



## Subject Analyzer: Módulo de Classificação Automática - Plataforma E-go

ANA MANUEL SOARES DOS SANTOS

Outubro de 2021

# **Subject Analyzer: Módulo de Classificação Automática - Plataforma E-goI**

**Ana Manuel Soares dos Santos**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Engenharia de Software**

**Orientador: Dra. Isabel Praça  
Supervisor: Dr. Ivo Pereira**

**Júri:**  
Presidente:

Vogais:

*“The price of anything,  
is the amount of life you exchange for it.*

*Henry David Thoreau*

# Dedicatória

Dedico esta dissertação a todos aqueles que, direta ou indiretamente, se cruzaram no meu percurso e fizeram com que esta etapa fosse possível.



# Resumo

Com a constante evolução do mundo digital e tendo em consideração a situação pandémica atual, a utilização de *emails* como meio de comunicação apresenta um crescimento notório, assim como o *email marketing*, que é considerado o canal que gera o maior *Return On Investment* (ROI).

Com as caixas de *email* sobrecarregadas, os utilizadores tendem a priorizar certos *emails* em detrimento de outros com base em algumas decisões, sendo a leitura do assunto um fator decisivo na sua abertura. Assim, pretende-se criar uma ferramenta de suporte aos editores das campanhas de *email*, que analisa e classifica um determinado assunto, de forma a aumentar a sua taxa de aberturas e conseqüente sucesso da campanha. Com base no histórico de *emails* enviados pela plataforma E-goí, procura-se criar um sistema que prevê o impacto de um determinado assunto, classificando-o de zero a cinco estrelas.

Iniciado pela introdução ao projeto, na qual se identifica o problema existente e os objetivos propostos, e se apresenta a empresa na qual o projeto se enquadra, o presente documento visa contextualizar o leitor quanto aos conceitos mais relevantes relativos ao projeto e às soluções concorrentes existentes no mercado. De seguida, analisam-se técnicas de *Natural Language Processing* (NLP), assim como algoritmos de *Machine Learning* (ML) e *Deep Learning* (DL), de forma a concluir qual a melhor estratégia a adotar. Logo após, são apresentadas as *frameworks* de apoio à implementação mais utilizadas. Posteriormente, é elaborada a análise de valor de negócio que visa analisar e avaliar o propósito do projeto. Posto isto, é analisada e modelada a solução na qual se explica o domínio, os requisitos pretendidos e se apresentam duas soluções alternativas de *design*, seguidas da avaliação e escolha da mais adequada. Uma vez definida a melhor abordagem arquitetural, apresenta-se o *design* detalhado da solução, seguida da implementação da mesma. Por último, de modo a avaliar o potencial a nível funcional, a qualidade e usabilidade do sistema e a satisfação do utilizador, apresentam-se métodos de avaliação baseados em testes de *software*, testes de hipóteses, inquéritos e no modelo *Quantitative Evaluation Framework* (QEF).

**Palavras-chave:** Campanha de *Email*, Assunto, Classificação, Taxa de Aberturas, *Natural Language Processing*, *Machine Learning*, *Deep Learning*



# Abstract

Regarding the constant evolution of the digital world and considering the current pandemic situation, the emails usage as a communication channel presents notable growth, as well as email marketing, which is considered the channel that generates the highest ROI.

With mailboxes overloaded, users tend to prioritize certain emails over others based on some decisions, being the reading of the subject a decisive factor in its opening. Thus, it is intended to create a support tool for the email campaigns editors, which analyzes and classifies a certain subject, in order to increase its openings rate and consequent success of the campaign. Based on the history of emails sent by the E-goi platform, an attempt is made to create a system that predicts the impact of a given subject, classifying it from zero to five stars.

Initiated by the introduction of the project, which identifies the existing problem and the proposed objectives, and presents the company in which the project fits, this document aims to contextualize the reader regarding the most relevant concepts related to the project and the competing solutions existing in the market. Then, NLP techniques are analyzed as well as ML and DL algorithms in order to conclude which is the best strategy to adopt. Soon after, the most used implementation support frameworks are presented. Subsequently, the business value analysis is performed to analyze and evaluate the purpose of the project. Therefore, the solution is analyzed and modeled, explaining the domain and the requirements and presenting two alternative design solutions followed by the evaluation and choice of the most appropriate one. Once the best architectural approach is defined, the detailed design of the solution is presented, followed by its implementation. Finally, in order to assess the functional potential, quality and usability of the system and user satisfaction, evaluation methods based on software tests, hypothesis tests, surveys and the QEF model are presented.

**Keywords:** Email campaign, Subject, Classification, Openings Rate, Natural Language Processing, Machine Learning, Deep Learning



# Agradecimentos

É certo que nada é tão difícil quando temos boas pessoas por perto. É com base nesta premissa que quero começar por agradecer a todos os que me acompanharam. Um agradecimento muito especial à minha família e amigos, por serem tudo que preciso para me exceder pessoalmente e profissionalmente e por nunca me faltarem com nada.

Pretendo agradecer à Dra. Isabel Praça e ao Dr. Ivo Pereira, pela orientação ao longo de todo o desenvolvimento desta dissertação. Estou certa de que este caminho se tornou tão enriquecedor devido às pessoas, imprescindíveis, que dele fizeram parte.

Por último, quero endereçar os meus sinceros agradecimentos ao Instituto Superior de Engenharia do Porto, a todos os docentes que me acompanharam ao longo da minha formação académica e, ainda, à E-goi, por me integrar e permitir o meu desenvolvimento pessoal e profissional.



# Conteúdo

<b>Lista de Figuras</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xix</b>
<b>Lista de Código</b>	<b>xxii</b>
<b>Lista de Acrónimos</b>	<b>xxiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contexto . . . . .	1
1.2 Apresentação da Empresa . . . . .	1
1.3 Problema . . . . .	3
1.4 Objetivos . . . . .	3
1.5 Estrutura do Documento . . . . .	4
<b>2 Contextualização</b>	<b>5</b>
2.1 Campanha de <i>Email</i> . . . . .	5
2.1.1 Relatório . . . . .	6
2.2 <i>E-goï Campaign Analyzer</i> . . . . .	7
2.3 Testes A/B . . . . .	8
2.4 Soluções Concorrentes de Análise de Assunto . . . . .	9
2.4.1 <i>Mailchimp Subject Line Helper</i> . . . . .	9
2.4.2 <i>Send Check It</i> . . . . .	10
2.4.3 <i>Mizy</i> . . . . .	12
2.4.4 <i>Omnisend Email Subject Line Tester</i> . . . . .	13
2.4.5 <i>Touchstone</i> . . . . .	14
2.4.6 <i>IsItWP Email Subject Analyzer</i> . . . . .	15
2.4.7 Análise Comparativa . . . . .	16
<b>3 Estado da Arte</b>	<b>19</b>
3.1 <i>Natural Language Processing</i> . . . . .	19
3.2 <i>Machine Learning</i> . . . . .	20
3.2.1 Tipos de <i>Machine Learning</i> . . . . .	21
3.2.2 <i>Deep Learning</i> . . . . .	22
3.2.3 Métricas de Avaliação . . . . .	23
3.2.4 Algoritmos de Classificação de Texto . . . . .	25
<i>Naïve Bayes</i> . . . . .	25
<i>Decision Tree</i> . . . . .	25
<i>Support Vector Machine</i> . . . . .	26
<i>Artificial Neural Network</i> . . . . .	27
<i>Recurrent Neural Network</i> . . . . .	28

3.2.5	Conclusão . . . . .	29
3.3	Tecnologias . . . . .	30
3.3.1	<i>TensorFlow</i> . . . . .	31
3.3.2	<i>Keras</i> . . . . .	31
3.3.3	<i>PyTorch</i> . . . . .	31
3.3.4	<i>Caffe</i> . . . . .	31
3.3.5	Análise Comparativa . . . . .	32
<b>4</b>	<b>Análise de Valor</b>	<b>35</b>
4.1	Orientação . . . . .	35
4.1.1	Identificação de Oportunidade . . . . .	36
4.1.2	Análise de Oportunidade . . . . .	37
4.2	Valor da Solução . . . . .	40
4.2.1	Valor, Valor para o Cliente e Valor Percecionado . . . . .	40
4.2.2	Proposta de Valor . . . . .	41
4.3	Identificação e Análise Funcional . . . . .	42
4.4	Conceção de Alternativas . . . . .	43
4.5	Análise e Avaliação . . . . .	44
<b>5</b>	<b>Análise e Design da Solução</b>	<b>49</b>
5.1	Modelo de Domínio . . . . .	49
5.2	Requisitos Funcionais . . . . .	51
5.3	Requisitos Não Funcionais . . . . .	53
5.3.1	<i>Functionality</i> . . . . .	53
5.3.2	<i>Usability</i> . . . . .	53
5.3.3	<i>Reliability</i> . . . . .	53
5.3.4	<i>Performance</i> . . . . .	53
5.3.5	<i>Supportability</i> . . . . .	54
5.3.6	Restrições de Implementação . . . . .	54
5.3.7	Restrições de Interface . . . . .	54
5.4	Abordagens Possíveis e Escolha da Arquitetura . . . . .	54
5.4.1	<i>Subject Analyzer</i> com Segregação de Responsabilidades e Exposição na API Pública . . . . .	54
5.4.2	<i>Subject Analyzer</i> com Centralização de Responsabilidades . . . . .	55
5.4.3	Comparação de Alternativas . . . . .	56
5.5	Design Detalhado . . . . .	56
5.5.1	Diagrama de Componentes Detalhado do Sistema . . . . .	56
5.5.2	Diagrama de Implantação do Sistema . . . . .	58
5.5.3	Vistas de Cenários do Sistema . . . . .	59
5.5.4	Mock-ups . . . . .	61
<b>6</b>	<b>Implementação</b>	<b>63</b>
6.1	Extract, transform, load . . . . .	63
6.1.1	Abordagens Possíveis e Escolha . . . . .	64
Execução Sequencial . . . . .	65	
Execução Paralela com Recurso a Processos . . . . .	65	
Execução Paralela com Recurso a Processos e <i>Threads</i> . . . . .	66	
Comparação de alternativas . . . . .	67	
6.2	Pré-processamento dos dados . . . . .	67

6.3	Construção do modelo . . . . .	72
6.4	Application Programming Interface . . . . .	79
6.5	Integração com a plataforma E-goi . . . . .	81
<b>7</b>	<b>Experimentação e Avaliação</b>	<b>83</b>
7.1	Indicadores de Avaliação . . . . .	83
7.2	Definição de Hipóteses . . . . .	83
7.3	Metodologia de Avaliação . . . . .	84
7.3.1	Testes de <i>Software</i> . . . . .	84
7.3.2	<i>Quantitative Evaluation Framework</i> . . . . .	85
7.3.3	Inquérito de Usabilidade . . . . .	86
7.3.4	Inquérito de Satisfação . . . . .	86
7.3.5	Testes de Hipóteses . . . . .	86
7.4	Avaliação dos Resultados . . . . .	87
7.4.1	Testes de Software . . . . .	87
	Testes Unitários . . . . .	87
	Testes de Integração . . . . .	89
	Testes de Sistema . . . . .	89
	Testes de Aceitação . . . . .	91
7.4.2	Quantitative Evaluation Framework . . . . .	92
7.4.3	Inquérito de Usabilidade . . . . .	92
7.4.4	Inquérito de Satisfação . . . . .	94
<b>8</b>	<b>Conclusão</b>	<b>97</b>
8.1	Objetivos Alcançados . . . . .	97
8.2	Limitações . . . . .	99
8.3	Trabalho Futuro . . . . .	99
	<b>Bibliografia</b>	<b>101</b>
<b>A</b>	<b>Documentação Representational State Transfer (REST) Application Programming Interface (API)</b>	<b>111</b>
<b>B</b>	<b>Testes de Sistema</b>	<b>113</b>
<b>C</b>	<b>Testes de Aceitação</b>	<b>115</b>
<b>D</b>	<b>Indicadores de Avaliação do Modelo QEF</b>	<b>117</b>
<b>E</b>	<b>Inquérito de Usabilidade</b>	<b>121</b>
<b>F</b>	<b>Inquérito de Satisfação</b>	<b>123</b>
<b>G</b>	<b>Respostas Obtidas aos Inquéritos Realizados</b>	<b>125</b>
<b>H</b>	<b>Avaliação do Inquérito de Usabilidade</b>	<b>127</b>
<b>I</b>	<b>Avaliação do Inquérito de Satisfação</b>	<b>131</b>



# Lista de Figuras

1.1	Crescimento relativo ao número de trabalhadores da E-goi . . . . .	2
1.2	Crescimento relativo ao número de contas da E-goi . . . . .	2
2.1	Métricas gerais num relatório de <i>email</i> da E-goi . . . . .	7
2.2	Mapa do mundo consoante as aberturas . . . . .	7
2.3	Módulo de código do <i>Campaign Analyzer</i> . . . . .	8
2.4	Módulo de <i>spam</i> do <i>Campaign Analyzer</i> . . . . .	8
2.5	<i>Subject Line Helper</i> [18] . . . . .	9
2.6	<i>Performance</i> das últimas cinco campanhas de <i>email</i> enviadas [18] . . . . .	10
2.7	Classificação geral retornada pelo <i>Send Check It</i> [19] . . . . .	10
2.8	Análise de comprimento efetuada pelo <i>Send Check It</i> [19] . . . . .	10
2.9	Análise do nível de leitura efetuada pelo <i>Send Check It</i> [19] . . . . .	11
2.10	Análise à perceção efetuada pelo <i>Send Check It</i> [19] . . . . .	11
2.11	Análise de sentimento efetuada pelo <i>Send Check It</i> [19] . . . . .	11
2.12	Exemplo de integração do <i>Send Check It</i> num produto [22] . . . . .	12
2.13	<i>Mizy</i> integrada na plataforma [23] . . . . .	12
2.14	<i>Mizy</i> via <i>web</i> [23] . . . . .	12
2.15	Classificação geral do <i>Omnisend Subject Line Tester</i> [27] . . . . .	13
2.16	<i>Touchstone</i> [29] . . . . .	14
2.17	<i>IsItWP Email Subject Analyzer</i> [30] . . . . .	16
3.1	Categorias principais do ML [43] . . . . .	21
3.2	Tendência de pesquisas <i>online</i> pelo conceito DL [44] . . . . .	21
3.3	Camadas sucessivas de representações num modelo de classificação de dígito [46] . . . . .	23
3.4	Funcionamento de uma rede neuronal [46] . . . . .	23
3.5	<i>Confusion matrix</i> (adaptado de [47]) . . . . .	23
3.6	Rede <i>Naïve Bayes</i> (NB) [55] . . . . .	25
3.7	<i>Decision Tree</i> (DT) [55] . . . . .	26
3.8	Exemplo de criação de um hiperplano [55] . . . . .	27
3.9	Estrutura de uma <i>Artificial Neural Network</i> (ANN) [57] . . . . .	27
3.10	Estrutura básica de uma <i>Recurrent Neural Network</i> (RNN) [53] . . . . .	28
3.11	<i>Long Short-Term Memory</i> (LSTM) com unidades de controlo [53] . . . . .	29
3.12	Número de estudos publicados para cada área de NLP (adaptado de [58]) . . . . .	29
3.13	Desempenho de algoritmos ML e DL em função do número de amostras [59] . . . . .	30
3.14	Tendência de pesquisas <i>online</i> das <i>frameworks</i> [79] . . . . .	32
3.15	Número de artigos publicados na plataforma <i>Medium</i> relativos às <i>frameworks</i> [66] . . . . .	33
3.16	Número de empregos <i>online</i> relativos às <i>frameworks</i> [66] . . . . .	33
3.17	Número de artigos publicados na plataforma <i>arXiv</i> relativos às <i>frameworks</i> [66] . . . . .	34

3.18	Atividade no <i>GitHub</i> relativa às <i>frameworks</i> [66]	34
4.1	Processo de Análise de Valor (VA) [80]	35
4.2	Processo de inovação [82]	36
4.3	Matriz SWOT	37
4.4	Taxa de aberturas de campanhas de <i>email</i> [88]	39
4.5	Gráfico representativo da tendência de pesquisas <i>online</i> pelo conceito <i>email subject</i> [89]	39
4.6	Gráfico representativo da tendência de pesquisas <i>online</i> pelo conceito <i>open rate</i> [90]	40
4.7	Proposta de valor [93]	41
4.8	Fases do <i>Quality Function Deployment</i> (QFD) [95]	42
4.9	<i>House of Quality</i> (HOF)	43
4.10	Estrutura hierárquica do método <i>Analytic Hierarchy Process</i> (AHP) [98]	45
5.1	Modelo de domínio	49
5.2	Diagrama de casos de uso	51
5.3	<i>System Sequence Diagram</i> (SSD) do caso de uso Treinar Modelo de Classificação	52
5.4	SSD do caso de uso Analisar Assunto	52
5.5	Diagrama de componentes do sistema com segregação de responsabilidades e exposição na API pública	55
5.6	Diagrama de componentes do sistema com segregação de responsabilidades e exposição na API pública	55
5.7	Diagrama de componentes do sistema detalhado	57
5.8	Diagrama de implantação do sistema	58
5.9	Vista de cenários do caso de uso Treinar Modelo de Classificação	60
5.10	Vista de cenários do caso de uso Analisar Assunto	61
5.11	<i>Mock-up</i> do sistema de classificação de assunto	61
5.12	<i>Mock-up</i> da análise de restrições de boas práticas	62
5.13	<i>Mock-up</i> do histórico de consultas e seleção de assunto	62
6.1	Tabela <i>maxmailing_arquivo</i>	64
6.2	Tabela <i>egoi_message_email_count</i>	64
6.3	Estrutura da tabela <i>subject_data</i>	64
6.4	Execução do processo de forma sequencial	65
6.5	Execução do processo de forma paralela com recurso a processos	66
6.6	Execução do processo de forma paralela com recurso a processos e <i>threads</i>	67
6.7	Representação gráfica do pré-processamento	70
6.8	Divisão do conjunto de dados em dados de treino e dados validação	71
6.9	Distribuição dos dados	72
6.10	<i>Loss score</i> do modelo base	72
6.11	Arquitetura do modelo base	72
6.12	<i>Accuracy</i> do modelo base	73
6.13	<i>Loss score</i> do modelo base	73
6.14	<i>Accuracy</i> do modelo base com <i>Early Stopping</i>	74
6.15	<i>Loss score</i> do modelo base com <i>Early Stopping</i>	74
6.16	Arquitetura do modelo base com <i>embedding layer</i>	75
6.17	<i>Accuracy</i> do modelo base com <i>embedding layer</i>	75
6.18	<i>Loss score</i> do modelo base com <i>embedding layer</i>	75

6.19	<i>Accuracy</i> do modelo com a arquitetura LSTM . . . . .	76
6.20	<i>Loss score</i> do modelo com a arquitetura LSTM . . . . .	76
6.21	<i>Accuracy</i> do modelo com a arquitetura <i>Bidirectional LSTM</i> (BiLSTM) . . . . .	76
6.22	<i>Loss score</i> do modelo com a arquitetura BiLSTM . . . . .	76
6.23	Arquitetura do modelo com BiLSTM . . . . .	78
6.24	Documentação da API do <i>Subject Analyzer</i> . . . . .	80
6.25	Interface gráfica que representa a funcionalidade de análise de assunto de uma campanha . . . . .	82
7.1	Arquitetura do modelo <i>V-Model</i> (adaptado de [125]) . . . . .	84
7.2	Espaço de qualidade tridimensional [121] . . . . .	85
7.3	Modelo QEF . . . . .	92
A.1	Pedido GET responsável por devolver a classificação de um assunto . . . . .	111
A.2	Pedido POST responsável por treinar o modelo . . . . .	111
D.1	Indicadores de avaliação referentes ao fator classificação da dimensão funcionalidade . . . . .	117
D.2	Indicadores de avaliação referentes ao fator histórico da dimensão funcionalidade . . . . .	118
D.3	Indicadores de avaliação referentes ao fator integração da dimensão funcionalidade . . . . .	118
D.4	Indicadores de avaliação referentes ao fator adaptabilidade da dimensão manutenção . . . . .	118
D.5	Indicadores de avaliação referentes ao fator desempenho da dimensão eficiência . . . . .	119
D.6	Indicadores de avaliação referentes ao fator usabilidade da dimensão eficiência . . . . .	119
G.1	Respostas obtidas ao inquérito de usabilidade . . . . .	125
G.2	Respostas obtidas ao inquérito de satisfação . . . . .	126



# Lista de Tabelas

2.1	Análise comparativa de soluções . . . . .	17
3.1	Resumo das categorias principais de ML . . . . .	22
4.1	Matriz de comparação entre critérios . . . . .	45
4.2	Matriz normalizada de comparação entre os critérios e vetor de prioridades relativas . . . . .	46
4.3	Matriz de comparação de alternativas para o critério A . . . . .	46
4.4	Matriz de comparação de alternativas para o critério B . . . . .	46
4.5	Matriz de comparação de alternativas para o critério C . . . . .	47
4.6	Matriz de prioridades relativas das alternativas . . . . .	47
4.7	Prioridades compostas . . . . .	47
4.8	Matriz de índice aleatório . . . . .	48
4.9	Matriz de valores . . . . .	48
6.1	Códigos de personalização, descrição e identificador . . . . .	68
6.2	Representação parcial do índice de palavras . . . . .	71
6.3	Resultados do desempenho do modelo com a variante LSTM e BiLSTM . . . . .	77
6.4	Processo de <i>tuning</i> para os hiperparâmetros <i>optimizer</i> , <i>loss function</i> e <i>batch size</i> . . . . .	78
6.5	Resultados do desempenho do modelo de classificação . . . . .	79
7.1	Associação entre indicador e método de avaliação . . . . .	84
7.2	Teste de Sistema para Treinar Modelo de Classificação . . . . .	90
7.3	Teste de Sistema para Analisar Assunto . . . . .	90
7.4	Teste de Aceitação para Selecionar Assunto . . . . .	91
B.1	Teste de Sistema para Analisar Restrições de Boas Práticas . . . . .	113
B.2	Teste de Sistema para Visualizar Histórico de Consultas . . . . .	114
B.3	Teste de Sistema para Selecionar Assunto . . . . .	114
C.1	Teste de Aceitação para Treinar Modelo . . . . .	115



# Lista de Código

6.1	Implementação da função responsável pelas técnicas <i>Stop Word Removal</i> , <i>Lowercasing</i> e remoção da pontuação . . . . .	69
6.2	Código responsável pela criação do índice de palavras . . . . .	70
6.3	Código para a geração de sequências numéricas para cada elemento do conjunto de dados . . . . .	71
6.4	Implementação do <i>Early Stopping</i> . . . . .	73
6.5	Código responsável por correr a <i>cron-job</i> de treino do modelo mensalmente, de forma automática . . . . .	79
6.6	Função implementada para o pedido GET da classificação de assunto . . . . .	81
6.7	Função responsável por fazer a ligação entre o <i>Subject Analyzer</i> e o <i>Services</i> , via API . . . . .	82
7.1	Teste unitário implementado para a criação do índice de palavras . . . . .	88
7.2	Cobertura de testes do <i>Subject Analyzer</i> . . . . .	88
7.3	Teste à integração entre o <i>BO-UI</i> e o <i>Services</i> . . . . .	89
7.4	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 1 do inquérito de usabilidade . . . . .	93
7.5	Teste de <i>Wilcoxon</i> aplicado à questão 1 do inquérito de usabilidade . . . . .	94
7.6	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 5 do inquérito de usabilidade . . . . .	94
7.7	Teste de <i>Wilcoxon</i> aplicado à questão 5 do inquérito de usabilidade . . . . .	94
7.8	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 1 do inquérito de satisfação . . . . .	95
7.9	Teste de <i>Wilcoxon</i> aplicado à questão 1 do Inquérito de Satisfação . . . . .	95
H.1	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 2 do inquérito de usabilidade . . . . .	127
H.2	Teste de <i>Wilcoxon</i> aplicado à questão 2 do inquérito de usabilidade . . . . .	127
H.3	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 3 do inquérito de usabilidade . . . . .	128
H.4	Teste de <i>Wilcoxon</i> aplicado à questão 3 do inquérito de usabilidade . . . . .	128
H.5	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 4 do inquérito de usabilidade . . . . .	128
H.6	Teste de <i>Wilcoxon</i> aplicado à questão 4 do inquérito de usabilidade . . . . .	128
H.7	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 6 do inquérito de usabilidade . . . . .	129
H.8	Teste de <i>Wilcoxon</i> aplicado à questão 6 do inquérito de usabilidade . . . . .	129
H.9	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 7 do inquérito de usabilidade . . . . .	129
H.10	Teste de <i>Wilcoxon</i> aplicado à questão 7 do inquérito de usabilidade . . . . .	130
H.11	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 8 do inquérito de usabilidade . . . . .	130
H.12	Teste de <i>Wilcoxon</i> aplicado à questão 8 do inquérito de usabilidade . . . . .	130

I.1	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 2 do inquérito de satisfação . . . . .	131
I.2	Teste de <i>Wilcoxon</i> aplicado à questão 2 do inquérito de satisfação . . . . .	131
I.3	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 3 do inquérito de satisfação . . . . .	132
I.4	Teste de <i>Wilcoxon</i> aplicado à questão 3 do inquérito de satisfação . . . . .	132
I.5	Teste de normalidade <i>Shapiro-Wilk</i> aplicado à questão 4 do inquérito de satisfação . . . . .	132
I.6	Teste de <i>Wilcoxon</i> aplicado à questão 4 do inquérito de satisfação . . . . .	132

# Lista de Acrónimos

AAA	<i>Arrange-Act-Assert.</i>
AHP	<i>Analytic Hierarchy Process.</i>
AI	<i>Artificial Intelligence.</i>
ANN	<i>Artificial Neural Network.</i>
API	<i>Application Programming Interface.</i>
B2B	<i>Business-to-business.</i>
B2C	<i>Business-to-consumer.</i>
BBC	<i>British Broadcasting Corporation.</i>
BiLSTM	<i>Bidirectional LSTM.</i>
CSAT	<i>Customer Satisfaction Score.</i>
CSS	<i>Cascading Style Sheets.</i>
DL	<i>Deep Learning.</i>
DT	<i>Decision Tree.</i>
DTTL	<i>Deloitte Touche Tohmatsu Limited.</i>
ETL	<i>Extract, transform, load.</i>
FEE	<i>Fuzzy Front End.</i>
HOF	<i>House of Quality.</i>
HTML	<i>Hypertext Markup Language.</i>
IA	<i>Índice Aleatório.</i>
IC	<i>Índice Consistência.</i>
JSON	<i>JavaScript Object Notation.</i>
LSTM	<i>Long Short-Term Memory.</i>
ML	<i>Machine Learning.</i>
NB	<i>Naïve Bayes.</i>
NLP	<i>Natural Language Processing.</i>
NPD	<i>New Product Development.</i>
QEF	<i>Quantitative Evaluation Framework.</i>
QFD	<i>Quality Function Deployment.</i>

RC	Razão de Consistência.
REST	<i>Representational State Transfer.</i>
RNN	<i>Recurrent Neural Network.</i>
ROI	<i>Return On Investment.</i>
RSS	<i>Really Simple Syndication.</i>
SLA	<i>Service Level Agreement.</i>
SMS	<i>Short Message Service.</i>
SoC	<i>Separation Of Concerns.</i>
SSD	<i>System Sequence Diagram.</i>
SUS	<i>System Usability Scale.</i>
SVM	<i>Support Vector Machine.</i>
UML	<i>Unified Modeling Language.</i>
VA	Análise de Valor.

# Capítulo 1

## Introdução

Com a constante evolução digital, os meios de comunicação tradicionais, como os jornais e a rádio, têm vindo a ser comutados por meios digitais. Esta evolução impacta positivamente a vida das pessoas, reformando processos a nível da educação, da saúde, da produtividade, e, por último, da comunicação. Inerente à reformulação dos processos comunicativos, destaca-se a utilização de *emails*, que se considera ser o meio de comunicação principal no mundo empresarial atual [1].

### 1.1 Contexto

Na atualidade, a utilização de *emails* é vasta e encontra-se em constante evolução, pelo que se estima que, no presente ano, o número total de *emails* enviados e recebidos exceda os 306 bilhões e, ainda, que em 2024 haja um aumento de 18%. Outro fator revelador deste crescimento é o número de utilizadores de *email* que, em 2020, é aproximadamente de 4 bilhões e deverá crescer até 4,4 bilhões até ao final de 2024 [2].

No contexto imposto pela atual situação de pandemia, o uso de *emails* revela-se ainda mais significativo de um ponto de vista económico. Também neste ramo é notória a evolução, uma vez que o mercado global de *email marketing* é estimado em 6.34 bilhões de euros no ano de 2020 e projeta-se que, em 2027, se atinja os 15.15 bilhões, representando um crescimento de 13.3% em sete anos [3].

De acordo com VentureBeat [4], o *email marketing* é o canal mais eficaz para retenção, consciencialização e conversão de clientes, resultando no canal que gera o maior *Return On Investment* (ROI) para os profissionais de *marketing*.

Presentemente, existem diversos produtos de *email marketing* que disponibilizam a criação e gestão de campanhas de *email*, auxiliando os profissionais de *marketing* a alcançar os seus objetivos. O E-goí [5] é um exemplo destes produtos, que oferece ao cliente, entre outras funcionalidades, a criação de listas e segmentos associados, permitindo assim a segmentação do mercado, e a criação, gestão e automatização de diversas campanhas de *marketing*, entre elas, campanhas de *email*, campanhas de *Short Message Service* (SMS), campanhas de *Push*, campanhas de *WebPush* e campanhas de voz.

### 1.2 Apresentação da Empresa

A E-goí é uma empresa portuguesa de *marketing* digital, sediada em Matosinhos, fundada por Miguel Gonçalves. Fornece uma ferramenta de *marketing* multicanal automatizado,

prestando assim serviços para diferentes setores de atividade, nomeadamente comércio, ensino, entretenimento, imprensa e serviços públicos.

A empresa conta, atualmente, com mais de 100 trabalhadores (figura 1.1) e, ainda, elevados números de utilizadores, envios de *emails* e envios de SMS, que, à data de 2019 são, aproximadamente, de 400 mil (figura 1.2), 90 mil milhões e 600 milhões, respetivamente. Com mais de 15 anos de experiência, a E-goi é considerada uma referência no mundo do *marketing*, especialmente nos mercados português, brasileiro, espanhol, colombiano e da América Latina.

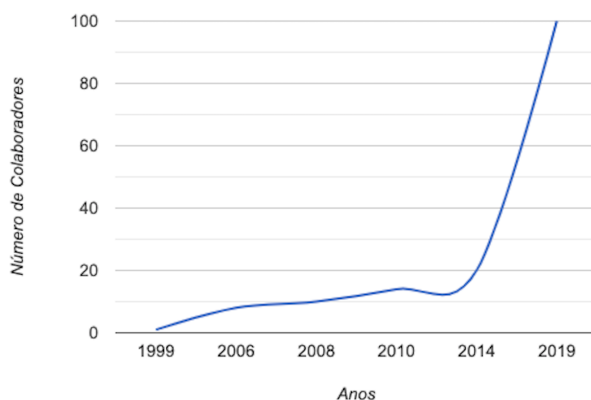


Figura 1.1: Crescimento relativo ao número de trabalhadores da E-goi

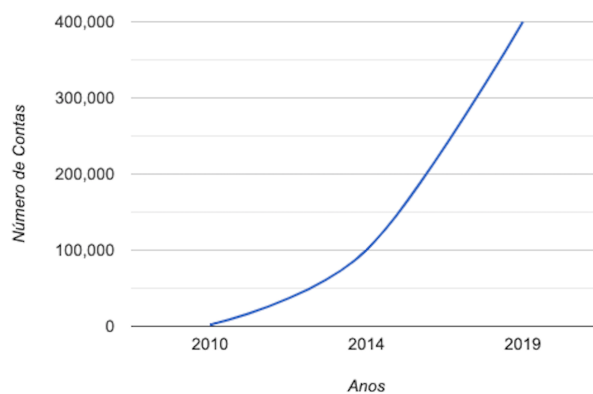


Figura 1.2: Crescimento relativo ao número de contas da E-goi

A missão da E-goi consiste na criação de soluções de comunicação digital eficazes, com uma usabilidade intuitiva. Defendem valores como inovação, criatividade, dinamismo, relevância e sustentabilidade. Relativamente à visão da mesma, a E-goi pretende “[...] ser o parceiro que consiga obter por parte dos consumidores o reconhecimento do valor das marcas”.

Para atingir os objetivos a que se propõe, a E-goi conta com uma equipa de colaboradores subdivididos em departamentos, tais como o departamento de apoio ao cliente, o comercial e o de desenvolvimento *web*, onde se enquadra este projeto. Relativamente aos clientes, a empresa fornece serviços, por exemplo, à Primavera, Salsa, Jerónimo Martins, Continente e SIBS.

Em 2013, a E-goi ganhou um dos prêmios do *Technology Fast 500™ EMEA*, um *ranking* anual elaborado pela *Deloitte Touche Tohmatsu Limited* (DTTL), que distingue as quinhentas empresas do setor tecnológico das regiões da Europa, Médio Oriente e África, que apresentam as maiores taxas de crescimento de volume de negócios nos últimos cinco anos [6]. Para além deste, a E-goi foi premiada, em 2019, pela Revista Exame Informática com o prêmio *Fast Mover*, como a empresa que mais cresceu em número de colaboradores, garantindo-lhe um lugar entre as 100 melhores empresas para trabalhar [7].

### 1.3 Problema

Tendo em conta a vasta utilização de *emails*, as caixas de correio dos utilizadores ficam sobrecarregadas diariamente, o que pode prejudicar o trabalho diário das pessoas que usam o *email* como principal meio de comunicação, uma vez que é impraticável dar atenção a todo o conteúdo.

Devido a esta afluência, os utilizadores priorizam certos *emails* em detrimento de outros com base em algumas decisões. Entre estas, encontra-se a atenção que a especificidade do assunto do *email* desperta ao utilizador [8][9][10].

Considerando que a leitura da linha de assunto de um *email* é a primeira interação entre o sistema e o utilizador, pode considerar-se que é a fase fulcral do processo de captação de interesse do utilizador pela campanha enviada.

Deste modo, torna-se necessário maximizar o factor impactante que do assunto advém. Para isto, torna-se oportuna a existência de uma ferramenta, preferencialmente automática, baseada em métodos inteligentes, que auxilie este processo, que indique o valor final e que vise ensinar o utilizador. A aplicação desta ferramenta é ainda mais útil para quem faz uso de *email marketing* no seu quotidiano, uma vez que esta fornece ao utilizador uma noção prévia do impacto que o assunto e, conseqüentemente, a campanha de *email* terão, de forma a captar e/ou reter clientes para a sua rede de contactos.

### 1.4 Objetivos

De forma a auxiliar o utilizador da E-goi a maximizar o impacto do assunto relativo à sua campanha de *email*, pretende-se implementar uma ferramenta para classificação de assunto, que analisará o campo de assunto de *emails*, dando pontuações e sugestões para o melhorar. O âmbito deste projeto passará por ser capaz de analisar assuntos de campanhas de *email* de forma autónoma, atuando como um sistema de classificação automático incorporado na plataforma E-goi.

Deste modo, encontram-se os objetivos a ter em conta, mais detalhadamente:

- Investigar e analisar ferramentas de análise de texto;
- Desenvolver um sistema baseado na abordagem de análise de texto investigada, capaz de devolver uma classificação em tempo real;
- Integrar o sistema na plataforma E-goi;
- Avaliar a eficácia do sistema.

## 1.5 Estrutura do Documento

O presente documento encontra-se subdividido em nove capítulos, respetivamente Introdução, Contextualização, Estado da Arte, Análise de Valor, Análise, *Design*, Implementação, Experimentação e Avaliação e Conclusão. De seguida, apresenta-se uma breve descrição dos mesmos:

- **Introdução:** Neste capítulo foi introduzido o projeto, sendo definido pelas secções Contexto, Apresentação da Empresa, Problema, Objetivos e Estrutura do documento;
- **Contextualização:** Neste capítulo foi realizada uma breve contextualização relativa ao projeto, nomeadamente aos conceitos de *marketing* relevantes, e às soluções concorrentes existentes no mercado;
- **Estado da Arte:** Neste capítulo foram abordadas técnicas e algoritmos que dão resposta ao problema em questão e, ainda, são apresentadas tecnologias de apoio ao desenvolvimento do projeto;
- **Análise de Valor:** Neste capítulo foi apresentada a oportunidade de negócio identificada juntamente com a sua análise. É também abordado o valor da solução por intermédio da definição do valor percebido e da proposta de valor. De seguida, é feita uma identificação e análise funcional e, por último, apresentam-se as fases de conceção de alternativas e da análise e avaliação das mesmas;
- **Análise:** Neste capítulo foi apresentado o negócio que envolve o projeto, com recurso a um modelo de domínio, seguido dos requisitos funcionais e não funcionais;
- **Design:** Neste capítulo foram apresentadas duas alternativas de *design*, seguidas da avaliação e posterior escolha da que mais se adequa. Uma vez definida a melhor solução, é apresentado o *design* detalhado, nomeadamente, o diagrama de componentes, diagrama de implantação, vistas de cenários e *mock-ups* do sistema;
- **Implementação:** Neste capítulo foi descrito todo o processo de implementação da solução desenhada;
- **Experimentação e Avaliação:** Neste capítulo foram apresentados os indicadores de avaliação, as hipóteses definidas para os mesmos e a metodologia de avaliação aplicada, seguido de uma avaliação dos resultados obtidos;
- **Conclusão:** Neste capítulo foram apresentados os objetivos alcançados, bem como as limitações da solução e o trabalho futuro.

## Capítulo 2

# Contextualização

O *email marketing* fornece um dos melhores métodos de comunicação com o consumidor [11], possibilitando empresas o desenvolvimento de campanhas de *email* para atrair novos clientes, encorajar a fidelidade dos mesmos e, ainda, anunciar sobre novas ofertas e/ou eventos [12]. Deste modo, torna-se relevante perceber os conceitos relacionados com o projeto em questão, bem como, dar a conhecer algumas soluções concorrentes existentes no mercado.

### 2.1 Campanha de Email

A plataforma E-goi permite a criação e envio de uma campanha de *email*. O fluxo inicia com a definição do nome, a seleção do editor e escolha da forma da campanha, existindo as seguintes opções:

- Criar campanha a partir do zero;
- Criar campanha a partir de um modelo E-goi;
- Criar campanha a partir de um modelo do próprio cliente;
- Criar campanha a partir de importação.

A campanha a partir do zero, tal como é expectável, refere-se à criação de uma campanha completamente vazia. Por outro lado, a E-goi disponibiliza ao utilizador um conjunto de modelos, agrupados pelas diversas categorias, entre elas, a categoria Carnaval e Aniversário, de modo a facilitar a criação de uma campanha cujo tema é recorrente, como uma época festiva. A par dos modelos da E-goi, é possível também criar uma campanha através de modelos do próprio cliente. Este tipo de modelos serve o propósito de auxiliar o trabalho dos clientes que possuem características próprias, como o formato de um *email*. Por último, a E-goi oferece a possibilidade de criação de uma campanha a partir da importação de uma página *web* e, ainda, a partir de um ficheiro *ZIP*.

Seguidamente, o utilizador é redirecionado para a página de opções da campanha de *email*, onde define o assunto e o remetente, com possibilidade de segmentação. A segmentação viabiliza a definição de diferentes opções para diferentes segmentos da lista de contactos do cliente, podendo focar-se mais nas necessidades de cada cliente, deixando-os mais satisfeitos [13]. Adicionalmente, mas não obrigatório, o cliente pode ainda integrar a sua campanha com o *Google Analytics* [14].

Após a criação da campanha e configuração das opções, o utilizador tem a possibilidade de editar o conteúdo, através do editor previamente escolhido no primeiro passo. A E-goi disponibiliza dois editores:

- *Editor Builder*: Editor *drag & drop* que permite a adição e/ou edição de elementos como texto, imagens, botões e colunas [15];
- Editor Avançado: Junção do editor visual com o editor *Hypertext Markup Language* (HTML) [15]. O editor visual é muito semelhante aos editores comuns, onde é possível adicionar conteúdo através das opções disponíveis na barra superior. Por outro lado, o outro editor permite a criação de uma campanha através de código HTML. O editor avançado é muito prático uma vez que trabalha com os dois editores anteriormente mencionados em conjunto, isto é, as alterações efetuadas no editor visual manifestam-se no editor HTML e vice-versa.

Posteriormente à edição da campanha, o utilizador pode definir as configurações de envio da mesma, sendo que pode definir a altura do envio (no momento ou agendada), a velocidade com que envia (a título de exemplo, o utilizador pode definir que pretende enviar apenas 10% dos *emails* por hora), e, ainda, a lista e possíveis segmentos a quem a campanha se destina.

Por último, o utilizador é redirecionado para uma página de confirmação, onde é apresentada a informação da campanha de forma resumida e onde é feita uma análise ao conteúdo do *email* através do *E-goi Campaign Analyzer*, que será mencionado na secção 2.2. Uma vez que o utilizador pode editar as configurações e o próprio conteúdo da campanha as vezes que desejar, esta página serve também para alertar o utilizador do que poderá estar em falta para o envio. Deste modo, a plataforma apenas deixa prosseguir com o envio caso os campos obrigatórios estejam todos preenchidos e se o *E-goi Campaign Analyzer* não apresentar nenhuma limitação.

### 2.1.1 Relatório

Para cada campanha de *email* enviada, é gerado um relatório que contém toda a informação relevante, desde os dados de envio, como o assunto e data de envio, aos dados relativos ao impacto da mesma. Deste modo, as métricas analisadas no relatório são:

- Número de *emails* enviados;
- Número de *emails* entregues e respetiva taxa;
- Número de aberturas, aberturas únicas<sup>1</sup> e respetivas taxas;
- Número de cliques, cliques únicos<sup>2</sup> e respetivas taxas;
- Número de remoções<sup>3</sup> e respetiva taxa;

---

<sup>1</sup>A abertura única refere-se a uma única abertura por utilizador. Isto é, se o utilizador abrir duas vezes a mesma campanha, o número de aberturas corresponde a 2, mas o número de aberturas únicas corresponde apenas a 1.

<sup>2</sup>À semelhança da abertura única, o clique único refere-se a um único clique por utilizador. Assim sendo, se o utilizador clicar três vezes no mesmo botão, o número de cliques únicos é 1.

<sup>3</sup>Uma remoção corresponde à remoção do subscritor da lista correspondente à campanha enviada.

- Número de *bounces*<sup>4</sup> e respetiva taxa;
- Número de queixas e respetiva taxa.

O relatório encontra-se dividido em seis separadores diferentes, sendo que a primeira página corresponde às métricas mencionadas acima, de uma forma clara e geral, como se pode ver na figura 2.1. Os restantes separadores apresentam a informação de forma mais orientada, de acordo com métricas específicas, consoante os critérios localização, *links* presentes na campanha, leitores de *email* e domínios dos destinatários.

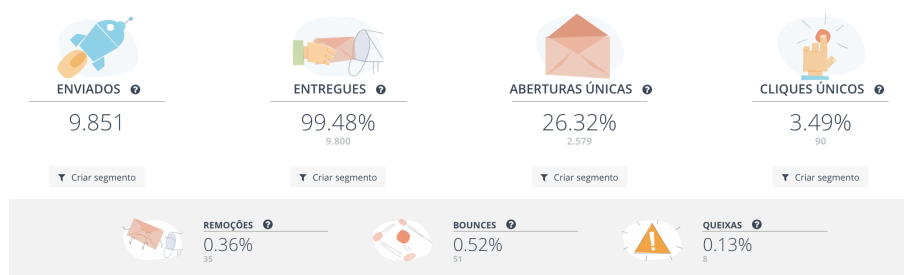


Figura 2.1: Métricas gerais num relatório de *email* da E-goi

A título de exemplo, e tendo em conta o critério localização, é mostrado um mapa, juntamente com uma tabela (figura 2.2), com os países onde existiram aberturas da campanha, de forma a ter uma ideia visual dos países onde a campanha teve mais impacto.

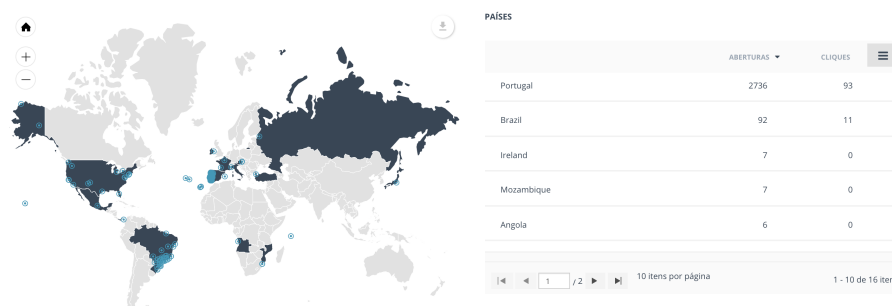


Figura 2.2: Mapa do mundo consoante as aberturas

## 2.2 E-goi Campaign Analyzer

*E-goi Campaign Analyzer* é um serviço de análise da campanha de *email*, presente no último passo do envio da campanha, como referido na secção 2.1, composto por dois módulos distintos: módulo de código e módulo de *spam*. O módulo de código, visível na figura 2.3, analisa o HTML, através de *links* não funcionais, e o *Cascading Style Sheets* (CSS), tendo em conta as boas práticas de programação CSS. Relativamente ao módulo de *spam* que se encontra ilustrado na figura 2.4, este baseia-se num filtro de *spam* de *email* popular, denominado *Apache SpamAssassin* [16], aplicado em diversas empresas [15]. Este baseia-se

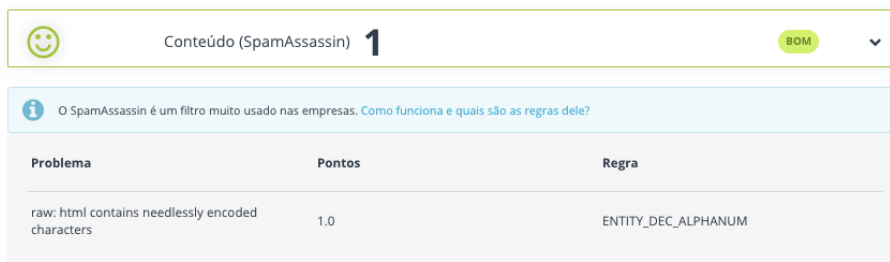
<sup>4</sup>Um *bounce* significa que o *email* foi rejeitado na caixa de *email*. Podem ser caracterizados como *soft bounces* ou *hard bounces*, sendo que o primeiro é temporário e acontece devido à caixa de entrada se encontrar cheia ou existir uma falha temporária de acesso, e o segundo é permanente e corresponde a endereços de *email* que já não existem. [15]

em regras e pontuações próprias, sendo que uma pontuação final acima de dois é considerada uma pontuação alta, existindo assim a possibilidade de *spam*.



Pontos	Tipo	Conteúdo	Problema	Solução
1	link	<code>&lt;img href="legoi_certified_link"&gt;</code>	O link parece não funcionar.	Verifique se o link está correcto. Se o link tiver um código de personalização do E-goi, é normal aparecer este aviso, mas confirme que é mesmo esse o código que quer usar.
1	link	<code>&lt;img href="linkremoval" style="font-family: &amp;quot;Open Sans&amp;quot;; font-size: 12px; color: rgb(54, 70, 86); text-align: center; text-decoration: underline; margin: 0px;&gt;</code>	O link parece não funcionar.	Verifique se o link está correcto. Se o link tiver um código de personalização do E-goi, é normal aparecer este aviso, mas confirme que é mesmo esse o código que quer usar.
1	link	<code>&lt;img href="viewlink" style="font-family: &amp;quot;Open Sans&amp;quot;; font-size: 10px; color: rgb(144, 148, 153); text-align: right; text-decoration: underline; margin: 0px;&gt;</code>	O link parece não funcionar.	Verifique se o link está correcto. Se o link tiver um código de personalização do E-goi, é normal aparecer este aviso, mas confirme que é mesmo esse o código que quer usar.

Figura 2.3: Módulo de código do *Campaign Analyzer*



Problema	Pontos	Regra
raw: html contains needlessly encoded characters	1.0	ENTITY_DEC_ALPHANUM

Figura 2.4: Módulo de *spam* do *Campaign Analyzer*

Cada módulo devolve uma pontuação juntamente com uma cor, considerando que a cor verde corresponde a uma análise positiva, a cor amarela a uma análise razoável e, por fim, a cor vermelha, a uma análise negativa, impedindo, esta última, o utilizador de prosseguir com o envio. A fim de auxiliar o utilizador, a ferramenta fornece algumas dicas de código, de modo a melhorar a qualidade do conteúdo do *email* e dicas de segurança para evitar que a campanha seja considerada *spam*.

## 2.3 Testes A/B

A E-goi oferece a possibilidade de utilizar testes A/B, que é uma das formas conhecidas de medição da qualidade e de resposta do cliente face a uma campanha. Deste modo, a técnica verifica como pequenas alterações no mesmo *email* impactam os resultados de uma campanha [15].

Assim, a plataforma E-goi envia duas ou mais versões diferentes do mesmo *email* apenas para uma amostra aleatória da lista de contactos seleccionada. Após o envio, é realizada uma análise durante o tempo definido pelo utilizador aquando da criação do teste. Por fim, a versão com mais sucesso, baseada no critério definido perante os existentes relativos a aberturas únicas, aberturas, cliques únicos e cliques, é enviada a todos os outros contactos.

## 2.4 Soluções Concorrentes de Análise de Assunto

A fim de obter uma amostra realista das soluções existentes hoje em dia relativas à análise de assunto, foram identificadas diversas ferramentas tendo por base a sua popularidade. Estas podem fragmentar-se em duas grandes categorias: ferramentas de análise relativas a uma plataforma de *marketing* como as ferramentas *Mailchimp Subject Line Helper*, *Mizy* e *Omnisend Email Subject Line Tester*, e ferramentas independentes que agregam as ferramentas *Send Check It*, *Touchstone* e *IsItWp Email Subject Analyzer*.

### 2.4.1 Mailchimp Subject Line Helper

O *Mailchimp* [17], uma plataforma de automação de *marketing*, oferece aos seus utilizadores uma ferramenta denominada *Subject Line Helper*, que verifica se o assunto escrito pelo utilizador segue as dicas de boas práticas fornecidas. Estas foram definidas pela equipa de análise do *Mailchimp* e baseadas no desempenho de taxa da abertura de centenas de milhões de *emails* enviados pela plataforma [18]. A adoção de personalização através de códigos, como o nome, o uso cuidado de *emojis*, e a limitação da pontuação são algumas das práticas aconselhadas.

Podemos verificar, pela figura 2.5, o funcionamento da ferramenta. Para cada uma das práticas, o *Mailchimp* mostra a cor verde às que foram cumpridas com sucesso, e um tom alaranjado às que necessitam de ajustes, sugerindo o que o utilizador deve alterar para alcançar um maior impacto.



Figura 2.5: *Subject Line Helper* [18]

Adicionalmente, o *Mailchimp* oferece ao utilizador a possibilidade de visualizar os assuntos referentes às últimas cinco campanhas enviadas, com as respetivas taxas de abertura, como se pode ver pela figura 2.6. Assim, de uma forma ágil e fácil, o utilizador possui uma visão geral do impacto que as suas últimas campanhas tiveram e quais os assuntos adotados.

Recent subject line performance	
Big deals on big ferns: Up to 40% off	65% opens
It's Friday!!!!	28.3% opens
This week's deals	52.7% opens
January flash sale!	72.7% opens
Great news	18.2% opens

Figura 2.6: Performance das últimas cinco campanhas de *email* enviadas [18]

Apesar do *Mailchimp* possuir quatro planos de subscrição da plataforma - *Premium*, *Standard*, *Essentials* e *Free* - com variadas funcionalidades, o *Subject Line Helper* encontra-se presente em todos eles, de forma gratuita.

## 2.4.2 Send Check It

*Send Check It* [19] tem como principal propósito apresentar ao utilizador uma classificação em tempo real, como se pode ver pela figura 2.7, oferecendo dicas relacionadas com a melhoria de um assunto de *email*, de forma gratuita.

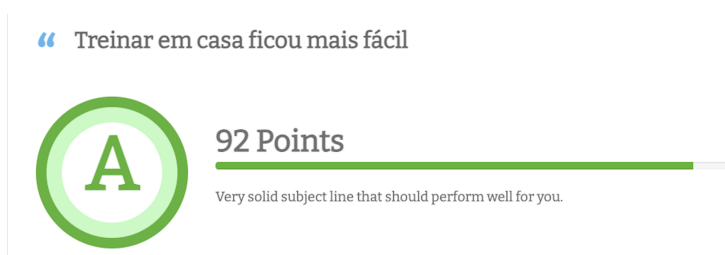


Figura 2.7: Classificação geral retornada pelo *Send Check It* [19]

Assim sendo, utiliza métricas como o comprimento do assunto (figura 2.8), verificando e analisando a extensão deste, o nível de leitura (figura 2.9), que tem como principal objetivo analisar a escolha de palavras, a complexidade de cada uma e a sua dimensão e, utiliza, ainda, uma métrica relativa à perceção (figura 2.10), que é um conceito que diz respeito à facilidade com que um assunto é percecionado por um utilizador aquando da sua visualização [20].

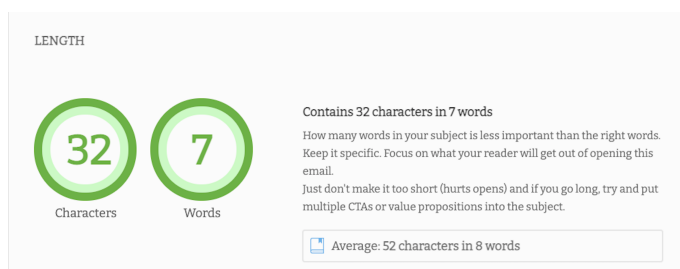


Figura 2.8: Análise de comprimento efetuada pelo *Send Check It* [19]

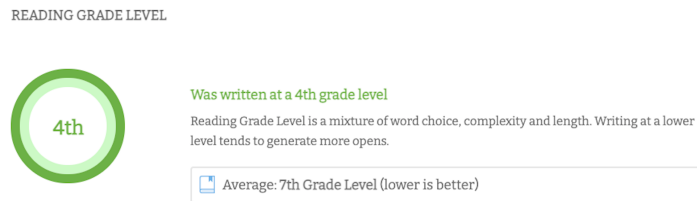


Figura 2.9: Análise do nível de leitura efetuada pelo *Send Check It* [19]

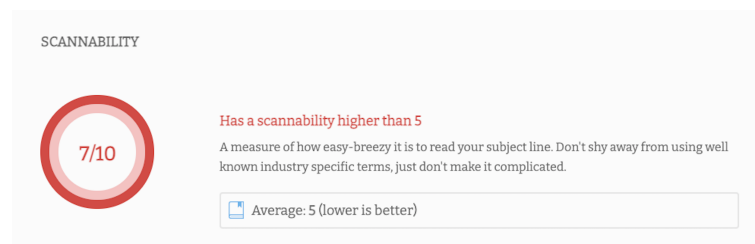


Figura 2.10: Análise à perceção efetuada pelo *Send Check It* [19]

Adicionalmente, e como se pode verificar pela figura 2.11, a ferramenta efetua uma análise de sentimento<sup>5</sup>. Para além destas, verifica a utilização de certas palavras, bem como o uso de pontuação, letras maiúsculas, personalização e *emojis*.

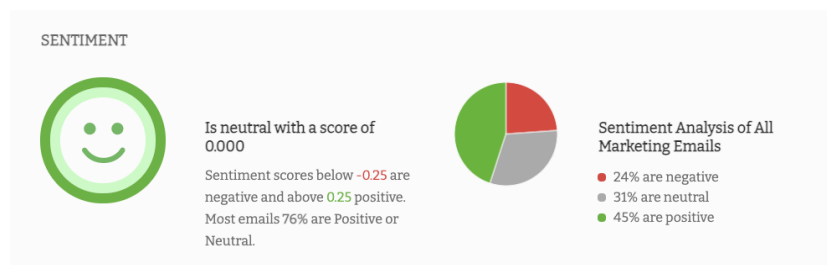


Figura 2.11: Análise de sentimento efetuada pelo *Send Check It* [19]

Como podemos verificar pelas imagens, os vários critérios são classificados consoante uma cor, existindo a cor vermelha, amarela e verde, que representa um nível baixo, médio ou alto de conformidade com o critério, respetivamente. Para além disto, cada critério tem a sua própria classificação textual, sendo que para a perceção a classificação segue uma escala de 0 a 10, enquanto que, para a métrica do comprimento do assunto, são apresentados ao utilizador o número de caracteres e o número de palavras, separadamente. No entanto, para cada critério é fornecida uma descrição com uma breve explicação do mesmo e a média aritmética, consoante os dados utilizados para a análise.

O utilizador pode fazer uso desta ferramenta de duas formas diferentes:

- Via *web*;
- Via *Application Programming Interface* (API);

<sup>5</sup>A análise de sentimento visa determinar se a opinião que foi expressada num determinado texto é positiva ou negativa [21].

O possível uso da ferramenta *Send Check It* via API, torna viável a sua integração com outro produto. Desta forma, apresenta-se, na figura 2.12, um exemplo da interface visual do produto com a integração.

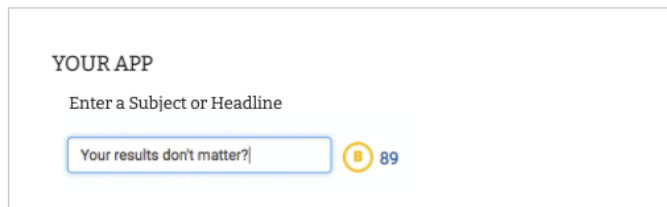


Figura 2.12: Exemplo de integração do *Send Check It* num produto [22]

### 2.4.3 Mizy

*Mizy* [23] é uma ferramenta que permite a análise de assuntos de campanhas de *email*. Desenvolvida pela *Automizy* [24], encontra-se integrada na plataforma de *email marketing* aquando da criação de uma campanha (figura 2.13), e ainda oferece o produto, de forma gratuita, via *web* (figura 2.14), sem obrigar o utilizador a criar uma conta caso não tenha intenções de utilizar a ferramenta mas sim de apenas analisar variados assuntos.

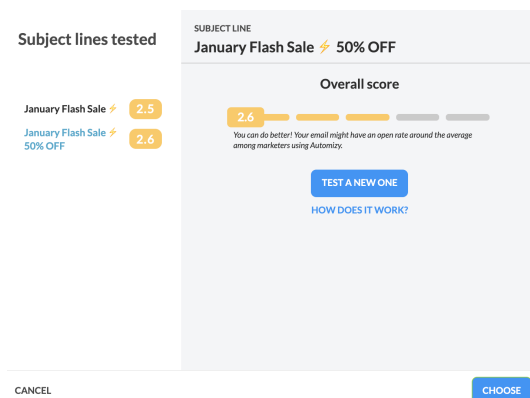


Figura 2.13: *Mizy* integrada na plataforma [23]

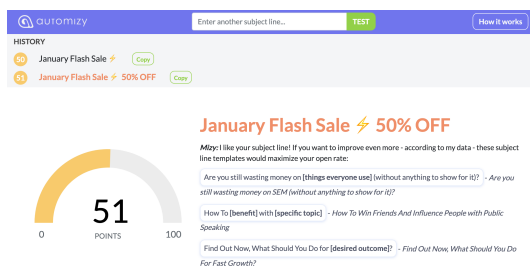


Figura 2.14: *Mizy* via *web* [23]

A ferramenta foi desenvolvida recorrendo a *Machine Learning* (ML) e *Deep Learning* (DL) de forma a analisar um volumoso conjunto de dados, atualizar os critérios de análise de forma automática e fazer previsões cada vez mais precisas sobre cada assunto. Os dados

utilizados, provenientes de campanhas reais, são compostos pelo assunto da campanha em conjunto com as taxas de aberturas e de cliques [25].

Relativamente ao serviço integrado na *Automizy*, este fornece uma pontuação visual de 0 a 5, guardando todas as consultas de assuntos efetuadas com a respetiva classificação. Por outro lado, o sistema presente na *web* oferece uma pontuação de 0 a 100 que, devido à maior largura da escala, torna mais precisa a classificação. Para além disto, apresenta, à semelhança do serviço integrado, um histórico de consultas, e ainda, mostra ao utilizador exemplos de assuntos que podem vir a ter uma alta taxa de aberturas, de acordo com o modelo de análise.

#### 2.4.4 Omnisend Email Subject Line Tester

A *Omnisend* [26] é uma plataforma de automação de *marketing* orientada ao *E-commerce*, que permite a criação e automação de campanhas de diversos canais de comunicação, nomeadamente, o *email*. Apesar de não estar integrado na plataforma e no fluxo de criação da campanha, a empresa oferece um analisador de assuntos de campanhas de *email* - *Omnisend Email Subject Line Tester* [27].

Para incentivar o uso da ferramenta, a *Omnisend* defende que [27]:

- 35% dos utilizadores abrem a campanha de *email* apenas baseados na linha de assunto;
- 69% dos clientes marcam o *email* como *spam* com base no assunto, o que afeta a capacidade de entrega e a reputação do remetente;
- 59% dos clientes alegam que as campanhas de *email* influenciam a sua decisão de compra.

Após a inserção do assunto a testar, a ferramenta faculta uma pontuação de desempenho geral, em forma de percentagem, como se verifica na figura 2.15, juntamente com sugestões de melhoria, a fim de potenciar a linha de assunto e, conseqüentemente, aumentar a taxa de aberturas.



Figura 2.15: Classificação geral do *Omnisend Subject Line Tester* [27]

A ferramenta analisa a eficácia das linhas de assunto considerando quatro métricas, sendo que estas se encontram categorizadas entre comprimento, redação, identificação de *spam* e perceção. No que diz respeito à métrica responsável pela análise do comprimento, a ferramenta em questão considera que um bom assunto não deverá conter mais do que quarenta caracteres, assim como não deverá ser constituído por um conjunto de palavras superior a sete. Relativamente à métrica da redação, a ferramenta apresenta uma análise sobre as palavras úteis e as palavras negativas do assunto, identificando as palavras a alterar,

não apresentando alternativas. A identificação de *spam* baseia-se na existência de caracteres e palavras de *spam*, e pontuação excessiva. No exemplo apresentado na figura 2.15, a ferramenta considera que as palavras *great* e *lose* podem classificar o *email* como *spam*, aconselhando o utilizador a testar e optar por outras alternativas. Por último, a perceção, avalia os critérios *sentence case* e a utilização de números. Quanto ao critério *sentence case*, este avalia se o assunto desenvolvido se encontra com a capitalização correta para que se considere uma frase bem escrita. Caso contrário, diminuirá a pontuação do assunto e avisará o utilizador. A utilização de números é promovida pela ferramenta, considerando um ponto bastante positivo se esta utilização se encontrar bem contextualizada com o modelo de negócio a que se propõe o assunto.

### 2.4.5 Touchstone

*Touchstone* [28] é uma plataforma de análise de assuntos de *email* que permite, através de importação, simular virtualmente a base de dados do cliente relativa a dados reais das campanhas já enviadas, comparar as variadas linhas de assunto com vinte e um bilhões de *emails*, e, ainda, equiparar o assunto aos restantes já utilizados relativamente ao setor em que se enquadra.



Figura 2.16: *Touchstone* [29]

Como se pode observar pela figura 2.16, a ferramenta apresenta, inicialmente, o número de *emails* utilizados para a análise, bem como a quantidade de *emails* já enviados que apresentem características similares ao testado. Posteriormente, são exibidas métricas relativas às taxas de aberturas, cliques e entrega, sob forma de gráficos de barras. Estes gráficos servem o propósito de comparar o assunto inserido aos restantes assuntos já utilizados consoante a base de dados importada, a base de dados referente ao setor ao qual o *email* se destina, e à base de dados total da *Touchstone*. Para além disto, é ainda fornecida a contagem de palavras, comprimento do assunto e a média de tamanho de cada palavra, juntamente com gráficos que mostram as aberturas, cliques e taxa de entrega relativas a cada valor.

Através das diferentes comparações, a *Touchstone* pretende oferecer um serviço mais preciso ao cliente, consoante as suas necessidades. No entanto, esta ferramenta não é gratuita. Assim, ela dispõe de três tipos de serviço:

- Um dia ilimitado: Custo de \$19.99;
- Plano para bases de dados até duzentas e cinquenta mil entradas: Custo de \$499 por mês;
- Plano personalizado: Custo dependente do plano definido consoante as necessidades do cliente.

#### 2.4.6 IsItWP Email Subject Analyzer

*IsItWP Email Subject Analyzer* [30] é uma ferramenta gratuita de análise de assuntos de *email*, que disponibiliza ao utilizador o histórico de todas as consultas efetivadas, bem como a sua respetiva classificação, em formato percentual. A análise assenta em sete fatores como se pode ver pela figura 2.17:

- Utilização de letras maiúsculas: A ferramenta considera que a utilização de *title case*<sup>6</sup> potencia a linha de assunto, considerando que se deve sempre desenvolver um assunto como tal;
- Contagem de palavras: Verifica se o número de palavras é o correto e informa que os assuntos que contêm entre quatro a seis palavras são os que possuem um melhor desempenho;
- Presença de números: Verifica se o assunto possui números, uma vez que defende que os assuntos com a sua presença têm mais aberturas;
- Contagem de caracteres: Indica que linhas de assunto com cerca de vinte caracteres no total são conhecidas como sendo as mais eficientes em termos de taxas de aberturas;
- Contagem de *emojis*: A ferramenta indica que o *email* terá mais possibilidades de ser bem sucedido se contiver, pelo menos, um *emoji*;
- Palavras que aumentam aberturas: Indica quais palavras presentes na linha de assunto são associadas a taxas de aberturas superiores;
- Palavras que diminuem aberturas: Indica quais palavras presentes na linha de assunto podem, eventualmente, diminuir a taxa de aberturas.

Com base nos sete fatores supratranscritos, a ferramenta calcula a classificação total do assunto e apela a que se ambicione um valor nunca menor do que 70%.

<sup>6</sup>Estilo de escrita no qual se utiliza letra maiúscula no início de cada palavra da frase [31].

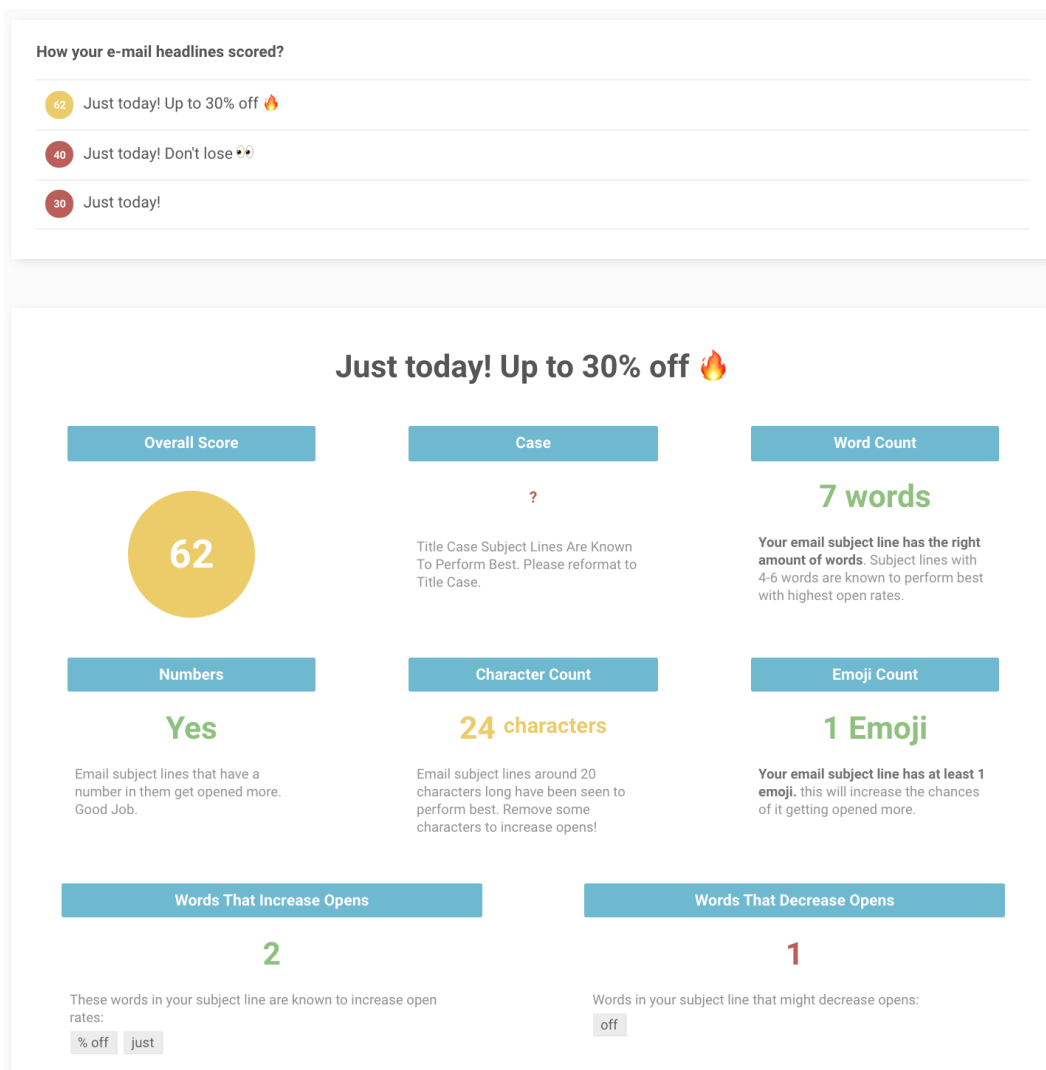


Figura 2.17: *IsItWP Email Subject Analyzer* [30]

### 2.4.7 Análise Comparativa

Uma vez apresentadas as soluções concorrentes, torna-se relevante o desenvolvimento de uma análise comparativa de modo a que se perceba em que critérios diferem as soluções, justificando tal diferenciação.

Com base nisto, apresenta-se a tabela 2.1 na qual se comparam todas as soluções de análise de assuntos abordadas anteriormente e, ainda, a solução relativa ao *E-goi Subject Analyzer* pretendida.

Tabela 2.1: Análise comparativa de soluções

	Fornecer API de integração pública	Integrada com plataforma de <i>marketing</i>	Utilização de dados reais	Classificação em tempo real	Histórico de consultas de assuntos	Gratuito
<i>Mailchimp Subject Line Helper</i>		X	X	X		X
<i>Send Check It</i>	X		X			X
<i>Mizy</i>		X	X	X	X	X
<i>Omnisend Email Subject Line Tester</i>						X
<i>Touchstone</i>			X		X	
<i>IsItWP Email Subject Analyzer</i>					X	X
<i>E-goi Subject Analyzer</i>		X	X	X	X	X

O primeiro critério de comparação diz respeito ao fornecimento de uma API de integração do analisador e, tal como se pode verificar, apenas o *Send Check It* fornece esta funcionalidade, a fim de ser possível a integração do analisador desenvolvido noutras plataformas. Por outro lado, a *E-goi* pretende desenvolver uma API de integração mas exclusivamente para uso interno.

Dos produtos concorrentes abordados, o *MailChimp Subject Line Helper*, o *Mizy* e o *Omnisend Email Subject Line Tester* pertencem a plataformas de *marketing* que possuem a funcionalidade de criação e envios de campanhas de *email*. No entanto, apenas as duas primeiras disponibilizam a funcionalidade de forma integrada neste fluxo, sendo que a *Omnisend* obriga o utilizador a usar de forma externa, via *web*, caso pretenda usufruir do produto. No que diz respeito ao *E-goi Subject Analyzer*, este, para além de oferecer a API, também se pretende que esteja integrado no fluxo de criação de campanhas da plataforma.

No que diz respeito ao terceiro critério aplicado, o *Omnisend Email Subject Line Tester* e o *IsItWP Email Subject Analyzer* são os únicos que não utilizam dados reais na sua análise, baseando-se apenas no que se consideram as boas práticas de *marketing* relativamente à escrita de um assunto. Apesar do *Mailchimp* analisar o assunto através de boas práticas, estas foram definidas pela própria equipa num estudo que envolveu inúmeras campanhas enviadas. Este processo tende a ser custoso uma vez que, para se manter a par das tendências, novos estudos terão de ser realizados.

A classificação em tempo real faz com que a produção do melhor assunto para cada campanha seja um processo mais simples e orgânico por parte do utilizador, ao invés do mesmo necessitar de exercer alguma ação, como o clique num botão, para obter a classificação. Das soluções abordadas apenas o *Mailchimp Subject Line Helper*, o *E-goi Subject Analyzer* e o *Mizy* devolvem a classificação do assunto em tempo real.

No que se refere ao quinto critério, somente o *Mizy*, o *Touchstone*, o *IsItWP Email Subject Analyzer* e o *E-goi Subject Analyzer* fornecem um histórico de consultas de assuntos. Esta

funcionalidade facilita a experimentação de várias hipóteses para que, no final, possa ter uma escolha consciente sobre qual o melhor assunto a definir para a campanha.

Por último, todos os serviços são gratuitos excetuando o *Touchstone*, que permite o pagamento por um dia, usufruindo de todas as funcionalidades durante vinte e quatro horas, e a adesão a um plano base ou um plano personalizado.

A título conclusivo, denota-se relevância no desenvolvimento do projeto na medida em que se considera fulcral para a E-goi o desenvolvimento de uma funcionalidade de análise de assuntos que compita com as soluções concorrentes existentes no mercado. Deste modo, a solução desenvolvida pela E-goi cumpre todos os critérios apresentados, excetuando a API para integração com produtos externos, ainda que se pretenda o desenvolvimento de uma API para uso interno.

## Capítulo 3

# Estado da Arte

Devido aos avanços tecnológicos que têm convergido num potenciamento das capacidades de armazenamento de dados, as empresas, nomeadamente a E-goi, possuem uma grande quantidade de dados existentes nas suas bases de dados. Ainda que se reconheça o valor destes dados, a obtenção de informações úteis, de modo a extrair conhecimento, é um processo desafiante, na medida em que é considerado complexo o processamento e a gestão dos dados. Deste modo, a utilização de um sistema autónomo facilitará o reconhecimento do valor dos dados [32]. Não obstante, considerando que o mundo digital se encontra em constante mudança [33], este sistema torna-se mais relevante caso se mantenha a par desta evolução, modelando-se e adaptando-se às novas tendências através de técnicas de ML.

De modo a que se criem sistemas que aprendam de forma autónoma, tornando-se cada vez mais inteligentes conforme a sua experiência, combinam-se algoritmos de ML com *Natural Language Processing* (NLP) [34]. Este define-se como sendo o processo que visa obter um melhor conhecimento no que diz respeito à linguagem natural através da utilização de computadores, de forma a melhorar processos de classificação e de recolha de informação [35].

### 3.1 Natural Language Processing

NLP é uma área que providencia às máquinas a capacidade de ler e entender a linguagem natural humana. Este conceito tem várias áreas de aplicação mas a sua utilização tem vindo a crescer ao longo do tempo. Apesar de ser um conceito relevante por si só, esta tecnologia, que representa o tratamento automático de linguagem natural, tem ainda mais valor quando aplicada a casos reais. Entre estes, casos de uso como tratamento e previsão de doenças, baseados na fala, desenvolvimento de produtos que ofereçam assistência cognitiva lembrando o utilizador de algo que este já não se lembra mas que em tempos referiu, análise de sentimento baseada em informação presente, por exemplo em redes sociais ou, ainda, mecanismos de combate ao *spam* através da classificação de *emails* utilizando NLP são só uma pequena percentagem das grandes áreas de atuação desta área tecnológica.

O pré-processamento de texto tem o objetivo de transformar os dados a fim de serem mais facilmente analisados pelo algoritmo e de prever resultados mais precisos, sendo considerada uma etapa importante no que se refere a tarefas de NLP [36]. Existem diversas técnicas de pré-processamento [37][38], entre elas:

- *Tokenization*: Processo que segmenta uma frase numa sequência de *tokens*, de acordo com a sua pontuação ou espaços em branco. Assim, é possível remover certos caracteres como a pontuação;

- *Lowercasing*: Mapeia todos os dados de texto em minúsculas;
- *Stemming*: Processo de redução das palavras através da remoção de afixos;
- *Lemmatization*: À semelhança da técnica *Stemming*, remove inflexões e mapeia uma palavra até à sua forma raiz. A diferença reside na forma como o faz. Enquanto a *Stemming* corta as palavras, a *Lemmatization* procura e vincula as palavras aos seus *lemmas*<sup>1</sup>;
- *Stop Word Removal*: Filtra e exclui palavras de linguagem comum, como pronomes e preposições, que não são informativos sobre o texto correspondente.

## 3.2 Machine Learning

O termo ML foi usado pela primeira vez no livro *Some Studies in Machine Learning Using the Game of Checkers*, de Arthur Lee Samuel [39], que, em 1987, foi congratulado com o prémio de pioneiro da computação, pela *IEEE Computer Society* [40]. Arthur Samuel descreve ML como “[...] um campo de estudo que dá ao computador a capacidade de aprender sem ser explicitamente programado”<sup>2</sup>.

Já Tom Mitchell propôs, em 1997, uma definição mais técnica e formal - “Diz-se que um programa de computador aprende com a experiência E com relação a alguma classe de tarefas T e medida de desempenho P, se o seu desempenho nas tarefas T, medido por P, melhorar com a experiência”<sup>3</sup> - isto é, um programa evolui e obtém melhores resultados numa determinada tarefa à medida que obtém mais dados e se torna, conseqüentemente, mais experiente [41].

No que concerne à resolução de problemas relativos à área, torna-se essencial tomar conhecimento dos seguintes conceitos:

- *Feature*: Característica individual mensurável ou atributo das entradas do conjunto de dados;
- *Label*: Atributo usado para classificar um conjunto de dados com base nas suas características.

Atualmente, a área de ML é considerada uma das áreas de Inteligência Artificial que mais evolui, sendo considerada o principal fator de sucesso no progresso das empresas que a utilizam [42]. Esta área segmenta-se em três categorias principais, apresentadas na figura 3.1 - *Supervised Learning*, *Unsupervised Learning* e *Reinforcement Learning* [43] -, e, cada uma, possui vários métodos que podem ser aplicados.

---

<sup>1</sup>Forma canônica de uma palavra.

<sup>2</sup>Tradução livre do autor. No original “[...] a field of study that gives computer the ability to learn without being explicitly programmed.”

<sup>3</sup>Tradução livre do autor. No original “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

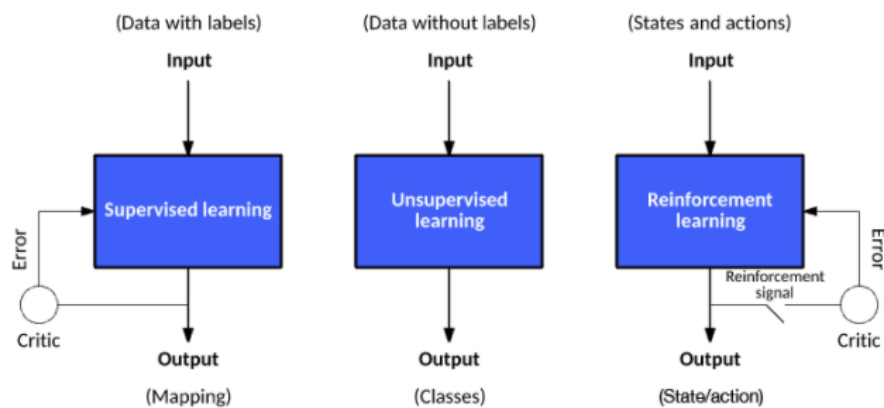


Figura 3.1: Categorias principais do ML [43]

Para além destes métodos, existem outros métodos de DL, um subconjunto de ML que tem vindo a crescer ao longo do tempo, como se pode analisar pelo gráfico de tendências presente na figura 3.2. Este gráfico é baseado em dados reais presentes na plataforma *Google Trends*, relativos às pesquisas efetuadas no motor de pesquisa *Google*, à escala mundial, desde 2004 até ao presente.

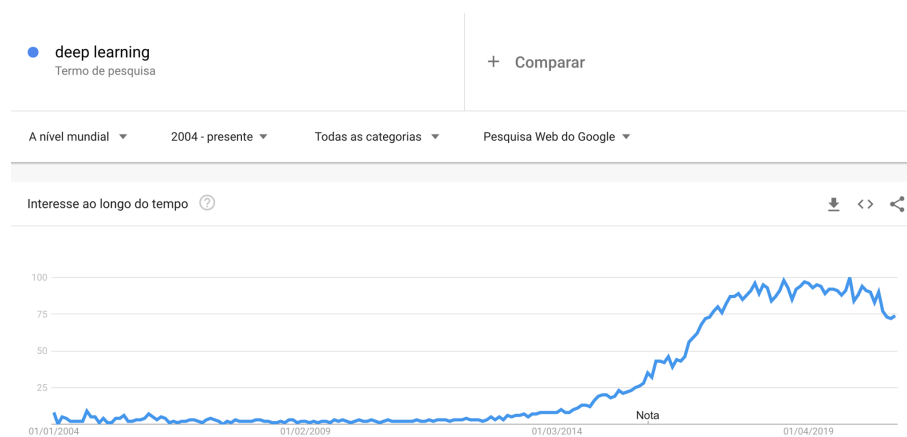


Figura 3.2: Tendência de pesquisas *online* pelo conceito DL [44]

### 3.2.1 Tipos de Machine Learning

Na *Supervised Learning*, o modelo é construído com base nos dados de entrada e apresentado sob a forma de pares ordenados - entrada e saída esperada. Isto é, os dados são rotulados, uma vez que sabemos de antemão a saída esperada para cada entrada de dados. O treino do modelo consiste em fornecer ao algoritmo um número suficiente de instâncias de entradas e saídas, ajustando-se ao longo do tempo para que defina uma regra geral que mapeie as entradas nas saídas corretamente, construindo o modelo final.

Os modelos de aprendizagem supervisionada podem ser sub-agrupados em casos de classificação e regressão. A classificação é, normalmente, aplicada quando o rótulo usado para classificar os dados de entrada é um conjunto finito de valores discretos [45], isto é, quando

a variável de saída é uma categoria. No que diz respeito à regressão, esta é adotada quando a variável de saída consiste em uma ou mais variáveis contínuas [45].

Na *Unsupervised Learning* os dados de entrada do algoritmo não são rotulados. Ao invés de prever a saída, como na *Supervised Learning*, o sistema explora os dados e faz inferências de conjuntos de dados para descrever estruturas ocultas, encontrando padrões anteriormente desconhecidos num conjunto de dados sem rótulos pré-existentes [45]. Esta abordagem pretende resolver problemas de associação e *clustering*.

*Reinforcement Learning* é um tipo de aprendizagem que se baseia na interação com o ambiente, fazendo com que os algoritmos encontrem as ações a tomar, de forma a maximizar a taxa de cumprimento de um determinado objetivo. Deste modo, os algoritmos usam os erros estimados como recompensas ou penalidades. Isto é, se o erro for grande, a penalidade é alta e a recompensa baixa. Por outro lado, se o erro for pequeno, a penalidade é baixa e a recompensa alta.

A título de resumo, apresenta-se a tabela 3.1 que tem como objetivo sintetizar informação relativa ao tipo dados e problema, à definição do tipo de algoritmos, à abordagem e, por fim, ao objetivo de cada categoria.

Tabela 3.1: Resumo das categorias principais de ML

	<b>Supervised Learning</b>	<b>Unsupervised Learning</b>	<b>Reinforcement Learning</b>
<b>Tipos de dados</b>	Dados rotulados	Dados não rotulados	Dados não predefinidos
<b>Definição</b>	Aprende a partir de dados rotulados	Aprende através de dados não rotulados, sem qualquer interação	Aprende interagindo com o meio ambiente
<b>Tipos de problemas</b>	Classificação e regressão	Associação e <i>clustering</i>	Problemas baseados em recompensa
<b>Abordagem</b>	Mapeia as entradas rotuladas para as saídas conhecidas	Entende padrões e descobre a saída	Segue o método de tentativa e erro
<b>Objetivo</b>	Calcular resultados	Descobrir padrões subjacentes	Aprender um conjunto de ações

### 3.2.2 Deep Learning

DL é um subcampo do ML e recorre a modelos baseados em redes neuronais complexas, com várias camadas internas. O número de camadas sucessivas de representações define a profundidade do modelo [46].

As redes neuronais consistem num elevado número de neurónios artificiais interconectados, semelhantes ao cérebro humano, que aplicam uma função ao valor da entrada, produzindo, deste modo, outro valor que é posteriormente recebido pelos demais neurónios nas camadas seguintes. Deste modo, a camada final é responsável por avaliar os valores presentes nos neurónios e fornecer a saída final [46].

Na figura 3.3 encontra-se presente uma rede com várias camadas de representações. Esta rede transforma a imagem para reconhecer o dígito definido, através da transformação em representações cada vez mais diferentes da imagem original e mais informativas sobre o resultado final.

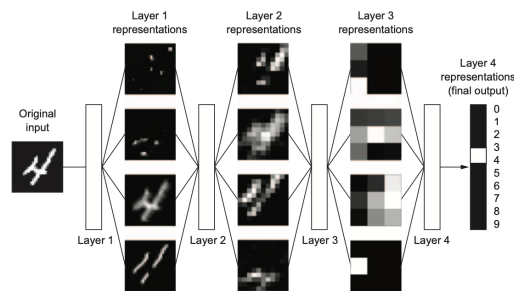


Figura 3.3: Camadas sucessivas de representações num modelo de classificação de dígito [46]

No que diz respeito ao funcionamento das redes neurais, representado na figura 3.4, a transformação implementada por uma camada é parametrizada por pesos. Inicialmente, os pesos da rede são atribuídos aleatoriamente, o que leva à implementação de um conjunto de transformações aleatórias e, conseqüentemente, a um *loss score* muito elevado. No entanto, a cada iteração do treino do modelo os pesos são ajustados, diminuindo o *loss score* e minimizando a *loss function*.

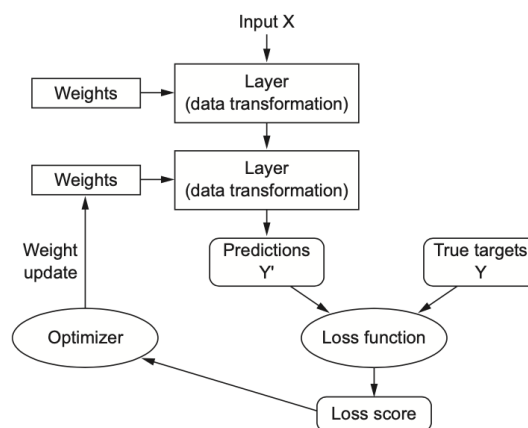


Figura 3.4: Funcionamento de uma rede neuronal [46]

### 3.2.3 Métricas de Avaliação

No que diz respeito à avaliação de desempenho de um modelo, surge o conceito de *confusion matrix*, representado na figura 3.5. Este é caracterizado por uma tabela que retrata o desempenho de um modelo, indicando os erros e os acertos em comparação com o resultado esperado [47].

		Assigned class	
		Positive	Negative
Real class	Positive	TP	FN
	Negative	FP	TN

Figura 3.5: *Confusion matrix* (adaptado de [47])

Da análise da figura 3.5, surgem os termos:

- Verdadeiros positivos (TP): Número de instâncias positivas corretamente classificadas pelo modelo;
- Falsos negativos (FN): Número de instâncias positivas erroneamente classificadas como negativas pelo modelo;
- Falsos positivos (FP): Número de instâncias negativas erroneamente classificadas como positivas pelo modelo;
- Verdadeiros negativos (TN): Número de instâncias negativas corretamente classificadas pelo modelo.

Uma vez apresentados os conceitos advindos da *confusion matrix*, surgem as métricas relativas à avaliação do modelo, nomeadamente *accuracy*, *precision*, *recall* e *F1-Score*.

No que diz respeito à *accuracy*, esta representa uma métrica de avaliação geral do desempenho do modelo, caracterizada pela relação entre o número de previsões corretas e o total de previsões efetuadas, tal como se pode verificar pela equação 3.1, em que  $l$  representa o número de classes:

$$Accuracy = \frac{\sum_{i=1}^l \left( \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i} \right)}{l} \quad (3.1)$$

Por outro lado, a *precision* reflete a proporção dos dados classificados corretamente como positivos, isto é, quão preciso é o modelo ao classificar instâncias positivas. Assim, apresentase, de seguida, a equação que a traduz:

$$Precision = \frac{\sum_{i=1}^l TP_i}{\sum_{i=1}^l (TP_i + FP_i)} \quad (3.2)$$

Relativamente à métrica *recall*, esta simboliza a percentagem de dados classificados como positivos comparado com a quantidade real de dados positivos que existem na amostra (calculada a partir da equação 3.5).

$$Recall = \frac{\sum_{i=1}^l TP_i}{\sum_{i=1}^l (TP_i + FN_i)} \quad (3.3)$$

Por último, *F1-Score* é definida como a média ponderada entre cada uma das classes:

$$F1-Score = \frac{\sum_{i=1}^l (F1-Score_i * No.Samples_i)}{\sum_{i=1}^l No.Samples_i} \quad (3.4)$$

Em que,  $F1-Score_i$ :

$$F1-Score_i = 2 * \frac{Precision_i * Recall_i}{Precision_i + Recall_i} \quad (3.5)$$

### 3.2.4 Algoritmos de Classificação de Texto

Uma vez que o propósito deste projeto passa por atribuir uma classificação a um assunto, tendo em conta dados previamente rotulados, categoriza-se este projeto como sendo parte integrante de um problema de classificação. Deste modo, descrevem-se alguns dos algoritmos mais usados como solução a problemas deste tipo - *Naïve Bayes* (NB), *Decision Tree* (DT), *Support Vector Machine* (SVM), *Artificial Neural Network* (ANN) e *Recurrent Neural Network* (RNN) [48][49][50][51][52][53].

#### Naïve Bayes

Baseado no teorema de Bayes, o algoritmo NB é um classificador probabilístico baseado em suposições de independência. Essas suposições de independência fazem com que a ordem das *features* seja irrelevante e, consequentemente, que a presença de uma *feature* particular não tenha relação com outra [54].

Matematicamente, o teorema de Bayes determina a probabilidade de um evento ocorrer, dada a probabilidade de outro evento já ter ocorrido, e é representado pela seguinte equação [54]:

$$P(Y|X) = \frac{P(X|Y) * P(Y)}{P(X)} \quad (3.6)$$

Uma rede NB é considerada um modelo para relações de probabilidade entre um conjunto de *features*. É representada sob a forma de um grafo acíclico direcionado, em que cada característica raiz tem uma probabilidade de ocorrência e cada elemento filho tem várias probabilidades de ocorrência com base na probabilidade dos elementos pai [55]. Um exemplo ilustrativo de uma rede NB encontra-se na figura 3.6.

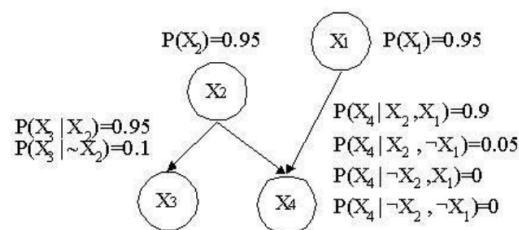


Figura 3.6: Rede NB [55]

#### Decision Tree

As DT são estruturas em árvore (figura 3.7) baseadas na estratégia de divisão e conquista. Assim sendo, seguem o princípio da decomposição, aplicando regras, de forma recursiva, para dividir um problema considerado complexo em problemas mais simples [55].

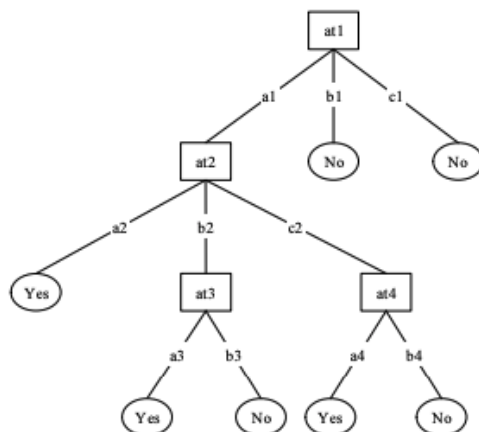


Figura 3.7: DT [55]

No que diz respeito à sua composição, encontram-se os seguintes elementos:

- Nós (representados na figura 3.7 pela forma rectangular): Cada nó tem o objetivo de dividir um conjunto de dados em subconjuntos, de acordo com um teste específico;
- Folhas (representadas na figura 3.7 pela forma ovalada): Representa os nós finais, isto é, os nós que contêm o conjunto de dados mais homogêneo;
- Ramos: Responsáveis pela interligação entre nós ou entre nós e folhas.

Assim, para que se classifique uma instância, o mecanismo de DT inicia-se pela raiz da árvore. Uma vez testada essa *feature* segue-se o ramo correspondente ao valor da *feature* dada, repetindo-se para a sub-árvore cuja raiz é um novo nó. A título conclusivo, ao fluir pela árvore desde o nó inicial, é possível atribuir uma classificação a uma instância, seguindo os ramos definidos na árvore que terminam no nó folha, nó este que contém a solução ao problema inicial.

### Support Vector Machine

SVM é uma técnica que almeja a divisão do espaço de classificação pertencente a um conjunto de instâncias de dados, a fim de os classificar. Deste modo, a finalidade passa por descobrir o hiperplano ideal num espaço N-dimensional, em que N representa o número de *features*, de modo a classificar distintamente um conjunto de instâncias de dados [55].

Tendo em consideração que as instâncias mais próximas da margem são denominadas vetores de suporte, o hiperplano ótimo é definido, entre um universo de hiperplanos possíveis, através da maximização da distância, isto é, através do hiperplano que apresenta a distância máxima entre os vetores de suporte de ambas as classes [55].

Considerando as duas classes, quadrado e círculo, presentes no exemplo de criação de um hiperplano (figura 3.8), pode verificar-se que o hiperplano ótimo encontra-se caracterizado pela cor vermelha. Este é considerado uma fronteira tal que todos os quadrados residem de um lado do hiperplano e todos os círculos residem do outro, tendo em conta que o hiperplano ótimo é aquele que maximiza a margem entre as duas classes com o auxílio dos vetores de suporte.

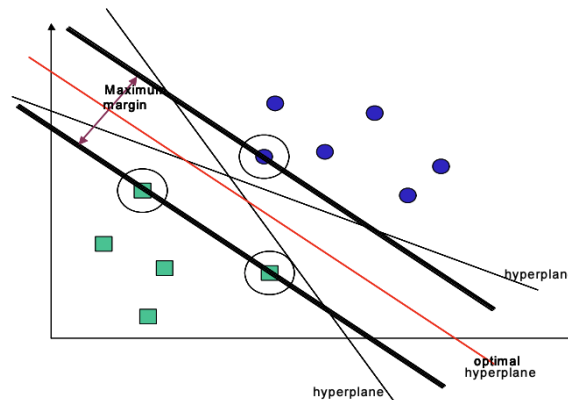


Figura 3.8: Exemplo de criação de um hiperplano [55]

### Artificial Neural Network

ANN é um modelo que visa processar os dados de forma semelhante ao cérebro humano, sendo considerada uma abstração da rede neural biológica. Composta por um conjunto de neurónios conectados, é organizada em várias camadas distintas - uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída [56].

A figura 3.9 representa um exemplo de uma rede. No que diz respeito à primeira camada, esta é responsável pela introdução dos dados, a fim de serem processados pelas camadas subsequentes da rede artificial, enquanto as camadas ocultas processam um grupo de entradas ponderadas, por intermédio da aplicação de uma função de ativação e, por último, a camada de saída encontra-se encarregue de produzir o resultado.

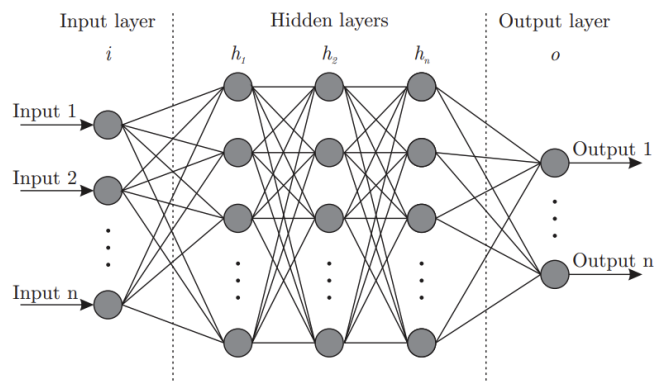


Figura 3.9: Estrutura de uma ANN [57]

No que diz respeito ao funcionamento da rede, cada neurónio é responsável pela soma da multiplicação dos dados de entrada pelos pesos sinápticos que, no momento da inicialização da rede, são atribuídos aleatoriamente. Posteriormente, o neurónio aplica uma função de ativação à soma das entradas ponderadas e fornece o resultado para os neurónios presentes na camada seguinte. O objetivo da função de ativação é limitar a amplitude de saída do neurónio, dado um conjunto de entradas. As funções *Sigmoid*, *Tan-h*, *Softmax*, *ReLU* e *Leaky ReLU* são algumas das funções de ativação que podem ser aplicadas, dependendo do problema em questão [56].

De forma a minimizar o erro, é utilizado um método de retropropagação, que compara o valor obtido na saída com o valor desejado, através da equação 3.7, em que  $j$  representa o neurónio,  $rr$  o valor esperado e  $ro$  o valor obtido. Caso o resultado não se encontre no espectro de um padrão aceitável, a rede propaga a correção para as demais camadas até à entrada, ajustando assim os pesos sinápticos até se atingir um valor aceitável [56].

$$e_j = ro_j - rr_j \quad (3.7)$$

### Recurrent Neural Network

No que se refere às ANN, as predições dos dados dependem exclusivamente das entradas, isto é, os neurónios apenas transmitem a informação da camada de entrada em direção à camada de saída, não possuindo capacidade de memória. É neste contexto que surgem as RNN, cujo objetivo é mimetizar a capacidade do cérebro humano de memorizar acontecimentos passados.

À semelhança das ANN, uma RNN possui uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, destinadas ao processamento de sequências de dados. A diferença chave diz respeito à existência de um ciclo associado aos neurónios das camadas ocultas, capacitado para preservar as informações da sequência ao longo do tempo. Deste modo, a figura 3.10 demonstra uma arquitetura RNN composta por conexões recorrentes entre os neurónios presentes na camada oculta, produzindo uma saída a cada passo de tempo [53].

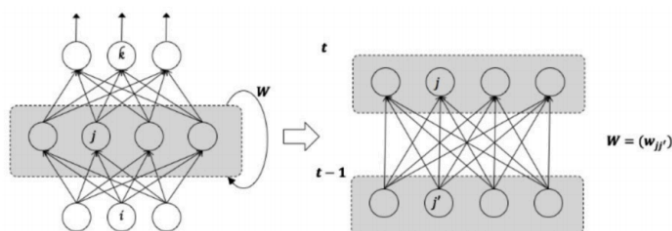


Figura 3.10: Estrutura básica de uma RNN [53]

A *Long Short-Term Memory* (LSTM) pretende superar a dificuldade em aprender dependências a longo prazo existente na RNN que, à medida que aumenta o número de etapas a serem consideradas, as informações dos eventos anteriores desaparecem exponencialmente. Deste modo, a LSTM permite o escalonamento de sequências muito mais longas que o RNN [53].

Um bloco LSTM encontra-se representando na figura 3.11 e é composto por uma célula de memória e três unidades de portas multiplicativas: uma porta de entrada (identificada por  $i_t$ ), uma porta de saída (identificada por  $O_t$ ) e uma porta de esquecimento (identificada por  $f_t$ ). Relativamente à célula de memória, esta é responsável por transmitir valores em intervalos de tempo arbitrários em que, cada porta, é responsável por executar operações de gravação, leitura e redefinição para as células [53]. Deste modo, ao longo de várias etapas de tempo, as portas determinam os dados que devem ser adicionados ou excluídos/esquecidos com base na importância aprendida, tendo em consideração que o algoritmo aprende com o tempo a fim de ajustar, adequadamente, a atribuição da importância.

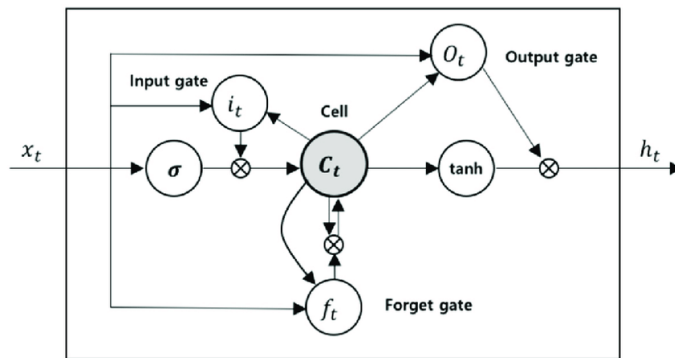


Figura 3.11: LSTM com unidades de controle [53]

### 3.2.5 Conclusão

O processo de NLP envolve diversas áreas, entre elas a classificação de texto, na qual se enquadra o presente projeto. É possível concluir, a partir da análise da figura 3.12, que todas as áreas testemunharam um crescimento ao longo dos anos, representando uma tendência crescente de estudos de NLP a nível global. Consta-se ainda que a área na qual se categoriza o *Subject Analyzer* é a área que conta com o maior número de estudos à data de 2018. Apesar de não serem dados relativos à atualidade imediata, a tendência crescente é notória e constante.

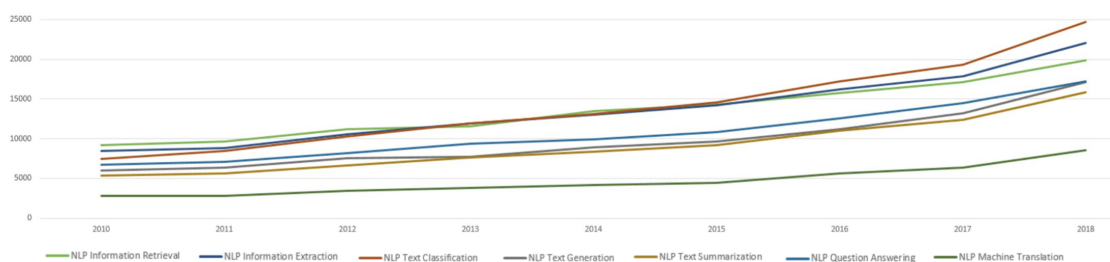


Figura 3.12: Número de estudos publicados para cada área de NLP (adaptado de [58])

Considerando que as técnicas de ML têm sido substituídas, ao longo dos anos, por técnicas DL em tarefas de NLP [58] e que os algoritmos de DL são providos de uma melhor *performance* quando comparados com algoritmos de ML [58][59], conforme se pode verificar através da análise da figura 3.13, optou-se pela utilização de sistemas baseados em algoritmos de DL. Uma vez que os dados armazenados pela E-goi são muito volumosos, com tendência crescente, contendo dados relativos a aproximadamente 90 mil milhões de campanhas enviadas por todos os clientes registados, e tendo em consideração que a *performance* é um requisito explícito, denota-se uma necessidade de se adotar algoritmos DL.

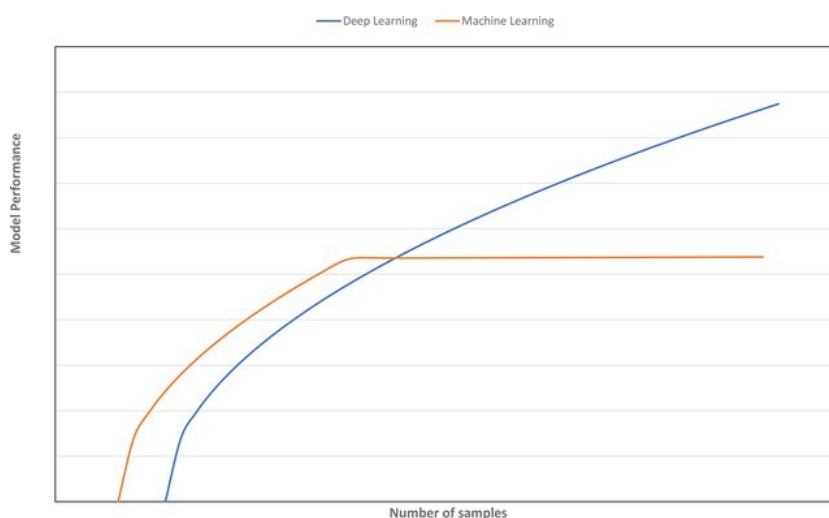


Figura 3.13: Desempenho de algoritmos ML e DL em função do número de amostras [59]

Nos últimos anos, o uso de algoritmos de DL aplicados a tarefas de NLP tem aumentado consideravelmente, conforme se pode verificar através de estudos prévios [58][60], começando em grande parte pelo projeto pioneiro *Natural Language Processing (Almost) from Scratch* de Collobert e seus co-autores [61].

Tendo em conta o problema a abordar, soluções baseadas em algoritmos de redes neurais, como ANN e RNN, podem ser aplicadas. Considerando que uma linha de assunto é composta por um conjunto de palavras contextualizadas, a relação de interdependência toma um papel fulcral. Posto isto, as possíveis disposições das palavras numa frase podem convergir em sentidos de percepção distintos, pelo que a adoção do algoritmo a utilizar deve contemplar sequências de dados [58].

Deste modo, a RNN é considerada uma das arquiteturas mais aplicadas em tarefas de NLP, uma vez que é capaz de processar texto com um comprimento variável, devido à sua estrutura recorrente, principalmente com a adoção da variante LSTM [62]. Como referido na subsecção 3.2.4, a LSTM permite o escalonamento de sequências de dados muito mais longas que o RNN.

A título exemplificativo, encontra-se referenciado um estudo que convergiu num projeto relativo à classificação de texto, mais especificamente à classificação de notícias relacionadas com a *British Broadcasting Corporation* (BBC) [63], a mais antiga emissora britânica, que utiliza o algoritmo LSTM, em conjunto com as tecnologias *Tensorflow* e *Keras* [64].

### 3.3 Tecnologias

A utilização de tecnologias representam um grande auxílio no que diz respeito à aplicação de algoritmos de DL, uma vez que ajudam na criação do modelo, através da aplicação do algoritmo [65]. Deste modo, destacam-se as tecnologias *Tensorflow*, *Keras*, *PyTorch* e *Caffe* com base no seu nível de popularidade [66][67].

### 3.3.1 TensorFlow

*TensorFlow* [68] é uma *framework open source* desenvolvida pela *Google Brain*, um segmento da *Google* orientado à *Artificial Intelligence (AI)* e direcionado ao desenvolvimento de soluções de ML e DL.

O *TensorFlow* baseia-se na implementação de grafos de fluxo de dados, em que conjuntos de dados representam tensores e são processados por intermédio de uma série de algoritmos descritos pelos gráficos mencionados.

Além da facilidade de execução, por intermédio da invocação de funções já implementadas, sendo apenas necessário definir as suas configurações, a ferramenta oferece o *TensorBoard* [69] que disponibiliza um conjunto de serviços, entre eles:

- Acompanhamento e visualização de métricas como perda e precisão;
- Visualização do grafo do modelo;
- Visualização de histogramas de pesos ou outros tensores.

Deste modo, o *TensorBoard* permite inspecionar e interpretar visualmente qualquer estatística relativa ao modelo, como parâmetros de treino, e, ainda, a estrutura do grafo, através de uma página *web*.

### 3.3.2 Keras

*Keras* [70] é uma API *open source* de alto nível, escrita em *Python*, que serve como abstração para *frameworks* de DL como o *Tensorflow*.

Desenvolvida para criar e treinar modelos, por Francis Chollet, é considerada [70]:

- Fácil de usar: Possui uma API simples e consistente e fornece *feedback* claro e prático para os erros cometidos por parte do utilizador;
- Modular: Os modelos são implementados conectando elementos configuráveis, com poucas restrições;
- Fácil de estender: Os novos módulos são fáceis de incorporar.

### 3.3.3 PyTorch

*PyTorch* [71] é uma ferramenta *open source* de ML, baseada no *Torch* [72] e desenvolvida pela *Facebook AI Research*. Enquanto *Torch* é uma *framework* baseada na linguagem *Lua* [73], o *PyTorch* é executado em *Python*.

Fornecendo um gráfico em tempo real, esta *framework* pode ser utilizada para construir redes neuronais profundas e executar cálculos de tensores de alta complexidade, permitindo flexibilidade e rapidez na construção. À semelhança do *TensorFlow*, possui um ecossistema rico e oferece suporte ao desenvolvimento.

### 3.3.4 Caffe

*Caffe* [74] é uma *framework* de DL implementada de acordo com três princípios: arquitetura expressiva, velocidade e modularidade [74]. Desenvolvida pela *Berkeley AI Research*, é uma

ferramenta *open source*, sob a licença BSD, escrita na linguagem C++, com a interface em *Python*.

A tecnologia oferece suporte a vários tipos de arquitetura, tais como RNN, inicialmente com o foco de classificação de imagens e segmentação [67]. O conceito relativo à expressão incentiva a aplicação e inovação, a velocidade faz com que a *framework* tenha a capacidade de processar sessenta milhões de imagens por dia e, por último, a modularidade permite que o projeto possa ter contribuições com mudanças significativas.

### 3.3.5 Análise Comparativa

Considerando estudos prévios [66][75] e de modo a constatar qual a melhor tecnologia a adotar para o desenvolvimento do *Subject Analyzer* optou-se por uma análise comparativa, baseada nos seguintes critérios: volume de pesquisa no *Google*, artigos da *Medium* [76], requisitos para emprego *online*, artigos no *arXiv* [77] e atividade no *GitHub* [78].

Relativamente ao primeiro critério de análise, e tendo em consideração que as pesquisas na *web* representam um bom indicador de popularidade, recorreu-se ao uso da plataforma *Google Trends*. Como se pode analisar pela figura 3.14, foi efetuada uma pesquisa de tendências comparando as diferentes tecnologias, à escala mundial, desde 2004 até ao presente relativamente à categoria *Machine Learning Artificial Intelligence*. Embora o *PyTorch* tenha tido uma evolução notória nos últimos tempos, o *Tensorflow* continua a ser a *framework* que detém um rácio de pesquisas mais elevado.

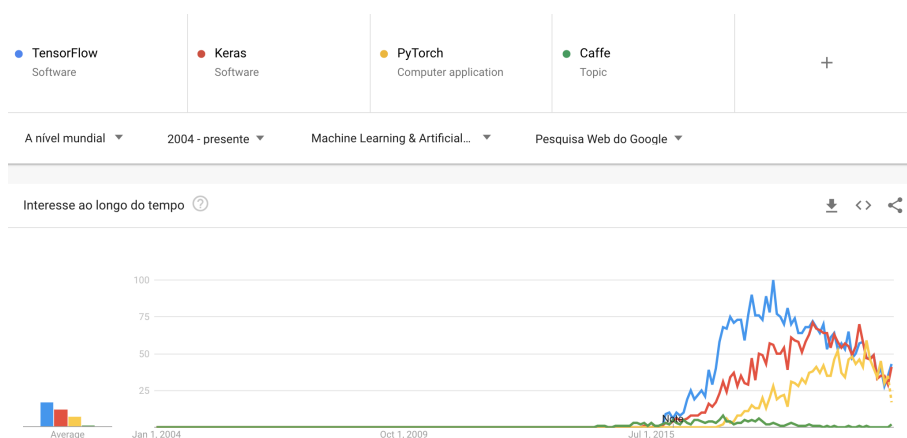


Figura 3.14: Tendência de pesquisas *online* das *frameworks* [79]

Relativamente ao segundo critério, referente aos artigos da *Medium*, uma plataforma de partilha dos mesmos, denota-se, através da análise da figura 3.15 que a *framework* *Keras* é aquela que mais artigos relacionados tem, seguindo-se o *Tensorflow* e o *PyTorch*.

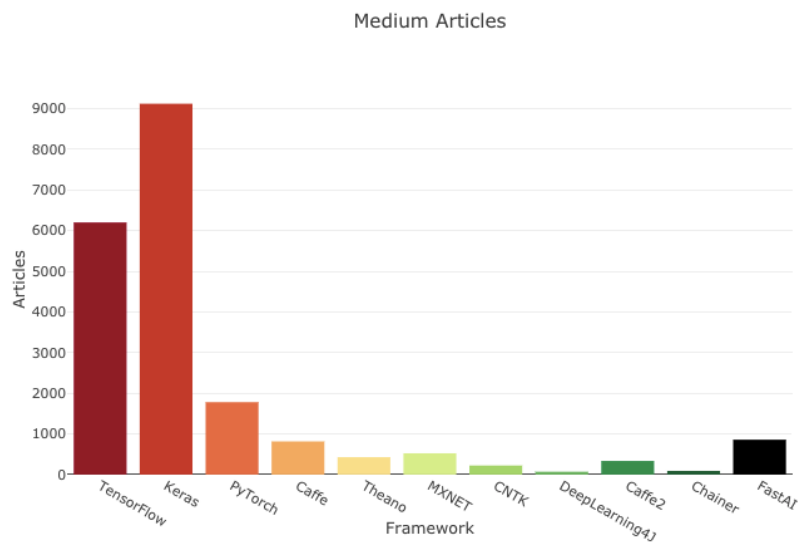


Figura 3.15: Número de artigos publicados na plataforma *Medium* relativos às *frameworks* [66]

O critério relativo aos requisitos para emprego *online* foi construído com base nas listas de empregos publicadas no *LinkedIn*, *Indeed*, *Simply Hired*, *Monster* e *Angel List*. Deste modo, é possível verificar pelo gráfico representativo da listagem em relação às diversas *frameworks*, presente na figura 3.16, que o *Tensorflow* é a ferramenta mais requisitada quando se trata de empregos orientados ao termo ML.

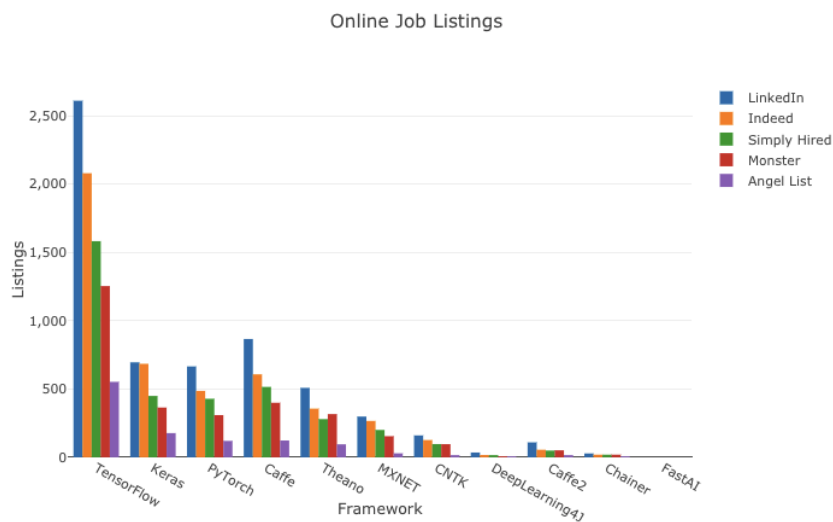


Figura 3.16: Número de empregos *online* relativos às *frameworks* [66]

No que se refere aos artigos públicos no *arXiv*, um repositório *online* onde a maioria dos artigos académicos relativos a ML são publicados, denota-se uma descida significativa quanto ao *Keras*, destacando-se novamente o *Tensorflow* (figura 3.17).

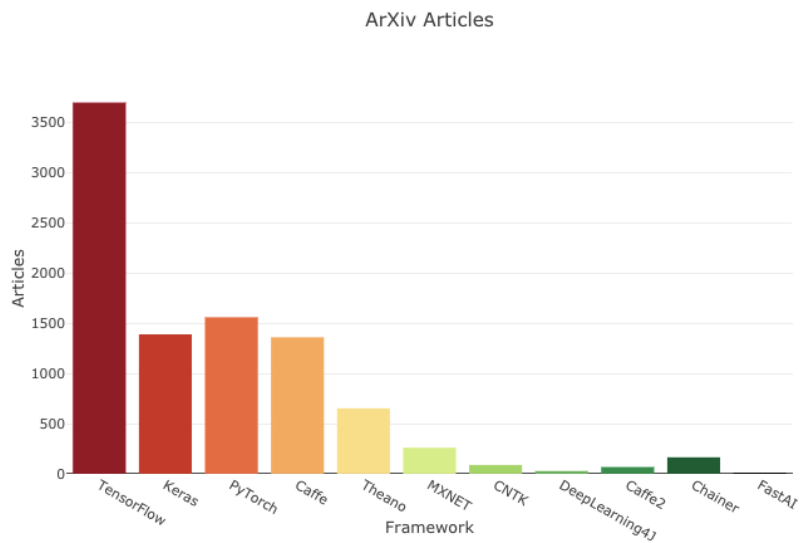


Figura 3.17: Número de artigos publicados na plataforma *arXiv* relativos às *frameworks* [66]

No que diz respeito à atividade no *GitHub*, plataforma que providencia serviços de hospedagem *online* para desenvolvimento de *software* e controlo de versões, esta relaciona-se com diferentes indicadores: número de estrelas, *forks*, observadores e contribuidores. Conclui-se que o *Tensorflow* lidera qualquer uma das métricas previamente definidas. Um aspeto interessante observado diz respeito aos níveis de contribuidores, que se encontram mais próximos para as diversas *frameworks*, do que qualquer uma das outras três métricas.

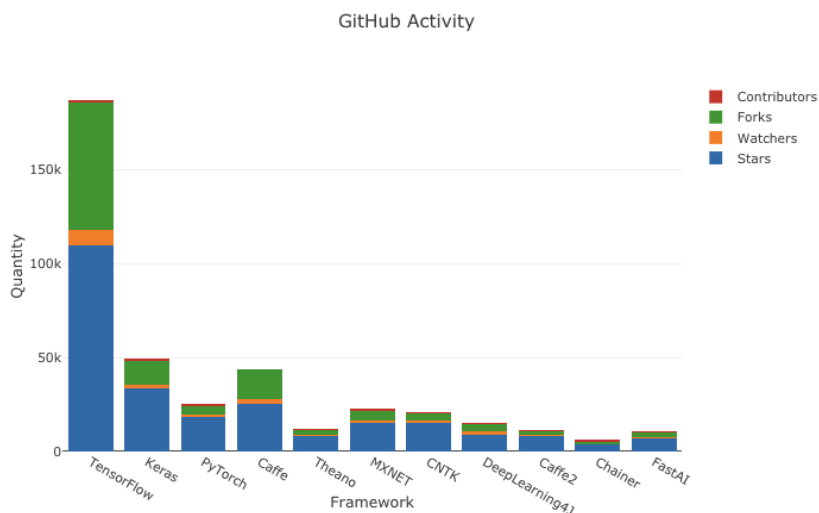


Figura 3.18: Atividade no *GitHub* relativa às *frameworks* [66]

A título conclusivo, é notório o predomínio do *Tensorflow* em comparação com as restantes tecnologias, no que diz respeito aos critérios analisados, excetuando para o critério referente ao número de artigos publicados no *Medium*, em que a *framework* *Keras* se destacou.

## Capítulo 4

# Análise de Valor

Aquando o desenvolvimento de um novo produto, torna-se essencial a sua análise e avaliação, de modo a que seja viável o aumento do seu valor com o menor custo possível, sem descuidar a qualidade, permitindo assim oferecer ao cliente um produto com qualidade superior e desenvolvido de forma mais eficiente [80].

O processo de Análise de Valor (VA), presente na figura 4.1, é composto por cinco etapas: Orientação, Análise Funcional, Criação de Alternativas, Análise e Avaliação, e Implementação [80].

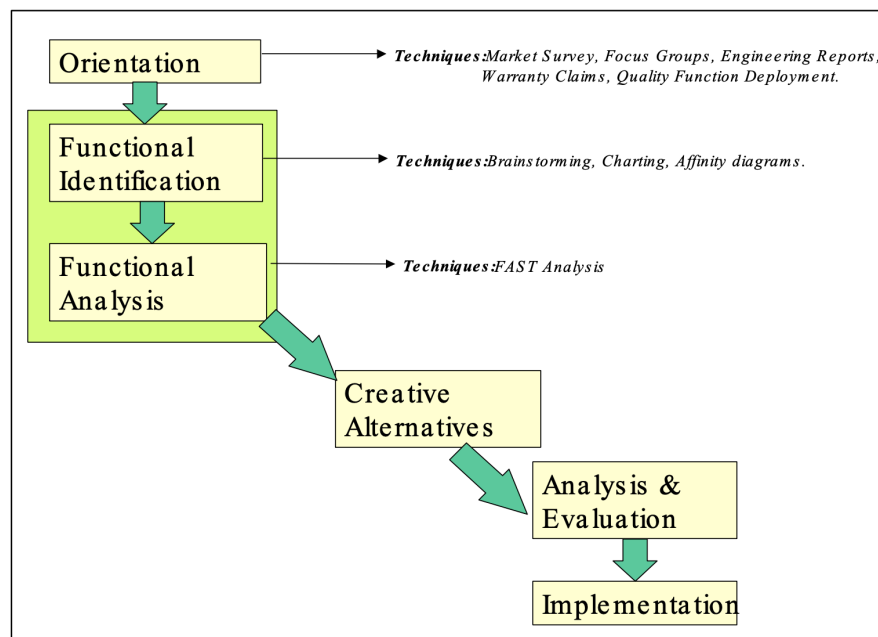


Figura 4.1: Processo de VA [80]

### 4.1 Orientação

Tendo em conta a evolução do mundo tecnológico, torna-se essencial inovar constantemente produtos e/ou processos com o objetivo primordial de criar valor às empresas e aos seus clientes.

O processo de inovação, representado no livro *The PDMA TookBook for New Product Development* [81], de autoria Peter A. Koen e seus co-autores, é um processo que segue o fluxo desde a criação à comercialização de um certo produto.

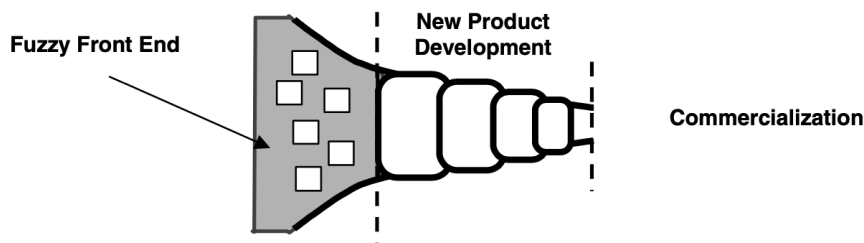


Figura 4.2: Processo de inovação [82]

Deste modo, pela análise da figura 4.2 podem verificar-se as três fases que caracterizam o processo:

- *Fuzzy Front End* (FEE);
- *New Product Development* (NPD);
- Comercialização.

Reinertsen e Smith introduziram o termo FEE em 1991 [83], que tem como objetivo o desenvolvimento de um conceito advindo de uma oportunidade previamente identificada. No que diz respeito à fase de NPD, esta é subdividida em novas fases, como a fase de *design*, desenvolvimento e *marketing*, com o objetivo de aumentar a participação de uma empresa no mercado. Por último, a fase de Comercialização é constituída pela venda do produto desenvolvido.

#### 4.1.1 Identificação de Oportunidade

A capacidade de atingir clientes num curto espaço de tempo, a fácil recuperação do investimento e a rapidez em obter relatórios e estatísticas precisas sobre uma campanha de *email* fazem com que o *email marketing* seja um dos métodos prediletos de contacto pelas empresas [84].

Uma pré-condição necessária para que qualquer campanha de *email marketing* seja considerada bem sucedida é a sua abertura e visualização por parte do utilizador, dado que elevadas taxas de abertura tornam-se um fator crítico para o sucesso do negócio [9].

Como referido na secção 1.3 denota-se um problema relativo à elevada quantidade de *emails* recebidos diariamente, por parte dos consumidores, o que aumenta a competição pela atenção e resulta em diversos *emails* sem qualquer interação.

Em resposta a essa competição e como solução ao problema, as empresas, nomeadamente a E-goí, procuram uma forma de maximizar a percentagem de destinatários que abrem o *email*, aumentando a taxa de aberturas e potenciando a taxa de cliques, isto é, aumentando o número de assinantes que clicaram em pelo menos um *link* da campanha de *email marketing*.

A atenção captada por um assunto bem definido é uma das formas principais de maximização de atenção por campanhas via *email* (c.f secção 1.3). Desta forma, a E-goí considera que

a implementação de um analisador de assunto é uma oportunidade e uma mais valia no mercado de *email marketing*.

#### 4.1.2 Análise de Oportunidade

Após a identificação da oportunidade, é necessário analisá-la a fim de confirmar a viabilidade da ideia. Para tal, existem diversas técnicas que auxiliam esta análise, entre elas, a análise SWOT.

A análise SWOT é um processo de gestão estratégica que envolve quatro componentes em duas dimensões - análise externa e análise interna [85]. Ao realizar uma análise externa, são identificadas as oportunidades e ameaças do meio ambiente em que a ideia se insere. Enquanto esta se concentra em fatores externos, a análise interna foca-se nos fatores internos, auxiliando a identificação dos pontos fortes e fracos advindos da ideia. Dito isto, a análise SWOT visa a compreensão de quais os aspetos prováveis de serem fontes de vantagem competitiva.

No que diz respeito à oportunidade identificada - implementação de um analisador de texto incorporado na plataforma E-goi - efetuou-se a análise SWOT caracterizada pela matriz presente na figura 4.3, com o objetivo primordial de apreender se a oportunidade identificada oferece valor à empresa.

	Benéfico	Prejudicial
Análise Interna	<ul style="list-style-type: none"> <li>● Base de clientes previamente adquirida</li> <li>● Análise baseada em dados reais</li> <li>● Utilização de códigos de personalização</li> <li>● Integração com a criação de uma campanha de <i>email marketing</i></li> <li>● Sistema de classificação visual do assunto da campanha</li> <li>● Dicas de boas práticas</li> <li>● Histórico de consultas de assuntos</li> </ul>	<ul style="list-style-type: none"> <li>● Complexidade de integração com o sistema</li> <li>● Custo associado à produção do analisador de assunto</li> </ul>
Análise Externa	<ul style="list-style-type: none"> <li>● Utilização de <i>email marketing</i> bastante adotada nas empresas</li> <li>● Acréscimo de utilização de campanhas de <i>email</i> consequentes da situação pandémica atual</li> <li>● Impacto advindo da atenção pelo assunto de <i>email</i> que se traduz num aumento de interações</li> </ul>	<ul style="list-style-type: none"> <li>● Concorrência (seja ela do mesmo modelo de negócio ou com soluções mais específicas de analisadores de assunto)</li> </ul>

Figura 4.3: Matriz SWOT

Do ponto de vista interno e benéfico, foram identificados sete pontos fortes relativos ao desenvolvimento do *Subject Analyzer*. O facto de ser um serviço integrante do produto E-goi, confere ao analisador uma forte vantagem relativamente a alguns concorrentes, uma vez

que irá ser disponibilizado e utilizado por uma base de clientes extensa já adquirida. Como consequência, a base de dados utilizada sob a qual o analisador irá trabalhar será composta por dados reais, advindos da junção de vários assuntos de campanhas já enviadas, tendo em consideração as respetivas taxas de aberturas. A integração do *Subject Analyzer* com a plataforma previamente existente permite ainda a utilização de códigos de personalização<sup>1</sup>, traduzindo-se assim numa força para o produto, uma vez que a utilização de informações específicas do destinatário, nomeadamente o seu nome, encontra-se associada a um aumento da probabilidade de aberturas e, conseqüentemente, ao sucesso da campanha [86].

Uma vez que os utilizadores, atualmente, não têm noção do impacto que a sua campanha de *email* terá para os seus destinatários, nem quais os aspetos mais relevantes a ter em consideração aquando da definição do assunto, torna-se pertinente o desenvolvimento de um sistema que permita uma análise visual automática e em tempo real e que ofereça dicas de boas práticas de *marketing* relativas à escrita de um assunto com qualidade. Adicionalmente, o facto de se incorporar no fluxo de criação de uma campanha de *email* potencia uma melhoria na usabilidade, uma vez que não é necessário interromper o fluxo para utilizar serviços externos de análise.

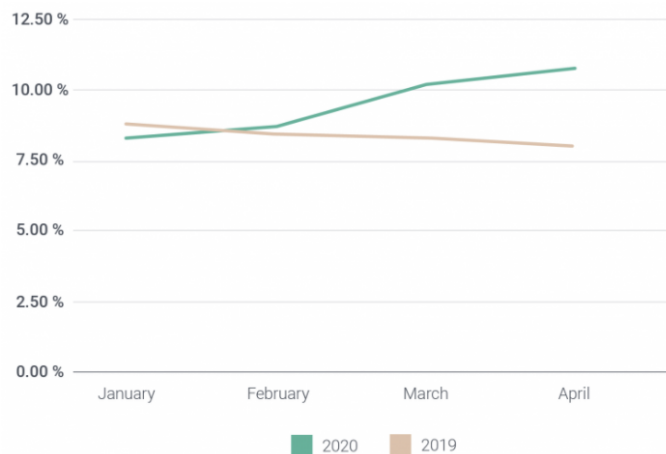
A última força identificada diz respeito ao histórico de consultas de assuntos que auxilia o fluxo de trabalho do utilizador, facilitando a experimentação das várias hipóteses e a visualização e posterior escolha da que melhor se adapta, uma vez que persiste os assuntos já testados em base de dados e a sua respetiva classificação.

Tendo em conta que o sistema E-goí já existente é vasto, a integração do *Subject Analyzer* com o mesmo pode ser um processo custoso, uma vez que existem já vários projetos integrados, de tecnologias distintas, que poderão causar incompatibilidades no sistema.

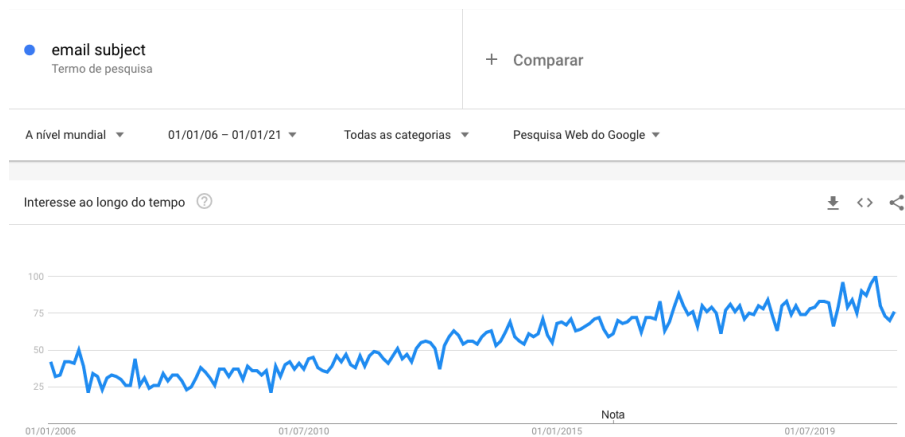
Relativamente ao aspeto externo e benéfico, identificaram-se três oportunidades fundamentais. A vasta adoção de *email marketing* por parte de empresas a nível mundial [84] considera-se um dos fatores principais à viabilidade do projeto em questão. A par com este crescimento, dada a situação pandémica que se vivencia à data de escrita deste documento [87], as tecnologias ganham ainda mais ênfase, convergindo numa maior utilização de plataformas digitais que facilitam a comunicação, conforme se pode verificar na secção 1.1, em que se projeta um crescimento de 13.3% do mercado global de *email marketing* em sete anos. Deste modo, existe um acréscimo na utilização de campanhas de *email* e, conseqüentemente, evidencia-se um aumento da taxa de aberturas de *emails* quando comparados os anos 2019 e 2020, assim como se pode verificar através da análise da figura 4.4.

---

<sup>1</sup>Códigos de personalização são códigos utilizados para personalizar o conteúdo com informação pessoal dos contactos de cada cliente [15].

Figura 4.4: Taxa de aberturas de campanhas de *email* [88]

A última oportunidade identificada, encontra-se relacionada com a proporção existente entre a atenção dada a assuntos de *email* e a interação que advém dessa atenção. Ou seja, um assunto mais apelativo ao cliente específico que o receberá terá uma taxa de interações inevitavelmente melhor. Conforme se pode verificar pela análise das figuras 4.5 e 4.6, os conceitos referentes à linha de assunto de um *email* e à taxa de aberturas têm vindo a ser cada vez mais pesquisados nos últimos quinze anos, denotando-se uma tendência de interesse crescente.

Figura 4.5: Gráfico representativo da tendência de pesquisas *online* pelo conceito *email subject* [89]

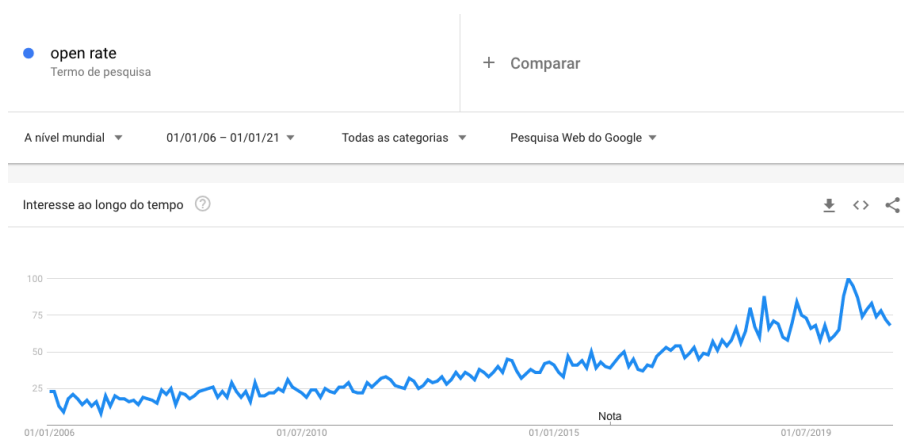


Figura 4.6: Gráfico representativo da tendência de pesquisas *online* pelo conceito *open rate* [90]

Por último, no que diz respeito às ameaças identificadas, reconhece-se apenas a concorrência existente já portadora de produtos que pretendem endereçar o mesmo problema, retratadas na secção 2.4.

## 4.2 Valor da Solução

O valor de um produto ou serviço é fundamental para qualquer empresa uma vez que é o fator fulcral para a retenção de clientes a médio e longo prazo, distinguindo-o dos demais concorrentes [91]. Deste modo, torna-se importante analisar, aquando da proposta de um produto ou serviço, o seu valor, nomeadamente o valor percecionado, com recurso à proposta de valor.

### 4.2.1 Valor, Valor para o Cliente e Valor Percecionado

Tendo em conta que a maioria das receitas de uma empresa advêm dos clientes, é importante analisar o valor que o produto terá para os mesmos. O valor para o cliente consiste no valor recebido pelo cliente final, que poderá ser um único indivíduo - *Business-to-consumer* (B2C) -, ou uma organização com vários indivíduos - *Business-to-business* (B2B). Independentemente do tipo de cliente final, o valor deriva da perceção pessoal do cliente, consoante a sua experiência e as suas necessidades, sendo que, o mesmo produto pode traduzir-se em valores distintos para diferentes clientes.

A noção de valor percecionado pelo cliente, por Kotler e Keller, representa “[...] a diferença entre a avaliação potencial de todos os benefícios e todos os custos de uma oferta e as alternativas percecionadas”<sup>2</sup> [92], sendo representada pela equação matemática:

$$CPV = Total_{CustomerValue} - Total_{CustomerCosts} \quad (4.1)$$

<sup>2</sup>Tradução livre do autor. No original “[...] the difference between the prospective customer’s evaluation of all the benefits and all the costs of an offering and the perceived alternatives.”.

No âmbito do *Subject Analyzer* da E-goi, é expectável que os clientes sintam um aumento de valor no produto, uma vez que representa uma adição de funcionalidade, de forma gratuita, que lhes permite ter uma melhor noção do impacto que as suas campanhas terão, representando assim um aumento das métricas das campanhas de *email*, nomeadamente, da taxa de aberturas e da taxa de cliques.

### 4.2.2 Proposta de Valor

Para que se proceda a uma conceção e um desenvolvimento de um novo produto, visando apresentar o produto e/ou os serviços que este oferece, quais os problemas que este propõe suprir e ainda o que será oferecido ao cliente final para que este atinga os objetivos a que se definiu aquando da aquisição do produto, é imprescindível o desenvolvimento de uma proposta de valor. Os benefícios que advêm da obtenção do produto devem também ser explícitos para o cliente.

Assim sendo, com o propósito de organizar as ideias envolvidas no processo de criação de valor, utiliza-se a ferramenta *Value Proposition Canvas* [93], mundialmente adotada na área tecnológica e que pretende auxiliar nesta fase de conceção.

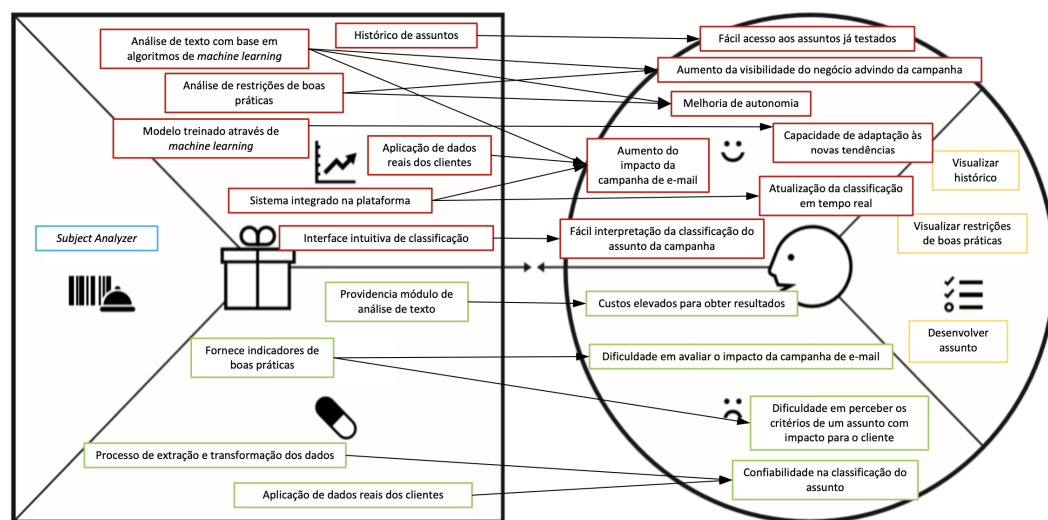


Figura 4.7: Proposta de valor [93]

Após análise da informação presente no *Value Proposition Canvas* desenvolvido na figura 4.7, denota-se uma ligação clara entre tudo que o produto/serviço tenciona oferecer e tudo o que o cliente pretende resolver. Deste modo, a proposta de valor deste projeto consiste no desenvolvimento de um serviço de análise de texto, nomeadamente, de análise de assunto de uma campanha de *email*, providenciando ao utilizador - todos os clientes E-goi - uma noção prévia do impacto que a sua campanha terá e, conseqüentemente, no aumento das interações dos clientes com a mesma, tornando o negócio mais visível. Tendo em conta que a E-goi não possui nenhuma ferramenta de análise de assunto de *email*, o *Subject Analyzer* torna-se imprescindível, uma vez que se ambiciona a realização da análise de acordo com os dados provenientes das campanhas de *email* enviadas pelos clientes da E-goi e que seja um serviço integrado na plataforma.

### 4.3 Identificação e Análise Funcional

A fase de identificação e análise funcional é composta pela identificação dos requisitos mais relevantes do produto [80]. Para tal, as empresas podem recorrer a métodos que facilitem esta análise, tal como o *Quality Function Deployment* (QFD).

O QFD tenta eliminar a lacuna existente entre o que o cliente pretende e o que o produto realmente oferece, garantindo a identificação dos principais requisitos [80], reduzindo falhas e erros. Deste modo, a sua efetivação permite à empresa colocar no mercado, com mais rapidez e menor custo, produtos de maior qualidade e orientados aos clientes [94].

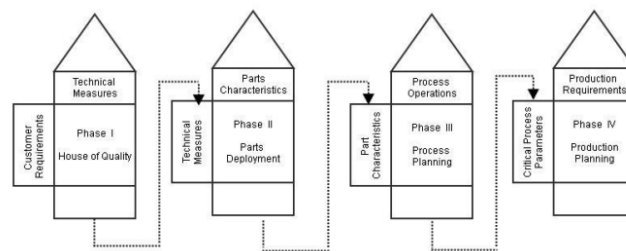


Figura 4.8: Fases do QFD [95]

O processo é composto por quatro fases, representadas na figura 4.8, sendo que cada fase é composta por uma matriz evolutiva. *House of Quality* (HOF) corresponde à primeira fase e é considerada a matriz mais importante do QFD. Esta relaciona as necessidades qualitativas do cliente aos requisitos técnicos do projeto, resultando nos requisitos técnicos mais importantes em relação aos requisitos do cliente e à análise competitiva [95].

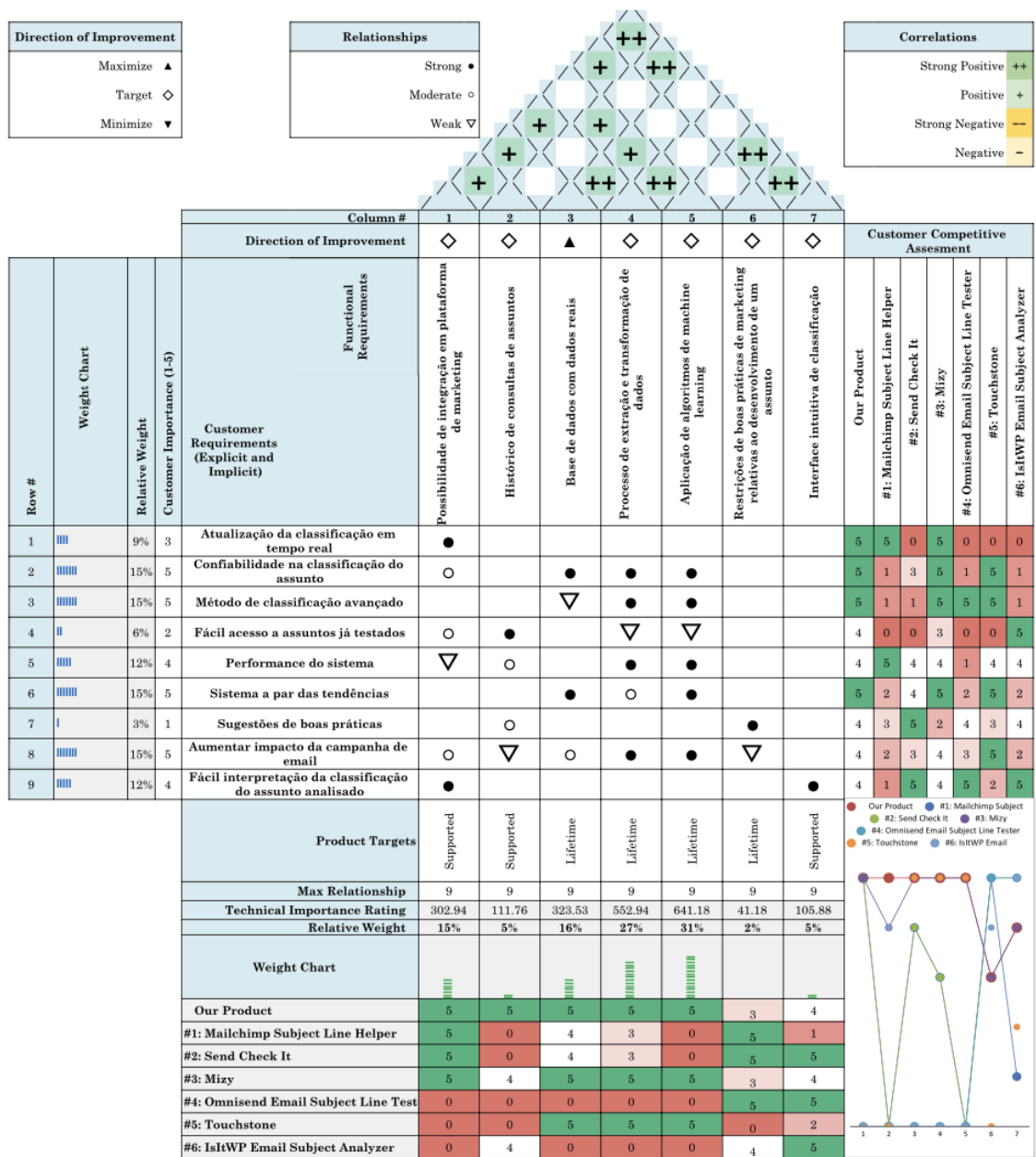


Figura 4.9: HOF

Analisando a HOF do presente projeto, representada na figura 4.9, é possível identificar, através do peso relativo, que o uso de algoritmos de ML, o processo de extração e transformação dos dados, a base de dados com dados reais dos clientes e a integração do produto com a plataforma E-goi são as características de qualidade mais importantes evidenciadas dos requisitos do cliente, uma vez que possuem os pesos de 31%, 27%, 16% e 15%, respetivamente.

### 4.4 Conceção de Alternativas

Depois de demonstrada a validade da oportunidade identificada, segue-se a fase de conceção de alternativas. Esta é uma fase iterativa que diz respeito à geração, desenvolvimento e

maturação de uma ideia concreta [82].

A fim de concluir a melhor abordagem de um analisador de texto que cumprisse os requisitos, foi realizada uma sessão de *brainstorming* interna, de forma a serem analisadas e discutidas, com base nos requisitos definidos, quais as alternativas mais viáveis. Deste modo, as alternativas finais obtidas foram:

- Integração de analisador de texto já existente com a plataforma;
- Desenvolvimento de um analisador de texto de raiz;
- Utilização de um analisador de texto sem integração.

## 4.5 Análise e Avaliação

Findada a fase de conceção de alternativas, torna-se necessário analisar as alternativas existentes a fim de dar uma solução concreta à oportunidade identificada. Tendo em conta que a seleção da alternativa é um processo fundamental para o futuro do negócio [82], optou-se pela aplicação de um método multicritério de apoio à decisão, o *Analytic Hierarchy Process* (AHP). Este, proposto por Thomas L. Saaty, segue uma abordagem racional, baseada na utilização de técnicas matemáticas, e no cruzamento entre as alternativas e os critérios definidos [96].

Este é um método baseado no método newtoniano e cartesiano de pensar, isto é, segue a lógica da decomposição e divisão do problema em fatores, até ao nível mais baixo [97], de forma a que estes possam ser associados e sintetizados. Pode ser dividido em três etapas principais: divisão hierárquica, definição de prioridades e consistência lógica [97].

A divisão hierárquica auxilia a compreensão e avaliação do problema, sendo que o primeiro nível da hierarquia corresponde ao propósito geral do problema, o segundo aos critérios e, por último, o terceiro às alternativas [97]. A fim de atingir a solução final e tendo em consideração que o propósito geral é a definição da abordagem mais adequada de desenvolvimento de um analisador de assunto de campanhas de *email*, esta deve ser avaliada com base nos seguintes critérios:

- Utilização de dados reais dos clientes E-goi;
- Sistema a par das tendências dos clientes;
- Integração com o fluxo de criação e envio de uma campanha de *email*, no sistema E-goi.

O primeiro ponto, referente à utilização de dados reais dos clientes E-goi, é imprescindível dado que potencia o impacto da campanha de *email*, uma vez que a base de dados utilizada pelo sistema será composta por vários assuntos de campanhas reais já enviadas e as respetivas taxas de aberturas, conferindo-lhe uma maior confiabilidade em comparação com um sistema baseado em boas práticas. O segundo critério diz respeito a um sistema a par das tendências dos clientes, em virtude de se tornar relevante um sistema que acompanhe a evolução do mundo e as tendências dos clientes. Por último, é fundamental que o analisador seja parte integrante do sistema E-goi, isto é, que pertença ao fluxo de criação e envio de uma campanha de *email*, de forma a que o utilizador não sinta necessidade de interromper o fluxo para utilizar serviços externos.

Na figura 4.10 apresenta-se um gráfico representativo da estrutura hierárquica do método AHP a aplicar, considerando o objetivo final, as alternativas identificadas na secção 4.4 e os critérios a aplicar.

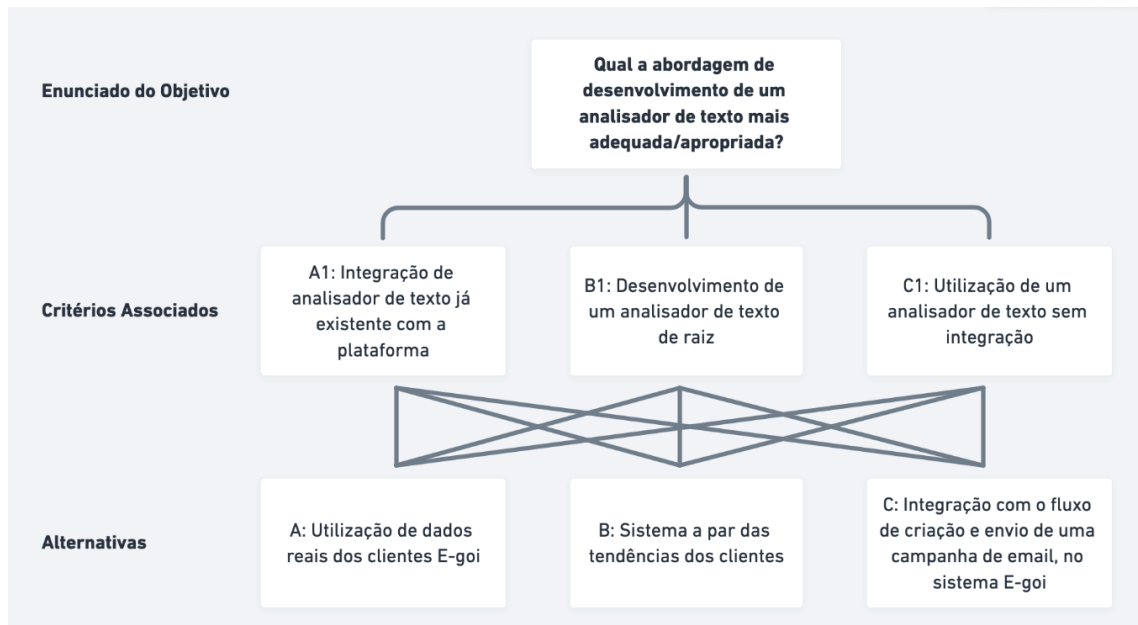


Figura 4.10: Estrutura hierárquica do método AHP [98]

Uma vez terminada a divisão hierárquica, segue-se a segunda fase do método AHP, onde é necessário definir prioridades entre os diferentes pares de critérios escolhidos. Para esta definição utilizou-se a escala de Saaty [99], proposta por Thomas L. Saaty aquando da apresentação do AHP, que consiste numa escala numerada de 1 a 9, onde 1 corresponde a uma igualdade de importância entre critérios, e 9 corresponde a uma diferença de importância absoluta. Assim sendo, representa-se na tabela 4.1 a matriz de comparação entre critérios, considerando que:

- A: Utilização de dados reais dos clientes E-goi;
- B: Sistema a par das tendências dos clientes;
- C: Integração com o fluxo de criação e envio de uma campanha de *email*, no sistema E-goi.

Tabela 4.1: Matriz de comparação entre critérios

	A	B	C
A	1	3	5
B	$\frac{1}{3}$	1	3
C	$\frac{1}{5}$	$\frac{1}{3}$	1

Analisando a matriz de comparação pode concluir-se que o critério A, definido pela utilização de dados reais dos clientes E-goi, é levemente mais importante e prioritário que o sistema a par das tendências e é, também, fortemente mais importante que a integração com o fluxo

de criação e envio de uma campanha de *email*. Também é possível observar que o sistema a par das tendências é levemente mais importante que a integração.

Após a definição da matriz de comparação entre critérios, é importante identificar a prioridade relativa de cada critério, obtendo o vetor de prioridades, através da normalização dos valores da matriz. A matriz normalizada de comparação entre critérios e respectivo vetor de prioridades relativas encontra-se presente na tabela 4.2, sendo a prioridade relativa calculada através da média aritmética de cada linha da matriz normalizada.

Tabela 4.2: Matriz normalizada de comparação entre os critérios e vetor de prioridades relativas

	A	B	C	Prioridade Relativa
A	$\frac{17}{26}$	$\frac{9}{13}$	$\frac{5}{9}$	0.63
B	$\frac{11}{51}$	$\frac{3}{13}$	$\frac{1}{3}$	0.26
C	$\frac{3}{23}$	$\frac{1}{13}$	$\frac{1}{9}$	0.11

Após a priorização dos critérios, é necessário priorizar as soluções, recorrendo a matrizes de comparação que cruzam todas as alternativas existentes com cada critério de forma individual.

Considerando:

- A1: Integração de analisador de texto já existente com a plataforma;
- B1: Desenvolvimento de um analisador de texto de raiz;
- C1: Utilização de um analisador de texto sem integração.

Tabela 4.3: Matriz de comparação de alternativas para o critério A

	A1	B1	C1	Prioridade Relativa
A1	1	$\frac{1}{3}$	9	0.32
B1	3	1	9	0.63
C1	$\frac{1}{9}$	$\frac{1}{9}$	1	0.05

Tabela 4.4: Matriz de comparação de alternativas para o critério B

	A1	B1	C1	Prioridade Relativa
A1	1	$\frac{1}{3}$	3	0.33
B1	3	1	$\frac{1}{2}$	0.37
C1	$\frac{1}{3}$	2	1	0.30

Tabela 4.5: Matriz de comparação de alternativas para o critério C

	A1	B1	C1	Prioridade Relativa
A1	1	1	5	0.44
B1	1	1	7	0.49
C1	$\frac{1}{5}$	$\frac{1}{7}$	1	0.08

Pela análise das matrizes de comparação das alternativas, presentes nas tabelas 4.3, 4.4 e 4.5, pode concluir-se que, para os três critérios definidos, a alternativa relativa ao desenvolvimento de um analisador de texto de raiz é a mais adequada, uma vez que apresenta uma priorização mais elevada em comparação com as restantes alternativas - 0.63 para a utilização de dados reais dos clientes E-goí, 0.37 para o sistema a par das tendências dos clientes e, por fim, 0.49 para a integração com o fluxo de criação e envio de uma campanha de *email*.

Seguidamente à priorização das várias alternativas, segue-se a definição da prioridade composta para estas, através da junção dos vetores de prioridades das matrizes de comparação de alternativas por critérios, numa só matriz:

Tabela 4.6: Matriz de prioridades relativas das alternativas

	A	B	C
A1	0.32	0.33	0.44
B1	0.63	0.37	0.49
C1	0.05	0.30	0.08

A prioridade composta é obtida através da multiplicação da matriz de prioridades relativas dos critérios, presente na tabela 4.2, pela matriz de prioridades relativas das alternativas. A matriz de prioridades compostas encontra-se representada na tabela 4.7.

Tabela 4.7: Prioridades compostas

A1	0.33
B1	0.55
C1	0.12

Deste modo, é possível escolher qual a alternativa mais apropriada. No entanto, é necessário garantir a consistência, que representa a terceira fase do processo. Para isto, é necessário calcular a Razão de Consistência (RC) e verificar se o resultado é menor do que 0.1, através da seguinte fórmula, em que IC corresponde ao índice de consistência e IA ao índice aleatório:

$$RC = \frac{IC}{IA} \quad (4.2)$$

O Índice Aleatório (IA) é obtido através da matriz de índice aleatório, presente na tabela 4.8, consoante a dimensão da matriz. Tendo em conta que a dimensão da matriz é 3, o valor de IA será 0.58.

Tabela 4.8: Matriz de índice aleatório

Dimensão da matriz	1	2	3	4	5	6	7	8	9	10
Índice de consistência	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.19

Tendo o valor de IA, é necessário calcular o valor correspondente ao Índice Consistência (IC). Para tal, aplica-se a seguinte fórmula:

$$IC = \frac{(\lambda_{max} - n)}{(n - 1)} \quad (4.3)$$

Deste modo, calculou-se a matriz de valores obtida, que se representa na tabela 4.9, através da multiplicação da matriz de comparação com o vetor de prioridades, a fim de ser possível o cálculo do valor do  $\lambda_{max}$ .

Tabela 4.9: Matriz de valores

A	1.94
B	0.79
C	0.32

De seguida, calculou-se o valor de  $\lambda_{max}$ , através da divisão da média dos valores obtidos na matriz pelos valores do vetor de prioridades relativas:

$$\lambda_{max} = \frac{1.94/0.63 + 0.79/0.26 + 0.32/0.11}{3} = 3.04 \quad (4.4)$$

Obtido o  $\lambda_{max}$ , prossegue-se o cálculo do IC:

$$IC = \frac{3.04 - 3}{3 - 1} = 0.02 \quad (4.5)$$

Por fim, calculadas todas as variáveis, aplica-se a fórmula da RC:

$$RC = \frac{0.02}{0.58} = 0.03 \quad (4.6)$$

Uma vez que  $0.03 < 0.1$ , conclui-se que os critérios estão consistentes e que o resultado do processo AHP é válido. Por isto, e tendo em consideração os valores obtidos nas prioridades compostas presentes na tabela 4.7, em que:

- Solução A: 0.33
- Solução B: 0.55
- Solução C: 0.12

É possível comprovar que a melhor solução a adotar para o problema identificado é a B, o desenvolvimento de um analisador de texto de raiz, uma vez que foi o que obteve maior resultado - 0.55.

## Capítulo 5

# Análise e Design da Solução

Este capítulo tem como propósito principal apresentar os conceitos de negócio que envolvem o projeto em questão, com recurso ao modelo do domínio e à definição dos requisitos funcionais e não funcionais. A par com isto, apresenta-se, também, os diagramas de componentes e de implantação, assim como vistas de cenários e *mock-ups* que são exemplos de artefactos resultantes do *design*, que representa um processo de criação a fim de oferecer às partes interessadas artefactos de *software* [100].

### 5.1 Modelo de Domínio

O modelo de domínio tem como principal propósito dar a conhecer, através de uma representação visual estruturada, os conceitos de negócio alusivos ao projeto [101]. Deste modo, apresenta-se na figura 5.1, o modelo de domínio definido com recurso à notação *Unified Modeling Language* (UML), no qual se encontram as classes conceptuais organizadas de acordo com a sua interligação e respetivas cardinalidades.

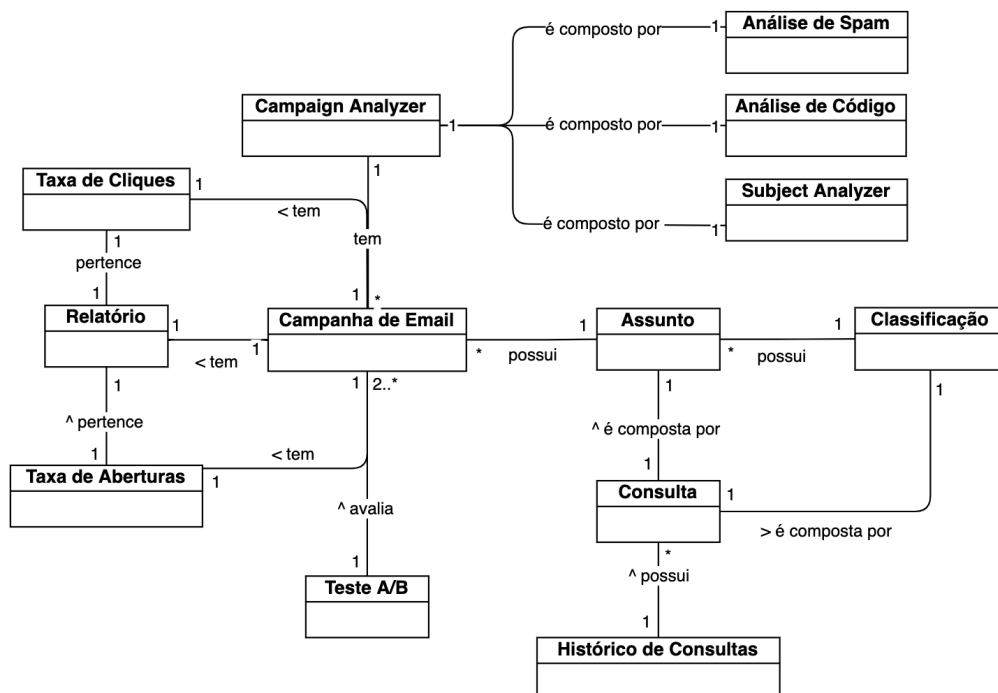


Figura 5.1: Modelo de domínio

Assim, apresenta-se, de seguida, uma breve descrição das classes identificadas:

- Campanha de *Email*: Entidade representativa de uma campanha de *email* (conceito abordado na secção 2.1) que pode ser criada e editada na plataforma E-goi pelos clientes, de forma a oferecer conteúdo e ofertas relevantes a um conjunto de contactos definidos pelos mesmos;
- Relatório: Entidade representativa do relatório (conceito abordado na subsecção 2.1.1) gerado após o envio de uma campanha de *email*, composto por várias métricas de envio da mesma, nomeadamente a taxa de aberturas e a taxa de cliques;
- Taxa de Aberturas: Entidade representativa da métrica relativa à taxa de aberturas presente no relatório gerado (conceito abordado na subsecção 2.1.1), definida pela relação entre o número de aberturas da campanha de *email* e o número de campanhas entregues;
- Taxa de Cliques: Entidade representativa da métrica relativa à taxa de cliques presente no relatório gerado (conceito abordado na subsecção 2.1.1), definida pela relação entre o número de cliques em quaisquer elementos da campanha de *email* e o seu número de aberturas;
- *Campaign Analyzer*: Entidade representativa do módulo de análise de campanha existente na plataforma E-goi e que faz parte da última fase do processo de criação e envio de uma campanha de *email* (conceito abordado na secção 2.2). É composto por submódulos relativos à análise de *spam*, análise de código HTML e CSS e análise de assunto - *Subject Analyzer*, representando, este, o presente projeto;
- Análise de *Spam*: Entidade representativa do módulo de análise de *spam* que se baseia no filtro de *spam SpamAssassin* (conceito abordado na subsecção 2.2);
- Análise de Código: Entidade representativa do módulo de análise de código que analisa o código HTML, através de *links* não funcionais e o código CSS, tendo em considerações as boas práticas de programação (conceito abordado na subsecção 2.2);
- *Subject Analyzer*: Entidade representativa do módulo de análise de assunto que visa classificar um assunto de 0 a 5 estrelas com base num modelo DL treinado com recurso a dados provenientes dos clientes E-goi;
- Teste A/B: Entidade representativa de um teste que mede a qualidade entre várias campanhas, consoante os critérios aberturas, aberturas únicas, cliques e cliques únicos (conceito abordado na secção 2.3);
- Assunto: Entidade representativa do assunto da campanha de *email* que pode ser analisado e classificado pelo *Subject Analyzer*;
- Classificação: Entidade representativa da classificação do assunto fornecida pelo *Subject Analyzer*, representada através de uma escala de valores discretos num intervalo de 0 a 5;
- Consulta: Entidade representativa de uma consulta efetivada ao sistema de análise, composta pelo assunto e a respetiva classificação, a fim de ser possível disponibilizar ao utilizador todos os assuntos analisados, através do histórico de consultas;
- Histórico de Consultas: Entidade representativa do conjunto de consultas de assuntos realizadas para uma campanha de *email*.

## 5.2 Requisitos Funcionais

De forma a representar os requisitos funcionais do sistema, recorreu-se a um digrama de casos de uso, em notação UML. Este encontra-se representado pela figura 5.2.

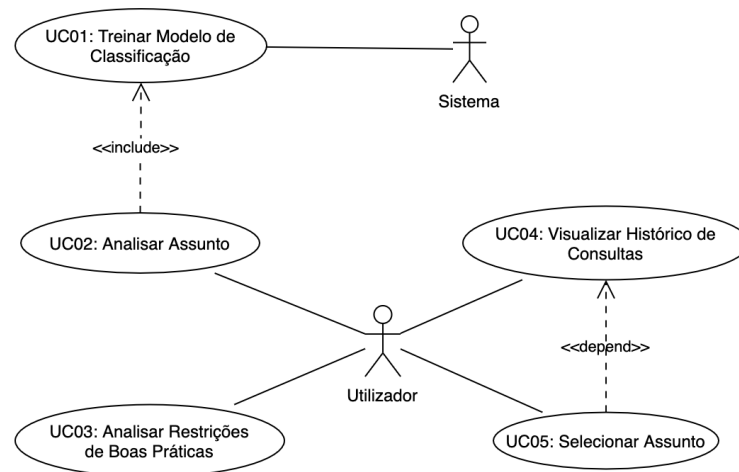


Figura 5.2: Diagrama de casos de uso

Os casos de uso apreendidos da análise da figura 5.2, encontram-se brevemente descritos de seguida, juntamente com o seu fluxo normal de execução:

- **UC01: Treinar Modelo de Classificação:** O sistema inicia o processo de treino do modelo de classificação, automaticamente. Primeiramente, extrai os dados provenientes das novas campanhas enviadas. De seguida, faz um pré-processamento dos dados lidos. Por último, treina o modelo com os dados resultantes.
- **UC02: Analisar Assunto:** O utilizador inicia o processo de desenvolvimento de um assunto. O sistema reconhece o pedido e inicia o módulo de análise do assunto inserido, classificando-o com base no modelo treinado, guardando em base de dados a consulta (assunto e classificação) e devolvendo *feedback* ao utilizador.
- **UC03: Analisar Restrições de Boas Práticas:** O utilizador inicia o processo de análise de restrições de boas práticas relativas ao assunto inserido. O sistema reconhece o pedido e inicia a análise relativa a métricas como a presença de *emojis*, contagem de caracteres e palavras, e pontuação, e devolve *feedback* ao utilizador.
- **UC04: Visualizar Histórico de Consultas:** O utilizador inicia o processo de visualizar o histórico de consultas. O sistema reconhece o pedido e apresenta todas as consultas efetuadas para a campanha específica, com o assunto e classificação obtida.
- **UC05: Selecionar Assunto:** O utilizador inicia o processo de seleção de assunto através do histórico de consultas. O sistema reconhece o pedido e altera o assunto da campanha para o escolhido.

Uma vez apresentados os casos de uso do sistema, torna-se relevante descrever os casos de uso arquiteturalmente mais relevantes, através de diagramas *System Sequence Diagram* (SSD).

Tendo em consideração que o *Subject Analyzer* se baseia em técnicas de DL para classificação, o caso de uso considerado arquiteturalmente mais relevante é o UC01 - **Treinar Modelo de Classificação**.

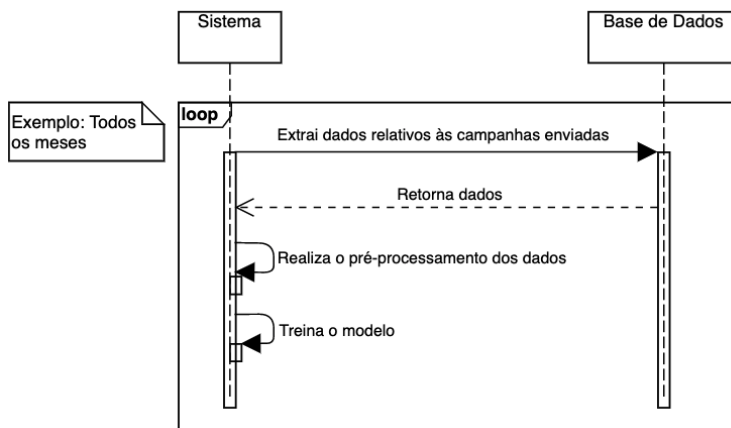


Figura 5.3: SSD do caso de uso Treinar Modelo de Classificação

De forma a manter o modelo a par das tendências dos clientes torna-se essencial treiná-lo constantemente. Assim, como se pode analisar pela figura 5.3, pretende-se que este caso de uso seja realizado automaticamente pelo sistema, de forma recorrente, por exemplo mensalmente, com recurso a uma *cron-job*<sup>1</sup>. Para tal, o sistema extrai os dados relativos às novas campanhas enviadas, especificamente a linha de assunto e a taxa de aberturas correspondente. Extraídos os dados, o sistema realiza o pré-processamento dos mesmos com recurso às técnicas de NLP mencionadas na secção 3.1. Por último, treina o modelo com base nos dados resultantes.

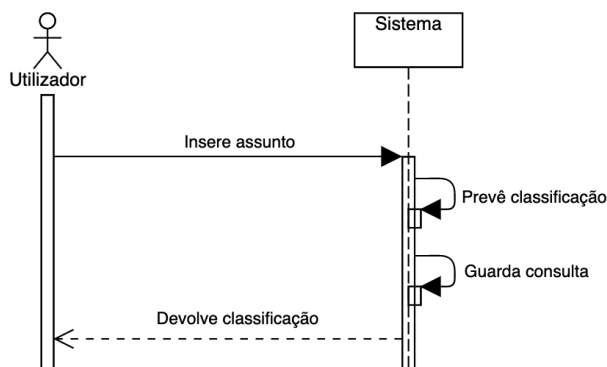


Figura 5.4: SSD do caso de uso Analisar Assunto

No que diz respeito ao ator utilizador, o caso de uso principal é o UC02 - **Analisar Assunto**. Apesar da figura 5.4 apresentar uma arquitetura simples, este baseia-se no modelo treinado (representado no UC01) para realizar previsões acerca da linha de assunto. Deste modo, o utilizador insere o assunto e o sistema prevê a classificação. De seguida, persiste a consulta efetuada na base de dados e, por fim, devolve a classificação ao utilizador.

<sup>1</sup> *Cron-job* é uma ferramenta de programação de tarefas baseados em intervalos de tempo [102].

O caso de uso **Analisar Assunto** pode ser iterado o número de vezes que o utilizador desejar. Pode ser utilizado em conjunto com os casos de uso **Visualizar Histórico de Consultas** e **Selecionar Assunto**, de forma a tornar o processo de desenvolvimento do assunto mais eficiente. Assim, o utilizador tem a possibilidade de visualizar o histórico de consultas, analisar as classificações já obtidas e optar por selecionar um assunto já testado ou voltar a executar o caso de uso até obter o assunto com a classificação que deseja.

## 5.3 Requisitos Não Funcionais

Os requisitos não funcionais são os requisitos que não interferem diretamente com o sistema. Ao invés disso, estabelecem a forma como o sistema se comporta em situações específicas. A fim da sua definição, recorreu-se ao modelo FURPS+, um acrónimo para *functionality, usability, reliability, performance* e *supportability*, onde o + possui o intuito de identificar as restrições de *design*, restrições de implementação, restrições de interface e restrições físicas, que representam os requisitos resultantes da evolução do modelo [101].

### 5.3.1 Functionality

*Functionality* representa todos os requisitos do sistema relativos à funcionalidade do mesmo, não representados pelos casos de uso [101], definidos na secção 5.2. Assim, apresentam-se os requisitos definidos:

- O sistema deve comunicar com o produto E-*goi* através de uma API;
- O treino do modelo deve ser realizado recorrentemente através de uma *cron-job*.

### 5.3.2 Usability

*Usability* representa os requisitos baseados na interação com o utilizador [101]. Assim, apresentam-se os requisitos definidos:

- A interface de comunicação com o utilizador deve ser simples de usar;
- O sistema deve tratar os erros fornecendo detalhes claros sobre o mesmos.

### 5.3.3 Reliability

*Reliability* diz respeito ao comportamento do sistema, nomeadamente à sua integridade, conformidade e interoperabilidade [101]. Assim, apresenta-se o requisito definido:

- O sistema deve manter-se disponível após a ocorrência de um erro.

### 5.3.4 Performance

*Performance* avalia os requisitos de desempenho do sistema sob diferentes cargas de uso, incluindo questões como tempo de resposta, tempo de carregamento e tempo recuperação de falha [101]. Assim, apresentam-se os requisitos definidos:

- O sistema deve devolver uma classificação em tempo real, não obrigando o utilizador a executar uma ação adicional para a obter;
- O tempo de resposta do sistema de análise de assunto da campanha deve ser adequado de modo a que não prejudique a experiência do utilizador.

### 5.3.5 Supportability

*Supportability* engloba requisitos relacionados com adaptabilidade, manutenção, escalabilidade, entre outros [101]. Assim, apresentam-se os requisitos definidos:

- O sistema deve ser compatível com a plataforma E-goi, estando integrado no fluxo de criação de campanhas;
- O sistema deve estar preparado para a substituição do modelo aplicado.

### 5.3.6 Restrições de Implementação

Restrições de implementação especificam e/ou restringem requisitos de implementação [101]. Assim, apresentam-se os requisitos definidos:

- O sistema deve ser desenvolvido de forma modular, de modo a que a sua integração com o E-goi e a sua manutenção sejam processos fáceis;
- A integração com a API deverá ser feita através do contrato *Representational State Transfer* (REST);
- O sistema deve ser desenvolvido com recurso às tecnologia *Tensorflow*.

### 5.3.7 Restrições de Interface

Restrições de interface especificam e/ou restringem as funcionalidades inerentes à interface. Assim, apresenta-se o requisito definido:

- A interface do sistema deve ser bastante intuitiva sendo que deve devolver uma classificação gráfica, de 0 a 5, sob o formato de estrelas e em tempo real.

## 5.4 Abordagens Possíveis e Escolha da Arquitetura

Assim sendo, são apresentadas de seguida duas possíveis abordagens de *design* com recurso a diagramas de componentes. Estes são apresentados numa perspetiva de granularidade grossa uma vez que não se pretende caracterizar em detalhe as especificidades de cada componente mas representar as decisões arquiteturais mais relevantes com o propósito de concluir qual a melhor abordagem a adotar.

### 5.4.1 Subject Analyzer com Segregação de Responsabilidades e Exposição na API Pública

Na figura 5.5 encontra-se o diagrama de componentes relativo à abordagem do *Subject Analyzer* com segregação de responsabilidades e exposição na API pública, assim como as interações entre os seus componentes.

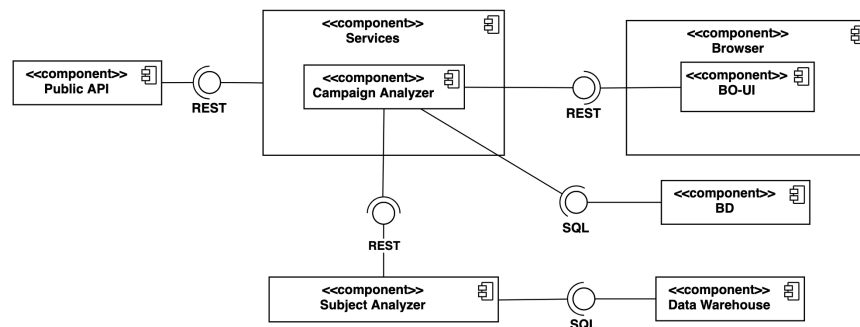


Figura 5.5: Diagrama de componentes do sistema com segregação de responsabilidades e exposição na API pública

De acordo com o diagrama, descrevem-se os seguintes componentes:

- *Public API*: Componente representativo da API pública [103] oferecida pela E-goi aos seus clientes que permite o usufruto das funcionalidades da plataforma via API;
- *Services*: Componente responsável por agregar todos os módulos de pedidos relativos ao negócio, nomeadamente o *Campaign Analyzer*;
  - *Campaign Analyzer*: Componente representativo dos pedidos relativos à análise de campanha, nomeadamente da análise do assunto.
- *BO-UI*: Componente representativo da interface gráfica da plataforma E-goi;
- *Subject Analyzer*: Componente responsável pelo módulo de análise de assunto;
- *BD*: Componente representativo da base de dados que tem a responsabilidade de armazenar os dados relativos às consultas efetuadas;
- *Data Warehouse*: Componente representativo da base de dados que tem a responsabilidade de alojar os dados que servem o modelo de classificação, nomeadamente os assuntos das campanhas e respetivas taxas de aberturas.

#### 5.4.2 Subject Analyzer com Centralização de Responsabilidades

Na figura 5.6 encontra-se o diagrama de componentes relativo à abordagem do *Subject Analyzer* com centralização de responsabilidades, assim como as interações entre os seus componentes.

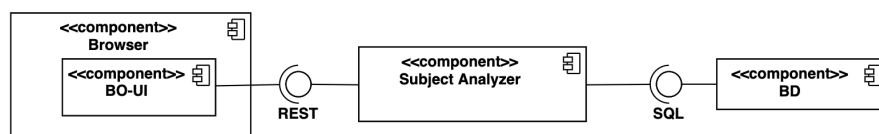


Figura 5.6: Diagrama de componentes do sistema com segregação de responsabilidades e exposição na API pública

O diagrama representado na figura acima é constituído pelos componentes já apresentados na subsecção 5.4.1. Contudo, nesta abordagem de *design* não se encontra presente o

componente *Services* e, conseqüentemente, o componente *Public API*, uma vez que o *Services* é o responsável por fornecer informação a fim de ser possível a utilização da API por parte dos utilizadores. Para além destes, o componente *Data Warehouse* também não se encontra ilustrado, passando o componente *BD* a garantir a persistência e gestão dos dados das consultas dos assuntos, bem como dos dados que servem o modelo de classificação, denotando-se assim uma centralização de responsabilidades relativa ao componente *BD*.

### 5.4.3 Comparação de Alternativas

A ideia de se centralizar a persistência dos dados numa só base de dados considera-se interessante uma vez que facilitaria a logística de alojamento das mesmas. No entanto, implicaria a utilização da mesma tecnologia de persistência para os dados a serem utilizados no *Subject Analyzer* e no *Services*. Tal não se revelaria caso a arquitetura presente na subsecção 5.4.1 fosse adotada, na medida em que nesta se pretende distribuir essa persistência em duas bases de dados distintas, pelo que, apesar da logística de alojamento de serviços distribuídos ser mais complexa, o ganho obtido no que diz respeito à escalabilidade e manutenção prevalece. A ideia inerente à primeira arquitetura encontra-se em conformidade com o princípio de *design Separation Of Concerns* (SoC), princípio este que visa subdividir um problema complexo em problemas mais simples, atribuindo responsabilidades a componentes menos complexos, de fácil gestão e manutenção. De acordo com Dijkstra, “[*Separation of Concerns*], mesmo que não seja perfeitamente possível, ainda é a única técnica disponível, que eu conheça, para ordenar eficazmente os pensamentos de alguém.”<sup>2</sup> [104].

Denota-se, ainda, uma diferença entre as duas abordagens quanto à utilização do componente *Services*. Este componente é utilizado pela API pública do sistema E-goi, podendo usufruir das funcionalidades do *Subject Analyzer* na plataforma e via API, sem que seja necessário seguir todo um fluxo de uma campanha para analisar um assunto, permitindo assim, como trabalho futuro (c.f secção 8.3), utilizar o *Subject Analyzer* para casos de uso distintos. Assim, considera-se que a utilização do componente *Services* seria uma vantagem quando comparadas as abordagens.

Dito isto, optou-se pela abordagem referente à segregação de responsabilidades com possibilidade de exposição do serviço na API pública do E-goi.

## 5.5 Design Detalhado

Na secção de *design* detalhado aborda-se o conceito de *design* mais específico, pormenorizando a solução a desenvolver. Assim, apresenta-se de seguida um desenho detalhado da solução arquitetural definida na subsecção 5.4.3, constituído pelo diagrama de componentes, diagrama de implantação, vistas de cenários e *mock-ups* do sistema.

### 5.5.1 Diagrama de Componentes Detalhado do Sistema

Uma vez analisadas e comparadas as arquiteturas possíveis, numa perspetiva de granularidade grossa, torna-se fundamental detalhar a arquitetura definida passando assim a uma granularidade fina. Assim sendo, encontram-se representados, na figura 5.7, os diversos componentes, interligações e dependências entre eles.

---

<sup>2</sup>Tradução livre do autor. No original “[*Separation of Concerns*], even if not perfectly possible, is yet the only available technique for effective ordering of one’s thoughts, that I know of.”.

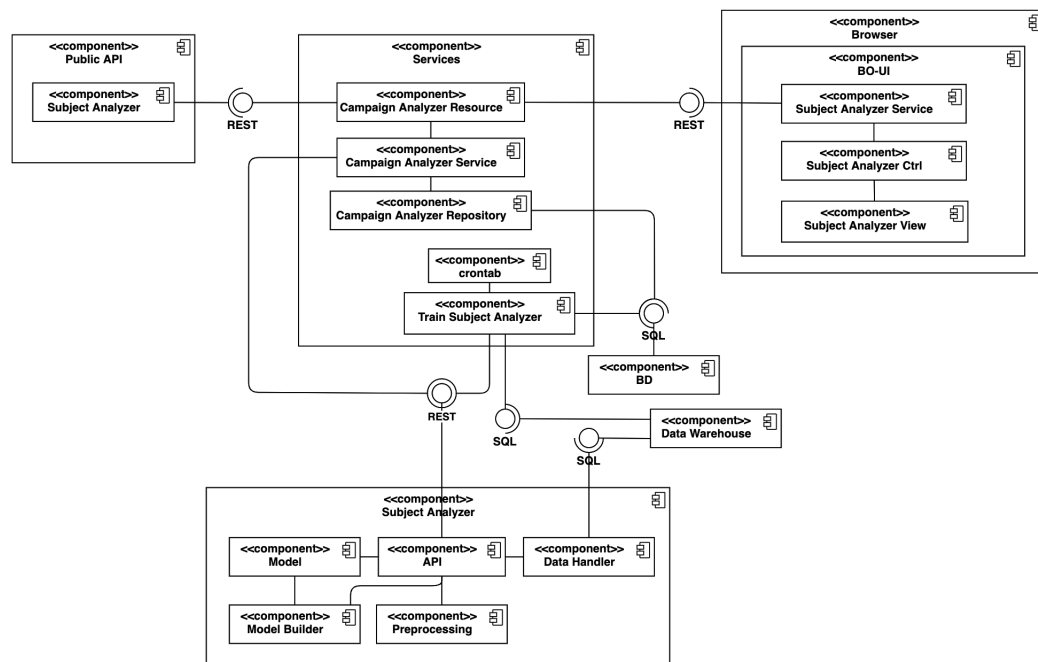


Figura 5.7: Diagrama de componentes do sistema detalhado

Através da análise do digrama é possível verificar a existência de componentes já descritos na subsecção 5.4.1, nomeadamente: *Public API*, *Services*, *BO-UI*, *Subject Analyzer*, *BD* e *Data Warehouse*. No entanto, dado o aumento do nível de detalhe, passa-se a explicar os restantes componentes que constituem o sistema:

- *Public API*
  - *Subject Analyzer*: Componente representativo do pedido relativo à análise de assunto presente no componente API pública.
- *Services*
  - *crontab*: Componente responsável por agregar todas as *cron-job* do sistema E-goji, incluindo a *cron-job* encarregue do treino do modelo;
  - *Train Subject Analyzer*: Componente representativo da *cron-job* responsável por treinar o modelo de classificação de assunto;
  - *Campaign Analyzer Resource*: Componente representativo do *resource* onde são realizados os pedidos relativos à análise da campanha, nomeadamente, a análise do assunto;
  - *Campaign Analyzer Service*: Componente representativo do serviço responsável por agregar toda a lógica referente à análise da campanha, nomeadamente, da chamada realizada à API do *Subject Analyzer* para obter a classificação do assunto a analisar;
  - *Campaign Analyzer Repository*: Componente representativo do repositório, responsável por realizar os pedidos às bases de dados, nomeadamente, por persistir a consulta efetuada aquando da análise de um assunto.
- *BO-UI*

- *Subject Analyzer Service*: Componente representativo da camada de serviços responsável por gerir os pedidos efetuados ao *Services* referentes à funcionalidade de classificação de assunto;
  - *Subject Analyzer Ctrl*: Componente representativo da camada de controladores, responsável por gerir a funcionalidade de classificação de assunto, fazendo a ligação entre o *Subject Analyzer Service* e o *Subject Analyzer View*;
  - *Subject Analyzer View*: Componente representativo da camada de vistas existente no *BO-UI*, responsável pela apresentação visual do conteúdo do classificador de assunto ao utilizador.
- *Subject Analyzer*
    - *API*: Componente representativo da API responsável pela gestão de assuntos e classificação;
    - *Model*: Componente representativo do modelo treinado de classificação;
    - *Model Builder*: Componente responsável pelo treino do modelo;
    - *Data Handler*: Componente responsável por manipular os dados;
    - *Preprocessing*: Componente responsável pelo pré-processamento dos dados.

### 5.5.2 Diagrama de Implantação do Sistema

De forma a representar a estrutura física do sistema, recorreu-se a um diagrama de implantação em notação UML, ilustrado na figura 5.8.

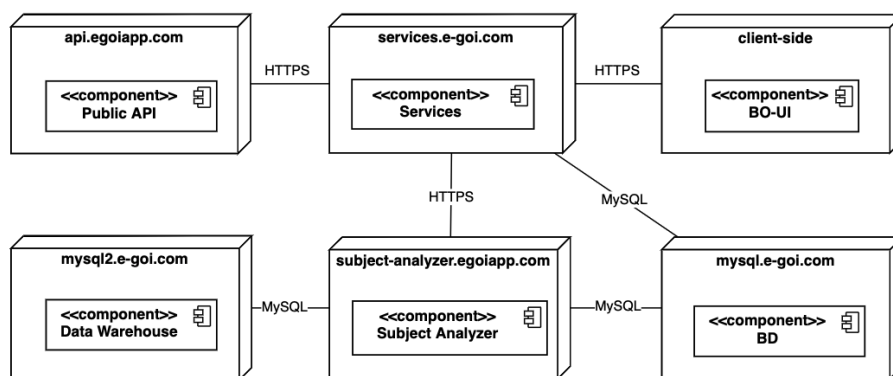


Figura 5.8: Diagrama de implantação do sistema

Assim, os dispositivos físicos que constituem o sistema são os seguintes:

- *api.egoapp.com*: Domínio responsável pela implantação da API pública do E-goi;
- *services.e-goi.com*: Domínio responsável pela implantação do componente *Services*;
- *client-side*: Máquina do cliente na qual está implantada a interface denominada *BO-UI*;
- *subject-analyzer.egoapp.com*: Domínio responsável pela implantação do *Subject Analyzer*;
- *mysql.e-goi.com*: Máquina responsável por agregar o componente *Data Warehouse*;

- *mysql2.e-goi.com*: Máquina responsável por agregar o componente BD.

Para o presente projeto, apesar de grande parte dos dispositivos já existirem como parte integrante do sistema E-goi, sentiu-se a necessidade de criar uma máquina dedicada ao *Subject Analyzer*, uma vez que o treino do modelo requer uma capacidade de *hardware* elevada e, ainda, uma máquina responsável por alojar o componente *Data Warehouse*.

A modularidade que caracteriza o sistema permite agilizar processos de manutenção tecnológica, uma vez que a atualização se reflete a nível individual, sem que seja necessário recorrer a uma reestruturação do sistema para casos mais severos de manutenção.

### 5.5.3 Vistas de Cenários do Sistema

As vistas de cenários têm como propósito a representação da interação funcional e das relações existentes entre os componentes do sistema e, para as representar, recorre-se à utilização de diagramas de sequência, em notação UML.

Devido à sua relevância arquitetural, na figura 5.9 encontra-se representada a vista de cenários do *UC01: Treinar Modelo*, no qual se retrata a interação entre os componentes intervenientes nesse mesmo caso de uso, permitindo, ainda, a identificação do fluxo de todo o processamento. O *UC01* inicia-se de modo automático e recorrente (c.f subsecção 5.3.1), com base numa *cron-job* que se encontra alojada no componente *crontab* que tem a responsabilidade de agregar todas as *cron-job* do sistema E-goi. Uma vez iniciado o processo, a responsabilidade assenta no componente *Train Subject Analyzer*, que trata de obter os dados relativos às campanhas enviadas e a todas as suas métricas, guardando os dados obtidos na *Data Warehouse*. Uma vez terminado o processo de obtenção dos dados, passa-se ao pré-processamento. Esta fase é composta por todas as funções que contêm a lógica de negócio associada ao pré-processamento dos dados (c.f secção 6.2). Após o tratamento dos dados, e a fim de terminar a execução deste caso de uso, treina-se o modelo através do componente *Model Builder* existente no *Subject Analyzer*. Aquando do término do caso de uso, apresenta-se uma representação visual do *feedback* do processo que, em condições normais, informa do sucesso da operação.

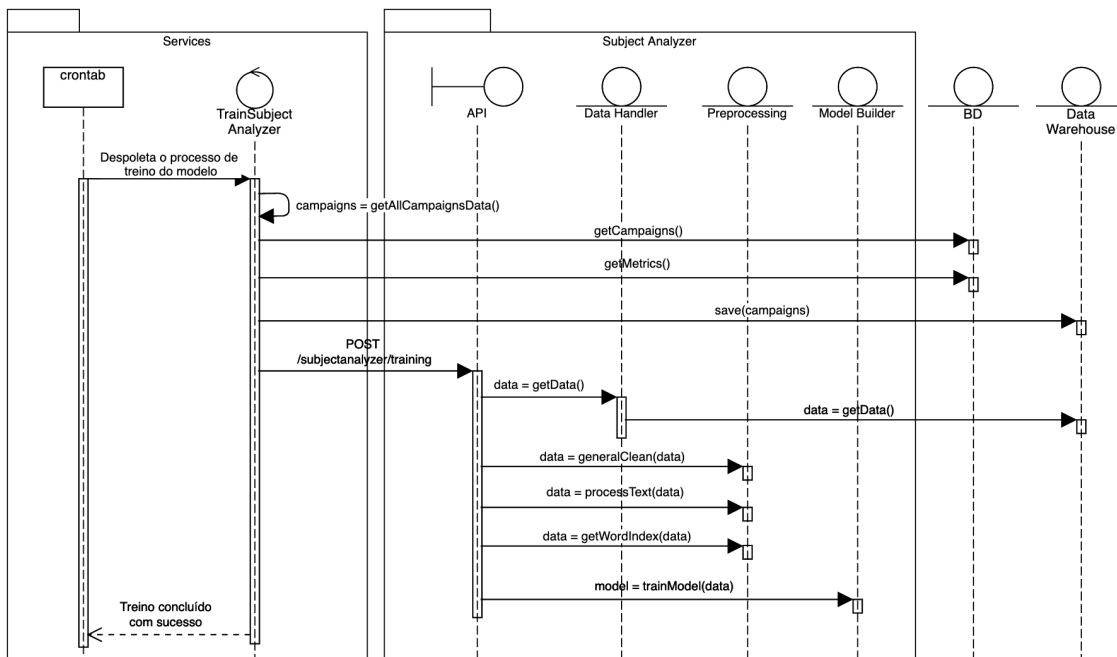


Figura 5.9: Vista de cenários do caso de uso Treinar Modelo de Classificação

Considerando que o *UC02: Analisar Assunto* é, também, um dos casos de uso de maior relevância a nível arquitetural, apresenta-se, na figura 5.10, a representação da vista de cenários desenvolvida para o caso de uso em questão. Este caso de uso inicia-se no ator Utilizador que decide a qualquer momento iniciar a análise do assunto. Toda a interação do utilizador é executada através da vista *Subject Analyzer View* que, por sua vez, utiliza o componente *Subject Analyzer Ctrl* que será o responsável por gerir a lógica da classificação de assunto e, por isso, comunica com o *Subject Analyzer Service* que efetuará o pedido ao serviço externo, alojado no *Services*. Uma vez efetuado o pedido, o componente *Campaign Analyzer Resource* trata de orquestrar o pedido para o serviço certo. Neste caso o pedido será mapeado para o serviço presente no componente *Campaign Analyzer Service* que, por sua vez, comunicará com a API desenvolvida presente no *Subject Analyzer* para que esta possa prever a classificação do assunto com base no modelo previamente criado e definido (c.f secção 6.3). Uma vez finalizada a classificação do assunto, grava-se a informação referente ao assunto, a par com a classificação que este obteve, utilizando o componente *Campaign Analyzer Repository*, que comunicará com a base de dados a fim de persistir os dados. Esta persistência torna-se necessária para dar resposta ao caso de uso relativo ao histórico de consultas, pelo que, gravando a informação relativa a todos os assuntos, assim como a classificação que estes obtiveram, torna-se possível, futuramente, consultar o histórico com base nestes dados e, ainda, selecionar um assunto a partir do histórico. Uma vez terminado o caso de uso, a classificação é apresentada ao utilizador.

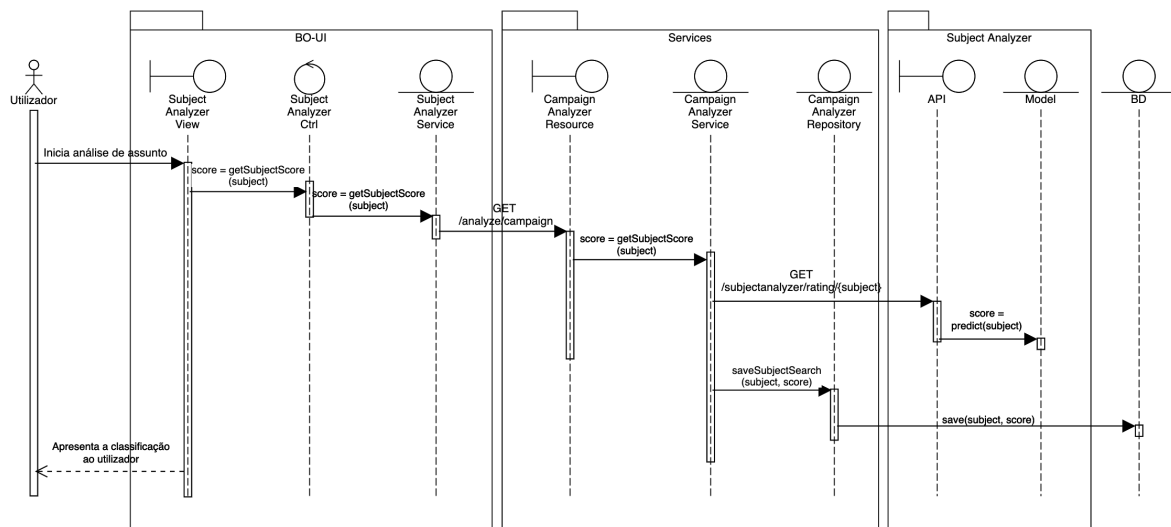


Figura 5.10: Vista de cenários do caso de uso Analisar Assunto

### 5.5.4 Mock-ups

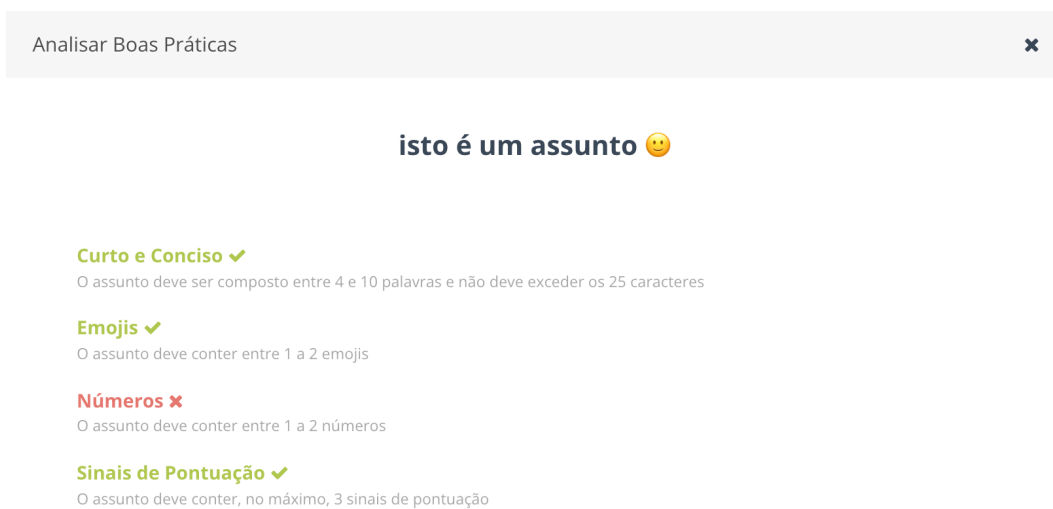
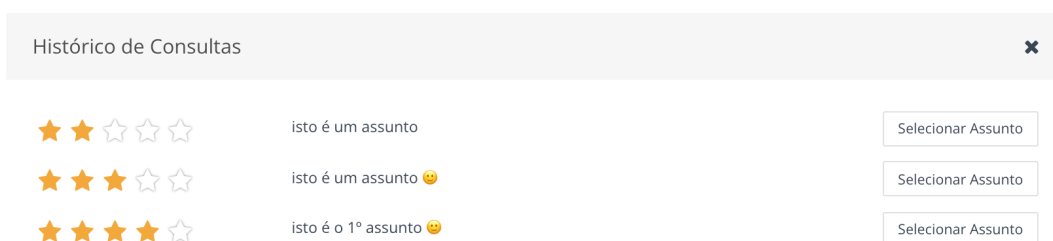
O *design* de *mock-ups* considera-se uma fase relevante para o desenvolvimento de novas funcionalidades, na medida em que permite a exploração de várias decisões de *design* numa etapa em que a mudança da interface ainda tem um custo reduzido. Deste modo, pretende-se evitar consequências mais custosas, como seriam caso as alterações surgissem aquando da implementação. Posto isto, apresenta-se, nas figuras 5.11, 5.12 e 5.13, uma representação em *mock-up* da interface gráfica definida para o sistema de classificação de assunto, da análise de boas práticas e do histórico de consultas em conjunto com a seleção de assunto, respetivamente.

The mock-up shows a form titled 'Opções' with the following fields and controls:

- Nome da Campanha:** A text input field containing 'Nome interno da campanha'.
- Assunto:** A text input field containing 'Isto é um assunto', followed by a star rating (3 stars selected), a search icon, and a refresh icon.
- Remetente:** A dropdown menu showing '1 -Emanuel(egomes+deploy@e-goi.com)' with a '+ Variação' button.
- Reply-to:** A dropdown menu showing '1 -Emanuel(egomes+deploy@e-goi.com)' with a '+ Variação' button.
- Avançadas:** A blue toggle switch.

At the bottom, there is a navigation bar with buttons: ANTERIOR, ESCOLHER, OPÇÕES (highlighted), EDITAR, PREPARAR, RESUMO, and AVANÇAR.

Figura 5.11: *Mock-up* do sistema de classificação de assunto

Figura 5.12: *Mock-up* da análise de restrições de boas práticasFigura 5.13: *Mock-up* do histórico de consultas e seleção de assunto

Através da análise das figuras 5.11, 5.12 e 5.13, denota-se que todas as decisões têm como contexto o produto E-goi no qual o *Subject Analyzer* se insere, pelo que decisões como o esquema de cores, iconografia e tipografia já seguem o padrão definido para o produto em questão. Quanto à disposição e representação gráfica, a figura 5.11 permite analisar que o *Subject Analyzer* segue um esquema de representação de avaliação baseado em estrelas, com uma escala de 0 a 5. De modo a aceder a casos de uso como *UC03: Analisar Restrições de Boas Práticas* e *UC04: Visualizar Histórico de Consultas* (c.f. secção 5.2), utiliza-se um esquema de ícones que segue uma disposição horizontal ao nível da caixa de texto de inserção de assunto, facilitando a leitura da interface, tornando-a mais intuitiva.

## Capítulo 6

# Implementação

Tendo em conta a enorme quantidade de dados relativos às campanhas enviadas pelos utilizadores E-goi e, considerando também, que estes mesmos dados se encontram divididos em diversas base de dados e tabelas, optou-se pela utilização do processo *Extract, transform, load* (ETL) a fim de centralizar os dados. Uma vez terminado o processo, torna-se necessário realizar um pré-processamento dos dados resultantes a fim de servir o modelo. Posto isto, passa-se à fase de criação do modelo, na qual se demonstra todas as decisões tomadas na criação do mesmo. Por último, apresenta-se a secção de criação da API, que tem como principal objetivo a exposição do *Subject Analyzer*, para que este seja integrado com a plataforma E-goi.

### 6.1 Extract, transform, load

ETL é um processo constituído por três fases que visa simplificar a análise de um conjunto de dados. A primeira fase deste processo é caracterizada pela extração de dados provenientes de várias fontes. Uma vez terminada a extração, passa-se à segunda fase, sendo esta responsável pela transformação dos dados previamente extraídos através de um conjunto de tarefas que objetivam corrigir dados errôneos e fornecer dados limpos e consistentes com o negócio ao utilizador final. Por último, a fase de carregamento permite concluir o processo, armazenando os dados numa estrutura denominada *Data Warehouse* [105][106][107][108].

No sistema E-goi, as base de dados dos clientes são distribuídas por vários servidores, sendo que, cada cliente possui uma base de dados dedicada que, por sua vez, é constituída pelas tabelas responsáveis pela persistência da informação necessária ao correto funcionamento da conta. Tendo isto em vista, existe uma tabela, denominada *maxmailing\_arquivo*, responsável por armazenar a informação relativa às campanhas, nomeadamente o assunto das mesmas (coluna *assunto*), conforme se pode verificar pela figura 6.1. No entanto, as métricas relativas ao envio da campanha não se encontram nesta tabela, havendo uma tabela específica para tal - *message\_email\_count* -, centralizando a informação necessária para a construção do relatório, tal como o número absoluto de aberturas totais (coluna *opens*) e aberturas únicas (coluna *opens\_unique*).

maxmailing_arquivo	
<b>id</b>	: int (11)
user_id	: int (11)
cliente_id	: int (11)
list	: int (11)
assunto	: text
corpo_texto	: text
corpo_html	: text
data_programada	: datetime
data_inicio	: datetime
data_fim	: datetime
mailing_hashcode	: varchar (255)
modelo	: text
canceled	: tinyint (1)
deleted	: tinyint (1)
re_auto	: tinyint (1)
remetente	: text
estado	: text
canal	: text
grupo	: int (11)
last_editor	: text
default_style	: text

Figura 6.1: Tabela *maxmailing\_arquivo*

ego_message_email_count	
<b>message_hash</b>	: varchar (255)
sent	: int (11)
clicks	: int (11)
clicks_unique	: int (11)
opens	: int (11)
opens_unique	: int (11)
delivered	: int (11)
bounces_hard	: int (11)
bounces_soft	: int (11)
complaints	: int (11)
social_actions	: int (11)
unsubscriptions	: int (11)
clicks_by_url	: mediumblob
social_actions_by_social_network	: mediumblob
by_location	: mediumblob
by_date	: mediumblob
by_domain	: mediumblob
by_reader	: mediumblob
unsubscriptions_by_method	: mediumblob
unsubscriptions_by_reason	: mediumblob

Figura 6.2: Tabela *ego\_message\_email\_count*

Desta forma, surgiu a necessidade de adotar o seguinte processo ETL:

1. Extração: Extrai da tabela *maxmailing\_arquivo* as colunas relativas ao identificador da conta do cliente e identificador do assunto da campanha de *email*. Extrai, ainda, da tabela *ego\_message\_email\_count* as colunas relativas ao número de envios, número de aberturas únicas e número de cliques únicos;
2. Transformação: Realiza os cálculos necessários para transformar os valores absolutos das métricas de envio da campanha de *email* em taxas, nomeadamente, taxa de entrega, taxa de aberturas únicas e taxa de cliques únicos;
3. Carregamento: Os dados são armazenados na tabela *subject\_data* criada com o único propósito de persistir e fornecer dados ao modelo de classificação de assunto. A estrutura da tabela criada encontra-se representada na figura 6.3.

subject_data	
<b>campaign_hash</b>	: varchar (255)
<b>client_id</b>	: int (11)
subject	: varchar(255)
delivery_rate	: double
unique_open_rate	: double
unique_click_rate	: double
created	: int (11)

Figura 6.3: Estrutura da tabela *subject\_data*

### 6.1.1 Abordagens Possíveis e Escolha

A necessidade de automatizar a execução do processo ETL colmatou na elaboração de várias abordagens. Deste modo, procede-se de seguida à enumeração e avaliação de todas as abordagens de execução do processo ETL existentes.

### Execução Sequencial

A abordagem sequencial, que se considera a abordagem menos complexa, consiste na execução do processo ETL de forma iterativa e sequencial para cada um dos servidores existentes. Como se pode verificar pela análise da figura 6.4, nesta abordagem não é possível replicar o processo pelos vários servidores em simultâneo nem é possível a execução do processo múltiplas vezes dentro de cada servidor, pelo que teria de ocorrer de forma singular e individual para cada um dos servidores.

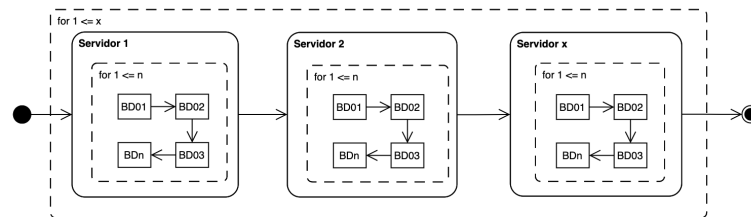


Figura 6.4: Execução do processo de forma sequencial

Assim, a performance do processo ETL seria afetada em função do número de servidores existentes. Ainda que a *performance* do processo seja afetada, nesta abordagem, o impacto causado ao servidor será reduzido, considerando que apenas um processo irá ser executado individualmente.

### Execução Paralela com Recurso a Processos

A abordagem de execução paralela com recurso a processos consiste na execução paralela do processo ETL em cada um dos servidores existentes. Deste modo, e conforme se pode verificar pela análise da figura 6.5, ainda que a execução seja paralela em cada um dos servidores, cada processo seria responsável por iterar sequencialmente, em cada servidor, todas as bases de dados e respetivas tabelas. Assim, a *performance* do processo ETL seria equivalente à *performance* do servidor mais lento, seja devido à sua capacidade de processamento como devido à quantidade de bases de dados que nele estão alojadas.

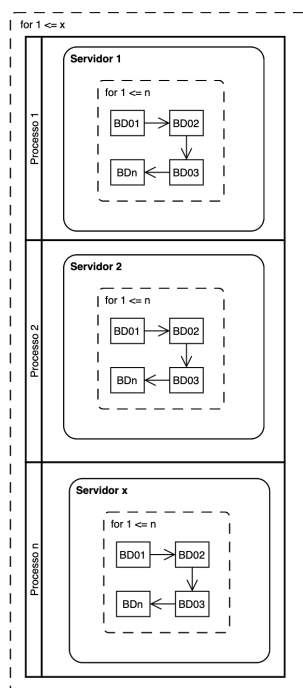


Figura 6.5: Execução do processo de forma paralela com recurso a processos

### Execução Paralela com Recurso a Processos e Threads

A abordagem de execução paralela com recurso a processos e *threads* considera-se a abordagem mais complexa das três enumeradas, uma vez que nesta, não só se pretende paralelizar a execução do processo ETL por todos os servidores, como também se pretende paralelizar a execução por todas as bases de dados existentes em cada servidor. Para que tal seja possível, conforme se pode analisar na figura 6.6, seria necessário que cada processo em cada servidor gerasse um conjunto de *threads*, cada uma responsável pela execução do processo ETL em cada uma das bases de dados.

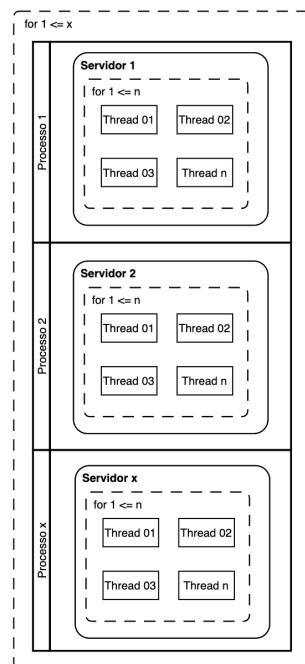


Figura 6.6: Execução do processo de forma paralela com recurso a processos e threads

### Comparação de alternativas

A escolha da abordagem final é de grande relevância na medida em que se pretende criar um equilíbrio entre a *performance* do processo ETL e a disponibilidade dos servidores do sistema E-goi de modo a não comprometer o *Service Level Agreement* (SLA) existente no E-goi. Assim, dada a possibilidade de se paralelizar a execução do processo ETL pelos servidores existentes no E-goi, e tendo em conta o número elevado de servidores atualmente existentes, considera-se de enorme relevância a escolha de uma abordagem que priorize a paralelização da execução do processo ETL por todos os servidores existentes. Deste modo, como na abordagem sequencial o impacto na *performance* do processo ETL é proporcional à expansão de servidores, exclui-se a abordagem sequencial. Relativamente à paralelização do processo por todas as bases de dados existentes em cada servidor, tal seria uma abordagem interessante considerando a *performance* do processo ETL, no entanto, o impacto que essa abordagem teria nos servidores poderia comprometer a sua disponibilidade. Considerando que a disponibilidade dos servidores se sobrepõe à obtenção da melhor *performance*, optou-se pela segunda alternativa - Execução Paralela com Recurso a Processos.

## 6.2 Pré-processamento dos dados

Uma vez terminado o processo ETL, que culminou no armazenamento dos dados numa *Data Warehouse* denominada *subject\_data*, passa-se à fase do pré-processamento dos dados na qual se efetuam as alterações necessárias ao conjunto de dados, de forma a que estes se tornem mais simples e adequados ao modelo de DL.

A fase de tratamento de dados deve ser iniciada por uma decisão relativa ao universo de assuntos que se pretendem tratar. Assim, a fim de se decidir se um certo assunto é mantido

ou descartado, procede-se a uma avaliação do mesmo com base num conjunto de expressões regulares pré-definidas enumeradas de seguida:

- `!extra_field_[0-9]*`: Expressão regular responsável por identificar o código de personalização (conceito mencionado na subsecção 4.1.2) relativo ao campo *extra field*. Este campo distingue-se dos restantes campos de personalização pelo facto de não só permitir a customização do valor consoante o contacto mas também a definição do contexto ao qual se destina o campo. Deste modo, por ser um código de personalização que permite alteração contextual, remover-se-ão todos os assuntos que dele fizerem uso;
- `(\{\{IF: |\{\{FEEDBLOCK\}| (\{ENDIF\}) | (\{ELSEIF\})`: Expressão regular responsável por identificar códigos *Really Simple Syndication* (RSS)<sup>1</sup> nas campanhas de *email*. Tal como o campo de personalização *extra field*, estes códigos não têm um valor nem um contexto comum ao longo do tempo, pelo que se removem os assuntos que contiverem estes códigos.

Uma vez descartados os assuntos que se consideram de menos interesse relativamente ao modelo que será treinado, passa-se à normalização e uniformização dos mesmos. No que diz respeito aos *emojis*, é feita uma normalização recorrendo à biblioteca *demoji* [109] que permite o reconhecimento de *emojis* num dado texto tendo como base o repositório de códigos *emojis* do *Unicode Consortium* [110]. Determinados os *emojis*, uniformizam-se os assuntos, substituindo-se o *emoji* respetivo pelo identificador global a utilizar para se identificar estes caracteres -  $\xi$ *emoji*.

De seguida, substituem-se os restantes códigos de personalização por identificadores mais globais de forma a manter a consistência entre os identificadores. Estes códigos de personalização encontram-se representados e descritos na tabela 6.1, assim como o identificador global pelo qual são substituídos.

Tabela 6.1: Códigos de personalização, descrição e identificador

Código de Personalização	Descrição	Identificador
!fname	Nome próprio do contacto	$\xi$ <i>fname</i>
!lname	Apelido do contacto	$\xi$ <i>lname</i>
!email	<i>Email</i> do contacto presente na lista de contactos à qual a campanha se destina	$\xi$ <i>email</i>
!dateadd	Data em que o contacto foi adicionado à lista à qual a campanha se destina	$\xi$ <i>dateadd</i>
!cellphone	Telemóvel do contacto	$\xi$ <i>cellphone</i>
!uid	Identificador do contacto	$\xi$ <i>uid</i>
!birth_date	Data de aniversário do contacto	$\xi$ <i>birth_date</i>
!lang	Idioma do contacto	$\xi$ <i>lang</i>

Ainda relacionado com a fase de pré-processamento, analisa-se cada palavra de forma individual para que se possa concluir se esta é realmente interessante para o treino do modelo. Deste modo, identificaram-se as moedas, as datas e as percentagens como sendo algo que pode ser tratado de uma forma mais genérica pelo que se substitui as moedas, as datas e percentagens por  $\xi$ *currency*,  $\xi$ *date* e  $\xi$ *percentage*, respetivamente.

<sup>1</sup>O sistema E-goï permite enviar campanhas tendo em conta o *feed* RSS de um *site* de notícias ou *blog*.

Deste modo, apresenta-se de seguida as expressões regulares utilizadas:

- `(?i)((eur+((o)s*)*( |\b))|reais( |\b))|[\$£€]|R\$"`: Expressão regular responsável por substituir todas as ocorrências de moeda, seja símbolo ou por extenso, pelo identificador `§currency`;
- `([0-9]{1,2}(\.|\|))[0-9]4/`: Expressão regular responsável por substituir todos os formatos de data existentes pelo identificador `§date`;
- `([0-9]*(\.|,|))?[0-9]+%`: Expressão regular responsável por substituir percentagens pelo identificador `§percentage`.

Finalizada esta fase do tratamento dos dados ao nível do domínio, aplicaram-se as técnicas *Stop Word Removal*, *Lowercasing* e *Tokenization*, mencionadas na secção 3.1.

Com o objetivo de identificar *stop words*, de forma a economizar tempo e espaço, utilizou-se a biblioteca *spaCy*[111]. Esta biblioteca contém uma lista padrão de trezentas e vinte e seis *stop words*, permitindo ainda a adição de novas palavras a esta lista. Para além disto, possibilita ainda a identificação de pontuação. No excerto de código 6.1 encontra-se representada a implementação da função responsável pela aplicação da técnica *Stop Word Removal*.

```

1 def process_text(data):
2     # Lowercasing
3     text = nlp(data.lower())
4
5     response = []
6     for token in text:
7         if token.text in nlp.Defaults.stop_words:
8             continue
9         if token.is_punct:
10            continue
11            result.append(token.lemma_)
12    return " ".join(response)

```

Excerto de Código 6.1: Implementação da função responsável pelas técnicas *Stop Word Removal*, *Lowercasing* e remoção da pontuação

Como se pode analisar pelo excerto de código, a função `process_text` para além de descartar *stop words* e pontuação identificadas com recurso à biblioteca *spaCy*, aplica também a técnica *Lowercasing* como primeiro passo, colocando todos os caracteres minúsculos.

Em suma, apresenta-se de seguida, na figura 6.7, uma representação gráfica de todo o processo, tomando como base um exemplo real.

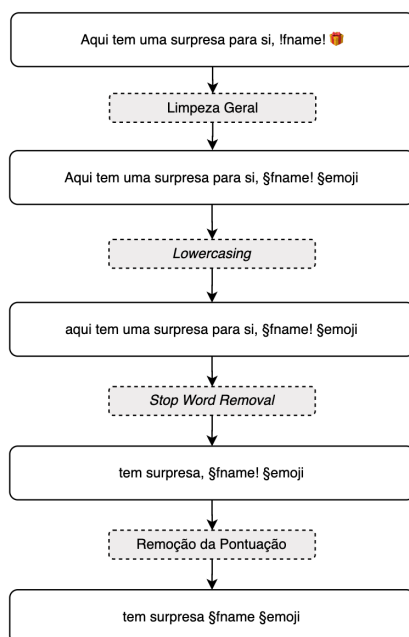


Figura 6.7: Representação gráfica do pré-processamento

Uma vez que os algoritmos de ML são fortemente baseados em procedimentos matemáticos e estatísticos, torna-se necessário transformar a série de palavras que compõem os assuntos em forma numérica. Para tal, numa primeira fase, é criado um índice onde cada palavra é mapeada com um número, através da API *Keras* de pré-processamento de texto. O código responsável por este processo encontra-se representado no excerto de código 6.2.

```

1 X = data['subject']
2 Y = to_categorical(data['unique_open_rate'])
3
4 tokenizer = tf.keras.preprocessing.text.Tokenizer(filters='\t\n\r',
5           lower=False, char_level=False)
6 tokenizer.fit_on_texts(X)
7 tokenizer.word_index

```

Excerto de Código 6.2: Código responsável pela criação do índice de palavras

A título exemplificativo, uma representação parcial do índice de palavras gerado encontra-se na tabela 6.2.

Tabela 6.2: Representação parcial do índice de palavras

Palavra	Índice
"semana"	81
"banco"	175
"§percentage"	14
"fim"	295
"encenação"	458252
"rastreo"	13652
"normalidade"	353635
"raspadinha"	533123
"startup"	733258

De seguida, torna-se necessário transformar cada assunto do conjunto de dados numa sequência de números com base no índice previamente gerado, com recurso, novamente, à API *Keras* de pré-processamento de texto. O código necessário à transformação encontra-se no excerto de código 6.3.

```
1 # Create array of word sequences  
2 X = tokenizer.texts_to_sequences(X)
```

Excerto de Código 6.3: Código para a geração de sequências numéricas para cada elemento do conjunto de dados

Na última etapa, os dados são divididos em dados de treino e dados de validação para que, durante o treino apenas sejam usados os dados de treino e, após a finalização do modelo, se utilizem os dados de validação para o testar. Deste modo, o conjunto de dados foi dividido em 70% e 30%, para treino e validação, respetivamente, conforme se pode verificar pela figura 6.8. Como podemos analisar pelo gráfico, o conjunto de dados é composto por aproximadamente um milhão e oitocentas mil entradas, sendo que, pouco mais de um milhão e duzentas fazem parte do conjunto de dados de treino e pouco mais de quinhentas mil entradas dizem respeito ao conjunto de dados de validação.

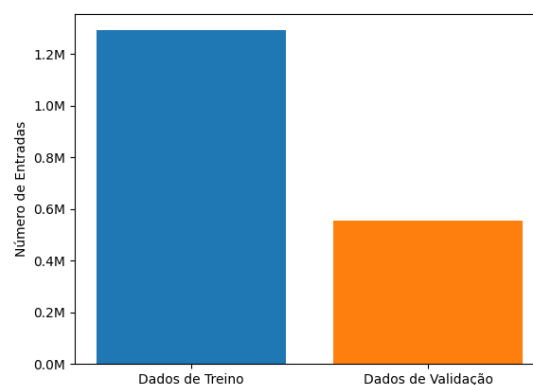


Figura 6.8: Divisão do conjunto de dados em dados de treino e dados de validação

Uma vez que se trata de um problema de classificação multiclasse, podendo um assunto obter uma classificação entre 0 e 5, encontra-se representada a distribuição dos dados, de uma forma homogênea, pelas classes tanto para os dados de treino (figura 6.9), como para os dados de validação (figura 6.10).

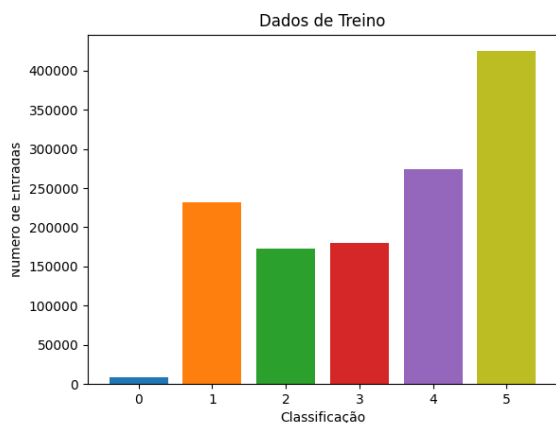


Figura 6.9: Distribuição dos dados

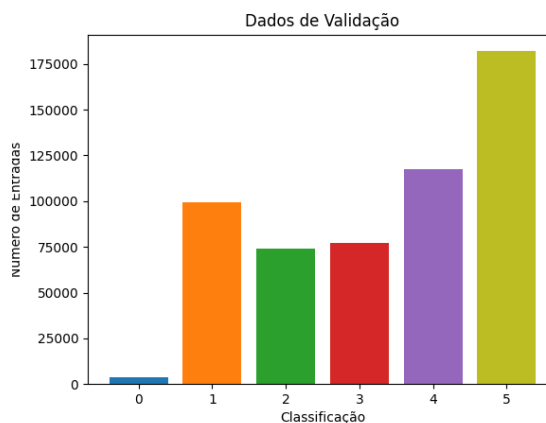


Figura 6.10: Loss score do modelo base

### 6.3 Construção do modelo

Os modelos de DL podem ser constituídos por diversas arquiteturas e hiperparâmetros, sendo, estes últimos, responsáveis pela configuração lógica do processo de aprendizagem. Assim, definem-se os seguintes hiperparâmetros:

- *Optimizer*: Adam com Nesterov momentum [112], uma vez que se verificou um desempenho superior do mesmo em tarefas de sequência com LSTM [113];
- *Loss Function*: Categorical cross-entropy loss, tendo em conta que é mais usada em problemas de classificação [114];
- *Batch size*: Batch size 64, dado que foi observado, na prática, que usar um batch size mais elevado resulta numa degradação significativa na qualidade dos modelos de DL [115].

Numa fase inicial, implementou-se um modelo base sequencial, com duas *dense layers* [70] e um nó de saída, conforme se pode verificar pela figura 6.11.

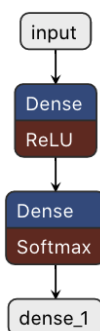


Figura 6.11: Arquitetura do modelo base

Uma vez definida a arquitetura, passou-se à fase do treino, representando, deste modo, a evolução do desempenho do modelo através das figuras 6.12 e 6.13, que retratam, graficamente, a *accuracy* e o *loss score* do modelo, respetivamente.

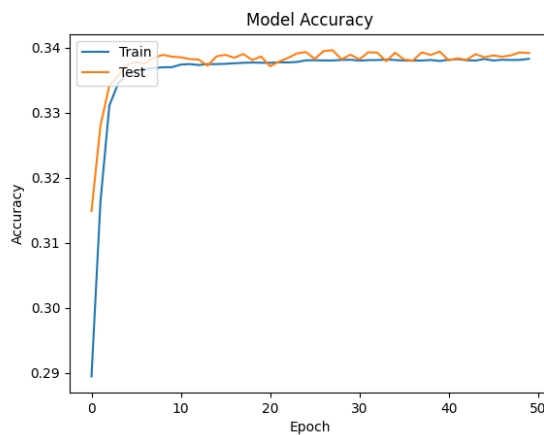


Figura 6.12: *Accuracy* do modelo base

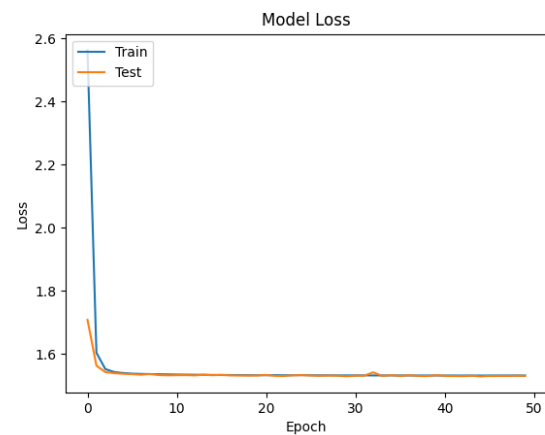


Figura 6.13: *Loss score* do modelo base

Uma das tarefas mais desafiantes na construção do modelo é a definição do número de épocas de treino, tendo em consideração que poucas épocas podem levar a um *underfitting*, ocorrência na qual o modelo não aprende tudo o que seria possível tendo em conta os dados de treino fornecidos. No entanto, definir muitas épocas pode levar a um *overfitting*, que se considera o acontecimento oposto ao descrito anteriormente.

Pela análise das figuras acima, figura 6.12 e figura 6.13, podemos constatar que o modelo não apresenta evoluções significativas passado algumas épocas de treino. Assim, de forma a evitar esforço computacional desnecessário, sem comprometer a precisão do modelo, foi implementada uma técnica de otimização, denominada *Early Stopping*. A implementação desta técnica encontra-se representada no excerto de código 6.4.

```

1 self.tensor_callbacks = [tf.keras.callbacks.EarlyStopping(
2     monitor='val_loss',
3     patience=3,
4     restore_best_weights=True)]

```

Excerto de Código 6.4: Implementação do *Early Stopping*

O sistema *Early Stopping* monitoriza o *loss score* e tem um valor de 3. Deste modo, permitimos que o treino continue durante três épocas adicionais após o ponto em que o *loss score* de validação começa a atenuar, fazendo com que o modelo grave o estado em que obteve a melhor precisão de validação. As vantagens deste método são demonstradas nas figuras 6.14 e 6.15.

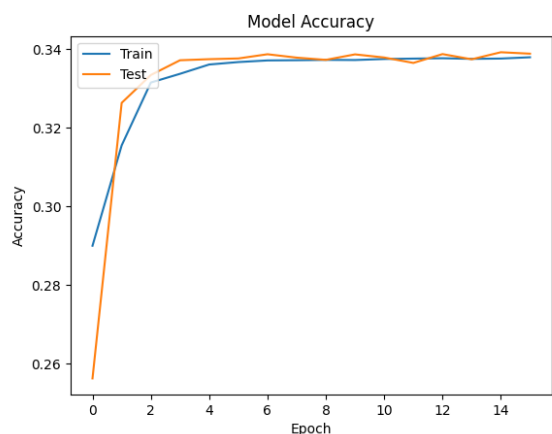


Figura 6.14: *Accuracy* do modelo base com *Early Stopping*

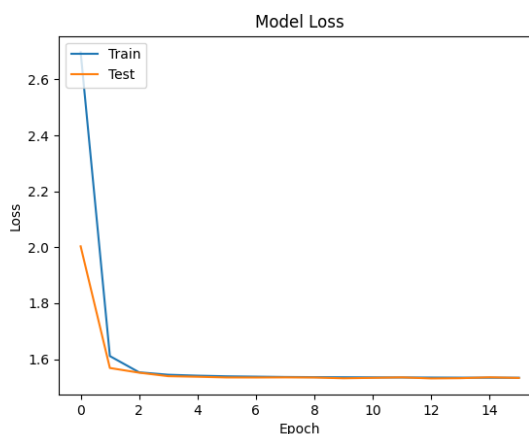


Figura 6.15: *Loss score* do modelo base com *Early Stopping*

Uma vez que se trata de uma tarefa de NLP, em que o contexto de cada palavra é importante, aplicou-se uma *embedding layer* [70] ao modelo, responsável por transformar índices em vetores. Neste caso, cada palavra é representada através de um número do índice de palavras previamente gerado, que se encontra na secção 6.2. Deste modo, a *Keras embedding layer* é inicializada com pesos aleatórios e com os seguintes parâmetros definidos:

- Dimensão da entrada: Tamanho do índice de palavras gerado;
- Dimensão da saída: Tamanho de espaço vetorial no qual as palavras se encontram. O valor definido foi 64;
- Comprimento da entrada: Comprimento da sequência de entrada. Neste caso, o valor é 25, uma vez que define o comprimento máximo permitido num assunto.

A representação arquitetural do modelo base com *embedding layer* encontra-se presente na figura 6.16. Pode-se verificar pela análise do grafo que existe uma camada adicional - *flatten layer* [70]. Uma vez que a saída de uma *embedding layer* é caracterizada por uma matriz bidimensional e a *dense layer* recebe, como entrada, um vetor unidimensional, tornou-se necessário adicionar esta camada com o objetivo de realizar a transformação necessária.

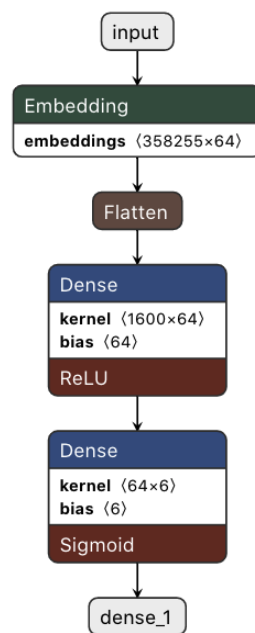


Figura 6.16: Arquitetura do modelo base com *embedding layer*

De seguida, apresentam-se nas figuras 6.17 e 6.18, a *accuracy* e *loss score* do modelo conseqüente. Pode-se facilmente verificar que a adição desta camada permitiu um crescimento na *accuracy*, passando de, aproximadamente, 0.33 para 0.42, nos dados de treino. No entanto, o mesmo não se verificou para os dados de validação, que divergem dos dados de treino.

Não obstante, é também possível observar que o modelo atinge o seu melhor estado sete épocas mais cedo, demorando assim metade do tempo do que o modelo anterior.

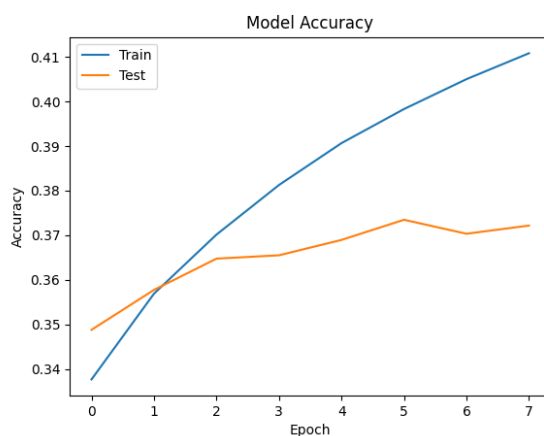


Figura 6.17: *Accuracy* do modelo base com *embedding layer*

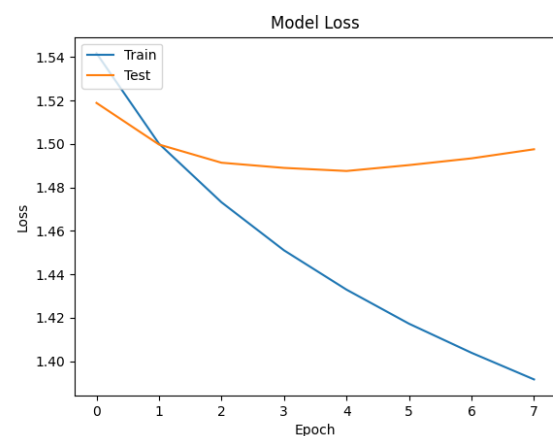


Figura 6.18: *Loss score* do modelo base com *embedding layer*

Tal como foi concluído na subsecção 3.2.5, a LSTM é uma das arquiteturas mais aplicadas em tarefas de NLP. Deste modo, foi testada a arquitetura LSTM, bem como a sua variante *Bidirectional LSTM* (BiLSTM). A principal diferença reside no facto de, ao usar BiLSTM,

o modelo tem a capacidade de memorizar informações tanto das camadas seguintes como das anteriores, ao contrário da variante LSTM que apenas possui a capacidade de preservar as informações do passado. A par da arquitetura foi também adicionada uma *pooling layer - GlobalMaxPooling1D* [70] - responsável por reduzir a dimensão da entrada através do valor máximo da mesma, fazendo assim com que deixe de ser necessário o uso da *flatten layer*. Não obstante, foi também adicionada uma *unit forget bias* à porta de esquecimento da rede LSTM, com o valor 1, conforme recomendado no artigo [116] e demonstrado como tendo um grande impacto [117]. Assim, encontram-se nas figuras 6.19 e 6.20 a *accuracy* e *loss score* relativamente à variante LSTM e, nas figuras 6.21 e 6.22 no que diz respeito à segunda variante, BiLSTM, respetivamente.

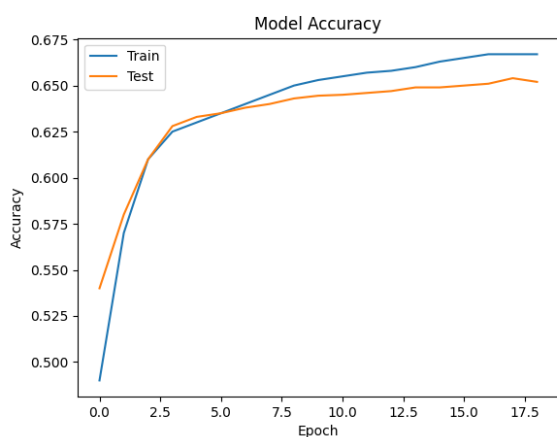


Figura 6.19: *Accuracy* do modelo com a arquitetura LSTM

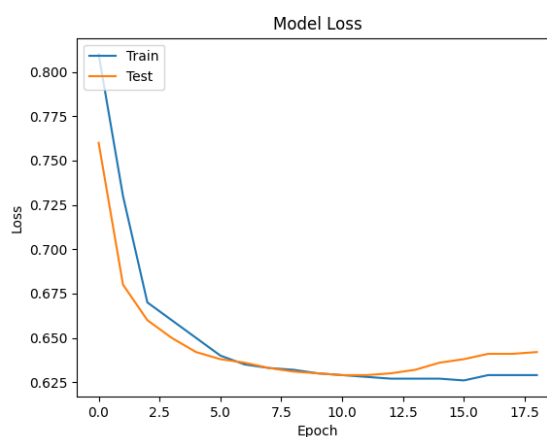


Figura 6.20: *Loss score* do modelo com a arquitetura LSTM

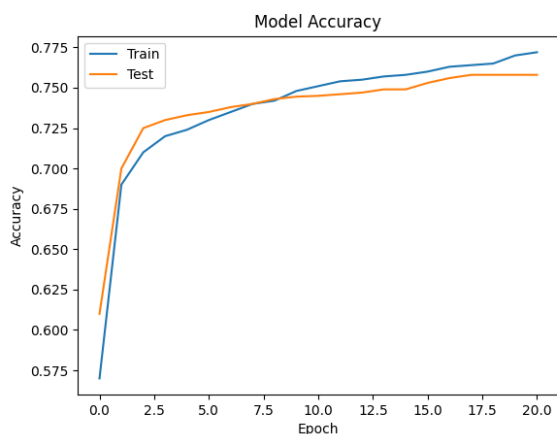


Figura 6.21: *Accuracy* do modelo com a arquitetura BiLSTM

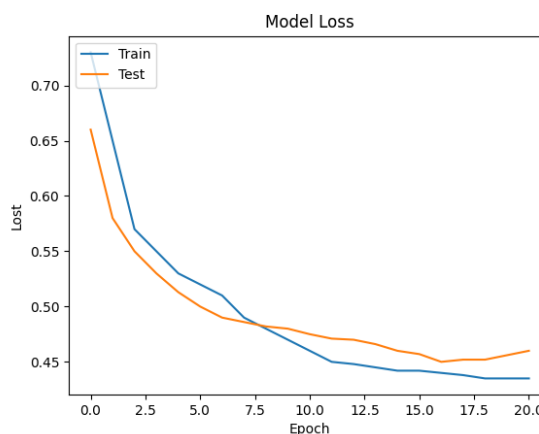


Figura 6.22: *Loss score* do modelo com a arquitetura BiLSTM

Através da análise dos gráficos podemos concluir que a variante BiLSTM apresenta melhores resultados, tanto ao nível da *accuracy* como do *loss score*. No que diz respeito à *accuracy*,

a aplicação da variante BiLSTM representa um aumento de 11%, tendo assim, aproximadamente o valor de 0.76. Quando ao *loss score*, a variante BiLSTM também se mostra mais adequada uma vez que o *loss score* é mais baixo comparativamente à arquitetura LSTM, minimizando assim a *loss function* do modelo.

A fim de complementar a análise, apresenta-se na tabela 6.3, os resultados do desempenho do modelo com as duas variantes. Atendendo a que se trata de uma classificação multiclasse, podendo assim um assunto obter uma classificação entre 0 e 5, apresenta-se tanto a métrica *precision* como a *recall* para cada uma das classes, sendo o *F1-Score* definido como a média ponderada de cada uma das seis classes. Assim, para além da *accuracy* apresentar um aumento quando aplicada a arquitetura BiLSTM, podemos constatar que também a métrica *F1-Score* apresenta melhores resultados. Em suma, a arquitetura adotada diz respeito à arquitetura que faz uso da variante BiLSTM, descartando deste modo a arquitetura LSTM unidirecional.

Tabela 6.3: Resultados do desempenho do modelo com a variante LSTM e BiLSTM

	Modelo LSTM	Modelo BiLSTM
<i>Accuracy</i>	65%	76%
<i>F1-Score</i>	63%	74%
<i>Precision class 0</i>	54%	70%
<i>Precision class 1</i>	58%	71%
<i>Precision class 2</i>	62%	73%
<i>Precision class 3</i>	67%	73%
<i>Precision class 4</i>	66%	75%
<i>Precision class 5</i>	64%	76%
<i>Recall class 0</i>	49%	74%
<i>Recall class 1</i>	62%	73%
<i>Recall class 2</i>	66%	73%
<i>Recall class 3</i>	54%	68%
<i>Recall class 4</i>	59%	74%
<i>Recall class 5</i>	66%	78%

Após a definição da melhor rede a adotar, vê-se a necessidade de adicionar uma *regularization layer*, uma vez que a arquitetura atinge o estado de *overfitting* relativamente cedo. Sabendo que existem diversas técnicas de regularização, optou-se pela aplicação de uma *dropout layer* [70]. Esta técnica fundamenta-se na eliminação aleatória de neurónios e respetivas conexões durante o processo de treino, de forma a evitar o *overfitting*.

Assim, apresenta-se, na figura 6.23, a arquitetura final do modelo resultante deste processo de construção que teve como finalidade a implementação do modelo que melhor se adapta ao contexto do problema de classificação em questão, que culminou numa *accuracy* de 76% e *F1-Score* de 74%.

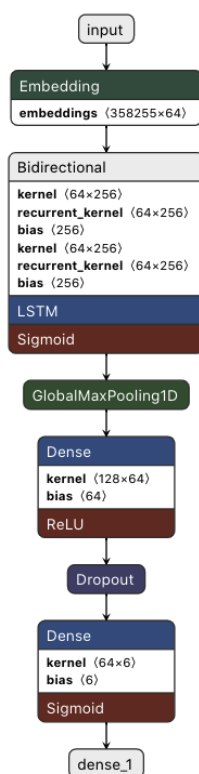


Figura 6.23: Arquitetura do modelo com BiLSTM

Uma vez definida a arquitetura do modelo de classificação e, de forma a comprovar a validade de aplicação dos hiperparâmetros definidos no início da secção, foi aplicado um processo de otimização, denominado *tuning*, que visa o ajuste dos hiperparâmetros a fim de obter os valores ideais para o modelo em questão. Deste modo, o processo de *tuning* foi implementado por intermédio do módulo *Grid Search* [118] que efetua uma pesquisa sobre todas as combinações de parâmetros possíveis e avalia o desempenho através da técnica *cross-validation*. De seguida, apresenta-se na tabela 6.4, o resultado do ajuste dos hiperparâmetros de acordo com os intervalos de valores definidos tendo em conta a *framework Keras* [70].

Tabela 6.4: Processo de *tuning* para os hiperparâmetros *optimizer*, *loss function* e *batch size*

Hiperparâmetro	Intervalo de Valores	Resultado
<i>Optimizer</i>	<code>['RMSprop', 'adam', 'nadam']</code>	Best accuracy: 0.766534 using {'optimizer': 'nadam', 'loss': 'categorical_crossentropy', 'batch_size': 128}
<i>Loss Function</i>	<code>['categorical_crossentropy', 'sparse_categorical_crossentropy', 'kullback_leibler_divergence']</code>	
<i>Batch Size</i>	<code>[32, 64, 128]</code>	

A análise da tabela 6.4 permite concluir que, para os hiperparâmetros *optimizer* e *loss function*, os valores aplicados inicialmente são os ideais. No entanto, o mesmo não se verifica para o hiperparâmetro *batch size*, onde se utilizou um *batch size* de 64 e se concluiu, através do processo de *tuning*, que o ideal seria de 128. Uma vez que se verifica que a *accuracy*, métrica utilizada para a avaliação do desempenho do modelo proveniente do

processo de *tuning*, é ligeiramente superior - aproximadamente 77% -, tornou-se relevante treinar e persistir o modelo com base nesta alteração. Assim, apresentam-se, na tabela 6.5, as métricas de desempenho relativas ao modelo final, com as alterações provenientes do processo de *tuning* implementado, que resultou numa *accuracy* de 77% e *F1-Score* de 75%.

Tabela 6.5: Resultados do desempenho do modelo de classificação

<i>Accuracy</i>	77%
<i>F1-Score</i>	75%
<i>Precision class 0</i>	72%
<i>Precision class 1</i>	73%
<i>Precision class 2</i>	72%
<i>Precision class 3</i>	73%
<i>Precision class 4</i>	75%
<i>Precision class 5</i>	77%
<i>Recall class 0</i>	74%
<i>Recall class 1</i>	74%
<i>Recall class 2</i>	72%
<i>Recall class 3</i>	73%
<i>Recall class 4</i>	74%
<i>Recall class 5</i>	78%

Assim, dado o término da definição da arquitetura e dos hiperparâmetros ideais do modelo, e de forma a mantê-lo atualizado e a par das tendências, foi implementada uma *cron-job*, conforme definido na subsecção 5.3.1, que, automaticamente, e com uma recorrência mensal, atualiza a *Data Warehouse* com as campanhas que foram recentemente enviadas e treina o modelo com recurso ao *endpoint* criado na API (c.f secção 6.4). O código responsável por executar a *cron-job* mensalmente encontra-se no excerto de código 6.5.

```
1 0 0 1 * * root APPLICATION_ENV=production /usr/bin/php /opt/services.e-
  goi.com/public/index.php trainSubjectAnalyzer
```

Excerto de Código 6.5: Código responsável por correr a *cron-job* de treino do modelo mensalmente, de forma automática

## 6.4 Application Programming Interface

De forma a tornar o *Subject Analyzer* acessível e também atender aos requisitos não funcionais definidos nas subsecções 5.3.1 e 5.3.6, foi implementada uma REST API com recurso à biblioteca *Falcon* [119], conhecida pela sua fácil adoção, livre de dependências.

No que diz respeito à documentação da API, a mesma foi gerada com recurso ao *Swagger* [120], interface usada para documentar REST API através de *JavaScript Object Notation* (JSON). Deste modo, a documentação gerada pode ser visualizada na figura 6.24.

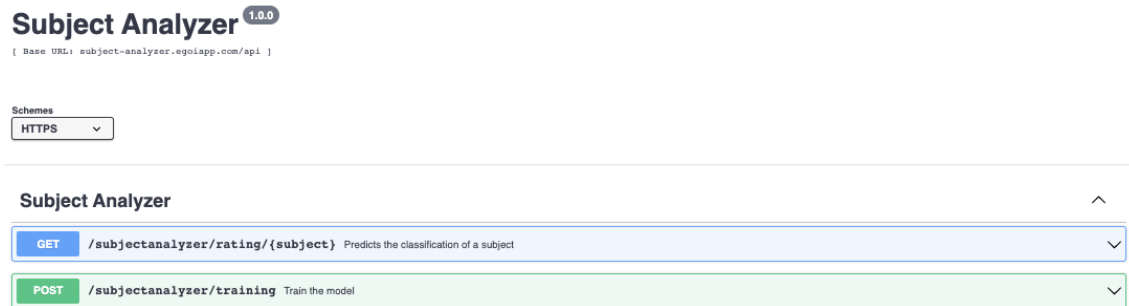


Figura 6.24: Documentação da API do *Subject Analyzer*

Como se pode verificar pela documentação, foram implementados dois *endpoints*, nomeadamente:

- GET `/subjectanalyzer/rating/{subject}`: Pedido responsável por devolver a classificação prevista, dado um assunto;
- POST `/subjectanalyzer/training`: Pedido responsável por iniciar o processo de treino do modelo.

Dada a frequência da sua utilização e considerando o propósito base do projeto em questão, considera-se que o pedido arquiteturalmente mais relevante é o pedido responsável por devolver a classificação de um assunto, pelo que, no excerto de código 6.6, se encontra representado, a título exemplificativo, o código desse pedido.

```
1 import falcon
2 import numpy as np
3 from tensorflow.compat.v1.keras.preprocessing import sequence
4
5 class RatingResource:
6
7     def on_get(self, req, resp):
8
9         # Retrieve parameters from url
10        subject = req.params['subject']
11
12
13        # Check if all required parameters were found
14        if not (subject):
15            raise falcon.HTTPBadRequest('Missing request data.', '
16            Missing \'subject\' on URL.')
17
18        # Convert subject to tokens
19        seq = self._tokenizer.texts_to_sequences([subject])
20
21        # Normalize token list length
22        X = sequence.pad_sequences(seq, maxlen=25)
23
24        # Predict subject score
25        prediction = self._model.predict(X)
26
27        # Get subject rating (from 0-5, None if not enough data)
28        rating = np.argmax(prediction[0])
29
30
31        resp.media = {
32            'rating': int(rating)
33        }
34        resp.status_code = falcon.HTTP_200
```

Excerto de Código 6.6: Função implementada para o pedido GET da classificação de assunto

A documentação detalhada dos pedidos da API encontra-se no Anexo A.

## 6.5 Integração com a plataforma E-goi

Considerando a arquitetura atualmente existente no sistema E-goi e considerando também a abordagem escolhida, apresentada na subsecção 5.4.3, pode-se verificar que a API do *Subject Analyzer* (API já abordada na secção 6.4) é acedida por um componente já existente, denominado *Services*, responsável por agregar todos os serviços da plataforma, internos ou externos, sendo considerado o ponto único de comunicação com a interface da plataforma E-goi (*BO-UI*).

Assim sendo, a função responsável por fazer a ligação entre o *Subject Analyzer* e o *Services* encontra-se representada no excerto de código 6.7.

```

1 public getSubjectScore($data)
2 {
3     if (empty($data['listId'] || empty($data['messageHash']))) {
4         return false;
5     }
6
7     // Checks if campaign data matches an entity
8     $entity = $this->getServiceLocator()->get('FactoryBroadcastEmail')->
        findOneBy(['hash' => $data['messageHash'], 'listId' => $data['listId'],
9         'channel' => 'email']);
10
11     if (!empty($entity)) {
12         $client = new Client();
13         $uri = $this->getServiceLocator()->get('config')['subjectAnalyzer']['
14             urlApp'];
15         $subject = json_decode($entity->getSubject());
16
17         if (json_last_error() !== JSON_ERROR_NONE) {
18             $subject = html_entity_decode($entity->getSubject());
19         }
20
21         $client->setUri($uri);
22         $client->setMethod('GET');
23         $client->setParameterGet(['subject' => $subject]);
24
25         try {
26             $data['score'] = $client->send()->getBody();
27         } catch (Exception $e) {
28             return false;
29         }
30     }
31     return $data;
32 }

```

Excerto de Código 6.7: Função responsável por fazer a ligação entre o *Subject Analyzer* e o *Services*, via API

Nesta fase, o *BO-UI* torna-se responsável por realizar pedidos ao *Services* de modo a obter a previsão de classificação do assunto para a campanha. De seguida, apresenta-se, na figura 6.25, uma imagem representativa da interface gráfica *BO-UI* onde se encontra representada a funcionalidade de análise de assunto de uma campanha.

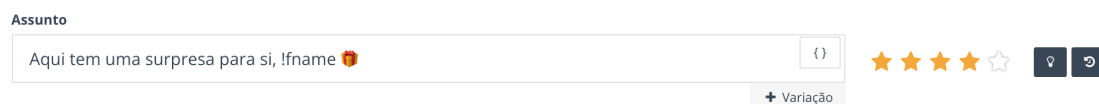


Figura 6.25: Interface gráfica que representa a funcionalidade de análise de assunto de uma campanha

A justificação da interface gráfica encontra-se na subsecção 5.5.4.

## Capítulo 7

# Experimentação e Avaliação

Após o desenvolvimento do sistema, torna-se necessário proceder à fase de avaliação de forma a que seja possível perceber se a solução atende às necessidades dos utilizadores, se o produto tem qualidade e se os problemas identificados inicialmente e que representam a causa do seu desenvolvimento foram mitigados. Deste modo, este capítulo tem como objetivo avaliar a solução desenvolvida tanto a nível técnico como a nível de usabilidade e satisfação do utilizador.

### 7.1 Indicadores de Avaliação

De forma a auxiliar a avaliação do sistema, definiram-se os seguintes indicadores:

- Cumprimento dos requisitos funcionais e não funcionais;
- Qualidade do sistema;
- Usabilidade do sistema;
- Satisfação do utilizador;

O primeiro indicador identificado permite verificar se os requisitos definidos, funcionais e não funcionais, foram alcançados. No que diz respeito à qualidade do sistema, esta tenciona garantir que o produto final atende às necessidades dos utilizadores [121]. Por outro lado, a usabilidade do sistema, segundo a norma ISO 9241-11 [122], permite aferir se um produto pode ser usado para atingir metas específicas como a eficácia, a eficiência e a satisfação, num contexto de utilização específico. Deste modo, surge o último indicador que diz respeito à satisfação do utilizador aquando da utilização do sistema e dos resultados obtidos através do mesmo.

### 7.2 Definição de Hipóteses

Uma vez determinados os indicadores de avaliação e com o propósito de os avaliar, definem-se as seguintes hipóteses:

- Cumprimento, na totalidade, dos requisitos funcionais e não funcionais;
- Nível de qualidade da solução a 100%;
- Grau de usabilidade da solução igual ou superior a 80%;
- Grau de satisfação dos utilizadores igual ou superior a 80%.

A primeira hipótese tenciona garantir que os requisitos foram cumpridos na sua totalidade. Para a segunda hipótese, pretende-se atingir o nível máximo de qualidade, sendo que esta depende diretamente da taxa de conclusão dos requisitos. No que diz respeito à terceira e quarta hipóteses definidas, estas indicam que a usabilidade e satisfação, respetivamente, devem atingir uma taxa igual ou superior a 80%. Este valor advém da adoção do sistema *Customer Satisfaction Score* (CSAT) que se define como um indicador chave da satisfação dos utilizadores, considerando que a satisfação dos mesmos é alcançada quando se atinge um grau igual ou superior a 80% [123].

### 7.3 Metodologia de Avaliação

Com o intuito de avaliar as hipóteses anteriormente mencionadas utilizam-se metodologias como os testes de *software*, o modelo *Quantitative Evaluation Framework* (QEF), o inquérito de satisfação, o inquérito de usabilidade e os testes de hipóteses.

Assim, apresenta-se na tabela 7.1 a associação entre os indicadores definidos e os métodos de avaliação.

Tabela 7.1: Associação entre indicador e método de avaliação

Indicador	Método de Avaliação
Cumprimento dos Requisitos	Testes de <i>Software</i>
Qualidade do Sistema	QEF
Usabilidade	Inquérito de Usabilidade e Testes de Hipóteses
Satisfação	Inquérito de Satisfação e Testes de Hipóteses

#### 7.3.1 Testes de Software

A fim de validar o correto funcionamento do sistema e o alcance dos requisitos propostos [124], aplicam-se os testes de *software*, capazes de avaliar e identificar falhas funcionais bem como falhas na especificação dos requisitos implementados. Para tal, recorreu-se ao modelo *V-Model* [125], representado na figura 7.1, caracterizado pelas relações estabelecidas entre cada fase de desenvolvimento e a fase de testes correspondente.

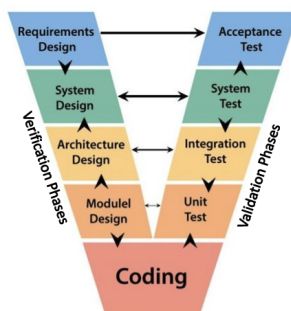


Figura 7.1: Arquitetura do modelo *V-Model* (adaptado de [125])

O modelo *V-Model* é constituído por duas fases principais: fase de verificação e fase de validação. A fase de verificação envolve todas as fases de desenvolvimento do projeto enquanto a fase de validação é composta por todas as fases de testes. No entanto, tendo em

conta o propósito da aplicação deste modelo, apenas se abordarão as categorias constituintes da fase de validação, nomeadamente, testes unitários, testes de integração, testes de sistema e testes de aceitação.

### 7.3.2 Quantitative Evaluation Framework

O modelo QEF, proposto por Paula Escudeiro e José Bidarra em 2006, representa um método de avaliação que visa melhorar a qualidade de uma solução de *software*, permitindo detetar e corrigir, em qualquer fase de desenvolvimento, eventuais falhas que possam ocorrer [121].

O QEF baseia-se num espaço de qualidade tridimensional (figura 7.2), em que cada dimensão ortogonal é composta por um conjunto de fatores que, por sua vez, são compostos por um conjunto de requisitos. Deste modo, um requisito é definido tendo em conta um grau de relevância específico, de acordo com uma escala compreendida entre os valores 2, 4, 6, 8 e 10, sendo que 10 é considerado o valor mais elevado. O requisito pode ser avaliado através dos valores 0, 25, 50, 75 e 100 que correspondem a uma descrição específica baseada em indicadores de cumprimento. Por sua vez, cada fator contém um peso relativo de importância, calculado pela soma dos graus de relevância dos requisitos que o representam, e um peso relativo de cumprimento, calculado a partir divisão da soma do cumprimento dos requisitos pelo valor total de cumprimento de requisitos desse fator.

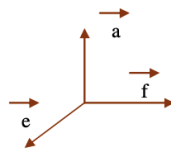


Figura 7.2: Espaço de qualidade tridimensional [121]

A fim de calcular a qualidade global do sistema, torna-se relevante entender dois conceitos distintos:

- Sistema ideal: Solução com qualidade máxima;
- Sistema real: Solução atual.

Sendo que, quanto menor for a distância euclidiana entre o sistema ideal e o sistema real, maior é a qualidade real da solução, esta é calculada a partir da fórmula [121]:

$$Q = 1 - \frac{D}{\sqrt{n}}, Q \in [0, 1] \quad (7.1)$$

Sendo que D, que representa a distância euclidiana, é traduzido pela fórmula [121]:

$$D = \sqrt{\sum_j \left(1 - \frac{Dim_j}{100}\right)^2} \quad (7.2)$$

### 7.3.3 Inquérito de Usabilidade

Segundo Jakob Nielsen, a usabilidade é um requisito que tem como alvo de estudo qualquer objeto, físico ou virtual, que permita a interação humana a fim de produzir algum resultado [126]. Aplicando-o ao presente projeto, a usabilidade representa um requisito não funcional do sistema (c.f secção 5.3), que se relaciona diretamente com a facilidade e adequação da interação do utilizador com o sistema.

De modo a avaliar a usabilidade do sistema, definiu-se, numa primeira fase, um guião composto por um conjunto de ações sequenciais que deve ser seguido pelo inquirido. Posteriormente, efetua-se um questionário formado por perguntas concretas e de resposta fechada. No que diz respeito às respostas, foi aplicada uma escala baseada no modelo *Likert Scale*, sendo esta constituída por cinco opções:

- Discordo totalmente;
- Discordo parcialmente;
- Indiferente;
- Concordo parcialmente;
- Concordo totalmente.

Relativamente às perguntas, estas são baseadas no inquérito *System Usability Scale* (SUS), criado por John Brooke em 1986 [127], que visam a medição da usabilidade de um sistema. Não obstante, de forma a complementar o inquérito, adaptaram-se ainda questões provenientes das dez heurísticas de Nielsen, criadas por Jakob Nielsen em 1994 [128], que têm como objetivo aumentar a usabilidade de um sistema.

### 7.3.4 Inquérito de Satisfação

Tendo em consideração que a satisfação do utilizador perante o serviço fornecido está diretamente relacionada com a utilização a longo prazo da plataforma, decidiu-se realizar uma análise através de um questionário que visa averiguar se o utilizador se encontra satisfeito com o sistema.

À semelhança do inquérito de usabilidade presente na subsecção 7.3.3, o questionário de satisfação segue uma estrutura composta por questões concretas, devido à ambiguidade que caracteriza esta medida, e à escala de *Likert Scale*.

### 7.3.5 Testes de Hipóteses

Os testes de hipóteses representam um método estatístico que, com base na informação advinda de uma amostra, auxilia na tomada de decisões, permitindo interpretar se uma premissa é válida ou não [129][130]. Consequente, um teste de hipóteses é composto por duas hipóteses [130]:

- $H_0$ : Hipótese nula que se pressupõe rejeitar;
- $H_1$ : Hipótese alternativa que se pretende verificar.

A fim de se determinar se a hipótese nula deve ser rejeitada recorre-se ao *p-value*, uma vez que considera o valor mínimo de significância que leva à rejeição da hipótese nula. Sendo  $\alpha$  o nível de significância do teste, rejeita-se  $H_0$  se [130]:

$$p\text{-value} \leq \alpha \quad (7.3)$$

Para os testes a realizar, assume-se o nível de significância como sendo de 0.05, valor este correspondente a um grau de confiança de 95%.

Os testes podem ser categorizados como testes paramétricos e testes não paramétricos. No que diz respeito aos testes paramétricos, estes seguem uma distribuição normal de amostras com dimensão igual ou inferior a trinta [130]. Por outro lado, os testes não paramétricos não necessitam de requisitos tão fortes, como a normalidade, sendo também indicados para amostras mais pequenas. No entanto, a principal desvantagem destes perante os testes paramétricos refere-se ao menor poder estatístico, sendo que não são tão capazes de levar a uma rejeição de  $H_0$  [130].

## 7.4 Avaliação dos Resultados

Após o desenvolvimento e integração do sistema, procede-se à avaliação dos resultados obtidos advindos de cada método de avaliação, tendo como base o indicador de avaliação e a hipótese definida.

### 7.4.1 Testes de Software

O primeiro método de avaliação analisado diz respeito ao indicador referente ao cumprimento dos requisitos da solução. Para tal, foram analisadas quatro categorias de testes de *software*, segundo o modelo *V-model*, enunciados na subsecção 7.3.1, a par com exemplos de testes desenvolvidos.

#### Testes Unitários

Os testes unitários são os primeiros testes a desenvolver, tal como representado no modelo *V-Model*, tendo como principal finalidade a validação de pequenas unidades de código caso estas se considerem testáveis, individuais, capazes de aceitar um *input* e devolver um *output* [131].

Ao longo do desenvolvimento dos testes unitários, utilizou-se o padrão de desenvolvimento de testes *Arrange-Act-Assert* (AAA) [132]. Este padrão consiste na divisão do teste em três blocos, nomeadamente:

- *Arrange*: Secção inicial composta pelas configurações necessárias;
- *Act*: Secção composta pelo comportamento alvo de teste;
- *Assert*: Secção onde se verifica o comportamento esperado.

Para o desenvolvimento dos testes foi utilizada a ferramenta de testes do *Tensorflow - tf.test* [133]. A título exemplificativo, encontra-se no excerto de código 7.1, o teste unitário implementado para a criação do índice de palavras, onde se denota a aplicação do padrão AAA.

```

1 def test_word_index(self):
2     # ARRANGE
3     subject = ['teste assunto', 'teste assunto 2']
4     tokenizer = tf.keras.preprocessing.text.Tokenizer(filters='\t\n\r',
5               lower=False, char_level=False)
6     tokenizer.fit_on_texts(subject)
7
8     # ACT
9     act = tokenizer.word_index['teste']
10
11    # ASSERT
12    assert isinstance(act, int)

```

Excerto de Código 7.1: Teste unitário implementado para a criação do índice de palavras

A par do teste unitário mencionado acima, foram desenvolvidos os testes que confirmam o sucesso das seguintes funcionalidades:

- Pré-processamento:
  - Leitura dos dados da *Data Warehouse*;
  - Limpeza geral dos dados;
  - Remoção de *stop words* dos dados;
  - *Lowercasing* dos dados;
  - Remoção de pontuação dos dados;
  - Divisão dos dados em dados de treino e validação.
- Modelo de classificação:
  - Treino do modelo;
  - Previsão de um assunto;
  - Persistir e carregar modelo.

No que diz respeito à cobertura de testes sob o código implementado, utilizou-se a ferramenta *coverage* [134]. Assim sendo, apresenta-se no excerto de código 7.2, os resultados obtidos para o presente projeto.

Name	Stmts	Miss	Cover	Missing
classifierTest.py	18	0	100%	
preprocessingTest.py	41	0	100%	
TOTAL	59		100%	

Excerto de Código 7.2: Cobertura de testes do *Subject Analyzer*

A análise do excerto de código 7.2, permite concluir que, para os testes unitários definidos, a cobertura dos mesmos é de 100%.

### Testes de Integração

Através da análise do modelo apresentado na subsecção 7.3.1, referente ao sistema de testes a ser utilizado, conclui-se que os testes de integração são os que sucedem os testes unitários na fase de validação. Assim, utilizam-se testes de integração para avaliar a conformidade de um sistema aquando da combinação de vários componentes individuais [135].

Dito isto, apresenta-se no excerto de código 7.3 um exemplo de um dos testes de integração desenvolvidos.

```
1 describe('SubjectAnalyzerService', () => {
2   let service: SubjectAnalyzerService;
3
4   beforeEach(() => {
5     TestBed.configureTestingModule({
6       providers: [SubjectAnalyzerService]
7     });
8
9     service = TestBed.inject(SubjectAnalyzerService);
10  });
11
12  it('Should return an integer value between 0 and 5', fakeAsync(() => {
13    service.getSubjectScore('subject test').subscribe((resp) => {
14      expect(resp.data.score >= 0 || resp.data.score <= 5).toBeTruthy()
15    };
16  });
17 }
```

Excerto de Código 7.3: Teste à integração entre o *BO-UI* e o *Services*

A análise do excerto de código 7.3 permite verificar a integração entre o *Services* e o *BO-UI*. Assim, o serviço denominado *SubjectAnalyzerService* (c.f subsecção 5.5.1) é responsável por aceder ao *Services*, através da função *getSubjectScore*, a fim de obter a classificação de assunto. Para verificar o sucesso da integração, o teste valida se a resposta do pedido efetuado é um valor entre 0 e 5.

À semelhança deste teste foi ainda efetuado o teste de integração entre o *Services* e o *Subject Analyzer* através da *API* mencionada na secção 6.4.

### Testes de Sistema

Os testes de sistema sucedem os testes de integração e têm como principal objetivo testar o sistema na totalidade, tendo por base a definição dos casos de uso [136], presente na secção 5.2. O sucesso de um teste de sistema é apenas concluído caso não falhe nenhum critério previamente definido de acordo com a definição do caso de uso. Assim, caso falhe pelo menos um critério, o teste falha.

Tendo em conta os casos de uso arquiteturalmente mais relevantes, apresentam-se nas tabelas 7.2 e 7.3, os testes de sistema para os casos de uso *UC01: Treinar Modelo de Classificação* e *UC02: Analisar Assunto*.

Tabela 7.2: Teste de Sistema para Treinar Modelo de Classificação

<b>Caso de Uso</b>	UC01
<b>Título</b>	Treinar Modelo de Classificação
<b>Cenário</b>	O sistema treina o modelo de classificação com uma recorrência mensal, através de uma <i>cron-job</i> .
<b>Procedimento</b>	O sistema inicia o processo de treino do modelo de classificação. De seguida, é executado o processo ETL que é seguido pelo pré-processamento dos dados. Por último, o sistema efetua o treino do modelo com os dados resultantes.
<b>Critérios</b>	<ol style="list-style-type: none"> <li>1. A <i>cron-job</i> executa mensalmente;</li> <li>2. O processo ETL é executado com sucesso, devolvendo um conjunto de dados atualizados relativos às campanhas;</li> <li>3. O sistema faz a leitura dos dados corretamente;</li> <li>4. O pré-processamento é realizado com sucesso;</li> <li>5. O treino do modelo é realizado com sucesso;</li> <li>6. O modelo treinado é persistido.</li> </ol>
<b>Resultados</b>	<ol style="list-style-type: none"> <li>1. Passou</li> <li>2. Passou</li> <li>3. Passou</li> <li>4. Passou</li> <li>5. Passou</li> <li>6. Passou</li> </ol>
<b>Resultado Final</b>	Passou

Tabela 7.3: Teste de Sistema para Analisar Assunto

<b>Caso de Uso</b>	UC02
<b>Título</b>	Analisar Assunto
<b>Cenário</b>	A análise de assunto é requisitada por parte do utilizador, uma vez iniciada a criação de uma campanha.
<b>Procedimento</b>	Após o utilizador criar uma campanha, o sistema disponibiliza a funcionalidade de análise de um assunto. Uma vez definido o assunto a analisar, o sistema devolve uma classificação para o assunto inserido.
<b>Critérios</b>	<ol style="list-style-type: none"> <li>1. O sistema devolve uma classificação do assunto;</li> <li>2. O sistema guarda a consulta (assunto e classificação).</li> </ol>
<b>Resultados</b>	<ol style="list-style-type: none"> <li>1. Passou</li> <li>2. Passou</li> </ol>
<b>Resultado Final</b>	Passou

Uma vez que ambos os testes passaram, pode-se concluir que os casos de uso *UC01: Treinar Modelo de Classificação* e *UC02: Analisar Assunto* foram desenvolvidos corretamente.

Os testes de sistema realizados para os restantes casos de uso encontram-se no anexo B.

### Testes de Aceitação

Os testes de aceitação constituem a etapa final da fase da avaliação e têm o propósito de avaliar se o produto final desenvolvido cumpre todos os requisitos propostos e definidos pelo cliente.

Para se proceder a esta avaliação e se determinar o resultado final de um teste, inicia-se o desenvolvimento de testes de aceitação pela definição de vários critérios relacionados com os requisitos do sistema e com a satisfação do cliente. Deste modo, para se considerar que um teste sucedeu, todos os critérios a ele correspondentes devem ser cumpridos com sucesso [137].

Na tabela 7.4 apresenta-se um dos testes de aceitação desenvolvidos, no qual se avalia um cenário de seleção de um assunto através do histórico de consultas. Nesta, apresentam-se também os critérios de satisfação definidos, advindos de toda a interação entre o utilizador e o sistema.

Tabela 7.4: Teste de Aceitação para Selecionar Assunto

<b>Cenário</b>	Selecionar assunto através do histórico de consultas após analisar diversos assuntos com recurso às dicas de boas práticas
<b>Descrição</b>	Inicialmente, o utilizador cria uma campanha e insere o seu assunto. O sistema efetua a análise do mesmo e, de seguida, o utilizador requisita a funcionalidade de assuntos para outros assuntos, recorrendo à funcionalidade de análise de boas práticas. Após a análise de todos os assuntos pretendidos pelo utilizador, o mesmo acede à funcionalidade de visualizar histórico de consultas, compara-os e seleciona o que melhor se adapta.
<b>Critérios</b>	<ol style="list-style-type: none"> <li>1. O sistema devolve uma classificação para o assunto inserido aquando da criação da campanha de <i>email</i>;</li> <li>2. O sistema guarda a consulta efetuada;</li> <li>3. O sistema permite a inserção de novo assunto;</li> <li>4. O sistema permite a análise de boas práticas;</li> <li>5. O sistema permite a visualização do histórico de consultas;</li> <li>6. O sistema permite a seleção do assunto da campanha através do histórico de consultas.</li> </ol>
<b>Resultados</b>	Todos os critérios foram cumpridos
<b>Resultado Final</b>	Aceite

Uma vez que todos os critérios foram bem sucedidos, considera-se que o teste de aceitação foi aceite.

Os restantes testes de aceitação realizados encontram-se no anexo C.

### 7.4.2 Quantitative Evaluation Framework

Tal como foi referido anteriormente, o modelo QEF tem como principal objetivo melhorar a qualidade de uma solução de *software*, através da deteção e consequente correção de eventuais falhas que possam ocorrer durante todo o processo de desenvolvimento. Assim, o modelo definido para avaliar a qualidade do *Subject Analyzer*, nomeadamente as dimensões, fatores, requisitos e pesos de avaliação referentes à taxa de cumprimento dos requisitos, encontra-se presente na figura 7.3.

Q	D	Qi	Dimensão	Qj	Wij (Peso Fator j na Dimensão i) [0,1]	Fator	rwjk (Peso Requisito k no Fator j) {2, 4, 6, 8, 10}	Requisito	wfk( % cumprimento requisito k) [0,100]
97%	0.11	100	Funcionalidade	100.00	0.64	Classificação	10.00	FC01 - Analisar assunto	100
							8.00	FC02 - Analisar restrições de boas práticas	100
							8.00	FC03 - Classificação em tempo real	100
							8.00	FC04 - Classificação gráfica, em forma de estrelas	100
							10.00	FC05 - Utilizar Tensorflow para o modelo de classificação	100
							10.00	FC06 - Utilização de dados relativos às campanhas enviadas pela E-goi	100
							10.00	FC07 - Treinar modelo com recurso a uma cron-job	100
		100.00	0.18	Histórico de consultas	10.00	FHC01 - Visualizar histórico	100		
					8.00	FHC02 - Selecionar assunto	100		
		100.00	0.18	Integração com a plataforma E-goi	10.00	FIPE01 - API REST de integração	100		
					10.00	FIPE02 - Sistema modular	100		
		100	Adaptabilidade	100.00	1.00	Manutenção	10.00	AM01 - Extração e transformação de dados	100
							10.00	AM02 - Sistema a par das tendências	100
							8.00	AM03 - Documentação da implementação	100
88.89	Eficiência	100.00	0.33	Desempenho	10.00	ED01 - Tempo de resposta adequado	100		
		83.33	0.67	Usabilidade	10.00	EU01 - Interface gráfica de classificação intuitiva	100		
							10.00	EU02 - O sistema deve tratar os erros	50

Figura 7.3: Modelo QEF

Assim, conforme se pode verificar, foram cumpridos os requisitos na totalidade, com a exceção do requisito *EU02 - O sistema deve tratar os erros*, que apresenta uma taxa de cumprimento de 50%, uma vez que, conforme os indicadores definidos, o sistema trata os erros mas não devolve informação sobre eles. Deste modo, a taxa de conclusão dos requisitos de forma a avaliar a qualidade do sistema traduz-se em 97%.

Os indicadores de avaliação referentes aos diversos fatores estabelecidos encontram-se presentes no anexo D.

### 7.4.3 Inquérito de Usabilidade

O inquérito de usabilidade, presente no anexo E, é composto por um guião que deve ser seguido pelo inquirido numa fase inicial, seguido por um questionário formado por oito questões. A avaliação das questões foi efetuada tendo em conta a hipótese relativa à usabilidade da solução - grau de usabilidade da solução igual ou superior a 80% -, assumindo um nível de significância de 0.05, com um grau de confiança de 95% (c.f subsecção 7.3.5).

Assim, o inquérito de usabilidade foi realizado a vinte e seis colaboradores E-goi. Ainda que se considere uma limitação (c.f secção 8.2), uma vez que os clientes E-goi não foram alvo do estudo, considera-se relevante a utilização e análise destes dados uma vez que os próprios colaboradores fazem uso da plataforma e têm contacto direto com o cliente.

Uma vez obtidas as respostas ao inquérito, presentes na figura G.1 do anexo G, realizaram-se os testes de hipóteses, na linguagem R, com recurso ao ambiente de desenvolvimento *RStudio* [138] de computação estatística. No entanto, antes de aplicar o teste de hipótese, torna-se necessário verificar se os dados de cada questão seguem, ou não, uma distribuição normal, sendo que estes constituem uma amostra independente, quantitativa e com uma dimensão de vinte e seis. Para tal, aplicam-se testes de normalidade *Shapiro-Wilk* [139].

Deste modo, dada a extensão das questões existentes neste inquérito, apresenta-se, de seguida, a avaliação das questões que se consideram mais relevantes tendo em consideração os objetivos definidos na secção 1.5.

No que diz respeito à questão 1 - Considera que usaria este sistema com frequência? - , aplica-se o teste *Shapiro-Wilk*, que se encontra no excerto de código 7.4, tendo em conta as seguintes hipóteses:

- $H_0$ : Os dados da amostra seguem uma distribuição normal;
- $H_1$ : Os valores da amostra não seguem uma distribuição normal.

```
1 > question1 <- c(2,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5)
2 > shapiro.test(question1)
3
4 Shapiro-Wilk normality test
5
6 data: question1
7 W = 0.6841, p-value = 3.237e-06
```

Excerto de Código 7.4: Teste de normalidade *Shapiro-Wilk* aplicado à questão 1 do inquérito de usabilidade

Pela observação do excerto de código 7.4, o teste é executado na segunda linha, recebendo como parâmetro um vetor, inicializado na primeira linha, com os dados das respostas. Obtido o valor de *p-value*,  $3.237e^{-06}$ , rejeita-se a hipótese  $H_0$ , uma vez que o valor é menor do que o valor de significância de 0.05, podendo assim concluir que a amostra não segue uma distribuição normal.

Uma vez que a amostra não segue uma distribuição normal, torna-se necessário aplicar um teste não paramétrico como o teste de *Wilcoxon*, não sendo possível aplicar um teste paramétrico, como, por exemplo, o *t-test* [140]. Assim sendo, definiram-se as seguintes hipóteses para o teste de *Wilcoxon* [141]:

- $H_0$ : A média das respostas obtidas é inferior ou igual a 4;
- $H_1$ : A média das respostas obtidas é superior a 4.

O valor 4 resulta da conversão do grau de usabilidade que se pretende (80%) para a escala utilizada no inquérito (escala de 1 a 5). Deste modo, a aplicação do teste de *Wilcoxon* à questão em avaliação encontra-se no excerto de código 7.5.

```

1 > wilcox.test(question1, mu = 4, alternative = "greater")
2
3   Wilcoxon signed rank test with continuity correction
4
5 data:  question1
6 V = 45, p-value = 0.02965
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código 7.5: Teste de *Wilcoxon* aplicado à questão 1 do inquérito de usabilidade

Uma vez que o valor de *p-value* é inferior ao nível de significância, rejeita-se a hipótese  $H_0$ . Desta forma pode-se concluir que, com um grau de confiança de 95%, a concordância dos utilizadores relativamente à questão 1 é superior a 80%.

Relativamente à questão 5 - Considera que o sistema não apresenta inconsistências? - encontra-se no excerto de código 7.6 a aplicação do teste *Shapiro-Wilk* para verificação da normalidade da distribuição.

```

1 > question5 <- c(3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question5)
3
4   Shapiro-Wilk normality test
5
6 data:  question5
7 W = 0.78337, p-value = 9.304e-05

```

Excerto de Código 7.6: Teste de normalidade *Shapiro-Wilk* aplicado à questão 5 do inquérito de usabilidade

Tendo em consideração que o valor obtido de *p-value* foi  $9.304e^{-05}$ , menor do que o valor do nível de significância, conclui-se que a amostra não segue uma distribuição normal, aplicando-se, deste modo, o teste de *Wilcoxon*, conforme se pode ver no excerto de código 7.7.

```

1 > wilcox.test(question5, mu = 4, alternative = "greater")
2
3   Wilcoxon signed rank test with continuity correction
4
5 data:  question5
6 V = 58.5, p-value = 0.04574
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código 7.7: Teste de *Wilcoxon* aplicado à questão 5 do inquérito de usabilidade

Analisando o valor de *p-value* advindo do teste de *Wilcoxon* é possível aferir que a usabilidade nesta questão é superior a 80%, com um grau de confiança de 95%.

No que se refere às restantes questões, nomeadamente as questões 2, 3, 4, 6, 7 e 8, foram igualmente analisadas seguindo o mesmo processo que as questões 1 e 5, encontrando-se os respetivos testes no anexo H.

#### 7.4.4 Inquérito de Satisfação

O inquérito de satisfação foi realizado a vinte e oito colaboradores da E-goi. Assim, à semelhança do inquérito de usabilidade considera-se a origem da amostra uma limitação (c.f

secção 8.2). No entanto, considera-se relevante proceder-se à análise da satisfação.

Uma vez obtidas as respostas ao inquérito, as mesmas foram analisadas recorrendo ao processo utilizado para a análise das respostas ao inquérito de usabilidade. Isto é, com recurso ao *software RStudio*, e assumindo um nível de significância de 0.05, com um grau de confiança de 95%, executam-se testes de *Shapiro-Wilk* para verificar a normalidade da distribuição da amostra. Posteriormente, caso siga uma distribuição normal aplica-se um *t-test* e, caso contrário, um teste *Wilcoxon*. Os dados obtidos das respostas ao inquérito encontram-se na figura G.2 do anexo G.

Dada a extensão do inquérito, apresenta-se, de seguida, a análise da questão mais relevante tendo em conta os objetivos definidos. Assim sendo, aplica-se o teste *Shapiro-Wilk* à questão 1 - Considera que o sistema influenciou o impacto das suas campanhas de *email*? (excerto de código 7.8) - atendendo às seguintes hipóteses:

- $H_0$ : Os dados da amostra seguem uma distribuição normal;
- $H_1$ : Os valores da amostra não seguem uma distribuição normal.

```

1 > q1 <- c(3,3,3,3,3,3,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(q1)
3
4 Shapiro-Wilk normality test
5
6 data: q1
7 W = 0.75674, p-value = 2.016e-05

```

Excerto de Código 7.8: Teste de normalidade *Shapiro-Wilk* aplicado à questão 1 do inquérito de satisfação

Pela análise do excerto de código 7.8, o teste de normalidade sob o vetor de respostas à questão 1 originou um *p-value* de  $2.016e^{-05}$ . Dado que, o valor de *p-value* é menor do que o valor de significância, pode-se concluir que a amostra não segue uma distribuição normal, rejeitando  $H_0$ . Consequente, aplica-se o teste de *Wilcoxon*, representado no excerto de código 7.9, tendo em consideração as seguintes hipóteses:

- $H_0$ : A média das respostas obtidas é inferior ou igual a 4;
- $H_1$ : A média das respostas obtidas é superior a 4.

```

1 > wilcox.test(q1, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data: q1
6 V = 147, p-value = 0.03857
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código 7.9: Teste de *Wilcoxon* aplicado à questão 1 do Inquérito de Satisfação

O excerto de código 7.9 demonstra que o valor de *p-value* é menor do que o nível de significância definido (0.05), pelo que se conclui que, com um grau de confiança de 95%, a satisfação nesta questão é superior a 80%.

A análise das restantes questões do inquérito de satisfação encontra-se no anexo I.



## Capítulo 8

# Conclusão

O presente capítulo tem como objetivo apresentar as conclusões finais da dissertação, evidenciando, deste modo, os objetivos alcançados, as limitações do projeto e, por último, o trabalho a desenvolver no futuro.

### 8.1 Objetivos Alcançados

De modo a desenvolver o projeto em questão, seguiram-se várias fases que se consideram indispensáveis aquando do desenvolvimento de um projeto íntegro e coerente. Assim, iniciou-se o desenvolvimento desta dissertação pela introdução do problema identificado, seguindo-se a definição dos objetivos propostos que se pretendem atingir ao longo de todo o projeto. Todo o conteúdo desenvolvido teve por base os objetivos definidos sendo que, todas as decisões de investigação e desenvolvimento, visaram dar-lhes resposta, proporcionando a obtenção de conclusões específicas relacionadas com cada um dos objetivos a título individual. Com o objetivo de evitar possíveis objeções ao desenvolvimento deste sistema, efetuou-se uma análise de valor da qual se comprovou a validade da ideia. De seguida, definiram-se os requisitos funcionais e não funcionais do sistema que resultaram na sua implementação, tendo como base a arquitetura modelada. A fim de avaliar a solução implementada e os objetivos propostos, aplicaram-se testes de *software*, testes de hipóteses, inquéritos e um modelo de avaliação de qualidade denominado QEF.

Assim, seguem-se os objetivos (c.f secção 1.5) que se pretendem qualificar:

- Investigar e analisar ferramentas de análise de texto;
- Desenvolver um sistema baseado na abordagem de análise de texto investigada, capaz de devolver uma classificação em tempo real;
- Integrar o sistema na plataforma E-go;
- Avaliar a eficácia do sistema.

No que diz respeito ao primeiro objetivo proposto, que visa a investigação e análise de ferramentas de análise de texto, e, tendo em conta que o problema se enquadra no universo de ML, foram considerados alguns dos algoritmos de classificação mais utilizados para esta categoria de problemas. Assim, algoritmos como NB, DT, SVM, ANN e RNN foram estudados permitindo concluir que o melhor algoritmo a implementar seria RNN com a variante LSTM, uma vez que é uma rede capaz de processar texto com um comprimento variável, com a facilidade de aprender dependências a longo prazo, devido à sua capacidade de memória. Uma vez definido o algoritmo a aplicar, efetuou-se uma análise comparativa entre diversas ferramentas de DL que visam auxiliar a implementação do modelo. Assim, tendo

em conta os critérios definidos na subsecção 3.3.5, as tecnologias *Tensorflow* e *Keras* prevaleceram, o que levou à sua adoção. Não obstante, após a identificação da oportunidade na subsecção 4.1.1 e a fim de concluir a melhor abordagem do analisador de texto, validando a oportunidade identificada, foram criadas alternativas que, recorrendo ao método AHP, colmataram na decisão relativa ao desenvolvimento de um analisador de texto de raiz. Assim, considera-se que se atingiu o primeiro objetivo que resultou na definição de um analisador de texto com recurso ao algoritmo LSTM e às ferramentas *Tensorflow* e *Keras*.

Relativamente ao segundo objetivo, alusivo ao desenvolvimento de um sistema baseado na abordagem de análise de texto investigada, surgiu a necessidade de desenvolver um processo ETL de modo a centralizar os dados que se encontram dispersos pelas várias tabelas e base de dados do ecossistema E-goi. Advindo da definição deste processo, foi necessário definir a execução do mesmo, pelo que se optou pela execução paralela com recurso a processos, cuja justificação se encontra na subsecção 6.1.1. Seguidamente, passou-se à implementação de um modelo capaz de classificar um assunto em seis níveis (de 0 a 5 estrelas). Durante a fase de implementação, foram testadas as variantes LSTM e BiLSTM, concluindo que, com base no desempenho do modelo, a melhor variante é BiLSTM (c.f secção 6.3). Após definição da melhor variante, desenvolveu-se e treinou-se um modelo de DL com base na variante BiLSTM, permitindo concluir que se cumpriu com sucesso o segundo objetivo.

Para dar resposta ao objetivo relacionado com a integração do sistema na plataforma E-goi, foram necessários alguns passos. Inicialmente, tornou-se necessária a projeção da melhor abordagem arquitetural que permitisse, não só a integração do *Subject Analyzer* com a plataforma E-goi já existente, mas também, uma possível integração futura com a API pública que a E-goi fornece aos seus clientes. Ainda que a integração do *Subject Analyzer* com a API pública do E-goi não esteja definida como um objetivo da presente dissertação, a abordagem arquitetural desenhada pela qual se optou, teve em vista essa futura extensibilidade, promovendo assim o baixo acoplamento e a alta coesão do sistema aquando dessa nova integração. A arquitetura desenhada tem por base uma abordagem na qual o sistema de classificação comunica com o produto através de uma nova REST API implementada no âmbito deste projeto (c.f secção 6.4), cumprindo assim a restrição de implementação do modelo FURPS+ imposta na subsecção 5.3.6. Uma vez descrita a arquitetura a utilizar, tanto ao nível da modelação de componentes (c.f secção 5.7) como ao nível da interligação e comunicação entre eles (c.f subsecção 5.5.3), tornou-se relevante o desenvolvimento de *mock-ups* que permitissem a integração visual do *Subject Analyzer* na plataforma E-goi, decidindo-se o posicionamento da funcionalidade no fluxo de criação de uma campanha de *email* (c.f subsecção 5.5.4). Uma vez definida a arquitetura e os *mock-ups*, passou-se à integração da API desenvolvida com o *Services* que, por sua vez, comunica com a interface gráfica de classificação de assunto que foi desenvolvida com base no *mock-up* definido. Posto isto, conclui-se com sucesso o objetivo de integração do sistema na plataforma E-goi.

Por último, o quarto objetivo pretende avaliar a eficácia do sistema. Assim, para tornar este objetivo dimensionável, criaram-se métricas que foram avaliadas aquando da construção do modelo de classificação (c.f secção 6.3) e desenvolveram-se, ainda, testes de *software*, nomeadamente testes unitários, testes de integração, testes de sistema e testes de aceitação, modelo QEF e inquéritos de usabilidade e satisfação. No que diz respeito às métricas avaliadas, considera-se que os valores obtidos de *accuracy* e *F1-Score*, 76% e 74% respetivamente, são positivos, uma vez que o problema abordado na dissertação se trata de um problema de classificação multiclasse, no qual a dificuldade de previsão aumenta consoante

a quantidade de classes. Relativamente aos testes de *software*, todas as fases do modelo *V-Model* proposto na subsecção 7.4.1 foram concluídas com sucesso. No entanto, quanto ao modelo QEF o mesmo não se verifica, resultando numa taxa de qualidade de 97% ao invés da hipótese definida de 100%. No que se refere aos inquéritos, todos os resultados obtidos das respostas aos mesmos, sejam eles relativos à usabilidade ou à satisfação, encontram-se acima da taxa de 80% estabelecida, pelo que evidencia uma tendência positiva. Em suma, o objetivo relativo à avaliação da eficácia do sistema foi, também, bem sucedido.

## 8.2 Limitações

Ainda que se considere que o projeto desenvolvido tenha cumprido com todos os objetivos propostos, tal não significa que esteja livre de qualquer limitação. Deste modo, nesta secção, identificam-se algumas limitações que poderão ser alvo de trabalho futuro.

Relativamente ao processo ETL, considera-se que há espaço de melhoria de modo a maximizar o desempenho do processo para que este possa ser utilizado numa abordagem de execução paralela com recurso a processos e *threads* (c.f subsecção 6.1.1), sem que coloque em causa o SLA dos servidores E-goi.

No que diz respeito aos inquéritos realizados, estes revelaram dados relevantes tanto ao nível de satisfação como ao nível de usabilidade. Contudo, ainda que se considere que a amostra obtida não tenha sido reduzida, tal é visto como uma limitação na medida em que se considera que os resultados seriam mais fiáveis e íntegros, caso fossem preenchidos por clientes finais em casos de uso reais relativos ao negócio de cada cliente. Esta limitação deve-se ao facto de que o projeto desenvolvido se encontra ainda em fase de testes internos, pelo que obter *feedback* advindo de clientes finais não seria algo tangível nesta fase.

## 8.3 Trabalho Futuro

Perante todo o trabalho desenvolvido, e com base nas limitações definidas, torna-se relevante definir qual o trabalho futuro que advém do desenvolvimento desta dissertação.

Conforme referido na secção 8.1 aquando da conclusão do objetivo referente à integração do sistema com a plataforma E-goi, considera-se pertinente a integração do *Subject Analyzer* com a API pública de modo a que a funcionalidade desenvolvida seja exposta aos clientes, permitindo a sua utilização não só via plataforma E-goi mas também via API.

Quanto à execução do processo ETL, apesar de se considerar uma boa abordagem a solução escolhida na subsecção 6.1.1, considera-se que seria interessante investigar uma solução que permitisse utilizar a abordagem da execução paralela com recurso a processos e *threads*, sem que esta interfira na disponibilidade dos servidores existentes. Tal só seria possível se algum dos seguintes cenários se concretizar:

- O desempenho do processo ETL é melhorado;
- A capacidade de processamento dos servidores existentes é amplificada

Referentemente ao desempenho do modelo de DL desenvolvido, ainda que os resultados obtidos sejam bastante interessantes tendo em conta o problema de classificação multiclasse (seis classes), considera-se que há margem de melhoria, uma vez que existe margem de progressão em todas as métricas obtidas. Para tal, reconhece-se que seria relevante prosseguir com a investigação de novas técnicas de pré-processamento, a fim de eliminar ainda mais

ruído presente nos dados obtidos, assim como continuar a alimentar o modelo com dados reais advindos de novas campanhas.

Considerando a hipótese de 100% relativa à qualidade do sistema (c.f secção 7.2), denota-se que o valor obtido no QEF se encontra aquém do esperado devido ao requisito de tratamento de erros possuir uma taxa de conclusão de 50%, uma vez que, tendo em conta os indicadores representados na figura D.6 do Anexo D, o sistema trata os erros mas não devolve qualquer tipo de *feedback*. Deste modo, é oportuno aprimorar o sistema de deteção e tratamento de erros, passando a fornecer informações relevantes sobre os mesmos.

Relativamente aos inquéritos desenvolvidos, ainda que se reconheça que os colaboradores E-goi tenham uma opinião bastante relevante relativa à funcionalidade, uma vez que fazem uso da mesma aquando da utilização da plataforma para o *marketing* empresarial da E-goi (assumindo o papel de cliente final, neste cenário), considera-se interessante a nova realização desses inquéritos para que se obtenham dados advindos de clientes finais, permitindo, assim, uma análise mais fiável quanto à satisfação e usabilidade em ambientes reais com outros modelos de negócio.

## Bibliografia

- [1] *Technology in Everyday Life* | JFG Inc. URL: <https://www.jfg-nc.com/technology-in-everyday-life/> (acedido em 24/11/2020).
- [2] «A TECHNOLOGY MARKET RESEARCH FIRM Email Statistics Report, 2020-2024,» rel. téc. URL: <http://www.radicati.com>.
- [3] *Global Email Marketing Industry*. URL: [https://www.reportlinker.com/p05900877/Global-Email-Marketing-Industry.html?utm\\_source=GNW](https://www.reportlinker.com/p05900877/Global-Email-Marketing-Industry.html?utm_source=GNW) (acedido em 09/11/2020).
- [4] *Email marketing: Learn the strategies that are achieving 300% ROI and more (VB Live)* | VentureBeat. URL: <https://venturebeat.com/2016/08/17/email-marketing-learn-the-strategies-that-are-achieving-300-roi-and-more-vb-live/> (acedido em 03/12/2020).
- [5] *E-goi - Email Marketing e Marketing Automation para todos!* URL: <https://www.e-goi.com/pt/> (acedido em 24/11/2020).
- [6] *Technology Fast 500 EMEA 2013 Portuguese Winners* | Deloitte Portugal | Technology, Media Telecommunications. URL: <https://www2.deloitte.com/pt/pt/pages/technology-media-and-telecommunications/articles/Technology-Fast-500-EMEA-2013-Winners-Portugal.html> (acedido em 10/02/2021).
- [7] *Nas 100 melhores empresas para trabalhar, tecnológica portuguesa já aponta ao topo* | Conteúdo comercial | PÚBLICO. URL: <https://www.publico.pt/2019/12/20/estudiop/artigo/100-melhores-empresas-trabalhar-tecnologica-portuguesa-ja-aponta-topo-1898005> (acedido em 11/02/2021).
- [8] (PDF) *Should I open this email?* URL: [https://www.researchgate.net/publication/221517023\\_Should\\_I\\_open\\_this\\_email](https://www.researchgate.net/publication/221517023_Should_I_open_this_email) (acedido em 09/11/2020).
- [9] *Learning to predict subject-line opens for large-scale email marketing*. URL: [https://www.researchgate.net/publication/283431373\\_Learning\\_to\\_predict\\_subject-line\\_opens\\_for\\_large-scale\\_email\\_marketing](https://www.researchgate.net/publication/283431373_Learning_to_predict_subject-line_opens_for_large-scale_email_marketing) (acedido em 03/12/2020).
- [10] L. Chittenden e R. Rettie, «An evaluation of e-mail marketing and factors affecting response,» *Journal of Targeting, Measurement and Analysis for Marketing*, vol. 11, n.º 3, pp. 203–217, jan. de 2003, issn: 1479-1862. doi: 10.1057/palgrave.jt.5740078.
- [11] G. Singh, H. Singh e S. Shrivastav, «Improving Email Marketing Campaign Success Rate Using Personalization,» em, Springer, Singapore, 2019, pp. 77–83. doi: 10.1007/978-981-13-1208-3\_8. URL: [https://link.springer.com/chapter/10.1007/978-981-13-1208-3\\_8](https://link.springer.com/chapter/10.1007/978-981-13-1208-3_8).
- [12] (PDF) *EMAIL MARKETING CAMPAIGNS: THE EASIEST PATH FROM ORGANIZATIONS TO CONSUMERS – AN EXPLORATORY ASSESSMENT*. URL: [https://www.researchgate.net/publication/227352908\\_EMAIL\\_MARKETING\\_CAMPAIGNS\\_THE\\_EASIEST\\_PATH\\_FROM\\_ORGANIZATIONS\\_TO\\_CONSUMERS\\_-\\_AN\\_EXPLORATORY\\_ASSESSMENT](https://www.researchgate.net/publication/227352908_EMAIL_MARKETING_CAMPAIGNS_THE_EASIEST_PATH_FROM_ORGANIZATIONS_TO_CONSUMERS_-_AN_EXPLORATORY_ASSESSMENT) (acedido em 27/01/2021).

- [13] *(PDF) Market Segmentation: Does it work?* URL: [https://www.researchgate.net/publication/266858293\\_Market\\_Segmentation\\_Does\\_it\\_work](https://www.researchgate.net/publication/266858293_Market_Segmentation_Does_it_work) (acedido em 27/01/2021).
- [14] *Analytics*. URL: <https://analytics.google.com> (acedido em 08/02/2021).
- [15] *E-goi Support*. URL: <https://helpdesk.e-goi.com/> (acedido em 02/03/2021).
- [16] *Apache SpamAssassin: Welcome*. URL: <https://spamassassin.apache.org/> (acedido em 01/02/2021).
- [17] *All-In-One Integrated Marketing Platform for Small Business | Mailchimp*. URL: <https://mailchimp.com/> (acedido em 30/01/2021).
- [18] *Best Practices for Email Subject Lines*. URL: <https://mailchimp.com/help/best-practices-for-email-subject-lines/> (acedido em 27/01/2021).
- [19] *Subject Line Tester*. URL: <https://sendcheckit.com/email-subject-line-tester> (acedido em 30/01/2021).
- [20] *Website Scannability for UX | Adobe XD Ideas*. URL: <https://xd.adobe.com/ideas/process/user-testing/process-user-testing-website-scannability-for-ux/> (acedido em 28/01/2021).
- [21] T. Luo, S. Chen, G. Xu, J. Zhou, T. Luo, S. Chen, G. Xu e J. Zhou, «Sentiment Analysis,» em *Trust-based Collective View Prediction*, Springer New York, 2013, pp. 53–68. doi: 10.1007/978-1-4614-7202-5\_4.
- [22] *Slack Email Subject Line Tester API*. URL: <https://sendcheckit.com/email-subject-line-tester/api> (acedido em 28/01/2021).
- [23] *Mizy's Free Email Subject Line Tester*. URL: <https://automizy.com/tools/free-email-subject-line-tester/> (acedido em 30/01/2021).
- [24] *Email Marketing Automation Software | Automizy*. URL: <https://automizy.com/> (acedido em 30/01/2021).
- [25] *How does Mizy's Subject Line Tester work? | Automizy Help Center*. URL: <https://help.automizy.com/en/articles/2516829-how-does-mizy-s-subject-line-tester-work> (acedido em 28/01/2021).
- [26] *Ecommerce Email Marketing and SMS Platform | Omnisend*. URL: <https://www.omnisend.com/> (acedido em 30/01/2021).
- [27] *Free Email Subject Line Tester | Boost Open Rates | Omnisend*. URL: <https://www.omnisend.com/subject-line-tester/> (acedido em 30/01/2021).
- [28] *Accurate Reliable results so FAST you can test every subject line every time*. URL: <https://www.touchstonetests.io/> (acedido em 30/01/2021).
- [29] *Can a new subject line tool eliminate A/B testing? | Smart Insights*. URL: <https://www.smartinsights.com/email-marketing/email-creative-and-copywriting/can-new-subject-line-tool-eliminate-ab-testing/> (acedido em 30/01/2021).
- [30] *Email Subject Line Analyzer Tool - Boost Your Email Open Rates!* URL: <https://www.isitwp.com/email-subject-line-analyzer/> (acedido em 30/01/2021).
- [31] *Title case capitalization*. URL: <https://apastyle.apa.org/style-grammar-guidelines/capitalization/title-case> (acedido em 30/01/2021).
- [32] P.-N. Tan, M. Steinbach e V. Kumar, «Introduction to Data Mining Instructor's Solution Manual,» rel. téc.
- [33] *Council Post: Digital Marketing Trends In An Era Of Constant Change*. URL: <https://www.forbes.com/sites/forbescommunicationscouncil/2018/06/14/digital-marketing-trends-in-an-era-of-constant-change/?sh=76c35efd6f57> (acedido em 07/02/2021).
- [34] *NLP, AI, and Machine Learning: What's The Difference?* URL: <https://monkeylearn.com/blog/nlp-ai/> (acedido em 20/01/2021).

- [35] B. Baharudin, L. H. Lee e K. Khan, «A Review of Machine Learning Algorithms for Text-Documents Classification,» *Journal of Advances in Information Technology*, vol. 1, n.º 1, fev. de 2010, issn: 1798-2340. doi: 10.4304/jait.1.1.4-20.
- [36] *NLP Text Preprocessing: A Practical Guide and Template | by Jiahao Weng | Towards Data Science*. URL: <https://towardsdatascience.com/nlp-text-preprocessing-a-practical-guide-and-template-d8087467e79> (acedido em 08/02/2021).
- [37] *All you need to know about text preprocessing for NLP and Machine Learning - KD-nuggets*. URL: <https://www.kdnuggets.com/2019/04/text-preprocessing-nlp-machine-learning.html> (acedido em 08/02/2021).
- [38] *Your Guide to Natural Language Processing (NLP) | by Diego Lopez Yse | Towards Data Science*. URL: <https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1> (acedido em 08/02/2021).
- [39] A. L. Samuel, «Some Studies in Machine Learning Using the Game of Checkers,» *IBM Journal of Research and Development*, vol. 3, n.º 3, pp. 210–229, jul. de 1959, issn: 0018-8646. doi: 10.1147/rd.33.0210. URL: <http://ieeexplore.ieee.org/document/5392560/>.
- [40] *Computer Pioneers - Arthur Lee Samuel*. URL: <https://history.computer.org/pioneers/samuel.html> (acedido em 03/01/2021).
- [41] *What is machine learning and types of machine learning — Part-1 | by chinmay das | Towards Data Science*. URL: <https://towardsdatascience.com/what-is-machine-learning-and-types-of-machine-learning-andrews-machine-learning-part-1-9cd9755bc647> (acedido em 01/02/2021).
- [42] *(PDF) Brief study on machine learning*. URL: [https://www.researchgate.net/publication/344748536\\_Brief\\_study\\_on\\_machine\\_learning](https://www.researchgate.net/publication/344748536_Brief_study_on_machine_learning) (acedido em 03/01/2021).
- [43] *Models for machine learning – IBM Developer*. URL: <https://developer.ibm.com/articles/cc-models-machine-learning/> (acedido em 16/01/2021).
- [44] *deep learning - Explorar - Google Trends*. URL: <https://trends.google.com/trends/explore?date=all&q=deep%20learning> (acedido em 31/01/2021).
- [45] M. Jordan, J. Kleinberg e B. Schölkopf, «Pattern Recognition and Machine Learning,» rel. téc.
- [46] F. Chollet, «Deep Learning with Python,» rel. téc., 2017.
- [47] *(PDF) About the reliability of diagnostic statements: fundamentals about detection rates, false alarms, and technical requirements*. URL: [https://www.researchgate.net/publication/328860691\\_About\\_the\\_reliability\\_of\\_diagnostic\\_statements\\_fundamentals\\_about\\_detection\\_rates\\_false\\_alarms\\_and\\_technical\\_requirements](https://www.researchgate.net/publication/328860691_About_the_reliability_of_diagnostic_statements_fundamentals_about_detection_rates_false_alarms_and_technical_requirements) (acedido em 14/10/2021).
- [48] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes e D. Brown, «Text Classification Algorithms: A Survey,» doi: 10.3390/info10040150. arXiv: 1904.08067v5.
- [49] T. Pranckevičius e V. Marcinkevičius, «Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification,» *Baltic Journal of Modern Computing*, vol. 5, n.º 2, 2017, issn: 2255-8950. doi: 10.22364/bjmc.2017.5.2.05.
- [50] B. Baharudin, L. H. Lee e K. Khan, «A Review of Machine Learning Algorithms for Text-Documents Classification,» *Journal of Advances in Information Technology*, vol. 1, n.º 1, fev. de 2010, issn: 1798-2340. doi: 10.4304/jait.1.1.4-20.

- [51] D. Xhemali, C. J. Hinde e R. G. Stone, «Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages,» *International Journal of Computer Science*, vol. 4, n.º 1, pp. 16–23, 2009, issn: 1694-0784 1694-0814. URL: <http://cogprints.org/6708/>.
- [52] E. Kirkos, C. Spathis e Y. Manolopoulos, «Support vector machines, decision trees and neural networks for auditor selection,» *Journal of Computational Methods in Sciences and Engineering*, vol. 8, n.º 3, pp. 213–224, 2008, issn: 14727978. doi: 10.3233/jcm-2008-8305.
- [53] H. Chung e K. S. Shin, «Genetic algorithm-optimized long short-term memory network for stock market prediction,» *Sustainability (Switzerland)*, vol. 10, n.º 10, out. de 2018, issn: 20711050. doi: 10.3390/su10103765.
- [54] B. Baharudin, L. H. Lee e K. Khan, «A Review of Machine Learning Algorithms for Text-Documents Classification,» *Journal of Advances in Information Technology*, vol. 1, n.º 1, fev. de 2010, issn: 1798-2340. doi: 10.4304/jait.1.1.4-20.
- [55] *Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and ... in Artificial Intelligence and Applications*) by Ilias G. Maglogiannis - PDF Drive. URL: <https://www.pdfdrive.com/emerging-artificial-intelligence-applications-in-computer-engineering-real-word-ai-systems-with-applications-in-ehealth-hci-information-retrieval-and-in-artificial-intelligence-and-applications-e185706956.html> (acedido em 03/02/2021).
- [56] *Artificial Neural Networks Explained* | by Aegeus Zerium | Good Audience. URL: <https://blog.goodaudience.com/artificial-neural-networks-explained-436fcf36e75> (acedido em 05/02/2021).
- [57] F. Bre, J. M. Gimenez e V. D. Fachinotti, «Prediction of wind pressure coefficients on building surfaces using artificial neural networks,» *Energy and Buildings*, vol. 158, n.º November, pp. 1429–1441, 2018, issn: 03787788. doi: 10.1016/j.enbuild.2017.11.045.
- [58] D. W. Otter, J. R. Medina e J. K. Kalita, «A Survey of the Usages of Deep Learning for Natural Language Processing,» rel. téc., 2019. arXiv: 1807.10854v3.
- [59] W. Ng, B. Minasny, W. de Sousa Mendes e J. A. Melo Demattê, «The influence of training sample size on the accuracy of deep learning models for the prediction of soil properties with near-infrared spectroscopy data,» *SOIL*, vol. 6, n.º 2, pp. 565–578, nov. de 2020, issn: 2199398X. doi: 10.5194/soil-6-565-2020.
- [60] Y. Goldberg, Y. Liu e M. Zhang, «Neural Network Methods for Natural Language Processing,» vol. 37, 2017. doi: 10.2200/S00762ED1V01Y201703HLT037.
- [61] R. Collobert, J. Weston, J. Com, M. Karlen, K. Kavukcuoglu e P. Kuksa, «Natural Language Processing (Almost) from Scratch,» rel. téc., 2011, pp. 2493–2537.
- [62] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao e B. Xu, «Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling,» rel. téc., 2016. arXiv: 1611.06639v1.
- [63] *BBC - Homepage*. URL: <https://www.bbc.com/> (acedido em 09/02/2021).
- [64] *PyCon-Canada-2019-NLP-Tutorial/BBC News\_LSTM.ipynb at master · susanli2016/PyCon-Canada-2019-NLP-Tutorial · GitHub*. URL: [https://github.com/susanli2016/PyCon-Canada-2019-NLP-Tutorial/blob/master/BBC%20News\\_LSTM.ipynb](https://github.com/susanli2016/PyCon-Canada-2019-NLP-Tutorial/blob/master/BBC%20News_LSTM.ipynb) (acedido em 09/02/2021).
- [65] A. Parvat, J. Chavan, S. Kadam, S. Dev e V. Pathak, «A survey of deep-learning frameworks,» em *Proceedings of the International Conference on Inventive Systems*

- and Control, ICISC 2017, Institute of Electrical e Electronics Engineers Inc., out. de 2017, isbn: 9781509047154. doi: 10.1109/ICISC.2017.8068684.
- [66] *Deep Learning Framework Power Scores 2018 | Kaggle*. URL: <https://www.kaggle.com/discdiver/deep-learning-framework-power-scores-2018> (accedido em 07/02/2021).
- [67] Y. Wu, L. Liu, C. Pu, W. Cao, S. Sahin, W. Wei e Q. Zhang, «A Comparative Measurement Study of Deep Learning as a Service Framework,» doi: 10.1109/TSC.2019.2928551. arXiv: 1810.12210v2.
- [68] *TensorFlow*. URL: <https://www.tensorflow.org/> (accedido em 07/02/2021).
- [69] *TensorBoard | TensorFlow*. URL: <https://www.tensorflow.org/tensorboard> (accedido em 07/02/2021).
- [70] *Keras: the Python deep learning API*. URL: <https://keras.io/> (accedido em 07/02/2021).
- [71] *PyTorch*. URL: <https://pytorch.org/> (accedido em 07/02/2021).
- [72] *Torch | Scientific computing for LuaJIT*. URL: <http://torch.ch/> (accedido em 07/02/2021).
- [73] *The Programming Language Lua*. URL: <http://www.lua.org/> (accedido em 07/02/2021).
- [74] *Caffe | Deep Learning Framework*. URL: <https://caffe.berkeleyvision.org/> (accedido em 11/02/2021).
- [75] *Which deep learning framework is the best? | Towards Data Science*. URL: <https://towardsdatascience.com/which-deep-learning-framework-is-the-best-eb51431c39a> (accedido em 09/02/2021).
- [76] *Medium – Where good ideas find you*. URL: <https://medium.com/> (accedido em 07/02/2021).
- [77] *arXiv.org e-Print archive*. URL: <https://arxiv.org/> (accedido em 07/02/2021).
- [78] *GitHub: Where the world builds software · GitHub*. URL: <https://github.com/> (accedido em 07/02/2021).
- [79] *TensorFlow, Keras, PyTorch, Caffe - Explore - Google Trends*. URL: <https://trends.google.com/trends/explore?cat=1299&date=all&q=%2Fg%2F11bwp1s2k3,%2Fg%2F11c1r2rvnp,%2Fg%2F11gd3905v1,%2Fg%2F11g6ym8nbt> (accedido em 05/03/2021).
- [80] N. Rich e M. Holweg, «Report produced for the EC funded project INNOREGIO: dissemination of innovation and knowledge management techniques VALUE ANALYSIS VALUE ENGINEERING,» rel. téc.
- [81] K. B. Kahn, S. E. Kay, R. J. Slotegraaf e S. Uban, «THE PDMA HANDBOOK OF NEW PRODUCT DEVELOPMENT THIRD EDITION,» rel. téc.
- [82] P. A. Koen, G. M. Ajamian, S. Boyce, A. Clamen, E. Fisher, S. Fountoulakis, A. Johnson, P. Puri e R. Seibert, «Fuzzy Front End : and Techniques,» *Industrial Research*, vol. pp, pp. 5–35, 1996.
- [83] P. Smith e D. Reinertsen, «Developing products in half the time,» *Choice Reviews Online*, vol. 28, n.º 10, jun. de 1991, issn: 0009-4978. doi: 10.5860/choice.28-5737.
- [84] R. D. Todor, «Promotion and communication through e-mail marketing campaigns,» *Economic Sciences*, vol. 10, n.º 1, pp. 61–66, 2017.
- [85] *(PDF) SWOT ANALYSIS: A THEORETICAL REVIEW*. URL: [https://www.researchgate.net/publication/319367788\\_SWOT\\_ANALYSIS\\_A\\_THEORETICAL\\_REVIEW](https://www.researchgate.net/publication/319367788_SWOT_ANALYSIS_A_THEORETICAL_REVIEW) (accedido em 07/12/2020).

- [86] N. S. Sahni, S. C. Wheeler e P. Chintagunta, «Personalization in email marketing: The role of noninformative advertising content,» *Marketing Science*, vol. 37, n.º 2, pp. 236–258, mar. de 2018, issn: 1526548X. doi: 10.1287/mksc.2017.1066.
- [87] R. E. Junusi, «Digital Marketing During the Pandemic Period; A Study of Islamic Perspective,» *Journal of Digital Marketing and Halal Industry*, vol. 2, n.º 1, p. 15, jul. de 2020, issn: 2716-4810. doi: 10.21580/jdmhi.2020.2.1.5717.
- [88] *COVID-19's Impact on Email Marketing Performance | Omnisend*. URL: <https://www.omnisend.com/resources/reports/covid-email-marketing-statistics/> (acedido em 31/01/2021).
- [89] *email subject - Explorar - Google Trends*. URL: <https://trends.google.com/trends/explore?date=2006-01-01%202021-01-01&q=email%20subject> (acedido em 31/01/2021).
- [90] *open rate - Explorar - Google Trends*. URL: <https://trends.google.com/trends/explore?date=2006-01-01%202021-01-01&q=open%20rate> (acedido em 31/01/2021).
- [91] A. Salem Khalifa, «Customer value: A review of recent literature and an integrative configuration,» *Management Decision*, vol. 42, n.º 5, pp. 645–666, jun. de 2004, issn: 00251747. doi: 10.1108/00251740410538497.
- [92] *Creating Customer Value, Satisfaction, and Loyalty PART II ...* URL: <https://www.yumpu.com/en/document/read/5593567/creating-customer-value-satisfaction-and-loyalty-part-ii-> (acedido em 02/02/2021).
- [93] *Value Proposition Canvas – Download the Official Template*. URL: <https://www.strategyzer.com/canvas/value-proposition-canvas> (acedido em 31/01/2021).
- [94] Warwick Manufacturing Group, «Product Excellence using Six Sigma: Quality Function Deployment,» *Product Excellence using Six Sigma*, n.º January, pp. 1–67, 2007. URL: [http://www2.warwick.ac.uk/fac/sci/wmg/ftmsc/modules/modulelist/peuss/slides/section\\_6a\\_qfd\\_notes.pdf](http://www2.warwick.ac.uk/fac/sci/wmg/ftmsc/modules/modulelist/peuss/slides/section_6a_qfd_notes.pdf).
- [95] L. O. Alrabghi, «QFD IN SOFTWARE ENGINEERING,» rel. téc., 2013.
- [96] T. L. Saaty, *The analytic hierarchy process : planning, priority setting, resource allocation*. McGraw-Hill International Book Co, 1980, p. 287.
- [97] C. S. Marins, D. d. O. Souza e M. d. S. Barros, «O Uso Do Método De Análise Hierárquica ( AHP ) Na Tomada De Decisões Gerenciais - Um Estudo,» *Xli Sbp*, p. 11, 2009.
- [98] *Whimsical: The Visual Workspace*. URL: <https://whimsical.com/> (acedido em 27/01/2021).
- [99] T. L. Saaty, «Decision making with the Analytic Hierarchy Process,» *Scientia Iranica*, vol. 9, n.º 3, pp. 215–229, 2002, issn: 10263098. doi: 10.1504/ijssci.2008.017590.
- [100] P. Ralph e Y. Wand, «A proposal for a formal definition of the design concept,» em *Lecture Notes in Business Information Processing*, vol. 14 LNBIP, Springer Verlag, 2009, pp. 103–136, isbn: 3540929657. doi: 10.1007/978-3-540-92966-6\_6. URL: [https://link.springer.com/chapter/10.1007/978-3-540-92966-6\\_6](https://link.springer.com/chapter/10.1007/978-3-540-92966-6_6).
- [101] *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process (2nd Edition): Larman, Craig: 0076092011491: Amazon.com: Books*. (acedido em 09/02/2021).
- [102] *cron-job.org - Free cronjobs - from minutely to once a year*. URL: <https://cron-job.org/en/> (acedido em 23/02/2021).
- [103] *E-goi - APIv3*. URL: <https://developers.e-goi.com/api/v3/#> (acedido em 22/02/2021).

- [104] *E.W. Dijkstra Archive: On the role of scientific thought (EWD447)*. URL: <https://www.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html> (acedido em 23/02/2021).
- [105] A. Kabiri e D. Chiadmi, «Survey on ETL processes,» *Journal of Theoretical and Applied Information Technology*, vol. 54, n.º 2, pp. 219–229, 2013, issn: 18173195.
- [106] R. Root e C. Mason, «ETL Processing with SQL,» *Pro SQL Server 2012 BI Solutions*, pp. 213–251, 2012. doi: 10.1007/978-1-4302-3489-0\_6.
- [107] J. C. Nwokeji, F. Aqlan, A. Apoorva e A. Olagunju, «Big data etl implementation approaches: A systematic literature review,» *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, vol. 2018-July, n.º July, pp. 714–715, 2018, issn: 23259086. doi: 10.18293/SEKE2018-152.
- [108] Thirumagal R, «ETL TOOLS IN DATA MINING A REVIEW,» *INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS*, vol. 2, n.º 1, 2014, issn: 2320-7345.
- [109] *demoji · PyPI*. URL: <https://pypi.org/project/demoji/> (acedido em 02/09/2021).
- [110] *Index of /Public/emoji*. URL: <https://unicode.org/Public/emoji/> (acedido em 02/09/2021).
- [111] *spaCy · Industrial-strength Natural Language Processing in Python*. URL: <https://spacy.io/> (acedido em 24/09/2021).
- [112] T. Dozat, «Incorporating Nesterov Momentum into Adam,» *ICLR Workshop*, n.º 1, pp. 2013–2016, 2016.
- [113] N. Reimers e I. Gurevych, «Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks,» rel. téc. arXiv: 1707.06799v2. URL: <https://arxiv.org/abs/1707.09861>.
- [114] *Common Loss functions in machine learning | by Ravindra Parmar | Towards Data Science*. URL: <https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23> (acedido em 29/09/2021).
- [115] N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. Tak e P. Tang, «ON LARGE-BATCH TRAINING FOR DEEP LEARNING: GENERALIZATION GAP AND SHARP MINIMA,» rel. téc. arXiv: 1609.04836v2.
- [116] F. A. Gers, J. Schmidhuber e F. Cummins, «Learning to forget: Continual prediction with LSTM,» *Neural Computation*, vol. 12, n.º 10, pp. 2451–2471, 2000, issn: 08997667. doi: 10.1162/089976600300015015.
- [117] *An empirical exploration of recurrent network architectures | Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. URL: <https://dl.acm.org/doi/10.5555/3045118.3045367> (acedido em 11/10/2021).
- [118] *sklearn.model\_selection.GridSearchCV - scikit - learn 1.0 documentation*. URL: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html) (acedido em 11/10/2021).
- [119] *The Falcon Web Framework — Falcon 3.0.1 documentation*. URL: <https://falcon.readthedocs.io/en/stable/> (acedido em 13/09/2021).
- [120] *API Documentation & Design Tools for Teams | Swagger*. URL: <https://swagger.io/> (acedido em 13/09/2021).
- [121] *(PDF) Quantitative Evaluation Framework (QEF)*. URL: [https://www.researchgate.net/publication/257579051\\_Quantitative\\_Evaluation\\_Framework\\_QEF](https://www.researchgate.net/publication/257579051_Quantitative_Evaluation_Framework_QEF) (acedido em 20/02/2021).

- [122] *ISO 9241-11:2018(en), Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*. URL: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> (acedido em 15/02/2021).
- [123] *Customer Satisfaction - How to Measure Satisfaction of Customers*. URL: <https://corporatefinanceinstitute.com/resources/knowledge/other/measuring-customer-satisfaction/> (acedido em 21/02/2021).
- [124] *Software Testing Tutorial - Tutorialspoint*. URL: [https://www.tutorialspoint.com/software\\_testing/index.htm](https://www.tutorialspoint.com/software_testing/index.htm) (acedido em 20/02/2021).
- [125] *What is V-model and W-model in Software Testing [What are They]*. URL: <https://www.testbytes.net/blog/v-model-and-w-model-software-testing/> (acedido em 17/02/2021).
- [126] *Prioritizing Web Usability - Jakob Nielsen, Hoa Loranger - Google Books*. URL: [https://books.google.pt/books/about/Prioritizing\\_Web\\_Usability.html?id=mL4QAQAAMAJ&redir\\_esc=y](https://books.google.pt/books/about/Prioritizing_Web_Usability.html?id=mL4QAQAAMAJ&redir_esc=y) (acedido em 17/02/2021).
- [127] *System Usability Scale (SUS) | Usability.gov*. URL: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html> (acedido em 18/02/2021).
- [128] *10 Usability Heuristics for User Interface Design*. URL: <https://www.nngroup.com/articles/ten-usability-heuristics/> (acedido em 18/02/2021).
- [129] *A Gentle Introduction to Statistical Hypothesis Testing*. URL: <https://machinelearningmastery.com/statistical-hypothesis-tests/> (acedido em 18/02/2021).
- [130] *Applied Statistics Using SPSS, STATISTICA, MATLAB and R - Joaquim P. Marques de Sá - Google Books*. URL: <https://books.google.pt/books?id=9Nf0vQAACAAJ> (acedido em 18/02/2021).
- [131] *Unit Testing: Tutorial, Types, Frameworks, Process, Techniques | Professionalqa*. URL: <https://www.professionalqa.com/unit-testing> (acedido em 01/10/2021).
- [132] *Arrange Act Assert | JustMock Documentation | Telerik JustMock*. URL: <https://docs.telerik.com/devtools/justmock/basic-usage/arrange-act-assert> (acedido em 04/10/2021).
- [133] *Module: tf.test | TensorFlow Core v2.6.0*. URL: [https://www.tensorflow.org/api\\_docs/python/tf/test](https://www.tensorflow.org/api_docs/python/tf/test) (acedido em 04/10/2021).
- [134] *Coverage.py — Coverage.py 5.1 documentation*. URL: <https://coverage.readthedocs.io/en/coverage-5.1/> (acedido em 04/10/2021).
- [135] *Testing in Software Development - British Computer Society. Working Group on Testing - Google Books*. URL: [https://books.google.pt/books?id=utFCImZOTEIC&pg=PA73&dq=integration+test&hl=en&sa=X&ei=4EpTV0vJMayu7Aak5YCIDA&redir\\_esc=y#v=onepage&q=integration%20test&f=false](https://books.google.pt/books?id=utFCImZOTEIC&pg=PA73&dq=integration+test&hl=en&sa=X&ei=4EpTV0vJMayu7Aak5YCIDA&redir_esc=y#v=onepage&q=integration%20test&f=false) (acedido em 29/09/2021).
- [136] *Data-warehouse of Software Testing | Professionalqa.com*. URL: <https://www.professionalqa.com/> (acedido em 24/09/2021).
- [137] *(PDF) User Acceptance Test*. URL: [https://www.researchgate.net/publication/300573862\\_User\\_Acceptance\\_Test](https://www.researchgate.net/publication/300573862_User_Acceptance_Test) (acedido em 29/09/2021).
- [138] *RStudio | Open source professional software for data science teams - RStudio*. URL: <https://www.rstudio.com/> (acedido em 01/10/2021).
- [139] S. S. Shapiro e M. B. Wilk, «An Analysis of Variance Test for Normality (Complete Samples),» *Biometrika*, vol. 52, n.º 3/4, p. 591, dez. de 1965. doi: 10.2307/2333709.

- 
- [140] STUDENT, «THE PROBABLE ERROR OF A MEAN,» *Biometrika*, vol. 6, n.º 1, pp. 1–25, mar. de 1908, issn: 0006-3444. doi: 10.1093/BIOMET/6.1.1. URL: <https://academic.oup.com/biomet/article/6/1/1/225634>.
- [141] F. Wilcoxon, «Individual Comparisons by Ranking Methods,» *Biometrics Bulletin*, vol. 1, n.º 6, p. 80, dez. de 1945. doi: 10.2307/3001968.



## Anexo A

# Documentação REST API

Neste anexo, apresentam-se os dois pedidos implementados de forma detalhada. Assim, apresenta-se na figura A.1 o pedido referente à classificação de um assunto e, na figura A.2, o pedido referente ao treino do modelo.

GET /subjectanalyzer/rating/{subject} Predicts the classification of a subject

Parameters Try it out

Name	Description
subject * required	Subject to classify
string (path)	<input type="text" value="subject"/>

Responses Response content type: application/json

Code	Description
200	Ok
405	Invalid Input
500	Internal Server Error

```
{ "rating": 0 }
```

Figura A.1: Pedido GET responsável por devolver a classificação de um assunto

POST /subjectanalyzer/training Train the model

Parameters Try it out

No parameters

Responses Response content type: application/json

Code	Description
200	Ok
500	Internal Server Error

Figura A.2: Pedido POST responsável por treinar o modelo



## Anexo B

# Testes de Sistema

Neste anexo, apresentam-se os restantes testes de sistema desenvolvidos. Assim, apresentam-se, nas tabelas B.1, B.2 e B.3, os testes de sistema que dizem respeito ao UC03, UC04 e UC05, respetivamente.

Tabela B.1: Teste de Sistema para Analisar Restrições de Boas Práticas

<b>Caso de Uso</b>	UC03
<b>Título</b>	Analisar Restrições de Boas Práticas
<b>Cenário</b>	A análise de restrições de boas práticas é requisitada por parte do utilizador, uma vez iniciada a criação de uma campanha.
<b>Procedimento</b>	Após o utilizador criar uma campanha, o sistema disponibiliza a funcionalidade de análise de restrições de boas práticas. Uma vez requisitada a análise de restrições de boas práticas, o sistema analisa o assunto de acordo com as boas práticas existentes, e apresenta <i>feedback</i> visual quanto à criação de um assunto.
<b>Critérios</b>	1. O sistema devolve uma representação visual do <i>feedback</i> quanto à análise de boas práticas do assunto.
<b>Resultados</b>	1. Passou
<b>Resultado Final</b>	Passou

Tabela B.2: Teste de Sistema para Visualizar Histórico de Consultas

<b>Caso de Uso</b>	UC04
<b>Título</b>	Visualizar Histórico de Consultas
<b>Cenário</b>	A visualização do histórico de consultas é requisitada por parte do utilizador, uma vez consultado, pelo menos, um assunto.
<b>Procedimento</b>	Após o utilizador definir um assunto e o sistema devolver uma classificação, o sistema disponibiliza a funcionalidade de visualizar o histórico de consultas. Uma vez requisitada a visualização das consultas, o sistema mostra uma lista de consultas, em que cada consulta é composta pelo assunto analisado e respetiva classificação.
<b>Critérios</b>	<ol style="list-style-type: none"> <li>1. O sistema não disponibiliza a funcionalidade caso não tenha sido analisado nenhum assunto;</li> <li>2. O sistema devolve uma lista de consultas, caso tenha sido analisado, pelo menos, um assunto;</li> <li>3. O sistema, para cada consulta, devolve o assunto e respetiva classificação.</li> </ol>
<b>Resultados</b>	<ol style="list-style-type: none"> <li>1. Passou</li> <li>2. Passou</li> <li>3. Passou</li> </ol>
<b>Resultado Final</b>	Passou

Tabela B.3: Teste de Sistema para Selecionar Assunto

<b>Caso de Uso</b>	UC04
<b>Título</b>	Selecionar Assunto
<b>Cenário</b>	A seleção do assunto é requisitada por parte do utilizador, uma vez visualizado o histórico de consultas.
<b>Procedimento</b>	Após o utilizador visualizar o histórico de consultas, o sistema disponibiliza a funcionalidade de selecionar um assunto a partir do histórico para a respetiva campanha. Uma vez selecionado, o assunto da campanha é atualizado.
<b>Critérios</b>	<ol style="list-style-type: none"> <li>1. O sistema permite a seleção de um assunto a partir do histórico de consultas;</li> <li>2. O sistema altera o assunto da campanha aquando da sua seleção.</li> </ol>
<b>Resultados</b>	<ol style="list-style-type: none"> <li>1. Passou</li> <li>2. Passou</li> </ol>
<b>Resultado Final</b>	Passou

## Anexo C

# Testes de Aceitação

Neste anexo, apresentam-se os restantes testes de aceitação desenvolvidos. Assim, apresenta-se, na tabela C.1, o teste de aceitação que diz respeito ao treino do modelo através da *cron-job* desenvolvida.

Tabela C.1: Teste de Aceitação para Treinar Modelo

<b>Cenário</b>	No dia 1/08 é expectável que a <i>cron-job</i> execute
<b>Descrição</b>	No dia 1/08 espera-se que a <i>cron-job</i> seja automaticamente despoletada pelo sistema.
<b>Critérios</b>	<ol style="list-style-type: none"> <li>1. O sistema despoleta a <i>cron-job</i>;</li> <li>2. O sistema executa o processo ETL;</li> <li>3. O sistema atualiza a tabela <i>subject_data</i> da <i>Data Warehouse</i>;</li> <li>4. O sistema lê os dados persistidos na tabela <i>subject_data</i>;</li> <li>5. O sistema faz o pré-processamento dos dados;</li> <li>6. O sistema treina o modelo;</li> <li>7. O sistema persiste o modelo treinado.</li> </ol>
<b>Resultados</b>	Todos os critérios foram cumpridos
<b>Resultado Final</b>	Aceite



## Anexo D

# Indicadores de Avaliação do Modelo QEF

Neste anexo encontram-se os indicadores de avaliação definidos para avaliar a taxa de cumprimento de cada requisito. Assim, apresentam-se nas figuras D.1, D.2, D.3, D.4, D.5 e D.6, os indicadores de avaliação referentes aos fatores classificação, histórico de consultas, integração com a plataforma E-goi, manutenção, desempenho e usabilidade, respetivamente.

Dimensão	Funcionalidade
Fator	Classificação

Requisito	Avaliação	Wfk - Cumprimento (%)				
		0	25	50	75	100
FC01 - Analisar assunto	O sistema deve ser capaz de analisar o assunto com base num modelo treinado e devolver uma classificação ao utilizador	Não implementado	-	-	-	Implementado na totalidade
FC02 - Analisar restrições de boas práticas	O sistema deve ser capaz de analisar o assunto com base em restrições de boas práticas e fornecer dicas ao utilizador	Não implementado	Fornecer sugestões de boas práticas sem analisar o assunto	Apenas são analisadas algumas das métricas relativas a boas práticas	Apenas são analisadas algumas das métricas relativas a boas práticas e retornadas sugestões	São analisadas todas as métricas relativas a boas práticas e retornadas sugestões
FC03 - Classificação em tempo real	O sistema deve ser capaz de devolver uma classificação em tempo real, não obrigando o utilizador a executar nenhuma ação para obter a classificação	Não implementado	-	-	-	Implementado na totalidade
FC04 - Classificação gráfica, em forma de estrelas	O sistema deve devolver uma classificação gráfica, em forma de estrelas	Não implementado	-	-	-	Implementado na totalidade
FC05 - Utilizar Tensorflow para o modelo de classificação	O modelo de classificação deve recorrer ao uso da tecnologia tensorflow	Não implementado	-	-	-	Implementado na totalidade
FC06 - Utilização de dados relativos às campanhas enviadas pela E-goi	Os dados utilizados no modelo de classificação devem ser relativos às campanhas enviadas pela E-goi	Não implementado	-	Utilização parcial dos dados provenientes de campanhas enviadas pela E-goi	-	Implementado na totalidade
FC07 - Treinar modelo com recurso a uma cron-job	O treino do modelo deve ser realizado recorrentemente através de uma cron-job	Não implementado	-	O treino do modelo é realizado recorrentemente mas sem recurso a uma cron-job	-	Implementado na totalidade

Figura D.1: Indicadores de avaliação referentes ao fator classificação da dimensão funcionalidade

Fator	Histórico de consultas
-------	------------------------

Requisito	Avaliação	Wfk - Cumprimento (%)				
		0	25	50	75	100
FHC01 - Visualizar histórico	O sistema deve permitir um histórico de consultas efetuadas para cada campanha de email	Não implementado		O sistema apenas devolve os assuntos já analisados, sem classificação		O sistema devolve os assuntos analisados juntamente com a sua classificação
FHC02 - Selecionar assunto	O sistema deve permitir a escolha de um assunto através do histórico de consultas	Não implementado	-	Aquando da seleção, fica guardado em memória	-	Aquando da seleção, o assunto é persistido na base de dados

Figura D.2: Indicadores de avaliação referentes ao fator histórico da dimensão funcionalidade

Fator	Integração com a plataforma E-goi
-------	-----------------------------------

Requisito	Avaliação	Wfk - Cumprimento (%)				
		0	25	50	75	100
FIPE01 - API REST de integração	Deve ser implementada uma API REST para integração com a plataforma E-goi	Não implementado	-	-	-	Implementado na totalidade
FIPE02 - Sistema modular	O sistema deve ser modular para facilitar a integração com a plataforma E-goi	Não implementado	-	-	-	Implementado na totalidade

Figura D.3: Indicadores de avaliação referentes ao fator integração da dimensão funcionalidade

Dimension	Adaptabilidade
Factor	Manutenção

Requirement	Metric Evaluation	Wfk - Fullfilment (%)				
		0	25	50	75	100
AM01 - Extração e transformação de dados	Deve ser criado um processo capaz de extrair e transformar os dados a fim de os fornecer ao modelo de classificação	Não Implementado	-	O processo extrai os dados mas não os transforma	-	Implementado na totalidade
AM02 - Sistema a par das tendências	O sistema deve manter-se a par das tendências dos clientes	Não Implementado	-	-	-	Implementado na totalidade
AM03 - Documentação da implementação	As decisões de arquitetura e o código devem ser documentados a fim de facilitar a deteção de erros	Não Implementado	-	Apenas se encontra documentado o código ou as decisões arquiteturais	-	Implementado na totalidade

Figura D.4: Indicadores de avaliação referentes ao fator adaptabilidade da dimensão manutenção

Dimension	Eficiência
Factor	Desempenho

Requirement	Metric Evaluation	Wfk - Fullfilment (%)				
		0	25	50	75	100
ED01 - Tempo de resposta adequado	O tempo de resposta do sistema deve ser adequado de forma a não comprometer a experiência do utilizador	O tempo de resposta não é adequado e compromete totalmente a experiência do utilizador	O tempo de resposta não é adequado e compromete parcialmente a experiência do utilizador	O tempo de resposta é adequado mas compromete a experiência do utilizador	O tempo de resposta é adequado mas compromete parcialmente a experiência do utilizador	O tempo de resposta é adequado e não compromete a experiência do utilizador

Figura D.5: Indicadores de avaliação referentes ao fator desempenho da dimensão eficiência

Factor	Usabilidade
--------	-------------

Requirement	Metric Evaluation	Wfk - Fullfilment (%)				
		0	25	50	75	100
EU01 - Interface gráfica de classificação intuitiva	A interface deve ser intuitiva permitindo a sua utilização de forma simples	A interface não é intuitiva	-	A interface é intuitiva mas a localização da funcionalidade no fluxo de criação da campanha deve ser repensada	-	A interface é intuitiva e a funcionalidade encontra-se na fase indicada do fluxo de criação da campanha
EU02 - O sistema deve tratar os erros	O sistema deve tratar os erros fornecendo detalhes claros sobre o mesmos	O sistema não trata os erros nem devolve detalhes	-	O sistema trata os erros mas não devolve informação	-	O sistema trata os erros e fornece detalhes

Figura D.6: Indicadores de avaliação referentes ao fator usabilidade da dimensão eficiência



## Anexo E

# Inquérito de Usabilidade

Neste anexo, apresentam-se o guião e o respetivo inquérito de usabilidade. Assim, antes do preenchimento do inquérito de usabilidade, o utilizador deve executar um conjunto de passos sequenciais representados no seguinte guião:

1. Crie uma campanha de *email* na plataforma E-goi.
2. Escolha o tipo de campanha (do zero, a partir de um modelo E-goi, a partir de um modelo próprio ou através de importação).
3. Escreva o assunto.
4. Verifique a classificação retornada.
5. Visualize as sugestões de boas práticas fornecidas.
6. Reproduza os passos 3 a 5 as vezes que desejar.
7. Visualize o histórico das consultas efetuadas.
8. Selecione o assunto que pretende com base nas classificações.

De seguida, apresenta-se o inquérito de usabilidade proposto.

Questão	Resposta				
	1	2	3	4	5
1. Considera que usaria este sistema com frequência?					
2. Considera o sistema intuitivo e fácil de usar?					
3. Considera que não precisaria de conhecimentos técnicos para dar uso ao sistema?					
4. Considera que o sistema não apresenta inconsistências?					
5. Considera que o sistema é fiável?					
6. Considera que o analisador de assunto se encontra bem localizado no fluxo de criação e envio da campanha?					
7. Considera que o sistema gráfico de classificação é adequado?					
8. Considera que a funcionalidade de seleção de assunto através do histórico é perceptível?					

**Legenda:** 1. Discordo Totalmente | 2. Discordo Parcialmente | 3. Indiferente | 4. Concordo Parcialmente | 5. Concordo Totalmente



## Anexo F

# Inquérito de Satisfação

Neste anexo encontra-se presente o inquérito de satisfação.

Questão	Resposta				
	1	2	3	4	5
1. Considera que o sistema influenciou o impacto das suas campanhas de <i>email</i> ?					
2. Considera útil a seleção de um assunto através do histórico?					
3. Considera que a apresentação do histórico de consultas de assuntos facilita o processo de análise de vários assuntos?					
4. Considera útil a sugestão de boas práticas?					

**Legenda:** 1. Discordo Totalmente | 2. Discordo Parcialmente | 3. Indiferente | 4. Concordo Parcialmente | 5. Concordo Totalmente



## Anexo G

# Respostas Obtidas aos Inquéritos Realizados

Neste anexo apresentam-se as respostas obtidas no inquérito de usabilidade (figura G.1) e no inquérito de satisfação (figura G.2). Assim, as respostas encontram-se agrupadas por questão, podendo deste modo, facilmente identificar o grupo de respostas que predominou - Concordo Parcialmente e Concordo Totalmente.

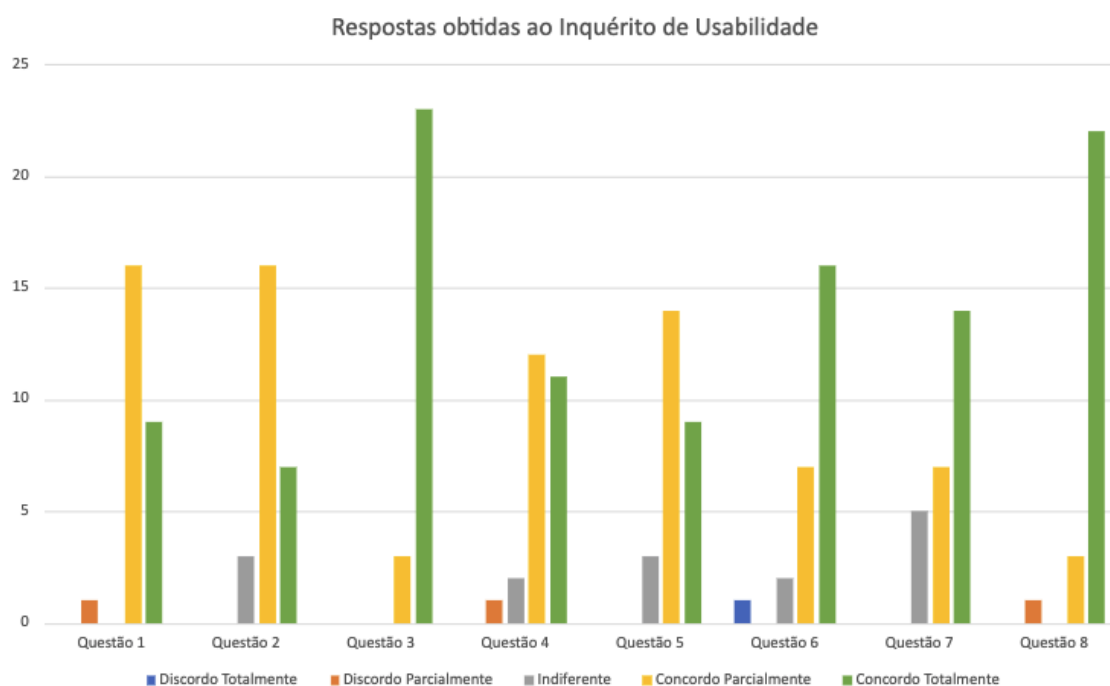


Figura G.1: Respostas obtidas ao inquérito de usabilidade

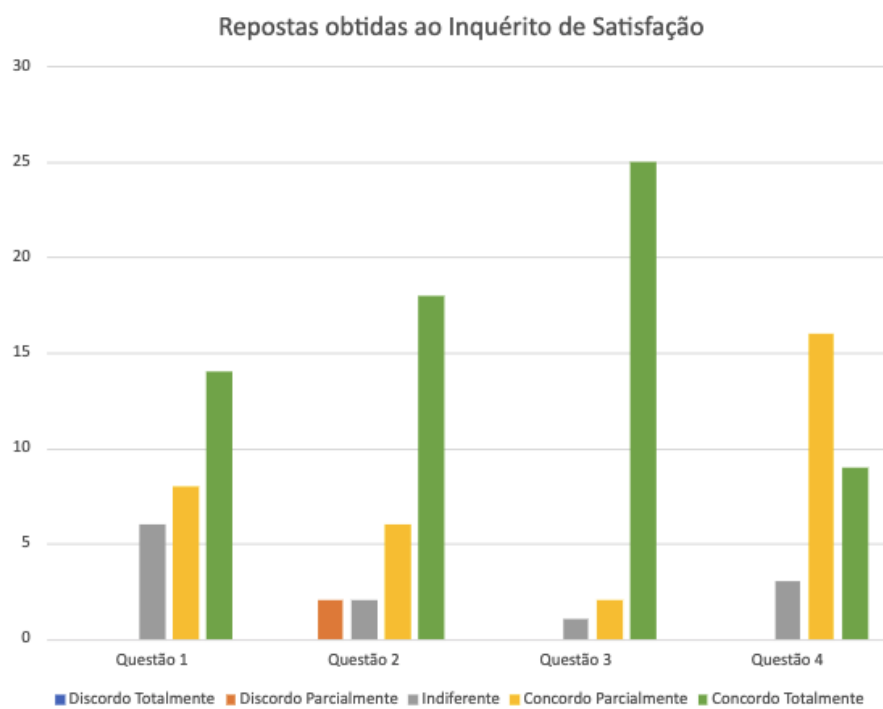


Figura G.2: Respostas obtidas ao inquérito de satisfação

## Anexo H

# Avaliação do Inquérito de Usabilidade

Neste anexo apresentam-se as análises às restantes questões do inquérito de usabilidade.

- **Questão 2:** Considera o sistema intuitivo e fácil de usar?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question2 <- c
   (3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5)
2 > shapiro.test(question2)
3
4 Shapiro-Wilk normality test
5
6 data:  question2
7 W = 0.76904, p-value = 5.475e-05

```

Excerto de Código H.1: Teste de normalidade *Shapiro-Wilk* aplicado à questão 2 do inquérito de usabilidade

Uma vez que o *p-value* é menor do que o nível de significância ( $5.475e^{-05} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question2, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data:  question2
6 V = 38.5, p-value = 0.1136
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código H.2: Teste de *Wilcoxon* aplicado à questão 2 do inquérito de usabilidade

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a usabilidade nesta questão é superior a 80%.

- **Questão 3:** Considera que não precisaria de conhecimentos técnicos para dar uso ao sistema?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question3 <- c
   (4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question3)
3
4 Shapiro-Wilk normality test
5
6 data: question3
7 W = 0.37594, p-value = 1.758e-09

```

Excerto de Código H.3: Teste de normalidade *Shapiro-Wilk* aplicado à questão 3 do inquérito de usabilidade

Uma vez que o *p-value* é menor do que o nível de significância ( $1.758e^{-09} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question3, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data: question3
6 V = 276, p-value = 8.832e-07
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código H.4: Teste de *Wilcoxon* aplicado à questão 3 do inquérito de usabilidade

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a usabilidade nesta questão é superior a 80%.

- **Questão 4:** Considera que o sistema não apresenta inconsistências?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question4 <- c
   (2,3,3,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question4)
3
4 Shapiro-Wilk normality test
5
6 data: question4
7 W = 0.78074, p-value = 8.428e-05

```

Excerto de Código H.5: Teste de normalidade *Shapiro-Wilk* aplicado à questão 4 do inquérito de usabilidade

Uma vez que o *p-value* é menor do que o nível de significância ( $8.428e^{-05} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question4, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data: question4
6 V = 77, p-value = 0.04815
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código H.6: Teste de *Wilcoxon* aplicado à questão 4 do inquérito de usabilidade

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a usabilidade nesta questão é superior a 80%.

- **Questão 6:** Considera que o analisador de assunto se encontra bem localizado no fluxo de criação e envio da campanha?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question6 <- c
   (1,3,3,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question6)
3
4 Shapiro-Wilk normality test
5
6 data:  question6
7 W = 0.65434, p-value = 1.334e-06

```

Excerto de Código H.7: Teste de normalidade *Shapiro-Wilk* aplicado à questão 6 do inquérito de usabilidade

Uma vez que o *p-value* é menor do que o nível de significância ( $1.334e^{-06} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question6, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data:  question6
6 V = 152, p-value = 0.005607
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código H.8: Teste de *Wilcoxon* aplicado à questão 6 do inquérito de usabilidade

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a usabilidade nesta questão é superior a 80%.

- **Questão 7:** Considera que o sistema gráfico de classificação é adequado?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question7 <- c
   (3,3,3,3,3,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question7)
3
4 Shapiro-Wilk normality test
5
6 data:  question7
7 W = 0.73947, p-value = 1.933e-05

```

Excerto de Código H.9: Teste de normalidade *Shapiro-Wilk* aplicado à questão 7 do inquérito de usabilidade

Uma vez que o *p-value* é menor do que o nível de significância ( $1.933e^{-05} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question7, mu = 4, alternative = "greater")
2
3   Wilcoxon signed rank test with continuity correction
4
5 data:  question7
6 V = 140, p-value = 0.02059
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código H.10: Teste de *Wilcoxon* aplicado à questão 7 do inquérito de usabilidade

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a usabilidade nesta questão é superior a 80%.

- **Questão 8:** Considera que a funcionalidade de seleção de assunto através do histórico é perceptível?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question8 <- c
2   (2,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5)
3 > shapiro.test(question8)
4
5   Shapiro-Wilk normality test
6
7 data:  question8
8 W = 0.41039, p-value = 3.537e-09

```

Excerto de Código H.11: Teste de normalidade *Shapiro-Wilk* aplicado à questão 8 do inquérito de usabilidade

Uma vez que o *p-value* é menor do que o nível de significância ( $3.537e^{-09} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question8, mu = 4, alternative = "greater")
2
3   Wilcoxon signed rank test with continuity correction
4
5 data:  question8
6 V = 253, p-value = 4.706e-05
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código H.12: Teste de *Wilcoxon* aplicado à questão 8 do inquérito de usabilidade

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a usabilidade nesta questão é superior a 80%.

## Anexo I

# Avaliação do Inquérito de Satisfação

Neste anexo apresentam-se as análises às restantes questões do inquérito de satisfação.

- **Questão 2:** Considera útil a seleção de um assunto através do histórico?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question2 <- c
   (2,2,3,3,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question2)
3
4 Shapiro-Wilk normality test
5
6 data:  question2
7 W = 0.66788, p-value = 1.034e-06

```

Excerto de Código I.1: Teste de normalidade *Shapiro-Wilk* aplicado à questão 2 do inquérito de satisfação

Uma vez que o *p-value* é menor do que o nível de significância ( $1.034e^{-06} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question2, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data:  question2
6 V = 189, p-value = 0.01333
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código I.2: Teste de *Wilcoxon* aplicado à questão 2 do inquérito de satisfação

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a satisfação nesta questão é superior a 80%.

- **Questão 3:** Considera que a apresentação do histórico de consultas de assuntos facilita o processo de análise de vários assuntos?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question3 <- c
   (3,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question3)
3
4 Shapiro-Wilk normality test
5
6 data: question3
7 W = 0.36568, p-value = 6.051e-10

```

Excerto de Código I.3: Teste de normalidade *Shapiro-Wilk* aplicado à questão 3 do inquérito de satisfação

Uma vez que o *p-value* é menor do que o nível de significância ( $6.051e^{-10} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question3, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data: question3
6 V = 337.5, p-value = 1.351e-06
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código I.4: Teste de *Wilcoxon* aplicado à questão 3 do inquérito de satisfação

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a satisfação nesta questão é superior a 80%.

- **Questão 4:** Considera útil a sugestão de boas práticas?

Aplicando o teste de *Shapiro-Wilk*:

```

1 > question4 <- c
   (3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5)
2 > shapiro.test(question4)
3
4 Shapiro-Wilk normality test
5
6 data: question4
7 W = 0.77522, p-value = 4.023e-05

```

Excerto de Código I.5: Teste de normalidade *Shapiro-Wilk* aplicado à questão 4 do inquérito de satisfação

Uma vez que o *p-value* é menor do que o nível de significância ( $4.023e^{-05} < 0.05$ ), aplica-se o teste de *Wilcoxon*:

```

1 > wilcox.test(question4, mu = 4, alternative = "greater")
2
3 Wilcoxon signed rank test with continuity correction
4
5 data: question4
6 V = 58.5, p-value = 0.04574
7 alternative hypothesis: true location is greater than 4

```

Excerto de Código I.6: Teste de *Wilcoxon* aplicado à questão 4 do inquérito de satisfação

Tendo em conta que o valor de *p-value* é menor do que 0.05 é possível concluir que a satisfação nesta questão é superior a 80%.