de
significância/2 = 0.05/2),
#visto que a distribuição é simétrica
desvio_pop = 14.5858964509
intervalo_amostrais_media_final = []
while amostras_quant < amostras_a_complt:
    intervalo_amostrais_media = []
    x1 = np.random.choice(frequencia, amostras_dimens).mean()
    int1 = x1 - z * desvio_pop/np.sqrt(amostras_dimens)
    intervalo_amostrais_media.append(int1)
    x2 = np.random.choice(frequencia, amostras_dimens).mean()
    int2 = x2 + z * desvio_pop/np.sqrt(amostras_dimens)
    intervalo_amostrais_media.append(int2)
    intervalo_amostrais_media_final.append(intervalo_amostrais_media)
    amostras_quant += 1

# Calcular o intervalo de confiança para a variância populacional
## Nível de confiança - 95%; Nível de significância - 5%; Dimensão da mostra - Grande
amostras_a_complt = 100
amostras_quant = 0
amostras_dimens = 50
xn2_a = 71.42 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1 =
50-1)
#e com uma probabilidade a 0.025 (nível de significância/2 = 0.05/2)
xn2_b = 32.357 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1
= 50-1)
#e com uma probabilidade a 0.975 (1 - (nível de significância/2) = 1 - (0.05/2))
intervalo_amostrais_var_final = []
while amostras_quant < amostras_a_complt:
    intervalo_amostrais_var = []
    var1 = np.random.choice(frequencia, amostras_dimens).var()
    int1 = ((amostras_dimens - 1) * var1) / (xn2_a)
    intervalo_amostrais_var.append(int1)
    var2 = np.random.choice(frequencia, amostras_dimens).var()
    int2 = ((amostras_dimens - 1) * var1) / (xn2_b)
    intervalo_amostrais_var.append(int2)
    intervalo_amostrais_var_final.append(intervalo_amostrais_var)
    amostras_quant += 1

# Calcular o intervalo de confiança para o desvio padrão populacional
## Nível de confiança - 95%; Nível de significância - 5%; Dimensão da mostra - Grande
amostras_a_complt = 100
amostras_quant = 0
amostras_dimens = 50
xn2_a = 71.42 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1 =
50-1)
#e com uma probabilidade a 0.025 (nível de significância/2 = 0.05/2)
xn2_b = 32.357 # Ir à tabela e encontrar o número que tenha os graus de liberdade igual a 49 (N-1
= 50-1)
#e com uma probabilidade a 0.975 (1 - (nível de significância/2) = 1 - (0.05/2))
intervalo_amostrais_desv_final = []
while amostras_quant < amostras_a_complt:
    intervalo_amostrais_desv = []
    var1 = np.random.choice(frequencia, amostras_dimens).var()
    int1 = ((amostras_dimens - 1) * var1) / (xn2_a)
    intervalo_amostrais_desv.append(np.sqrt(int1))
    var2 = np.random.choice(frequencia, amostras_dimens).var()
    int2 = ((amostras_dimens - 1) * var1) / (xn2_b)
    intervalo_amostrais_desv.append(np.sqrt(int2))
    intervalo_amostrais_desv_final.append(intervalo_amostrais_desv)
    amostras_quant += 1

print("-----Intervalos de confiança para o valor médio populacional
---\
-----")
print("\n")
for x in intervalo_amostrais_media_final:
    print(x)
print("\n")
print("Intervalos que não cumprem os requisitos")
print("\n")
for x in intervalo_amostrais_media_final:
    if (x[0] > 65.73343383999999) or (x[1] < 65.73343383999999):
        print(x)
print("\n")
print("-----Intervalos de confiança para a variância populacional-

```

```

-----\
-----"
print("\n")
for y in intervalo_amostrais_var_final:
    print(y)
print("\n")
print("Intervalos que não cumprem os requisitos")
print("\n")
for x in intervalo_amostrais_var_final:
    if (x[0] > 212.748375277) or (x[1] < 212.748375277):
        print(x)
print("\n")
print("-----Intervalos de confiança para o desvio padrão populacional-----\
-----"
print("\n")
for w in intervalo_amostrais_desv_final:
    print(w)
print("\n")
print("Intervalos que não cumprem os requisitos")
print("\n")
for x in intervalo_amostrais_desv_final:
    if (x[0] > 14.5858964509) or (x[1] < 14.5858964509):
        print(x)
print("\n")
print("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")

```

```

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

```

```

[59.99699577427446, 68.26300422572554]
[63.11699577427445, 70.46300422572554]
[57.99699577427445, 70.56300422572554]
[62.876995774274455, 71.58300422572555]
[61.75699577427445, 67.58300422572555]
[62.17699577427445, 72.00300422572553]
[63.096995774274454, 70.40300422572554]
[60.57699577427446, 67.42300422572555]
[60.33699577427445, 72.64300422572553]
[61.05699577427445, 67.46300422572554]
[63.19699577427445, 66.94300422572555]
[60.61699577427445, 70.80300422572554]
[61.236995774274455, 71.76300422572554]
[62.99699577427446, 70.18300422572554]
[61.05699577427445, 69.80300422572554]
[59.836995774274456, 68.82300422572554]
[59.67699577427445, 70.08300422572555]
[65.27699577427445, 70.82300422572554]
[65.29699577427446, 68.16300422572554]
[58.71699577427445, 71.90300422572554]
[59.336995774274456, 72.20300422572554]
[62.016995774274456, 72.28300422572553]
[63.83699577427445, 69.72300422572555]
[63.096995774274454, 71.20300422572554]
[64.15699577427446, 70.72300422572555]
[64.75699577427446, 70.66300422572554]
[62.67699577427445, 71.76300422572554]
[62.31699577427445, 70.44300422572555]
[59.596995774274454, 62.78300422572555]
[61.63699577427446, 68.94300422572555]
[59.916995774274454, 67.12300422572554]
[61.75699577427445, 67.78300422572555]
[59.836995774274456, 70.70300422572554]
[58.57699577427445, 67.04300422572554]
[62.99699577427446, 71.84300422572554]
[61.31699577427445, 72.08300422572555]
[61.31699577427445, 72.40300422572554]
[60.21699577427446, 68.28300422572553]
[61.91699577427445, 62.803004225725545]
[60.156995774274456, 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]
[62.35699577427446, 71.94300422572555]
[62.99699577427446, 66.46300422572554]
[62.03699577427445, 69.78300422572553]
[58.79699577427446, 68.52300422572554]
[62.53699577427445, 72.56300422572554]
[57.776995774274454, 69.72300422572555]
[62.81699577427445, 68.20300422572554]
[59.836995774274456, 68.16300422572554]
[62.39699577427445, 67.46300422572554]
[62.49699577427446, 71.62300422572554]
[62.736995774274455, 67.78300422572555]
[60.016995774274456, 68.94300422572555]
[62.43699577427446, 66.24300422572554]
[62.55699577427445, 70.38300422572554]
[62.33699577427445, 73.78300422572553]
[62.33699577427445, 63.983004225725544]
[63.45699577427445, 68.24300422572554]
[63.47699577427445, 69.92300422572553]
[63.03699577427445, 66.54300422572554]
[63.11699577427445, 70.64300422572553]
[66.07699577427447, 69.82300422572554]
[64.99699577427447, 68.80300422572554]
[62.596995774274454, 66.98300422572554]
[59.736995774274455, 70.86300422572553]
[64.17699577427446, 69.08300422572555]
[60.91699577427445, 67.70300422572554]
[64.65699577427446, 70.64300422572553]
[62.876995774274455, 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]
[62.736995774274455, 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, 62.803004225725545]
[62.33699577427445, 63.983004225725544]
[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]
[135.02896107532902, 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, 18.07091688046916]
[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, 14.686665048082585]
[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



G. CONCLUSÃO

Em síntese, desde o enquadramento teórico - com a explicação da definição do termo "Fake News" - dos procedimentos metodológicos - com a utilização do Google analytics para a recolha dos dados, bem como com a utilização da linguagem Python para uma melhor manipulação - da apresentação e discussão dos resultados - com a sua visualização, tanto a nível de identificar os grupos sociais, como

investigar o comportamento e conhecimento dos estudantes - até à análise exploratória dos dados - com a caracterização da amostra recolhida através da estatística descritiva e a caracterização da população inferida através da estatística inferencial e descritiva - 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”. Após de se ter utilizado a estatística inferencial para a caracterização da população dos estudantes da Faculdade de Engenharia - uma vez que as estatísticas amostrais são variáveis, pelo que as mesmas não vão ser iguais daquelas outras aquando da recolha de uma outra amostra diferente - constatou-se que o valor médio 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 um valor médio de 65.7, aproximadamente. 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 depararam no caminho. De facto, a experiência foi benéfica, uma vez que permitiu ao grupo 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 - com um modelo de comportamento informacional (por exemplo, do Tom Wilson) - 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, o grupo sentiu que 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

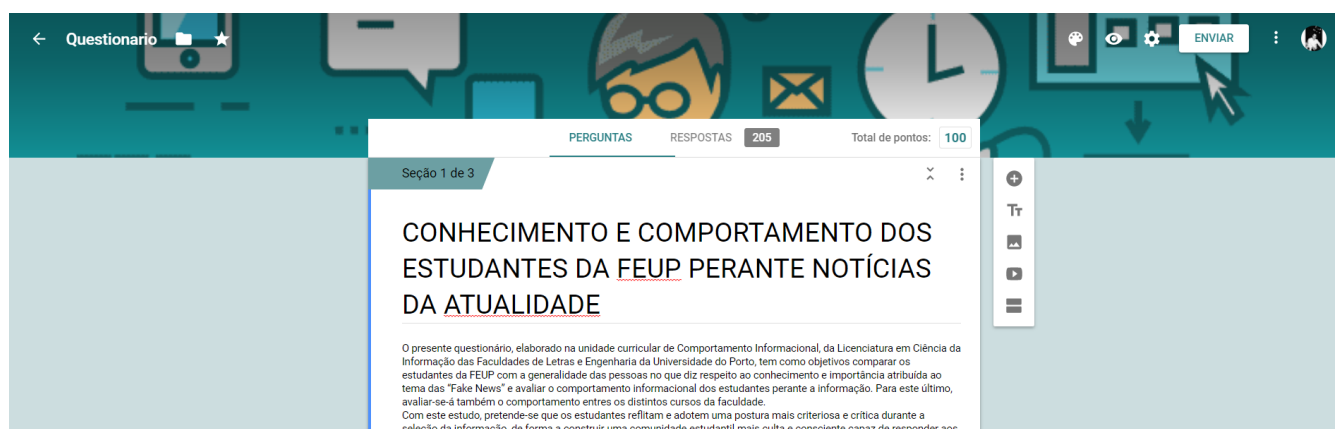
1. Academy, D. S. (2016). Python fundamentos para análise de dados. Retrieved May 22, 2019, from <https://www.datascienceacademy.com.br/start>
2. Community. (n.d.). Python documentation. Retrieved May 22, 2019, from <https://docs.python.org/3/>
3. Overflow, S. (2008). Stack Overflow - Where Developers Learn, Share, & Build Careers. Retrieved May 22, 2019, 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 enciclopédia livre. Retrieved 01:30, maio 1, 2019 from https://pt.wikipedia.org/w/index.php?title=Not%C3%ADcia_falsa&oldid=55000722.
7. Clickbait. (2018, junho 26). Wikipédia, a enciclopédia livre. Retrieved 22:00, junho 26, 2018 from <https://pt.wikipedia.org/w/index.php?title=Clickbait&oldid=52464537>.

I. ANEXOS

In [7]:

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

Out[7]:



constantas desatnos eminentes nas suas vidas pessoais, académicas e prtossionais.
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.

Após a seção 1 Continuar para a próxima seção

In [8]:

```
from IPython.display import Image  
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]:

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

Out[9]:

1.2 - Sexo *

☐ Masculino

☐ Feminino

1.3 - Habilitações Académicas *

☐ Ensino Secundário

☐ Licenciatura

☐ Mestrado

☐ Doutoramento

1.4 - Curso a frequentar *

Ex: Licenciatura em Ciência da Informação

Texto de resposta curta

2 - CONHECIMENTO E COMPORTAMENTO DOS ESTUDANTES

Descrição (opcional)

In [10]:

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

Out[10]:

2.1 - Tem acesso à Internet em casa? *

☐ Sim

☐ Não

2.2 - Costuma utilizar a Internet noutro 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]:

```
from IPython.display import Image
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]:

```
from IPython.display import Image
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"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Verifico outras fo...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Verifico o autor d...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Verifico o aspeto ...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Verifico a data da ...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Consulto uma pe...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

In [13]:

In [13]:

```
from IPython.display import Image
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]:

```
from IPython.display import Image
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"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Relevância	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Atualização	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Completa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Explicita	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<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

☐ Não

In [15]:

```
from IPython.display import Image
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