



**INSTITUTO  
FEDERAL**

Ceará

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ**

**CAMPUS FORTALEZA**

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES**

**MESTRADO ACADÊMICO EM ENGENHARIA DE TELECOMUNICAÇÕES**

**ELTON DIAS FREITAS**

**UMA ABORDAGEM PRÁTICA PARA IMPLEMENTAÇÃO DE ASSISTENTES  
VIRTUAIS UTILIZANDO LANGCHAIN E MODELOS DE LINGUAGEM:  
DESENVOLVIMENTO E AVALIAÇÃO DE DESEMPENHO DA ERIS, A ASSISTENTE  
VIRTUAL DA PROGEP DO IFCE**

**FORTALEZA**

**2023**

ELTON DIAS FREITAS

UMA ABORDAGEM PRÁTICA PARA IMPLEMENTAÇÃO DE ASSISTENTES VIRTUAIS  
UTILIZANDO LANGCHAIN E MODELOS DE LINGUAGEM: DESENVOLVIMENTO E  
AVALIAÇÃO DE DESEMPENHO DA ERIS, A ASSISTENTE VIRTUAL DA PROGEP DO  
IFCE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Telecomunicações do Programa de Pós-Graduação em Engenharia de Telecomunicações do Campus Fortaleza do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia de Telecomunicações. Área de Concentração: Micro-ondas e Óptica Integrada

Orientador: Prof. Dr. José Wally Mendonça Menezes

Coorientador: Prof. Dr. Robson da Silva Siqueira

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação  
Instituto Federal do Ceará - IFCE  
Sistema de Bibliotecas - SIBI  
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

---

- F862a Freitas, Elton Dias.  
UMA ABORDAGEM PRÁTICA PARA IMPLEMENTAÇÃO DE ASSISTENTES VIRTUAIS  
UTILIZANDO LANGCHAIN E MODELOS DE LINGUAGEM : AVALIAÇÃO DE DESEMPENHO DA  
ERIS, A ASSISTENTE VIRTUAL DA PROGEPI DO IFCE / Elton Dias Freitas. - 2023.  
84 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Engenharia de Telecomunicações,  
Campus Fortaleza, 2023.  
Orientação: Prof. Dr. José Wally Mendonça Menezes .  
Coorientação: Prof. Dr. Robson da Silva Siqueira.
1. Assistente virtual. 2. LLM. 3. LangChain. 4. Eris. I. Título.

CDD 621.382

---

ELTON DIAS FREITAS

UMA ABORDAGEM PRÁTICA PARA IMPLEMENTAÇÃO DE ASSISTENTES VIRTUAIS  
UTILIZANDO LANGCHAIN E MODELOS DE LINGUAGEM: DESENVOLVIMENTO E  
AVALIAÇÃO DE DESEMPENHO DA ERIS, A ASSISTENTE VIRTUAL DA PROGEF DO  
IFCE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Engenharia de Telecomunicações do Programa de Pós-Graduação em Engenharia de Telecomunicações do Campus Fortaleza do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia de Telecomunicações. Área de Concentração: Micro-ondas e Óptica Integrada

Aprovada em:

BANCA EXAMINADORA

---

Prof. Dr. José Wally Mendonça  
Menezes (Orientador)  
Instituto Federal de Educação, Ciência e Tecnologia  
do Ceará (IFCE)

---

Prof. Dr. Robson da Silva Siqueira (Coorientador)  
Instituto Federal de Educação, Ciência e Tecnologia  
do Ceará (IFCE)

---

Prof.<sup>a</sup>. Dr.<sup>a</sup>. Isabel Cristina Vieira Bento  
Instituto Federal de Educação, Ciência e Tecnologia  
do Ceará (IFCE)

---

Prof. Dr. Joel José Puga Coelho Rodrigues  
Faculdade Fecomércio

Dedico esta dissertação com profunda gratidão e amor à minha família: meus pais, meu irmão e à minha amada companheira, Virgínia Oliveira de Holanda. Expresso minha eterna gratidão pelo amor, paciência e constante incentivo.

## **AGRADECIMENTOS**

Agradeço a Deus, fonte inesgotável de força e inspiração, por me guiar com sabedoria e coragem ao longo desta jornada desafiadora do mestrado. Sua luz iluminou os caminhos mais escuros, dando-me a determinação necessária para superar obstáculos e alcançar esta conquista tão significativa.

À minha amada família: meu pai, minha mãe, meu irmão e à minha amada companheira, Virgínia Oliveira de Holanda, dedico palavras de gratidão pelo incessante incentivo, amor incondicional e apoio incansável. A importância de vocês nesse percurso não pode ser mensurada apenas em palavras, mas sim no coração, onde o calor do encorajamento de vocês sempre ardeu como uma chama constante.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), expresso minha profunda gratidão por proporcionar a oportunidade de realizar este mestrado, um capítulo enriquecedor em minha jornada acadêmica e profissional. A instituição não apenas me concedeu os recursos necessários, mas também criou um ambiente propício para o aprendizado e crescimento.

Aos respeitáveis professores José Wally Mendonça Menezes e Robson da Silva Siqueira, agradeço de coração pela orientação sábia, paciência inesgotável e pelo compartilhamento generoso de conhecimento. Sua dedicação a minha formação não apenas aprimorou meu entendimento acadêmico, mas também contribuiu para a minha evolução como profissional comprometido.

Aos meus amigos de trabalho na Superintendência de Tecnologia da Informação (STI), sou grato pelo suporte constante, compreensão e colaboração durante este período desafiador. Sua parceria e camaradagem foram fundamentais para o meu equilíbrio entre as responsabilidades profissionais e acadêmicas, tornando essa jornada mais fluida e enriquecedora.

“A ciência não trata do porquê. É sobre por que não?”

(Cave Johnson)

## RESUMO

A constante evolução tecnológica no campo da Inteligência Artificial (IA), especialmente a ascensão dos grandes modelos de linguagem, viabilizam a implementação de soluções robustas e sofisticadas para demandas em diversas áreas. Esta pesquisa propõe uma solução para aprimorar o suporte oferecido pela Pró-reitoria de Gestão de Pessoas (PROGEP) do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) aos seus servidores. A abordagem adotada envolve o desenvolvimento da assistente virtual Eris, utilizando o modelo de linguagem GPT-3.5, o *framework LangChain* e técnicas de *embeddings* de palavras. Além do processo de desenvolvimento, a pesquisa se concentra na avaliação de desempenho da Eris. Os critérios de avaliação incluem a qualidade das respostas geradas, o tempo de geração de resposta e o custo médio por geração de resposta. Através de uma escala *Likert* adaptada, o método de avaliação humana foi utilizado para pontuar as respostas geradas pela assistente. A Eris foi avaliada através de 246 perguntas sobre os temas: remoção, Programa de Gestão de Desempenho (PGD) e Plano de Desenvolvimento Pessoas (PDP). Os resultados revelam que 90,65% das respostas da Eris foram classificadas como de qualidade satisfatória ou superior, sendo que 76,02% foram consideradas equivalentes à qualidade de respostas fornecidas por especialistas humanos. No aspecto do tempo de geração de resposta, a Eris demonstrou um desempenho muito satisfatório, alcançando um tempo médio de 6,23 segundos por resposta. Ao analisar o custo médio por geração de resposta, foi obtido o valor de R\$ 0,2271 por resposta, indicando que a Eris é uma assistente virtual que está pronta para ser utilizada em um cenário real, estando apta a ser colocada em produção.

**Palavras-chave:** Assistente virtual. LLM. *LangChain*. Eris.

## ABSTRACT

The constant technological evolution in the field of Artificial Intelligence (AI), especially the rise of large language models, makes it possible to implement robust and sophisticated solutions for demands in different areas. This research proposes a solution to improve the support offered by the Dean of People Management (PROGEP) of the Federal Institute of Education, Science and Technology of Ceará (IFCE) to its employees. The approach adopted involves the development of Eris the virtual assistant, using the GPT-3.5 language model, the LangChain framework, and word embedding techniques. In addition to the development process, the research focuses on evaluating Eris' performance. Evaluation criteria include the quality of responses generated, response generation time and average cost per response generation. Using an adapted Likert scale, the human evaluation method was used to score the responses generated by the assistant. Eris was evaluated through 246 questions on the topics: removal, Performance Management Program (PGD) and People Development Plan (PDP). The results reveal that 90.65% of Eris' responses were classified as satisfactory or higher quality, with 76.02% being considered equivalent to the quality of responses provided by human experts. In terms of response generation time, Eris demonstrated very satisfactory performance, reaching an average time of 6.23 seconds per response. When analyzing the average cost per response generation, the value of R\$0.2271 per response was obtained, indicating that Eris is a virtual assistant that is ready to be used in a real scenario, being able to be put into production.

**Keywords:** Virtual assistant. LLM. LangChain. Eris.

## LISTA DE FIGURAS

|                                                                                                                                               |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1 – Agentes interagem com o ambiente através de sensores e atuadores . . . . .                                                         | 17 |
| Figura 2 – Arquétipo de assistente virtual . . . . .                                                                                          | 19 |
| Figura 3 – Exemplo de representação vetorial de uma palavra . . . . .                                                                         | 24 |
| Figura 4 – Treinamento de um modelo de aprendizado de máquina . . . . .                                                                       | 27 |
| Figura 5 – Avaliação de um modelo já treinado . . . . .                                                                                       | 27 |
| Figura 6 – Representação de uma rede neural simples . . . . .                                                                                 | 28 |
| Figura 7 – Comparação entre uma rede neural tradicional e uma rede neural profunda .                                                          | 29 |
| Figura 8 – Exemplos de utilização de modelos de aprendizado supervisionado e não supervisionado . . . . .                                     | 31 |
| Figura 9 – Camadas finais de um modelo pré-treinado substituídas após um novo treinamento em um conjunto de dados menor . . . . .             | 33 |
| Figura 10 – Visualização dos <i>tokens</i> através da ferramenta <i>Tokenizer</i> . . . . .                                                   | 35 |
| Figura 11 – Exemplo de <i>prompt</i> enviado a um <i>Large Language Model</i> (LLM) . . . . .                                                 | 36 |
| Figura 12 – Funcionamento de uma <i>Application Programming Interface</i> (API) <i>web</i> . . .                                              | 40 |
| Figura 13 – Exemplo de uma cadeia simples no <i>LangChain</i> . . . . .                                                                       | 43 |
| Figura 14 – Criação da base de conhecimento . . . . .                                                                                         | 48 |
| Figura 15 – Interface da Eris, versão <i>desktop</i> . . . . .                                                                                | 50 |
| Figura 16 – Interface da Eris, versão <i>mobile</i> , área principal (A) e versão <i>mobile</i> , com barra lateral em exibição (B) . . . . . | 51 |
| Figura 17 – Diagrama de funcionamento básico da Eris . . . . .                                                                                | 54 |
| Figura 18 – Utilização da ferramenta <i>Tokenizer</i> para verificar a quantidade de <i>tokens</i> . .                                        | 61 |
| Figura 19 – Comparação entre a pontuação da Eris (Remoção) e a pontuação de referência (máxima qualidade) . . . . .                           | 64 |
| Figura 20 – Respostas agrupadas pela pontuação recebida (Remoção) . . . . .                                                                   | 64 |
| Figura 21 – Comparação entre a pontuação da Eris (PGD) e a pontuação de referência (máxima qualidade) . . . . .                               | 66 |
| Figura 22 – Respostas agrupadas pela pontuação recebida (PGD) . . . . .                                                                       | 66 |
| Figura 23 – Comparação entre a pontuação da Eris (PDP) e a pontuação de referência (máxima qualidade) . . . . .                               | 68 |
| Figura 24 – Respostas agrupadas pela pontuação recebida (PDP) . . . . .                                                                       | 68 |

|                                                                                                                      |    |
|----------------------------------------------------------------------------------------------------------------------|----|
| Figura 25 – Comparação entre a pontuação da Eris (Geral) e a pontuação de referência<br>(máxima qualidade) . . . . . | 69 |
| Figura 26 – Respostas agrupadas pela pontuação recebida (Geral) . . . . .                                            | 70 |
| Figura 27 – Tempo de geração de respostas (Remoção) . . . . .                                                        | 72 |
| Figura 28 – Tempo de geração de respostas (PGD) . . . . .                                                            | 73 |
| Figura 29 – Tempo de geração de respostas (PDP) . . . . .                                                            | 74 |
| Figura 30 – Tempo de geração de respostas (Análise geral) . . . . .                                                  | 75 |

## LISTA DE TABELAS

|                                                                                                                    |    |
|--------------------------------------------------------------------------------------------------------------------|----|
| Tabela 1 – Documentação utilizada para compor a base de conhecimento . . . . .                                     | 46 |
| Tabela 2 – Exemplo do funcionamento da Eris . . . . .                                                              | 53 |
| Tabela 3 – Pontuação BLEU e METEOR . . . . .                                                                       | 56 |
| Tabela 4 – Exemplo de variações de pergunta frequente geradas via ChatGPT . . . . .                                | 58 |
| Tabela 5 – Pontuação da escala <i>Likert</i> modificada utilizada na avaliação humana . . . . .                    | 59 |
| Tabela 6 – Análise estatística da pontuação das respostas geradas pela Eris (Remoção) . . . . .                    | 63 |
| Tabela 7 – Análise estatística da pontuação das respostas geradas pela Eris (PGD) . . . . .                        | 65 |
| Tabela 8 – Análise estatística da pontuação das respostas geradas pela Eris (PDP) . . . . .                        | 67 |
| Tabela 9 – Análise estatística da pontuação das respostas geradas pela Eris(Geral) . . . . .                       | 69 |
| Tabela 10 – Análise do tempo de geração das respostas da Eris (Remoção) . . . . .                                  | 71 |
| Tabela 11 – Análise do tempo de geração das respostas da Eris(PGD) . . . . .                                       | 73 |
| Tabela 12 – Análise do tempo de geração das respostas da Eris (PDP) . . . . .                                      | 74 |
| Tabela 13 – Análise geral do tempo de geração das respostas da Eris . . . . .                                      | 75 |
| Tabela 14 – Quantidade de média de <i>tokens</i> contabilizados utilizando a ferramenta <i>Tokenizer</i> . . . . . | 76 |

## LISTA DE ABREVIATURAS E SIGLAS

|          |                                                                    |
|----------|--------------------------------------------------------------------|
| API      | <i>Application Programming Interface</i>                           |
| ARC      | <i>AI2 Reasoning Challenge</i>                                     |
| BERT     | <i>Bidirectional Encoder Representations from Transformer</i>      |
| BLEU     | <i>Bilingual Evaluation Understudy</i>                             |
| CICIAR   | Caixa de Intimação com Inteligência Artificial                     |
| DTL      | <i>Deep Transfer Learning</i>                                      |
| EUA      | Estados Unidos da América                                          |
| FAIR     | <i>Facebook AI Research</i>                                        |
| FAQ      | <i>Frequently Asked Questions</i>                                  |
| GPT      | <i>Generative Pre-trained Transformer</i>                          |
| GPT-2    | <i>Generative Pre-trained Transformer 2</i>                        |
| GPT-3    | <i>Generative Pre-trained Transformer 3</i>                        |
| gRPC     | <i>Google Remote Procedure Call</i>                                |
| HTTP     | <i>Hypertext Transfer Protocol</i>                                 |
| HTTPS    | <i>Hypertext Transfer Protocol Secure</i>                          |
| IA       | Inteligência Artificial                                            |
| IDF      | <i>Inverse Document Frequency</i>                                  |
| IFCE     | Instituto Federal de Educação, Ciência e Tecnologia do Ceará       |
| JSON     | <i>JavaScript Object Notation</i>                                  |
| JSON-RPC | <i>JavaScript Object Notation Remote Procedure Call</i>            |
| LiSA     | <i>Link Student Assistant</i>                                      |
| LLaMA    | <i>Large Language Model Meta AI</i>                                |
| LLM      | <i>Large Language Model</i>                                        |
| LLMs     | <i>Large Language Models</i>                                       |
| LSTM     | <i>Long Short Term Memory</i>                                      |
| METEOR   | <i>Metric for Evaluation of Translation with Explicit ORDERing</i> |
| MMLU     | <i>Massive Multitask Language Understanding</i>                    |
| OECD     | <i>Organisation for Economic Cooperation and Development</i>       |
| PCA      | <i>Principal Component Analysis</i>                                |
| PDF      | <i>Portable Document Format</i>                                    |
| PDP      | Plano de Desenvolvimento de Pessoas                                |

|        |                                                    |
|--------|----------------------------------------------------|
| PGD    | Programa de Gestão de Desempenho                   |
| PLN    | Processamento de Linguagem Natural                 |
| PNDP   | Política Nacional de Desenvolvimento de Pessoas    |
| PROGEP | Pró-reitoria de Gestão de Pessoas                  |
| RNN    | <i>Recurrent Neural Network</i>                    |
| RNP    | Rede Nacional de Pesquisa                          |
| SOAP   | <i>Simple Object Access Protocol</i>               |
| TF     | <i>Term Frequency</i>                              |
| TF-IDF | <i>Term Frequency - Inverse Document Frequency</i> |
| TIC    | Tecnologia da Informação e Comunicação             |
| XML    | <i>Extensible Markup Language</i>                  |

## SUMÁRIO

|                |                                                                                       |           |
|----------------|---------------------------------------------------------------------------------------|-----------|
| <b>1</b>       | <b>INTRODUÇÃO . . . . .</b>                                                           | <b>17</b> |
| <b>1.1</b>     | <b>Inteligência Artificial e Inovação . . . . .</b>                                   | <b>20</b> |
| <b>1.2</b>     | <b>Objetivos . . . . .</b>                                                            | <b>21</b> |
| <b>1.2.1</b>   | <b><i>Objetivo Geral . . . . .</i></b>                                                | <b>21</b> |
| <b>1.2.2</b>   | <b><i>Objetivos Específicos . . . . .</i></b>                                         | <b>21</b> |
| <b>1.3</b>     | <b>Organização do trabalho . . . . .</b>                                              | <b>21</b> |
| <b>2</b>       | <b>FUNDAMENTAÇÃO TEÓRICA . . . . .</b>                                                | <b>23</b> |
| <b>2.1</b>     | <b>Words Embeddings . . . . .</b>                                                     | <b>23</b> |
| <b>2.2</b>     | <b>Aprendizado de Máquina . . . . .</b>                                               | <b>25</b> |
| <b>2.2.1</b>   | <b><i>Redes Neurais e Aprendizado Profundo . . . . .</i></b>                          | <b>27</b> |
| <b>2.2.2</b>   | <b><i>Aprendizado Supervisionado . . . . .</i></b>                                    | <b>29</b> |
| <b>2.2.3</b>   | <b><i>Aprendizado Não-supervisionado . . . . .</i></b>                                | <b>30</b> |
| <b>2.2.4</b>   | <b><i>Aprendizado por Transferência . . . . .</i></b>                                 | <b>31</b> |
| <b>2.3</b>     | <b>Grandes Modelos de Linguagem . . . . .</b>                                         | <b>33</b> |
| <b>2.3.1</b>   | <b><i>Mecanismo de Atenção . . . . .</i></b>                                          | <b>34</b> |
| <b>2.3.2</b>   | <b><i>Token, Prompt e Parâmetros . . . . .</i></b>                                    | <b>35</b> |
| <b>2.3.2.1</b> | <b><i>Token . . . . .</i></b>                                                         | <b>35</b> |
| <b>2.3.2.2</b> | <b><i>Prompt . . . . .</i></b>                                                        | <b>36</b> |
| <b>2.3.2.3</b> | <b><i>Parâmetros . . . . .</i></b>                                                    | <b>36</b> |
| <b>2.3.3</b>   | <b><i>Zero-shot, Few-shot e Fine Tuning . . . . .</i></b>                             | <b>37</b> |
| <b>2.4</b>     | <b>Application Programming Interface . . . . .</b>                                    | <b>38</b> |
| <b>2.5</b>     | <b>Métodos de Avaliação de Agentes Conversacionais . . . . .</b>                      | <b>40</b> |
| <b>2.6</b>     | <b>LangChain . . . . .</b>                                                            | <b>42</b> |
| <b>3</b>       | <b>METODOLOGIA . . . . .</b>                                                          | <b>44</b> |
| <b>3.1</b>     | <b>Definição do Escopo e Criação da Base de Conhecimento . . . . .</b>                | <b>45</b> |
| <b>3.1.1</b>   | <b><i>Criação de índice de pesquisa com o FAISS . . . . .</i></b>                     | <b>47</b> |
| <b>3.2</b>     | <b>Desenvolvimento da Assistente Virtual Eris . . . . .</b>                           | <b>49</b> |
| <b>3.2.1</b>   | <b><i>Criação da interface da Eris . . . . .</i></b>                                  | <b>50</b> |
| <b>3.2.2</b>   | <b><i>Funcionamento da Eris . . . . .</i></b>                                         | <b>51</b> |
| <b>3.2.3</b>   | <b><i>Apresentação da primeira versão e atualizações após feedbacks da PROGEP</i></b> | <b>53</b> |

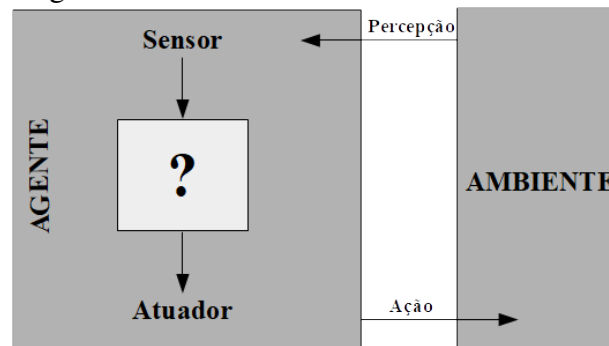
|              |                                                                                           |           |
|--------------|-------------------------------------------------------------------------------------------|-----------|
| <b>3.3</b>   | <b>Metodologia de Avaliação da Qualidade das Respostas Geradas . . . . .</b>              | <b>54</b> |
| <b>3.3.1</b> | <i>Avaliação humana . . . . .</i>                                                         | <i>57</i> |
| <b>3.4</b>   | <b>Metodologia de Avaliação do Tempo e Custo Médio para Geração de Resposta . . . . .</b> | <b>60</b> |
| <b>3.4.1</b> | <i>Avaliação do tempo de geração de resposta . . . . .</i>                                | <i>60</i> |
| <b>3.4.2</b> | <i>Avaliação do custo médio por geração de resposta . . . . .</i>                         | <i>61</i> |
| <b>4</b>     | <b>RESULTADOS E DISCUSSÃO . . . . .</b>                                                   | <b>63</b> |
| <b>4.1</b>   | <b>Qualidade das Respostas Geradas . . . . .</b>                                          | <b>63</b> |
| <b>4.1.1</b> | <i>Qualidade das respostas sobre Remoção . . . . .</i>                                    | <i>63</i> |
| <b>4.1.2</b> | <i>Qualidade das respostas das perguntas sobre (PGD) . . . . .</i>                        | <i>65</i> |
| <b>4.1.3</b> | <i>Qualidade das respostas das perguntas sobre PDP . . . . .</i>                          | <i>66</i> |
| <b>4.1.4</b> | <i>Qualidade geral das respostas geradas . . . . .</i>                                    | <i>68</i> |
| <b>4.2</b>   | <b>Tempo de Geração de Resposta . . . . .</b>                                             | <b>70</b> |
| <b>4.2.1</b> | <i>Tempo de geração de respostas das perguntas sobre Remoção . . . . .</i>                | <i>71</i> |
| <b>4.2.2</b> | <i>Tempo de geração de respostas das perguntas sobre (PGD) . . . . .</i>                  | <i>72</i> |
| <b>4.2.3</b> | <i>Tempo de geração de respostas das perguntas sobre (PDP) . . . . .</i>                  | <i>73</i> |
| <b>4.2.4</b> | <i>Tempo de geração de respostas geral . . . . .</i>                                      | <i>74</i> |
| <b>4.3</b>   | <b>Custo Médio por Geração de Resposta . . . . .</b>                                      | <b>76</b> |
| <b>5</b>     | <b>CONCLUSÃO . . . . .</b>                                                                | <b>77</b> |
|              | <b>REFERÊNCIAS . . . . .</b>                                                              | <b>80</b> |

## 1 INTRODUÇÃO

Inteligência Artificial (IA) tem sido um dos campos mais dinâmicos e transformadores da área da Tecnologia da Informação e Comunicação (TIC) nas últimas décadas. Desde sua concepção, onde Turing (1950) questiona se máquinas são capazes de pensar, até os avanços recentes, exemplificados pelos trabalhos de Chen et al. (2023), Zhang et al. (2022) e Guo et al. (2023), que introduziram, respectivamente, algoritmo de otimização para treinamento de redes neurais profundas, conjunto de decodificadores para transformadores pré-treinados e um estudo comparativo entre as respostas do ChatGPT e as respostas de especialistas humanos; a IA evoluiu de maneira extraordinária, viabilizando uma infinidade de aplicações nas mais variadas áreas. Essa revolução tecnológica está sendo impulsionada por diversas técnicas, algoritmos e modelos de aprendizado de máquinas.

A IA, como campo de estudo, tem suas raízes em tentativas iniciais de replicar a inteligência humana através de sistemas computacionais. Russell e Norvig (2013) definem IA como o estudo de agentes que recebem percepções do ambiente e a partir dessas percepções executam ações. Avanços substanciais em Aprendizado de Máquina, Processamento de Linguagem Natural (PLN), Visão Computacional e outros domínios da IA permitiram o desenvolvimento de sistemas que podem realizar tarefas complexas, como nos trabalhos revisados por Alharbi e Vakanski (2023) que apresentam métodos que utilizam aprendizado de máquina para realizar a classificação de câncer. Como consequência, a IA está cada vez mais presente em nossas vidas, seja em dispositivos móveis como *smartphones* e *tablets*, em plataformas de *streaming* e mídias sociais, recomendando conteúdo personalizado, assim como demonstrado pelo trabalho de Omeragić et al. (2023), ou em sistemas de direção autônoma em veículos, como destaca Murugan et al. (2022).

Figura 1 – Agentes interagem com o ambiente através de sensores e atuadores



Fonte: Adaptado de Russell e Norvig (2013, p. 35)

Nos últimos anos, uma das áreas de implementação de IA que mais tem se destacado é a de PLN, com foco especial na criação e aprimoramento de agentes conversacionais, como assistentes virtuais e *chatbots*. Assistentes virtuais estão se tornando altamente sofisticados e capazes de realizar tarefas que antes eram consideradas exclusivas da inteligência humana, como responder perguntas, realizar traduções, redigir textos, entre muitas outras funções. O impacto desses assistentes virtuais é evidente em uma variedade de setores, desde o atendimento automatizado ao cliente e uso em *e-commerces*, até a assistência na área da saúde e ensino à distância, transformando fundamentalmente a forma como interagimos com a tecnologia e acessamos informações.

Dahiya (2017) define agentes conversacionais como sendo programas projetados para falsificar uma comunicação inteligente via texto ou fala. Isto é, assistentes virtuais e *chatbots* são *softwares* projetados para interagir com os usuários de maneira natural, entender suas necessidades e responder com informações ou ações relevantes. Essas ferramentas desempenham um papel fundamental na automação de tarefas rotineiras, melhorando a eficiência operacional e a experiência do usuário na utilização de sistemas e recursos online.

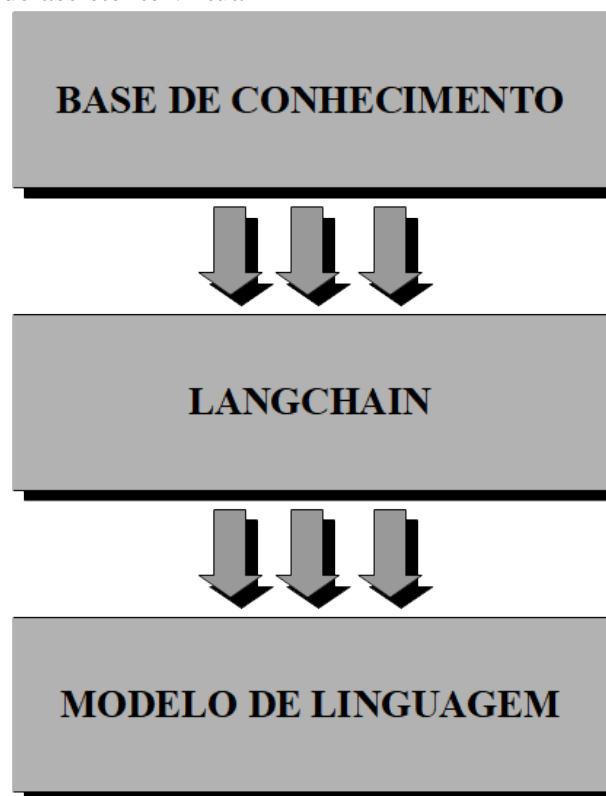
Recentemente o avanço mais notável no domínio de agentes conversacionais está diretamente relacionado ao surgimento dos *Large Language Models* (LLMs), como o *Generative Pre-trained Transformer 3* (GPT-3) e o *Large Language Model Meta AI* (LLaMA). Esses modelos de linguagem são treinados em grandes volumes de texto da internet e são capazes de realizar, com desempenho notável, as mais diversas tarefas referentes a PLN. A capacidade de compreender contexto, gerar respostas coerentes e até mesmo criar conteúdo original tornou esses modelos uma peça fundamental na criação e implementação de assistentes virtuais mais sofisticados e eficazes.

A utilização de assistentes virtuais para automatização do atendimento ao usuário está rapidamente se tornando uma tendência mundial. À medida que as instituições e organizações buscam maneiras eficientes de lidar com um volume cada vez maior de consultas e solicitações, os assistentes virtuais baseados em IA surgem como uma solução poderosa. Na área da educação o uso de *chatbots* e assistentes virtuais já é objeto de estudo. Dabitonto et al. (2018) criou o *Link Student Assistant* (LiSA), um *chatbot* desenvolvido com intuito de ajudar alunos que passam boa parte do seu dia nos campus das universidades, através do fornecimento de informações e serviços. Mais recentemente, Koivisto (2023) apresentou um estudo sobre a utilização de

*chatbots* e assistentes virtuais na realização dos serviços de tutoria e aconselhamento prestado por universidades aos seus estudantes.

No contexto do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), as instituições de ensino superior enfrentam uma série de desafios quando se trata de atender às necessidades de sua comunidade, que inclui discentes, docentes e servidores. Essas demandas variam de questões administrativas e pedagógicas, até questões pessoais. Nesse sentido, a motivação para o desenvolvimento deste trabalho foi a utilização de tecnologias emergentes para solucionar a necessidade de aprimoramento do suporte oferecido pela Pró-reitoria de Gestão de Pessoas (PROGEP) do IFCE aos seus servidores. Para isso, esta dissertação busca implementar um arquétipo de assistente virtual que utiliza uma base de conhecimento, o *framework Langchain* e um LLM.

Figura 2 – Arquétipo de assistente virtual



Fonte: autor (2023)

Este estudo apresenta uma aplicação prática do desenvolvimento e avaliação de desempenho da Eris, uma assistente virtual baseada em IA, em um cenário real de uma instituição de ensino superior, buscando contribuir para a compreensão abrangente das implicações da utilização e integração de LLMs no desenvolvimento de agentes conversacionais.

## 1.1 Inteligência Artificial e Inovação

Os conceitos e fundamentos da inovação, bem como seu papel no desenvolvimento socioeconômico, tem sido objeto de extensas pesquisas nas últimas décadas. No contexto contemporâneo, a inovação não é mais vista como uma ocorrência isolada, mas como um processo dinâmico e contínuo. A literatura atualizada destaca diversos tipos de inovação, Ribeiro et al. (2023) explica que há bastante ambiguidade sobre tipificação de inovação na literatura, porém destaca que dentre os tipos de inovação, temos: substancial, disruptiva, radical e incremental, sendo as duas últimas as mais populares. Já a *Organisation for Economic Cooperation and Development* (OECD) destaca que há dois tipos: inovação de produto e de processos de negócio (OECD; EUROSTAT, 2018).

Independente do tipo, um aspecto notável da inovação é sua forte conexão com as tecnologias emergentes e as formas de utilizá-las. Em um cenário onde a tecnologia avança a passos largos, a capacidade de uma organização de identificar, implementar ou integrar novas tecnologias como ferramenta para aprimorar ou reformular processos e produtos torna-se um fator determinante dentro do contexto de inovação. Conforme aponta Ribeiro et al. (2023), o uso de novas tecnologias digitais, como IA e seus mais variados meios de implementação, é denominado como Transformação Digital.

Vale ressaltar que a inovação não se limita apenas à esfera corporativa, estendendo-se para o âmbito governamental e social. A aplicação de tecnologias inovadoras tem o potencial de solucionar desafios e demandas complexas tanto nos setores privados quanto nos públicos, conforme demonstrado nos trabalhos de Parreiras et al. (2022), que apresenta a Caixa de Intimação com Inteligência Artificial (CICIAR) sendo utilizada na Defensoria Pública do Estado do Rio de Janeiro; e Campos e Figueiredo (2022), que realiza um estudo da análise de dados utilizando IA para apoiar a administração pública na resolução de problemas dentro do contexto de políticas públicas.

Por tanto, considerando a forte relação entre inovação e uso de tecnologias, também é objetivo deste estudo compreender os diferentes tipos de inovação que estão associados ao uso da assistente virtual desenvolvida para a PROGEP.

## 1.2 Objetivos

Considerando os pontos debatidos até o momento, seguem o objetivo geral e os objetivos específicos almejados neste trabalho.

### 1.2.1 *Objetivo Geral*

Desenvolver e avaliar o desempenho de uma assistente virtual para a PROGEP, que seja capaz de responder de maneira eficiente e precisa as dúvidas dos servidores.

### 1.2.2 *Objetivos Específicos*

- Implementar um arquétipo de assistente virtual que seja replicável para diversas áreas. Esse arquétipo utiliza uma base de conhecimento, técnicas de *word embedding*, o *framework LangChain* e um modelo de linguagem.
- Apresentar a nova ferramenta para a PROGEP para verificar os *feedbacks* sobre usabilidade da interface e aceitação de forma geral da assistente virtual desenvolvida.
- Avaliar o desempenho da assistente virtual desenvolvida, considerando três aspectos principais: qualidade das respostas geradas, tempo de geração das respostas e custo médio por geração de resposta.

## 1.3 Organização do trabalho

Esta dissertação está organizada em 5 capítulos, seguindo a ordem: Introdução, Fundamentação Teórica, Metodologia, Resultados e Conclusões.

O Capítulo 2 apresenta todo o embasamento teórico necessário para a compreensão deste estudo. Este capítulo detalha conceitos importantes que constituem a base do trabalho, incluindo Grandes Modelos de Linguagem, do inglês *Large Language Model* (LLM), *embeddings*, *tokens*, Interface de Programação de Aplicação, do inglês *Application Programming Interface* (API); e métodos de avaliação de agentes conversacionais. O entendimento desses conceitos é essencial para compreender plenamente a implementação, o funcionamento e a relevância do projeto, proporcionando uma base sólida para a análise e a interpretação dos resultados apresentados posteriormente na dissertação.

Na sequência, no Capítulo 3 da dissertação, é apresentada de forma detalhada a metodologia de implementação da assistente virtual Eris, bem como o seu funcionamento. Este capítulo descreve o processo de desenvolvimento, desde a criação da base de conhecimento, seleção de ferramentas e tecnologias, até a etapa de apresentação da primeira versão à PROGEP. Além disso, também é discutido o método de avaliação de desempenho da Eris, que se concentra em três aspectos: qualidade das respostas geradas, que mede a precisão e a relevância das informações fornecidas quando comparada com as respostas de especialistas humanos; o tempo de geração das respostas, que avalia a eficiência do sistema na entrega de respostas em tempo hábil; e o custo médio por geração de resposta, que demonstra o custo real da utilização da nova ferramenta desenvolvida.

No Capítulo 4 são apresentados os resultados detalhados da avaliação de desempenho da assistente virtual Eris. Essa seção do trabalho oferece uma análise minuciosa dos dados obtidos durante o processo de avaliação, apresentando-os de forma clara e visualmente informativa por meio de gráficos e tabelas. Este capítulo fornece evidências concretas da performance da Eris nos aspectos avaliados: qualidade das respostas geradas, tempo de geração das respostas e custo médio por geração de resposta.

Por fim, no Capítulo 5 são apresentadas as conclusões obtidas a partir deste trabalho de mestrado, destacando as principais contribuições para o campo de estudo e indicando sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Words Embeddings

*Word embeddings* são uma forma de representação de palavras ou frases em um espaço vetorial de alta dimensão. Essa representação é obtida através do uso de algoritmos de aprendizado de máquina especificamente treinados para reconhecer contextos semânticos e sintáticos de palavras em um grande conjunto de dados de texto. Os *embeddings* de palavras são capazes de capturar várias dimensões de significado de uma palavra, pois cada dimensão do vetor pode representar um aspecto diferente. Desta forma, uma dimensão do vetor pode identificar a classe da palavra: verbo, substantivo, adjetivo, etc., enquanto outra dimensão pode identificar se a palavra aplica algum sentido ao contexto em que está inserida: positivo, negativo, dúvida, etc. A semântica vetorial surgiu com a hipótese da incorporação da semântica linguística, onde as técnicas de *word embedding* são utilizadas para aprender a representar o significado das palavras a partir de suas distribuições nos textos (JURAFSKY; MARTIN, 2020).

Um exemplo ilustrativo de como representar em forma de vetor numérico uma determinada palavra pode ser visto na Figura 3. Considere as seguintes palavras em um vocabulário: Humano, Cachorro e Cavalo. Neste exemplo podemos utilizar como dimensões do vetor as seguintes informações: Número de pernas/patas, Peso usual, Expectativa de vida (dada em anos), Frequência do Termo no contexto geral, do inglês *Term Frequency* (TF) e Frequência inversa do documento, do inglês *Inverse Document Frequency* (IDF). A frequência do termo TF no contexto é dado pela razão entre o número de vezes que determinada palavra aparece em um texto, pelo o número total de palavras do referido texto. Logo, se a palavra “Humano” aparece 50 vezes em um texto com 1000 palavras, a sua frequência é de 0,05 (TF = 0,05). Quanto a frequência inversa do documento (IDF), é obtida a partir do logaritmo do número de documentos em um corpus dividido pelo número de documentos nos quais essa palavra aparece. Essa frequência pode ser expressa pela Equação (2.1), conforme a interpretação estatística realizada por Jones (1972):

$$IDF_p = \log\left(\frac{D}{|d : p \in d|}\right) \quad (2.1)$$

Onde  $D$  é o número de documentos no *corpus* e  $|d : p \in d|$  é o número de documentos  $d$  que contém a palavra  $p$  em seu vocabulário.

Seguindo o exemplo, se o corpus possui 5000 documentos e a palavra “Humano” aparece em 500 documentos, o IDF para a palavra “Humano” será o  $\log(5000/500) = \log 10$ . Nesse exemplo temos um IDF de 1. Outra informação que usualmente é apresentada, quando se trata de representação numérica de palavras, é a pontuação, ou peso, *Term Frequency - Inverse Document Frequency* (TF-IDF), que é basicamente o produto entre o TF e o IDF de uma determinada palavra em um determinado corpus. No exemplo dado, o TF-IDF de “Humano” seria de 0,05 multiplicado por 1, sendo assim teríamos um TF-IDF de 0,05.

Figura 3 – Exemplo de representação vetorial de uma palavra

|                        | Humano | Cachorro | Cavalo |
|------------------------|--------|----------|--------|
| Número de pernas/patas | 2      | 4        | 4      |
| Peso usual (em Kg)     | 80     | 20       | 450    |
| Expect. de vida (anos) | 75     | 15       | 25     |
| Term Frequency         | 0,05   | 0,01     | 0,03   |
| Inv. docum. Frequency  | 4      | 3        | 3      |

Fonte: autor (2023)

Os *embeddings* de palavras são úteis em muitas aplicações de PLN, como análise de sentimentos, tradução automática, geração de texto e sistemas de recomendação. Eles são especialmente úteis em tarefas que requerem compreensão semântica e sintática. Incitti, Urli e Snidaro (2023) afirmam que *embeddings* de palavras continua sendo uma ferramenta amplamente utilizada e uma das mais revolucionárias da área de PLN, no qual o principal objetivo é capturar o significado das palavras através de vetores numéricos. Desde a apresentação dos modelos *Word2Vec* de Mikolov et al. (2013) e, posteriormente, o *GloVe* de Pennington, Socher e Manning (2014), várias outras técnicas e algoritmos para gerar *embeddings* de palavras foram lançados, demonstrando o quão efetivo é o uso dessa técnica, especialmente com o recente crescimento dos modelos de linguagem e suas implementações.

## 2.2 Aprendizado de Máquina

O aprendizado de máquina, também conhecido como *machine learning* em inglês, é um subcampo da IA que se concentra no desenvolvimento de algoritmos e modelos estatísticos que permitem que sistemas computacionais aprendam e melhorem seu desempenho em uma tarefa específica com base em dados. O aprendizado de máquina se baseia na ideia de que os sistemas podem aprender com experiências passadas e, assim, aprimorar sua capacidade de realizar tarefas futuras sem serem explicitamente programados para tal. Zhu et al. (2022) explica que no aprendizado de máquina algoritmos são usados para analisar dados e tentar explorar padrões potenciais nos dados para prever novas informações, sendo amplamente utilizado em muitos campos devido à sua alta precisão, personalização flexível e extensibilidade conveniente.

Quatro conceitos são essenciais para o entendimento do aprendizado de máquina: dados, modelos de aprendizado de máquina, treinamento e avaliação. Os dados são a base fundamental do aprendizado de máquina. Eles podem ser estruturados, como tabelas ou bancos de dados, ou não estruturados, como texto, imagens ou áudio. Os dados são usados para treinar modelos, que aprenderão a reconhecer padrões e informações relevantes nos dados. Basicamente os dados são divididos em duas categorias: dados de entrada e dados de saída. Os dados de entrada podem ser entendidos como as informações do eixo X e os dados de saídas são os resultados apresentados no eixo Y. Os algoritmos aprendem a associar as entradas com suas respectivas saídas, dado um conjunto de treinamento de exemplos de entradas X e saídas Y (GOODFELLOW; BENGIO; COURVILLE, 2016).

Os modelos de aprendizado de máquina são algoritmos matemáticos que são treinados com base nos dados para realizar tarefas específicas. Existem vários tipos de modelos de aprendizado de máquina, incluindo regressão, classificação, *clustering* e redes neurais. Cada tipo de modelo é adequado para diferentes tipos de tarefas e dados. Sharifani e Amini (2023) explicam que os modelos de aprendizado de máquina normalmente envolvem o uso de algoritmos que utilizam de métodos estatísticos para aprender com os dados, com o objetivo de melhorar a precisão e eficiência de suas previsões ou decisões.

O treinamento de um modelo de aprendizado de máquina envolve alimentá-lo com um conjunto de dados de treinamento, que consiste em exemplos de entrada, ou seja, informações ou características importantes do eixo X, e a saída desejada, o resultado representado no eixo Y. O modelo ajusta seus parâmetros internos para aprender os padrões nos dados, de modo que possa

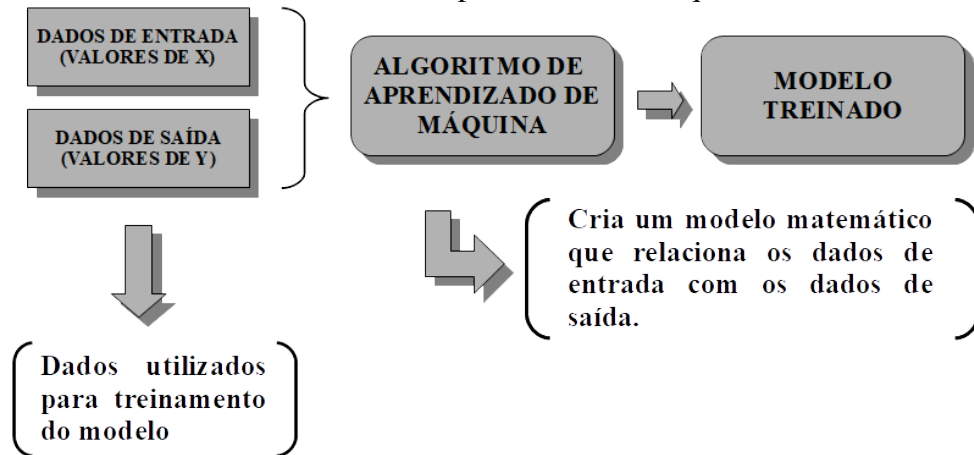
fazer previsões ou tomar decisões corretas em novos dados não vistos. É importante ressaltar que a qualidade e diversidade dos dados utilizados na fase de treinamento impactam muito na precisão e generalização dos modelos treinados (SHARIFANI; AMINI, 2023). Isso significa que se um modelo será treinado para classificar uma determinada situação, é essencial a existência de dados suficientes na etapa de treinamento para todos os possíveis tipos de classificação.

Após o treinamento, é necessário avaliar o desempenho do modelo. Isso ocorre durante a etapa de avaliação, que é realizada usando um conjunto de dados de teste, isto é, dados separados dos dados de treinamento. Em muitas aplicações práticas é comum a utilização da proporção de 70% à 80% dos dados existentes para treinamento e 20% à 30% para teste. Métricas de avaliação, como precisão, *recall* e *F1-score* são usadas para medir o quão bem o modelo está se saindo na tarefa. Nesta etapa o modelo é alimentado somente com os dados de entrada, como o modelo em questão já está treinado, será capaz de realizar a classificação e/ou predição. Os valores obtidos pelo modelo, a partir dos dados de teste, serão comparados com os valores reais.

Dependendo do tipo de aplicação, mesmo após o treinamento, o desempenho do modelo pode continuar a ser aprimorado. Isso é possível através do ajuste de hiperparâmetros, que são configurações específicas do modelo que não são aprendidas durante o treinamento. Alguns dos exemplos mais comuns de ajuste de hiperparâmetros são: alteração da proporção dos dados de treinamento e teste, escolha da função de ativação em redes neurais, escolha da função custo e definição da quantidade de camadas em redes neurais.

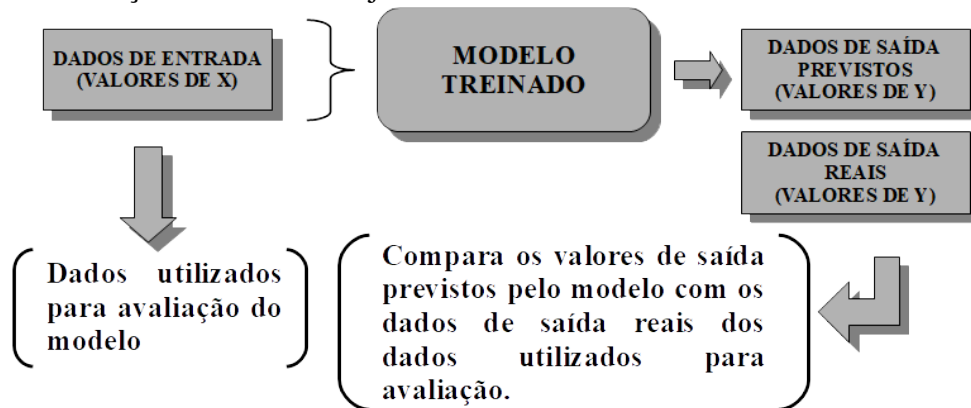
Um dos objetivos fundamentais do aprendizado de máquinas é a capacidade de generalizar. Barbiero, Squillero e Tonda (2020) apontam que é importante verificar até que ponto o modelo treinado é capaz de generalizar, isto é, verificar se o modelo é capaz de fornecer previsões significativas para novos dados e não apenas para os dados utilizados no treinamento. Isso significa que o modelo deve ser capaz de fazer previsões precisas em dados não vistos, extrapolando o conhecimento adquirido durante o treinamento.

Figura 4 – Treinamento de um modelo de aprendizado de máquina



Fonte: autor (2023)

Figura 5 – Avaliação de um modelo já treinado



Fonte: autor (2023)

### 2.2.1 Redes Neurais e Aprendizado Profundo

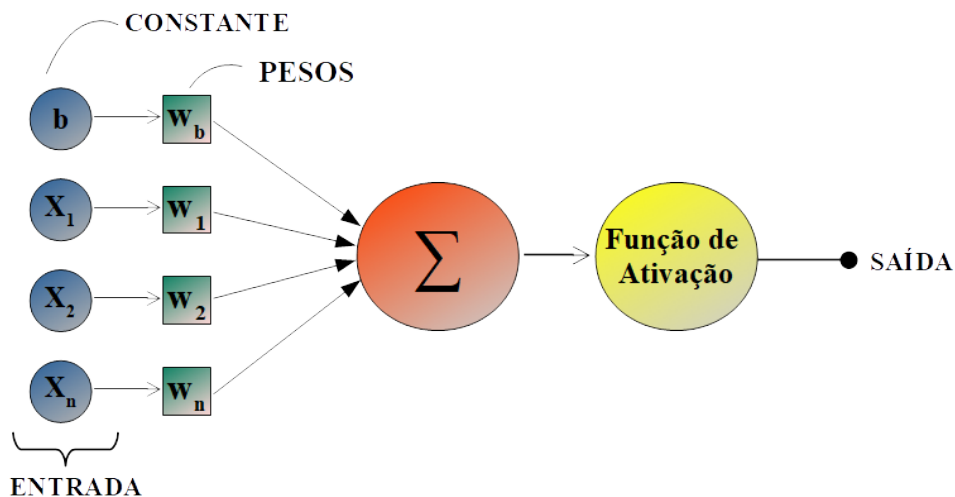
As redes neurais são um modelo matemático inspirado no funcionamento do cérebro humano. Elas consistem em uma coleção de unidades de processamento interconectadas, chamadas de neurônios artificiais ou *perceptrons*, que são unidades de processamento apresentadas por Frank Rosenblatt em 1958. Cada neurônio artificial recebe entradas ponderadas, aplica uma função de ativação não linear às entradas ponderadas e produz uma saída. A conexão entre neurônios é definida por pesos ajustáveis, que são otimizados durante o treinamento. A rede neural apresentada por Rosenblatt é definida como um classificador linear e pode ser expressa pela Equação 2.2:

$$\begin{aligned} f(x) &= 1, \text{ se } wx+b \geq 0 \\ f(x) &= 0, \text{ caso contrário} \end{aligned} \quad (2.2)$$

Onde  $x$  é o valor de uma entrada,  $w$  é um vetor de peso e  $b$  é uma constante que não depende do valor da entrada  $x$ .

Kufel et al. (2023) explica que as redes neurais compreendem múltiplos perceptrons ou neurônios artificiais, responsáveis por processar e transmitir informações. Os dados da entrada são processados através de camadas sucessivas de *perceptrons* até chegar à saída, cada camada contém vários *perceptrons*. A inspiração biológica para as redes neurais reside na estrutura do sistema nervoso, onde os neurônios estão conectados por sinapses. Assim como os neurônios biológicos transmitem sinais elétricos e químicos, os neurônios artificiais nas redes neurais transmitem informações por meio de operações matemáticas.

Figura 6 – Representação de uma rede neural simples



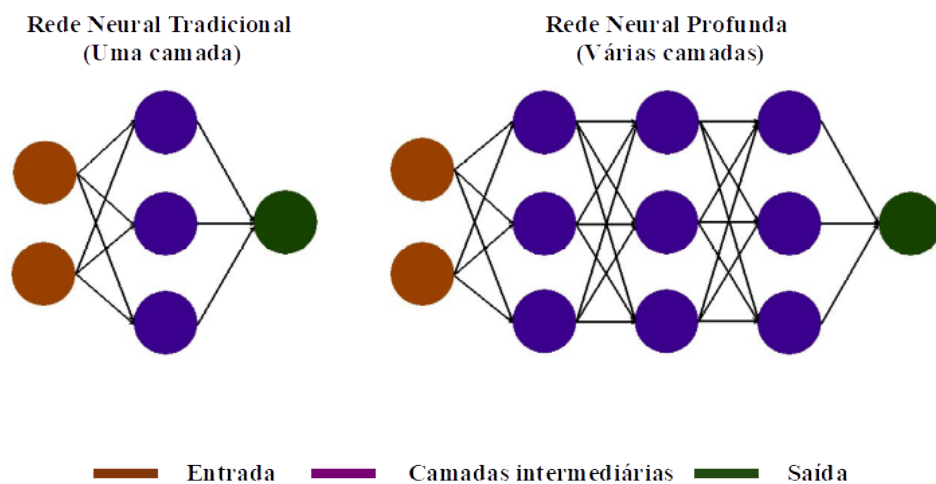
Fonte: Adaptado de Sharkawy (2020)

O aprendizado profundo, também conhecido como *deep learning*, é uma subárea das redes neurais que se concentra em redes com múltiplas camadas. Usualmente, a definição da quantidade de camadas que uma rede neural possui está relacionada com a complexidade da tarefa para a qual o modelo de aprendizado de máquina foi treinado para resolver. O termo “profundo” refere-se ao fato de que essas redes têm muitas camadas intermediárias entre a entrada e a saída. Pattanayak (2023) aponta que o aprendizado profundo supera os métodos tradicionais de aprendizado de máquina, especialmente em áreas como visão computacional e PLN. Isso ocorre pois à medida que as camadas da rede neural de aprendizado profundo aumentam, recursos cada vez mais complexos podem ser aprendidos.

As redes neurais profundas são capazes de aprender representações hierárquicas de dados, o que as torna extremamente eficazes em tarefas como reconhecimento de padrões, PLN e visão

computacional. O treinamento de redes profundas geralmente envolve o uso de grandes conjuntos de dados e algoritmos de otimização, como o gradiente descendente, para ajustar os pesos das conexões entre os neurônios artificiais que constituem a rede. Enquanto redes neurais tradicionais possuem uma ou duas camadas intermediárias, o aprendizado profundo envolve redes com muitas camadas intermediárias, permitindo a extração de representações complexas dos dados. Em redes neurais profundas é possível combinar camadas mais baixas para formar características abstratas em camadas mais altas, consequentemente, devido à complexidade adicional, esse tipo de rede geralmente requer mais dados e tempo de treinamento em comparação com redes neurais tradicionais.

Figura 7 – Comparação entre uma rede neural tradicional e uma rede neural profunda



Fonte: Adaptado de Kufel et al. (2023)

### 2.2.2 *Aprendizado Supervisionado*

O aprendizado supervisionado é uma abordagem na qual o modelo de aprendizado de máquina é treinado usando um conjunto de dados que contém exemplos rotulados. Cada exemplo de treinamento consiste em um par de entrada e saída desejada, onde a saída desejada, também chamada de rótulo ou etiqueta, representa a resposta correta ou o valor alvo associado a uma determinada entrada. No aprendizado supervisionado, o conjunto de dados de treinamento contém exemplos em que as saídas desejadas estão previamente conhecidas e são fornecidas como parte do treinamento. Goodfellow, Bengio e Courville (2016) destacam que, em vários casos, a saída ou alvo (Y) pode ser difícil de coletar automaticamente, precisando ser fornecido por um humano, que nessa situação será o “supervisor”, mas vale ressaltar que mesmo quando as saídas (Y) foram coletados automaticamente ainda se trata de aprendizado supervisionado.

O objetivo principal é aprender a mapear corretamente as entradas para as saídas com base nos exemplos de treinamento, de modo que o modelo possa fazer previsões precisas em novos dados. O aprendizado supervisionado é usado principalmente para tarefas de classificação e regressão. Na classificação, o modelo atribui uma classe a uma entrada, enquanto na regressão, ele prevê um valor numérico. Riccio et al. (2020) explica que em problemas de classificação o modelo de aprendizado de máquina tem que atribuir uma classe a uma determinada entrada, enquanto que na regressão o modelo deve realizar uma previsão que assume a forma de um ou mais valores contínuos.

Um exemplo de sistema que utiliza aprendizado supervisionado é um sistema de classificação de e-mails como “spam” ou “não spam” em um cliente de e-mail. Nesse cenário, o sistema é treinado com um grande conjunto de e-mails previamente classificados, onde cada e-mail é rotulado como “spam” ou “não spam”. O algoritmo de aprendizado supervisionado analisa as características dos e-mails, como palavras-chave, padrões de formatação e outras informações relevantes, e aprende a mapear essas características para as categorias “spam” ou “não spam” com base nos exemplos rotulados. Após o treinamento, o sistema pode automaticamente classificar novos e-mails recebidos pelo usuário.

### **2.2.3 *Aprendizado Não-supervisionado***

O aprendizado não supervisionado é uma abordagem em que o modelo é treinado em um conjunto de dados que não possui rótulos. Em vez disso, o modelo tenta aprender a estrutura ou padrões subjacentes nos dados sem orientação externa. Algoritmos não supervisionados são aqueles que conhecem apenas as entradas, logo não possuem conhecimento de uma saída rotulada, estando usualmente associado com a tentativa de extrair informações de uma distribuição que não requer trabalho humano para anotar exemplos (GOODFELLOW; BENGIO; COURVILLE, 2016).

O objetivo é encontrar estruturas, agrupamentos, padrões ou representações úteis nos dados, muitas vezes com o intuito de simplificar a análise ou a visualização dos dados. O aprendizado não supervisionado é frequentemente usado para tarefas de clusterização (agrupamento), redução de dimensionalidade e análise de componentes principais. Um algoritmo muito utilizado na área de IA para redução de dimensionalidade é o *Principal Component Analysis* (PCA). Salem e Hussein (2019) definem PCA como uma técnica de aprendizagem não supervisionada que

utiliza princípios matemáticos sofisticados para reduzir a dimensionalidade de grandes conjuntos de dados.

Um exemplo de sistema que utiliza aprendizado não supervisionado é um mecanismo de recomendação de conteúdo em uma plataforma de *streaming* de vídeo, como o *Netflix*. Nesse caso, o sistema processa dados de visualização e preferências de milhões de usuários, mas não depende de rótulos explícitos para categorizar os filmes ou séries. Em vez disso, ele aplica algoritmos de agrupamento (clusterização) para identificar padrões de visualização e preferências semelhantes entre os usuários. Com base nesses grupos, o sistema pode então sugerir filmes ou séries que são populares entre usuários com gostos semelhantes. Esse processo de agrupamento é realizado de forma não supervisionada, pois o sistema não recebe categorias predefinidas, mas descobre essas categorias automaticamente a partir dos dados de comportamento do usuário.

Figura 8 – Exemplos de utilização de modelos de aprendizado supervisionado e não supervisionado



Fonte: autor (2023)

#### 2.2.4 *Aprendizado por Transferência*

Aprendizado por Transferência, ou *Transfer Learning* em inglês, é um conceito fundamental no campo do aprendizado de máquina, que se refere à técnica de usar o conhecimento adquirido em um cenário, ou tarefa, para melhorar o desempenho em outro cenário, ou tarefa, relacionado, mas não necessariamente idêntico. O aprendizado por transferência é uma metodologia promissora que foca em transferir conhecimento através de domínios diferentes, com o intuito de solucionar o problema de coletar dados de treinamento suficientes, uma etapa que

costuma ser financeiramente dispendiosa, demorada e até mesmo irrealista em muitos cenários (ZHUANG et al., 2020).

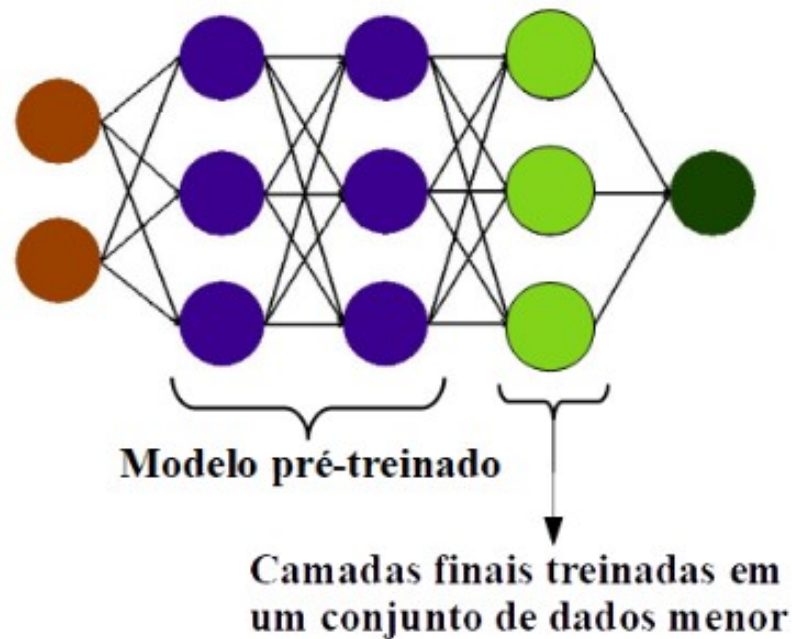
No aprendizado por transferência, um modelo pré-treinado é inicialmente treinado em um domínio específico, resultando em representações internas genéricas que capturam informações úteis sobre o domínio. Posteriormente, o modelo é adaptado para uma tarefa específica, substituindo suas camadas finais por novas camadas personalizadas treinadas com um conjunto de dados menor e específico. Quando esse processo ocorre em redes neurais profundas é chamado de Aprendizado Profundo por Transferência, do inglês *Deep Transfer Learning* (DTL).

Zhuang et al. (2020) ressalta que, no processo de aprendizagem por transferência, o conhecimento contido em um modelo pré-treinado para uma determinada atividade (domínio de origem) é transferido para a execução de uma nova atividade que, apesar de diferente, é relacionada à atividade do domínio de origem. Isso permite um melhor desempenho do modelo, mesmo que este não possua um grande volume de dados sobre a nova atividade.

Iman, Arabnia e Rasheed (2023) descrevem que o aprendizado profundo por transferência consiste em usar o conhecimento obtido de uma tarefa qualquer e um determinado conjunto de dados, mesmo um não fortemente relacionado à tarefa ou conjunto de dados de origem, para reduzir custos de aprendizagem de um novo modelo. Logo, a utilização do aprendizado por transferência reduz a necessidade de grandes quantidades de dados, uma vez que permite que modelos se beneficiem do conhecimento prévio, tornando-os capazes de alcançar um desempenho razoável com quantidades menores de dados de treinamento.

Os modelos pré-treinados já aprenderam representações úteis de características gerais, o que pode melhorar sua capacidade de generalização para tarefas específicas. Além disso, treinar um modelo a partir do zero pode ser computacionalmente caro e demorado. O aprendizado por transferência economiza tempo e recursos, pois começa com um modelo pré-treinado. Diante disso, o aprendizado por transferência é amplamente utilizado em várias áreas, incluindo visão computacional, PLN, reconhecimento de fala e classificação de texto. No contexto de PLN, modelos como o *Bidirectional Encoder Representations from Transformer* (BERT) e GPT-3 são exemplos populares da utilização de aprendizado por transferência.

Figura 9 – Camadas finais de um modelo pré-treinado substituídas após um novo treinamento em um conjunto de dados menor



Fonte: autor (2023)

### 2.3 Grandes Modelos de Linguagem

Os Grandes Modelos de Linguagem, do inglês *Large Language Models* (LLMs), são uma classe de modelos de aprendizado de máquina que se destacam na compreensão e geração de texto em linguagem natural. Mitchell e Krakauer (2023) definem LLMs como redes neurais profundas com bilhões a trilhões de parâmetros (pesos). Esses parâmetros são “pré-treinados” em um volume enorme de documentos escritos em linguagem natural, incluindo conteúdos da internet e coleções de livros online, totalizando *terabytes* de dados de treinamento. Os LLMs possuem uma ampla variedade de aplicações, como tradução, geração de respostas e sumarização de informações, por isso tem sido uma revolução no campo de PLN, tendo seu impacto debatido em diversos trabalhos, tais como Wardat et al. (2023), Kasneci et al. (2023) e Thirunavukarasu et al. (2023).

Os LLMs, como o *Generative Pre-trained Transformer* (GPT), são baseados na arquitetura de rede neural chamada *Transformer*. A arquitetura *Transformer* é uma arquitetura de rede mais simples, pois é baseada unicamente em mecanismos de Atenção, dispensando recorrências e convoluções (VASWANI et al., 2017). Além de mais simples, outra vantagem apontada por Vaswani et al. (2017) sobre o *Transformer* é que esta arquitetura é mais paralelizável e, por isso, requer significativamente menos tempo para treinar.

### 2.3.1 Mecanismo de Atenção

A Atenção, ou *Attention* em inglês, é o bloco de construção fundamental da arquitetura *Transformer* e é o que permite que o modelo entenda as relações entre as palavras em uma sequência. A Atenção é definida matematicamente da seguinte maneira: para cada palavra em uma sequência de entrada, a Atenção calcula um peso (pontuação de Atenção) que indica a importância das outras palavras na sequência em relação a essa palavra. Isso é feito calculando o produto escalar entre os vetores de consulta ( $Q$ ), chave ( $K$ ) e valor ( $V$ ) para todas as palavras na sequência. A Equação (2.3) apresentada por Vaswani et al. (2017) demonstra como é calculada a Atenção:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V \quad (2.3)$$

Onde  $Q$  é o vetor de consulta para a palavra atual,  $K$  é o vetor de chave para todas as palavras na sequência,  $V$  é o vetor de valor para todas as palavras na sequência,  $softmax$  é uma função de normalização que produz pesos de atenção e  $\sqrt{d}$  é um fator de escala para normalizar a pontuação de atenção.

Slechten (2023) explica que, ao contrário dos modelos *seq2seq*, como Redes Neurais Recorrentes, do inglês *Recurrent Neural Network* (RNN) e redes que utilizam Memória de Longo Prazo, do inglês *Long Short Term Memory* (LSTM), que processam palavras sequencialmente, a arquitetura *Transformer* permite o processamento paralelo de todas as palavras da sequência, resultando em redução do tempo de treinamento e na capacidade de calcular um contexto mais completo. Nos modelos *seq2seq*, a primeira palavra é inserida no codificador, e um novo estado é calculado com um estado inicial aleatório. O processo se repete para cada palavra subsequente, levando a um vetor de contexto de comprimento fixo. No entanto, as informações contextuais sobre estados anteriores podem ser perdidas, e o vetor de contexto pode não ser grande o suficiente. Em contraste, o *Transformer* consome toda a sequência e processa as palavras em paralelo, garantindo que o contexto seja calculado a partir de todas as palavras, resultando em um contexto mais completo e eficiência no treinamento.

Além do mecanismo de Atenção, outra inovação da arquitetura *Transformer* apresentada por Vaswani et al. (2017) é a Atenção Multicabeça, do inglês *Multi-Head Attention*, que permite que o modelo capture diferentes tipos de relações entre palavras. Isso é alcançado por meio da aplicação da Atenção em várias projeções dos vetores de consulta, chave e valor, e, em seguida,

combinando as saídas de diferentes cabeças. Conforme visto na Equação (2.4) demonstrada por Vaswani et al. (2017) para Atenção Multicabeça:

$$\text{Multi-Head Attention}(Q, K, V) = \text{concatenamento}(\text{Attention}(Q_i, K_i, V_i)) \quad (2.4)$$

Para  $i$  variando de 1 até  $h$ , onde  $h$  é o número de cabeças de Atenção.

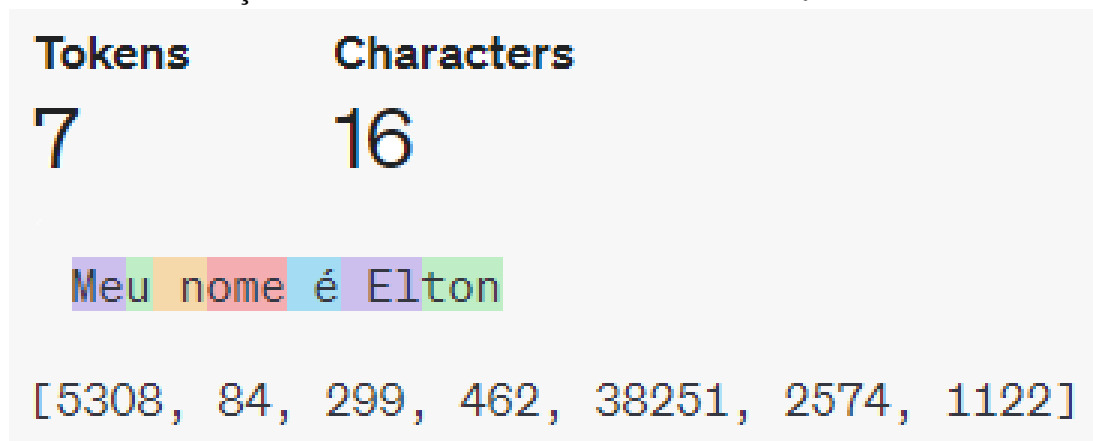
Liu, Liu e Han (2021) afirmam que a Atenção Multicabeça é uma das principais razões por trás do sucesso da arquitetura *Transformer*. Devido a sua capacidade de processar informações em paralelo e lidar com dependências de longo alcance, a Atenção Multicabeça provou ser altamente eficaz na modelagem de sequências. Atualmente, a arquitetura *Transformer* é a base de muitos LLMs do estado da arte, uma vez que essa arquitetura permite que os modelos capturem dependências de longo alcance, compreendam o contexto e gerem texto coerente e relevante em várias tarefas de PLN.

### 2.3.2 Token, Prompt e Parâmetros

#### 2.3.2.1 Token

*Tokens* são as unidades de texto em que um documento ou sentença é dividido. Ali et al. (2023) define que um *token* pode representar uma palavra, uma subpalavra ou mesmo um único caractere, onde cada *token* é representado no modelo por um vetor de *embedding*. Usualmente, ao se falar de textos, a unidade básica que utilizamos é a palavra ou o caractere, entretanto os LLMs utilizam como unidade básica o *token*. Por exemplo, na frase “Meu nome é Elton”, há 7 *tokens* e 16 caracteres, incluindo os espaços entre as palavras.

Figura 10 – Visualização dos *tokens* através da ferramenta *Tokenizer*



Fonte: autor (2023)

A tokenização é o processo de dividir um texto em *tokens*. Essa etapa é fundamental para o processamento de texto, pois transforma o texto para um formato em que os LLMs estão configurados para trabalhar. Ali et al. (2023) ressalta que, no que diz respeito à abordagem, o processo de tokenização pode ser classificado em três tipos: tokenização por palavra, por subpalavra e por caracteres. A tokenização por subpalavra é utilizada pelos LLMs do estado da arte, ao passo que a tokenização por caracteres é computacionalmente mais cara para a arquitetura *Transformer*, sendo assim, inviável para os LLMs. O número total de *tokens* únicos que um modelo de linguagem é capaz de entender é definido como vocabulário. Quanto maior o vocabulário, mais rico e diversificado é o conhecimento do modelo.

### 2.3.2.2 *Prompt*

Topsakal e Akinci (2023) descrevem um *prompt* como uma entrada enviada a um LLM, ou seja, um conjunto de alguns exemplos para ajudar o modelo de linguagem a gerar uma resposta melhor, ou instruções para o LLM sobre como processar a entrada que ele recebe. Os *prompts* podem ser gerados dinamicamente quando usados em uma aplicação que inclui a entrada do usuário (pergunta) como parte das instruções e/ou solicitação feita ao LLM. Lester, Al-Rfou e Constant (2021) explicam que os *prompts* normalmente são compostos de uma descrição de tarefa que pode ou não incluir vários exemplos que demonstram como determinada tarefa deve ser executada.

Figura 11 – Exemplo de *prompt* enviado a um LLM

```
template = """
Siga exatamente os passos a seguir:
1. Leia as informações recebidas neste prompt
2. Responda a Pergunta utilizando somente as informações existente neste prompt
3. Seja o mais detalhado e educado possível
"""
```

Fonte: autor (2023)

### 2.3.2.3 *Parâmetros*

Em aprendizado de máquina, o termo parâmetro refere-se aos pesos e configurações da rede neural que compõem o modelo. Uma rede neural é composta por camadas de neurônios interconectados, com pesos associados a cada conexão entre neurônios, que determinam a força da conexão. Durante o treinamento de um modelo de linguagem, esses pesos são ajustados

para que o modelo aprenda a relacionar palavras, entender contexto e gerar textos coerentes, entre outras tarefas de linguagem natural. Esses parâmetros resultam do treinamento em grandes volumes de dados textuais, representando o conhecimento e as habilidades do modelo em processar linguagem natural.

Topsakal e Akinci (2023) ressaltam que um modelo de linguagem é categorizado como um LLM, ou seja, um grande modelo de linguagem, a partir do número considerável de parâmetros em que este modelo é treinado. Os LLMs possuem grande números de parâmetros por serem treinados com gigantescos volumes de textos. Um exemplo de evolução na quantidade de parâmetros é o modelo apresentado por Radford e Narasimhan (2018) GPT, Radford et al. (2019) *Generative Pre-trained Transformer 2* (GPT-2) e Brown et al. (2020) GPT-3, onde o primeiro possuía 117 milhões de parâmetros, o segundo 1,5 bilhão de parâmetros e o terceiro 175 bilhões de parâmetros. Bender et al. (2021) aponta que uma das maiores tendências na área de PLN tem sido o tamanho crescente dos modelos de linguagem, caracterizado pelo aumento do número de parâmetros, que é um reflexo da quantidade de dados em que esses modelos são treinados. Modelos maiores, com mais parâmetros, geralmente têm uma capacidade maior de aprender e representar informações complexas. No entanto, também exigem mais recursos computacionais e memória para treinamento e uso em aplicações práticas.

### 2.3.3 *Zero-shot, Few-shot e Fine Tuning*

No contexto de aprendizado de máquinas, especialmente quando falamos de redes neurais como modelos de linguagem, os termos: *zero-shot*, *few-shot* e *fine tuning* estão relacionados às técnicas de treinamento e a adaptação de modelos à sua forma de utilização.

*Zero-Shot* refere-se a uma abordagem em que um modelo é capaz de realizar uma tarefa para a qual não foi explicitamente treinado. Ou seja, o modelo é capaz de realizar a tarefa ao usar informações semelhantes aprendidas durante o treinamento original, mesmo que a tarefa específica seja nova e não vista antes. Por exemplo, um modelo de linguagem treinado em tradução de idiomas pode realizar uma nova tarefa de tradução (*zero-shot*) entre dois idiomas não vistos durante o treinamento, com base na semelhança estrutural dos idiomas aprendidos. Brown et al. (2020) alerta que o termo *zero-shot* pode ser potencialmente ambíguo. Esse termo é utilizado no sentido de que nenhuma atualização de gradiente foi realizada para a execução da nova atividade, mas geralmente envolve fornecer algumas demonstrações de solução para o

modelo, então não é realmente aprender com zero exemplos. Bender et al. (2021) destaca que a otimização de *prompts* tende a oferecer um melhor desempenho em situações de *zero-shot*.

*Few-Shot* refere-se a uma abordagem em que o modelo recebe algumas demonstrações de como executar a tarefa solicitada. Brown et al. (2020) define que *few-shot* é o contexto de utilização onde é permitido disponibilizar quantas demonstrações de solução para a tarefa solicitada couberem na janela de contexto do modelo, ou seja, no *prompt*. Ele destaca que a definição dos termos *zero-shot* e *few-shot* depende de quantas demonstrações são fornecidas, mas normalmente no *few-shot* temos algo entre 10 a 100 demonstrações.

*Fine tuning* envolve a adaptação de um modelo pré-treinado a uma tarefa específica. Normalmente, utiliza-se um modelo, que foi treinado em uma grande quantidade de dados em uma tarefa geral, para realizar um treinamento adicional em um conjunto de dados menor e mais direcionado a nova atividade, conforme explicado no aprendizado por transferência. Brown et al. (2020) define que o *fine tuning* envolve a atualização dos pesos de um modelo pré-treinado por meio do treinamento em um conjunto de dados supervisionado específico para a tarefa desejada, destacando que essa abordagem tem sido a mais comum nos últimos anos.

## 2.4 Application Programming Interface

API, ou Interface de Programação de Aplicações, é um conjunto de regras e protocolos que permitem que diferentes sistemas de um *software* se comuniquem e interajam entre si. Ela define as formas pelas quais as diferentes partes de um *software* devem interagir e quais operações podem ser realizadas, fornecendo uma maneira estruturada e padronizada para que os desenvolvedores possam acessar funcionalidades ou dados de um sistema ou serviço.

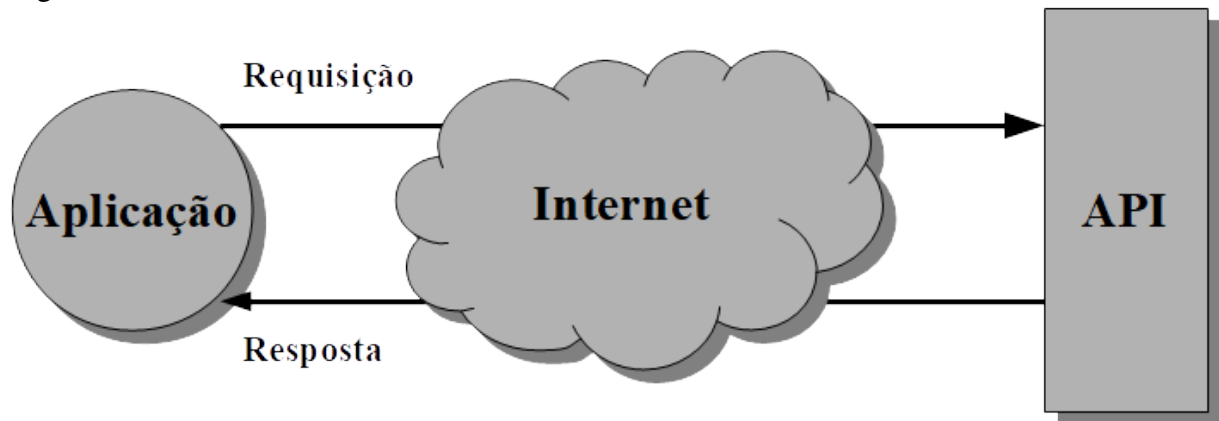
Uma API permite que diferentes sistemas, tais como: aplicativos, serviços *web* e bibliotecas, se comuniquem uns com os outros de forma eficaz, permitindo que compartilhem dados e funcionalidades. Isso é essencial para a integração de sistemas e a criação de aplicativos que utilizam serviços externos. Uma API abstrai as funcionalidades inerentes de um sistema, expondo apenas os recursos e operações relevantes para os desenvolvedores. Isso permite que os desenvolvedores interajam com um sistema sem precisar entender completamente sua implementação interna. APIs desempenham um papel de extrema importância no desenvolvimento de *softwares*, pois simplificam de maneira geral a execução do projeto (NAWAZ et al., 2022).

Uma API bem projetada é acompanhada por uma documentação detalhada que descreve como usá-la, incluindo informações sobre os *endpoints* (pontos de acesso), parâmetros, formatos de dados e exemplos de uso. A documentação é fundamental para que os desenvolvedores entendam como interagir com a API. Dominte (2023) explica que as APIs são desenvolvidas para serem consumidas por diferentes clientes e sistemas, sendo assim, para garantir que a interação com a API ocorra de forma simples, é essencial dedicar esforços no processo de documentação da API.

As APIs geralmente usam protocolos de comunicação específicos para permitir a transferência de dados entre sistemas. Alguns dos protocolos comuns incluem *Hypertext Transfer Protocol* (HTTP)/ *Hypertext Transfer Protocol Secure* (HTTPS) para APIs da *web*, *JavaScript Object Notation Remote Procedure Call* (JSON-RPC), *Google Remote Procedure Call* (gRPC) e *Simple Object Access Protocol* (SOAP). Existem três tipos principais de APIs, que são: *web*, biblioteca e sistema operacional. As APIs da *web* são acessadas por meio da internet e geralmente são usadas para interagir com serviços online, como redes sociais, sistemas de pagamento, previsões meteorológicas, etc. APIs de biblioteca são conjuntos de funções e procedimentos que podem ser incorporados em um *software* para fornecer funcionalidades específicas. Por exemplo, uma biblioteca gráfica que fornece funções para desenhar gráficos. APIs de sistema operacional permitem que aplicativos interajam com o sistema operacional em que estão sendo executados, realizando operações como acesso a arquivos e gerenciamento de recursos.

As APIs geralmente implementam mecanismos de autenticação e autorização para garantir que apenas os usuários autorizados possam acessar determinados recursos. Isso é fundamental para a segurança e proteção dos dados. Em APIs da *web*, o protocolo HTTP é frequentemente usado como protocolo de comunicação. Os principais métodos HTTP, como GET, POST, PUT e DELETE, são usados para solicitar e manipular recursos por meio da API. Os dados são frequentemente transmitidos em formatos como *JavaScript Object Notation* (JSON) ou *Extensible Markup Language* (XML).

Figura 12 – Funcionamento de uma API *web*



Fonte: autor (2023)

## 2.5 Métodos de Avaliação de Agentes Conversacionais

A avaliação de agentes conversacionais, como *chatbots* e assistentes virtuais, desempenha um papel fundamental no desenvolvimento e aprimoramento desses sistemas. Existem várias abordagens e métodos de avaliação disponíveis, que podem ser usados de forma complementar para obter uma compreensão completa do desempenho do agente conversacional. Aspectos como usabilidade, desempenho e naturalidade podem ser avaliados, dependendo da situação. Radziwill e Benton (2017) destacam que no que se refere ao conceito de usabilidade, três atributos de um agente conversacional são verificados: eficácia, eficiência e satisfação. Maroengsit et al. (2019) ressalta que a avaliação de agentes conversacionais devem considerar principalmente duas categorias: conteúdo, onde a avaliação foca no contexto da resposta gerada, e satisfação do usuário.

Na avaliação de usabilidade, tipicamente são conduzidos testes com usuários reais para avaliar a facilidade de uso do agente, onde os usuários fornecem *feedback* sobre a clareza das respostas, eficiência na resolução de problemas e navegabilidade. Além disso, na avaliação de desempenho, características como a precisão das respostas são verificadas, usando métricas como a precisão e o tempo de resposta. No que diz respeito à avaliação de naturalidade, aspectos como a coerência, coesão e naturalidade da linguagem são examinados, com um foco na adequação das respostas às perguntas dos usuários e na preferência por respostas que soem mais humanas.

Guo et al. (2023) realizou uma pesquisa comparativa entre as respostas de um agente conversacional, mais especificamente o ChatGPT, e as respostas de humanos. Nos resultados é

possível constatar que o ChatGPT demonstrou uma tendência a produzir respostas organizadas e logicamente estruturadas, muitas vezes detalhadas em sua extensão, enquanto os humanos frequentemente divergiram para outros tópicos em suas respostas. Além disso, o ChatGPT mostrou uma menor propensão a expressar preconceitos e informações prejudiciais, recusando-se a responder a perguntas sem conhecimento prévio, enquanto os humanos frequentemente introduziram expressões subjetivas em suas respostas. A formalidade foi outra distinção notável, com as respostas do ChatGPT sendo predominantemente formais, enquanto as dos humanos eram mais coloquiais. Essas diferenças também se refletiram no fator densidade, onde o vocabulário diversificado dos humanos se destacou em comparação ao ChatGPT.

É importante destacar que a escolha dos métodos de avaliação deve ser orientada pelos objetivos específicos do agente conversacional e pelas necessidades da demanda em questão. Uma combinação de métodos quantitativos e qualitativos é muitas vezes a abordagem mais eficaz para obter uma visão completa do desempenho do agente e identificar áreas de melhoria. Além disso, a avaliação deve ser um processo contínuo, pois os agentes conversacionais podem ser aprimorados ao longo do tempo com base no *feedback* e nos dados coletados.

Considerando o crescimento e a tendência de utilização de LLMs para implementação de agentes conversacionais, é de suma importância entender como esses modelos de linguagem são avaliados. Diversos métodos de avaliação foram apresentados no decorrer dos últimos anos, *frameworks*, *datasets* com coleções de perguntas e atividades diversas relacionadas à PLN evoluíram, refinando o processo de avaliação de LLMs. Clark et al. (2018) desenvolveu um conjunto de perguntas, corpus de texto e linhas de base focados em testar projetos de IA envolvendo a atividade de geração de respostas para questões avançadas, o *AI2 Reasoning Challenge* (ARC). O conjunto de dados do ARC consiste em uma coleção de 7.787 questões sobre ciências.

Zellers et al. (2019) demonstra que a inferência do senso comum ainda se mostra difícil até mesmo para LLMs do estado da arte. Isso foi provado através do *HellaSwag*, um *dataset* de desafios envolvendo inferência do senso comum, com perguntas que são triviais para humanos, mas que se mostraram desafiadoras para os modelos de linguagem. Hendrycks et al. (2021) propôs um teste para medir a precisão multitarefa dos modelos de linguagem. O *Massive Multi-task Language Understanding* (MMLU) abrange 57 tarefas, incluindo matemática elementar, história dos Estados Unidos da América (EUA), ciência da computação, leis, entre outros tópicos.

Para obter alta precisão neste teste, os modelos devem possuir um amplo conhecimento geral e a capacidade de resolução de problemas.

A disseminação de informações incorretas, bem como a temática crescente e recorrente de *fake news* ao redor do mundo, despertou a preocupação da comunidade científica. Lin, Hilton e Evans (2022) apresentaram o *TruthfulQA*, um teste para medir se um modelo de linguagem é verdadeiro na geração de respostas. O teste consiste em 817 perguntas que abrangem 38 categorias, tais como: saúde, direito, finanças e política. As perguntas existentes no *TruthfulQA* foram elaboradas para que mesmo os humanos eventualmente respondessem de forma errada, devido a uma crença falsa ou equívocos comumente espalhados na internet. Para ter um bom desempenho, os modelos de linguagem testados devem evitar a geração de respostas falsas devido a imitação de textos humanos que possuem informações incorretas.

É importante observar que a avaliação de LLMs é um campo em constante evolução, e novas métricas e abordagens continuam sendo desenvolvidas para lidar com desafios específicos, como detecção de viés e ética. Além disso, combinar diferentes métodos de avaliação, incluindo métricas automatizadas e avaliação humana, é muitas vezes a abordagem mais eficaz para obter uma compreensão completa do desempenho do modelo.

## 2.6 LangChain

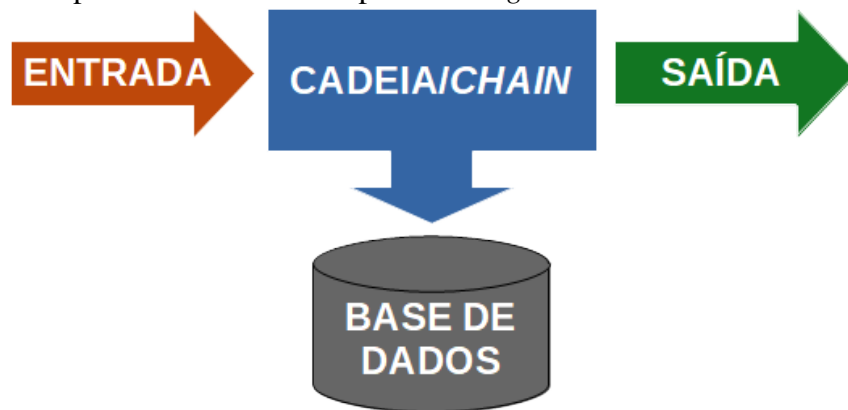
*LangChain* é um *framework open-source* versátil projetado para o desenvolvimento de aplicações que aproveitam o poder dos modelos de linguagem. É especialmente adequado para viabilizar que as aplicações recebam um reconhecimento de contexto. O *LangChain* conecta um LLM a diversas fontes de contexto (base de dados), com instruções imediatas, exemplos rápidos ou informações contextuais para fundamentar o processo de criação de respostas do LLM. Soygazi e Oguz (2023) explicam que o *LangChain* se concentra em aplicações que utilizam modelos de linguagem, tendo como conceito o desenvolvimento rápido de aplicações para vários domínios, aproveitando as habilidades dos LLMs.

Topsakal e Akinci (2023) apontam que o *LangChain* ganhou muita atenção da comunidade de IA, pois fornece soluções para o desenvolvimento de aplicações de IA que utilizam LLMs. O *LangChain* é constituído de diversos componentes principais, incluindo pacotes, tanto para a linguagem *Python* quanto para *JavaScript*, contendo interfaces e integrações. Os componentes são modulares, tornando simples de utilizar uma função específica do *framework*,

que também disponibiliza implementações prontas para uso de cadeias, agentes e templates, todos com o foco em agilizar o processo de desenvolvimento.

O objetivo é permitir que os desenvolvedores utilizem diversas fontes de dados para interagir com os LLMs. Para isso, o *LangChain* fornece componentes e cadeias, ou *chains* do inglês, que são a base mais importante do *LangChain* pois combinam uma série de informações que são enviadas ao LLM através do *prompt* (TOPSAKAL; AKINCI, 2023). Pesaru, Gill e Tangella (2023) ressaltam que o *LangChain* torna mais simples o processo de desenvolvimento de *chatbots* e aplicações escaláveis que utilizam IA e LLMs.

Figura 13 – Exemplo de uma cadeia simples no *LangChain*



Fonte: autor (2023)

### 3 METODOLOGIA

Este capítulo detalha a metodologia utilizada para o desenvolvimento e avaliação de desempenho da Eris, uma assistente virtual criada para a PROGEP do IFCE, com o objetivo de responder às dúvidas dos servidores. Inicialmente um *roadmap* simples foi criado para definir todas as etapas necessárias durante a execução do projeto, conforme segue:

1. **Definição do escopo de temas que a assistente virtual irá responder:** estabelecer claramente quais temas, informações e áreas de conhecimento a assistente virtual será capaz de abordar e responder de forma eficaz. Essa etapa envolve a identificação e delimitação dos assuntos que a assistente virtual será capaz de lidar, com o objetivo de proporcionar uma experiência coerente e útil aos servidores.
2. **Criação da base de conhecimento:** coletar, organizar e preparar todas as informações relevantes que servirão como o conjunto de conhecimento principal para a assistente virtual. A base de conhecimento é fundamental para que a assistente virtual possa responder com precisão às perguntas feitas pelos servidores.
3. **Seleção de tecnologias, bibliotecas, ferramentas e *frameworks* que serão utilizados:** escolher quais tecnologias (linguagem de programação, LLM, bibliotecas, *frameworks*, ferramentas, etc.) serão utilizadas para desenvolver a assistente virtual.
4. **Desenvolvimento e apresentação da primeira versão da assistente virtual:** desenvolver a assistente virtual, incorporando todas as decisões tomadas nas etapas anteriores. Apresentar esta primeira versão para PROGEP com o objetivo de coletar *feedbacks*.
5. **Atualização e melhorias de acordo com os *feedbacks* coletados:** analisar os *feedbacks* coletados para, com base nessa análise, realizar ajustes, melhorias, correções e implementações de novos recursos.
6. **Implementação de um método de avaliação da qualidade das respostas geradas pela assistente virtual:** estabelecer um processo formal e sistemático para avaliar a qualidade das respostas geradas pela assistente virtual. A implementação de um método de avaliação da qualidade das respostas é essencial para certificar que a assistente forneça respostas precisas e relevantes.
7. **Avaliação de desempenho considerando demais aspectos:** avaliar o desempenho da assistente virtual, levando em consideração os seguintes critérios: qualidade das respostas geradas, tempo de geração das respostas e custo médio por geração de resposta.

Vale ressaltar que, uma vez que o principal foco desta pesquisa é propor, através da utilização de IA, melhorias no suporte ao servidor do IFCE prestado pela PROGEP, este trabalho é de natureza aplicada. No tocante aos objetivos, a pesquisa possui características descritivas e exploratórias. Para Gil (2022) as pesquisas descritivas e exploratórias são normalmente realizadas por pesquisadores preocupados com atuação prática.

### **3.1 Definição do Escopo e Criação da Base de Conhecimento**

Durante a execução do projeto a PROGEP desempenhou um papel fundamental. Como Pró-reitoria responsável pela gestão de pessoas no IFCE, a PROGEP lida diariamente com diversas questões e solicitações dos servidores. A definição do escopo de temas foi fundamentada na análise das demandas recebidas pela PROGEP ao longo do tempo. Em outras palavras, a PROGEP identificou que a maioria das perguntas feitas pelos servidores do IFCE estão relacionadas aos seguintes temas: Remoção, Programa de Gestão de Desempenho (PGD) e Plano de Desenvolvimento de Pessoas (PDP).

A sazonalidade das demandas por respostas é um desafio notável no contexto da PROGEP. A natureza cíclica das necessidades de informação cria picos de interesse que podem ser seguidos por longos períodos de baixa interação. Por exemplo, durante um edital de remoção, a quantidade de perguntas sobre o processo pode aumentar significativamente, enquanto em períodos sem esse evento, a demanda diminui. Durante os picos, as respostas precisam ser fornecidas de maneira rápida e eficiente para atender à demanda.

Além disso, os temas tratados pela PROGEP são extensos e complexos, exigindo um conhecimento profundo dos processos. No entanto, é comum que as pessoas com expertise nesses assuntos estejam ocupadas com tarefas diárias que demandam a formalização desse conhecimento na análise de processos. O advento de um sistema como a assistente virtual Eris, apresentada neste projeto, oferece uma solução eficaz para superar esse desafio. Ao automatizar respostas e fornecer informações precisas durante os picos de demanda, esse sistema possibilita a prestação de um serviço importante com o mínimo envolvimento humano, permitindo que os especialistas concentrem-se em atividades mais complexas e estratégicas, enquanto as necessidades rotineiras são atendidas de maneira eficiente.

Posteriormente, com o auxílio da PROGEP, todas as leis, decretos, resoluções, instruções normativas, manuais, perguntas frequentes ou *Frequently Asked Questions* (FAQ), informações

disponíveis no sítio institucional, etc; que dizem respeito aos temas do escopo definido, foram organizadas e unificadas em uma base de conhecimento. Na Tabela 1 é apresentada toda a documentação, categorizada por tema, utilizada para formar a base de conhecimento.

Tabela 1 – Documentação utilizada para compor a base de conhecimento

| TEMA    | DOCUMENTAÇÃO UTILIZADA                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Remoção | Resolução CONSUP/IFCE nº 77, de 28 de junho de 2023;<br>Perguntas frequentes (FAQ);<br>Informações retiradas do sítio: < <a href="https://ifce.edu.br/progep/remocao">https://ifce.edu.br/progep/remocao</a> >.                                                                                                                                                                                                                                                                                                                              |
| PGD     | Decreto nº 11.072, de 17 de maio de 2022;<br>Instrução Normativa SGP/MPDG nº 2, de 12 de setembro de 2018;<br>Resolução CONSUP/IFCE nº 53, de 30 de agosto de 2022;<br>Nota Técnica GAB-PROGEP/PROGEP/REITORIA nº 2;<br>Ofício-Circular GAB-PROGEP/PROGEP/REITORIA-IFCE nº 1;<br>Ofício-Circular GAB-PROGEP/PROGEP/REITORIA-IFCE nº 73;<br>Portaria MEC nº 267, de 30 de abril de 2021;<br>Informações retiradas do sítio:<br>< <a href="https://ifce.edu.br/progep/programa-de-gestao">https://ifce.edu.br/progep/programa-de-gestao</a> >. |
| PDP     | Resolução CONSUP/IFCE nº 37;<br>Instrução Normativa SGP-ENAP/SEDGG/ME nº 21;<br>Decreto nº 9.991, de 28 de agosto de 2019;<br>Decreto nº 10.506, de 2 de outubro de 2020;<br>Manual PDP 2023 PROGEP/IFCE;<br>Perguntas frequentes (FAQ) retiradas do Portal do Servidor;<br>Informações retiradas do sítio:<br>< <a href="https://ifce.edu.br/progep/desenvolvimento-de-pessoas">https://ifce.edu.br/progep/desenvolvimento-de-pessoas</a> >.                                                                                                |

Fonte: autor (2023)

A documentação utilizada para criar a base de conhecimento foi salva localmente no formato *Portable Document Format* (PDF), totalizando 148 páginas de conteúdo. É importante destacar que não houve nenhum tipo de tratamento ou edição na documentação. Essa decisão foi tomada com o objetivo de garantir que qualquer novo documento que venha a surgir possa ser facilmente adicionado à base de conhecimento, sem a necessidade de etapas adicionais de pré-processamento. Isso significa que a base de conhecimento está preparada para crescer de forma orgânica e adaptar-se às mudanças de informações que possam surgir no futuro.

Ademais, manter os documentos em seu formato original garante, além de escalabilidade, confiabilidade à base de conhecimento. Os servidores podem confiar que as informações contidas na base de conhecimento refletem fielmente o conteúdo original dos documentos, o que é fundamental para garantir a precisão e a integridade das informações que serão disponibilizadas nas respostas geradas pela assistente virtual.

### 3.1.1 Criação de índice de pesquisa com o FAISS

Uma vez definida a base de conhecimento, é necessário criar uma representação compacta de todo o texto da documentação, para isso utilizamos técnicas de *word embeddings*. Posteriormente, através da biblioteca *FAISS*, cada representação é guardada em vetores que serão organizados em um índice de pesquisa. Isto se faz necessário para que as futuras pesquisas, comparações e recuperações de informações ocorram de forma rápida.

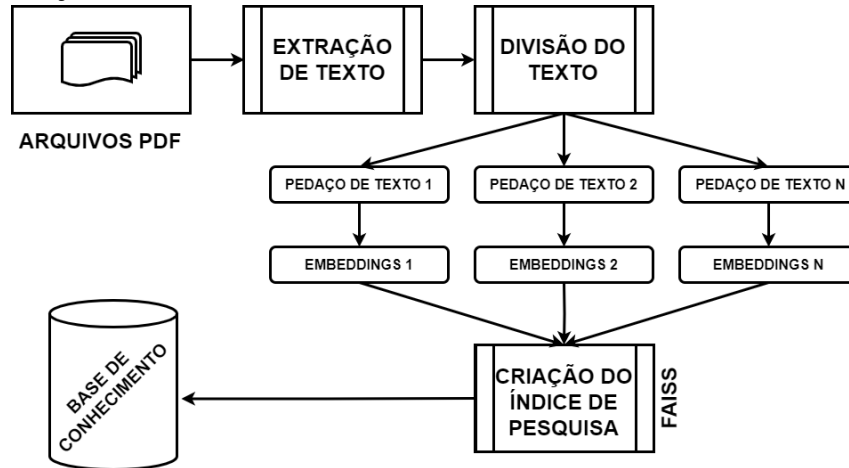
A biblioteca *FAISS*, desenvolvida pelo *Facebook AI Research* (FAIR), é uma ferramenta de código aberto projetada para facilitar a pesquisa eficiente em grandes conjuntos de dados de vetores de alta dimensionalidade. Especialmente otimizada para tarefas que envolvem a medição de similaridade entre vetores, como a distância cosseno ou Euclidiana. Dayton (2023) explica que a biblioteca *FAISS* foi lançada em 2017 para realizar buscas eficientes de similaridades e agrupamentos de vetores de alta dimensão, destacando-se pelo código extremamente eficiente e suporte multi-GPU para acelerar a pesquisa.

Em resumo, as etapas que seguem consistem em processar um arquivo PDF, dividir seu conteúdo em pedaços menores, incorporar esses pedaços em um índice de pesquisa e, finalmente, salvar esse índice localmente. Conforme, detalhado:

1. **Leitura de arquivo PDF:** utilizando o módulo *PyPDF2*, o arquivo PDF, ou seja, a base de conhecimento, é carregado, lido e armazenado.
2. **Extração de texto do PDF:** o texto de cada página do PDF é extraído e concatenado em uma única variável, que é a representação de texto puro do conteúdo do PDF.
3. **Divisão do texto em pedaços menores:** o próximo passo envolve dividir o conteúdo da variável em pedaços menores. Cada pedaço terá no máximo 1000 caracteres e haverá uma sobreposição de 200 caracteres entre um pedaço e outro. Esses pedaços menores são armazenados em uma lista.
4. **Criação de representações utilizando *embeddings*:** através da utilização de técnicas de *words embeddings* cada elemento da lista pode ser representado de forma mais compacta e em formato numérico.
5. **Criação de índice de pesquisa com *FAISS*:** usando a biblioteca *FAISS*, criamos um índice de pesquisa que associa cada elemento da lista a sua respectiva representação compacta, criada anteriormente.

6. **Salvar o índice de pesquisa localmente:** por fim, o índice de pesquisa é salvo localmente. Isso permite carregar e consultar o índice posteriormente sem a necessidade de reprocessar todo o PDF.

Figura 14 – Criação da base de conhecimento



Fonte: autor (2023)

Todas as etapas descritas anteriormente e ilustradas na Figura 14 ocorrem em aproximadamente 10 minutos e tem um custo de 2 centavos de dólar, referente ao uso do modelo de *embeddings* da *OpenAI*. Vale ressaltar que todo esse processo acontece de forma independente da assistente virtual. Em outros termos, esse processo é um arquivo de código fonte separado, que só precisa ser executado uma única vez ou caso ocorra remoção, adição ou alteração de informações na documentação que compõem a base de conhecimento. Por esse motivo, o custo referente à utilização do modelo de *embeddings* não será considerado no custo médio por geração de resposta.

O custo de 2 centavos de dólar associado a este projeto é um valor específico determinado pela extensão da base de conhecimento utilizada, composta por 148 páginas. É importante destacar que esse custo está intrinsecamente ligado ao tamanho da base de conhecimento, sendo proporcional ao aumento do número de páginas. Em projetos nos quais a base de conhecimento é substancialmente maior, esse custo também se expandirá em consonância.

Outro fator crítico a ser considerado é a frequência de atualização da base de conhecimento. Neste projeto, em que não houve atualizações periódicas, o custo permanece como um valor único e praticamente insignificante. No entanto, em projetos nos quais a base de conhecimento é atualizada regularmente, como, por exemplo, três ou quatro vezes por semana, esse custo deve ser cuidadosamente avaliado, pois pode se acumular ao longo do tempo.

### 3.2 Desenvolvimento da Assistente Virtual Eris

Para o desenvolvimento da Eris a linguagem de programação escolhida foi o *Python*, linguagem muito utilizada no mercado e em pesquisas científicas. Sarkar (2022) destaca *Python* com a linguagem de programação mais usada por profissionais da área de ciência de dados, *machine learning* e entusiastas de IA. Outro importante ponto é a comunidade muito ativa, que se reflete em uma grande variedade de recursos, documentação, bibliotecas e *frameworks*, tais como *TensorFlow*, *PyTorch*, *Streamlit* e *LangChain*. Além de prontas para uso e robustas, muitas das bibliotecas disponíveis para *Python* podem ser implementadas de forma eficaz no desenvolvimento de aplicações de uso geral, ciências da computação, visualização de dados e aprendizado de máquina (ARIF, 2020).

No que se refere ao LLM utilizado, a escolha foi o GPT-3, mas especificamente sua versão 3.5. GPT-3 é um modelo de linguagem autorregressivo com 175 bilhões de parâmetros, que apresenta resultados promissores em diversas tarefas que envolvem PLN. Brown et al. (2020) destaca que mesmo sem qualquer treinamento em base de dados específicas (*zero-shot*) ou com poucos exemplos de como executar determinadas tarefas (*few-shot*), diversos resultados alcançados pelo GPT-3 superam outros LLMs do estado da arte já treinados para tarefas específicas (*fine tuning*).

Abaixo segue a lista de todas as tecnologias, recursos, *frameworks*, ferramentas, módulos, bibliotecas e o detalhamento do *hardware* utilizados para o desenvolvimento da Eris:

- **Softwares utilizados:**

- Linguagem de programação: *Python*, versão 3.10.11;
- Ambiente de desenvolvimento: *Visual Studio Code* e *Google Colab*;
- Bibliotecas e Módulos: *PyPDF2*, *FAISS*, *Openai* e *Tiktoken*;
- *Frameworks*: *LangChain*, versão 0.0.171 e *Streamlit*, versão 1.26.0;
- LLM: GPT-3.5 (API da *OpenAI*), modelo *text-davinci-003*;

- **Hardware utilizado:**

- Notebook *Aspire A515-51G*;
- Processador *Intel Core i5-7200U*;
- Memória RAM de 12 GB DDR4 2133 MHz;

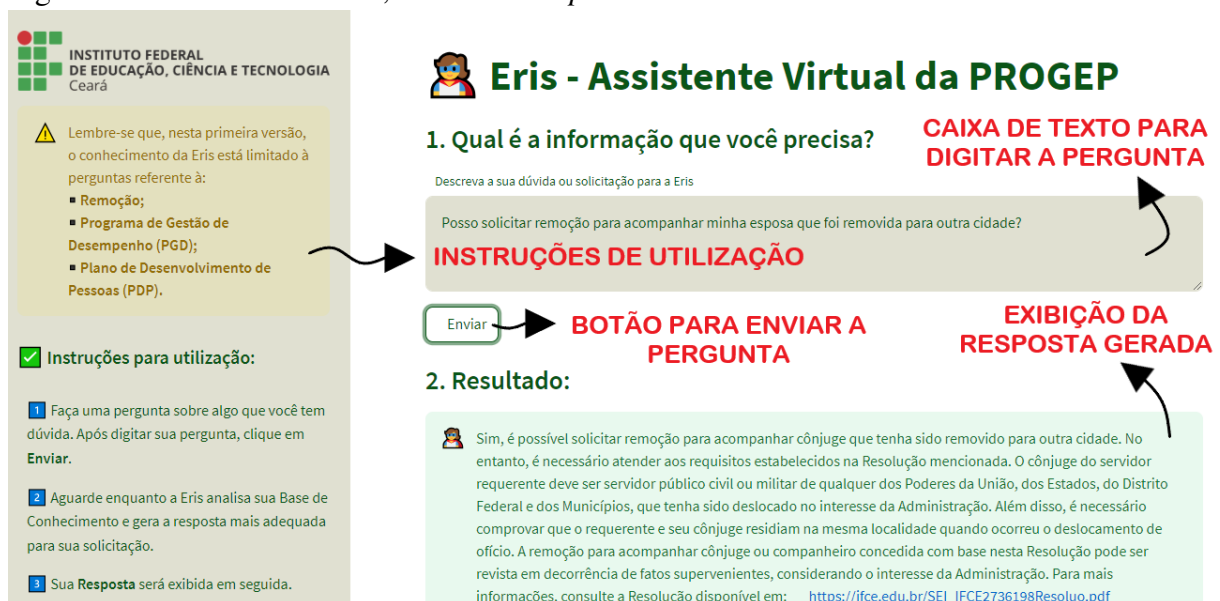
### 3.2.1 Criação da interface da Eris

A Eris foi idealizada para funcionar como uma aplicação *web*, assim como o ChatGPT, uma das implementações mais famosas do GPT-3.5. Desta forma, a Eris deve ser acessível de qualquer navegador e por qualquer dispositivo, o que inclui os dispositivos móveis. Logo, é imprescindível que a interface da Eris seja responsiva.

O *Streamlit*, um *framework open source*, foi utilizado no desenvolvimento da interface da Eris. O uso do *Streamlit* representa um dos métodos mais eficientes e simplificados para o desenvolvimento e a disseminação de aplicações baseadas em *Python*, possibilitando a utilização de sua infraestrutura em nuvem para a implantação, administração e distribuição das aplicações criadas (PATIL; LOKESHA, 2022).

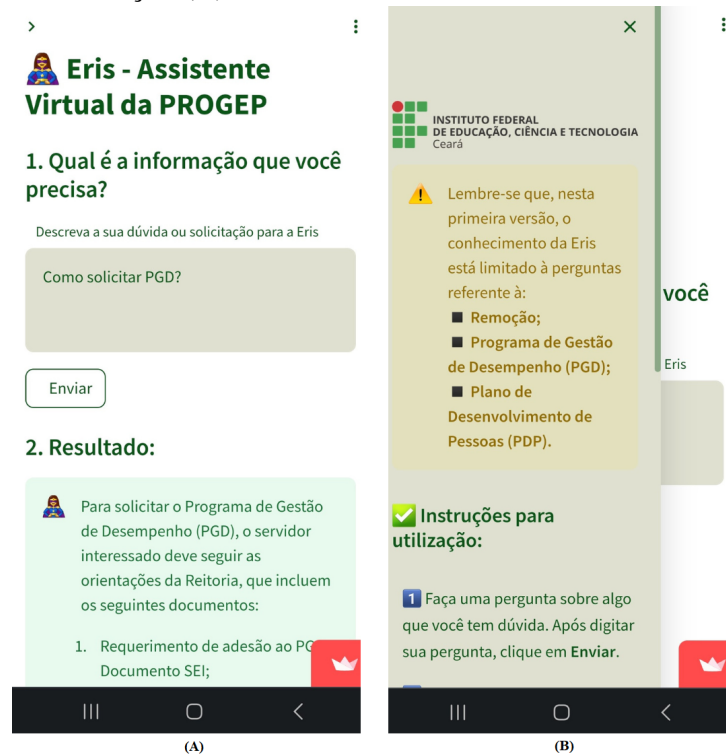
A interface da Eris apresenta uma organização simples para a interação com os servidores. Ela é constituída por uma barra lateral, que contém instruções objetivas sobre como utilizar a assistente, e uma área central, que apresenta todos os elementos principais de interação. Na área principal da interface os servidores encontrarão uma caixa de texto, onde podem digitar suas perguntas, e um botão para enviar as perguntas feitas. Logo abaixo, na mesma área, é apresentada a seção onde as respostas geradas pela assistente são exibidas. Essa organização foi criada para permitir que os servidores se comuniquem de forma intuitiva com a Eris, fornecendo perguntas e recebendo respostas.

Figura 15 – Interface da Eris, versão *desktop*



Fonte: autor (2023)

Figura 16 – Interface da Eris, versão *mobile*, área principal (A) e versão *mobile*, com barra lateral em exibição (B)



Fonte: autor (2023)

### 3.2.2 Funcionamento da Eris

O funcionamento básico da Eris consiste em um fluxo de tarefas que é iniciado a partir do envio da pergunta feita pelo servidor, e se encerra com a apresentação da resposta gerada em tela. Neste contexto, o *framework LangChain* e o LLM utilizado desempenham papéis de suma importância.

Apesar de muito recente, o *LangChain* já é amplamente reconhecido na comunidade de IA. Isto ocorre devido sua capacidade de interagir perfeitamente com várias fontes de informações distintas, permitindo que os desenvolvedores utilizem, convenientemente, as fontes de informações disponíveis para interagir com as aplicações em desenvolvimento (TOPSAKAL; AKINCI, 2023).

Através da função *similarity\_search*, o *LangChain* compara a pergunta feita pelo servidor com as informações existentes na base de conhecimento disponível, identificando e retornando as informações mais relevantes referentes à pergunta recebida. Isto significa que, após a comparação, o *LangChain* devolve os trechos de informações que provavelmente respondem à pergunta recebida.

A quantidade de trechos que serão retornados é configurada através do parâmetro  $K$  da função *similarity\_search*, que por padrão vale 4. Ou seja, para cada pergunta feita, o *LangChain* retorna os 4 trechos de texto mais relevantes da base de conhecimento. Cada trecho em questão possui 1000 caracteres, conforme configurado na criação da base de conhecimento deste projeto.

Por fim, tanto a pergunta feita pelo servidor quanto os trechos de informações retornados pelo *LangChain*, são enviados para o LLM utilizado, neste caso para o GPT-3.5, mais precisamente para o modelo de linguagem *text-davinci-003*. Esse envio é realizado via *prompt*, através da API da *OpenAI*. No *prompt* enviado, além das informações supracitadas, é essencial explicitar ao modelo que, para a gerar a resposta, apenas as informações existentes no *prompt* devem ser consideradas como contexto relevante.

Na ausência de uma orientação explícita no *prompt* de envio para a API da *OpenAI*, o modelo de linguagem tentará formular uma resposta com base em seu conhecimento prévio. Isso ocorre porque os LLMs, como o GPT-3, são treinados em uma grande variedade de textos e dados da internet e, por isso, possuem uma vasta base de conhecimento prévio (BROWN et al., 2020). Portanto, a orientação explícita no *prompt* é crucial para garantir que a resposta gerada esteja alinhada com as informações de contexto recebida.

Além das informações de orientação necessárias no *prompt* de envio, outros dois parâmetros muito importantes que devem ser passados para API: *temperature* e *max\_token*. O parâmetro *temperature* pode variar de 0 à 1, esse valor ajusta a criatividade e a fluidez das respostas geradas pela assistente. Valores mais baixos tornam as respostas mais focadas e determinísticas, enquanto valores mais altos introduzem aleatoriedade e diversidade nas respostas. Isso é fundamental para adaptar o estilo de comunicação. Para Eris, a fim de garantir respostas mais objetivas e evitar aleatoriedades, esse parâmetro foi configurado para 0. Quanto ao parâmetro *max\_token*, ele é utilizado para controlar o tamanho das respostas geradas, limitando o número máximo de *tokens* que serão utilizados na resposta. Para Eris, como existe a possibilidade de diversas perguntas feitas possuírem respostas com grande volume de informações, configuramos esse parâmetro para 2000 *tokens*.

Tabela 2 – Exemplo do funcionamento da Eris

| PERGUNTA                                      | INFORMAÇÕES RELEVANTES                                                                                             | PROMPT ENVIADO                                                                                  | RESPOSTA GERADA                                                |
|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|----------------------------------------------------------------|
|                                               |                                                                                                                    | Siga os 4 passos a seguir para gerar a resposta:                                                |                                                                |
|                                               |                                                                                                                    | 1. Leia as informações recebidas neste prompt;<br>=====                                         | “Sim, há um prazo para se inscrever no PGD [...]               |
|                                               |                                                                                                                    | 2. Utilize somente as informações deste prompt para gerar a resposta;<br>=====                  | o servidor deverá encaminhar às unidades de gestão de pessoas  |
| “Tenho prazo para solicitar a adesão ao PGD?” | “[...] o processo de solicitação de adesão ao PGD, com 15 dias de antecedência da data de início planejada. [...]” | 3. Seja detalhado;<br>=====                                                                     | o processo de solicitação de adesão ao PGD                     |
|                                               |                                                                                                                    | 4. Para casos em que a resposta não for encontrada, sua resposta padrão deve ser [...]<br>===== | com 15 dias de antecedência da data de início planejada [...]” |
|                                               |                                                                                                                    | { INFORMAÇÕES RELEVANTES }                                                                      |                                                                |
|                                               |                                                                                                                    | { PERGUNTA }                                                                                    |                                                                |

Fonte: autor (2023)

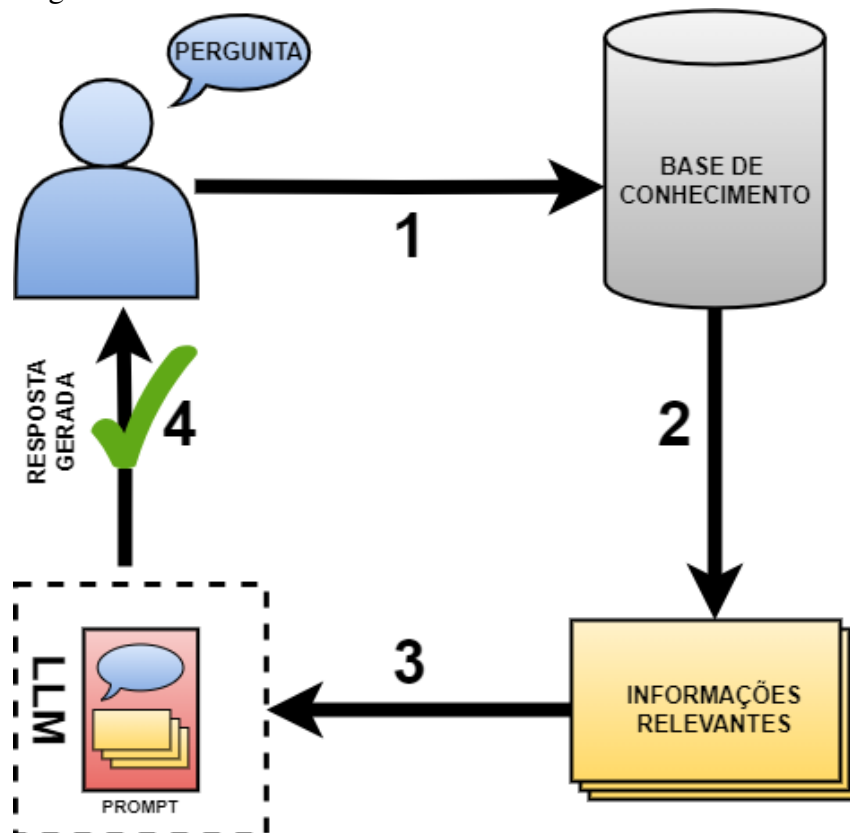
### 3.2.3 Apresentação da primeira versão e atualizações após *feedbacks* da PROGEP

Após o desenvolvimento da primeira versão da Eris, o passo seguinte foi apresentá-la à PROGEP, com o objetivo de coletar *feedbacks* sobre o andamento do projeto, bem como qualquer sugestão que aprimorasse a experiência de utilização da assistente virtual. Os *feedbacks* recebidos foram extremamente positivos.

Após a análise dos *feedbacks* e sugestões recebidas, algumas modificações foram implementadas na Eris, com destaque para:

- Alteração de um dos documentos da base de conhecimento que apresentava problemas na extração de informações.
- Revisão da resposta padrão para as situações em que a Eris não entende a pergunta ou não encontra uma resposta adequada. A nova resposta padrão acrescentou a sugestão de abertura de chamado no sistema SUAP, assim como solicitado pela PROGEP.
- Adição da informação de tempo necessário para gerar a resposta. Esta informação é dada em segundos, sendo exibida imediatamente abaixo da resposta gerada.

Figura 17 – Diagrama de funcionamento básico da Eris



Fonte: autor (2023)

### 3.3 Metodologia de Avaliação da Qualidade das Respostas Geradas

A avaliação da qualidade das respostas geradas pela assistente virtual Eris é uma tarefa essencial para garantir a utilidade da ferramenta desenvolvida. A eficácia de um sistema de assistente virtual é geralmente medida pela precisão, relevância e utilidade das respostas que ele fornece. Atualmente, uma tendência observada na indústria de tecnologia é a implementação de sistemas de assistente virtual por meio da utilização de LLMs, reconhecidos por seu poder de processamento de texto e linguagem natural. Isso permite a implementação de aplicações de

assistência e automação em várias áreas, como atendimento ao cliente e pesquisa de informações. No entanto, os métodos usuais de avaliação de LLMs do estado da arte podem não ser adequados para avaliar as respostas geradas pela Eris.

Os LLMs são frequentemente testados em uma variedade de tarefas, incluindo tradução, classificação de textos, sumarização de informações e interpretação de questões com raciocínio lógico. O *Hugging Face*, empresa de destaque na área de *machine learning*, avalia os LLMs através da utilização do *Evaluation Harness*, um *framework* apresentado por Gao et al. (2021) para testar modelos de linguagem generativos na execução de diferentes atividades. A avaliação é realizada nas 4 categorias abaixo:

- ***AI2 Reasoning Challenge*** (CLARK et al., 2018): consiste em avaliar as respostas de LLMs sobre questões de ciências do nível de ensino fundamental.
- ***HellaSwag*** (ZELLERS et al., 2019): consiste em atividades que normalmente são fáceis para humanos, mas desafiadoras para LLMs do estado da arte, pois envolvem inferência do senso comum.
- ***MMLU*** (HENDRYCKS et al., 2021): consiste em testes que medem a precisão de um LLM. O teste inclui atividades de diversas áreas, tais como matemática básica, ciência da computação e direito.
- ***TruthfulQA*** (LIN; HILTON; EVANS, 2022): consiste em um teste para medir o quanto um LLM é propenso a reproduzir equívocos populares, comumente encontrados online.

Os testes utilizados nesses tipos de avaliação não são aplicáveis à Eris, que é apenas uma implementação, isto é, uma forma de utilização de um LLM. A Eris foi projetada para ser uma assistente virtual que responde perguntas feitas pelos servidores sobre temas de competência da PROGEP. Assim, sua função é bem definida e limitada à referida atividade.

Métodos de avaliação como *Bilingual Evaluation Understudy* (BLEU) (PAPINENI et al., 2002) e *Metric for Evaluation of Translation with Explicit ORdering* (METEOR) (BANERJEE; LAVIE, 2005) também podem não ser adequados para avaliar a qualidade das respostas geradas pela Eris. Esses métodos são baseados em métricas de similaridade de n-gramas entre a saída do sistema, ou seja, a resposta gerada pela assistente, e uma resposta de referência dada por um ser humano. Essa metodologia de avaliação não captura adequadamente a qualidade da resposta em termos de relevância e utilidade para os servidores.

Tabela 3 – Pontuação BLEU e METEOR

| PERGUNTA                                                                                                               | RESPOSTA DE REFERÊNCIA                                                                                                                                                                                                                 | RESPOSTA GERADA                                                                                                                                                                                                              | BLEU  | METEOR |
|------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|--------|
| “Onde posso encontrar Editais de remoção passados no site do IFCE?”                                                    | “Prezado (a), os editais de remoção, bem como quaisquer informações referentes à remoção estão publicadas no site do IFCE no link abaixo.”                                                                                             | “Os editais, bem como os resultados e demais documentos dos concursos de remoção já realizados (desde 2011) podem ser encontrados em: <link>”                                                                                | 8,10  | 16,90  |
| “Quais os documentos necessários e o trâmite institucional para solicitação de remoção por acompanhamento de cônjuge?” | “O cônjuge do servidor requerente deve figurar como servidor público civil ou militar, de qualquer dos Poderes da União, dos Estados, do Distrito Federal e dos Municípios, que tenha sido deslocado no interesse da Administração...” | “O cônjuge do servidor requerente deve ser servidor público civil ou militar de qualquer dos Poderes da União, dos Estados, do Distrito Federal e dos Municípios, que tenha sido deslocado no interesse da Administração...” | 7,27  | 7,02   |
| “Quais os documentos necessários e o trâmite institucional para solicitação de remoção por acompanhamento de cônjuge?” | “O deslocamento de que trata o inciso I deve ter ocorrido em data posterior ao efetivo exercício do servidor requerente no IFCE no cargo que pretende a remoção.”                                                                      | “O deslocamento do cônjuge deve ter ocorrido após o servidor requerente ter iniciado efetivamente suas atividades no IFCE no cargo que pretende a remoção.”                                                                  | 34,04 | 60,52  |

Fonte: autor (2023)

Nos exemplos mostrados na Tabela 3, para calcular a pontuação BLEU foi utilizado o *Interactive BLEU score evaluator* da *Tilde Custom Machine Translation* e para calcular a pontuação METEOR foi utilizado o módulo *PyMETEOR* do *Python*. As duas pontuações variam de 0 a 100, onde 100 indica uma correspondência perfeita entre a resposta gerada e a resposta de referência. Conforme constatado, mesmo que as respostas geradas pela Eris tenham exatamente

as mesmas informações que a resposta de referência, a pontuação calculada pelos dois métodos não fica nem próximo da pontuação máxima, ficando muitas vezes bem aquém do esperado.

Desta forma, dada a natureza específica deste projeto e as limitações dos métodos de avaliação citados, a escolha mais coerente de avaliação da qualidade das respostas geradas é a avaliação humana. Este método permite uma avaliação mais qualitativa e contextualizada das respostas geradas pela Eris. A avaliação humana pode considerar fatores como a relevância da resposta para a pergunta, a precisão das informações fornecidas, clareza da linguagem usada e, principalmente, ponderação das diferenças de informações entre as respostas geradas e as respostas de referência. Para Reddy (2023) “a eficácia de um modelo de linguagem como ChatGPT é melhor avaliada por humanos que podem julgar a qualidade do texto gerado em termos de sua fluência, coerência e relevância para a tarefa em mão.”

### ***3.3.1 Avaliação humana***

Para realizar a avaliação humana das respostas geradas pela Eris é essencial a figura do especialista no tema em questão. Esse especialista desempenha um papel crítico, pois pode tanto analisar as respostas geradas pela assistente, como no trabalho de Perez-Pino et al. (2023), como também gerar respostas de referência. Assim como nos métodos de avaliação já descritos, as respostas de referência servem como um ponto de comparação fundamental para a avaliação, permitindo medir a qualidade e a precisão das respostas da assistente em relação às expectativas e ao conhecimento especializado do especialista. Isso garante que a assistente virtual seja capaz de fornecer informações corretas e relevantes.

Neste projeto, os especialistas são servidores do IFCE que têm competência e conhecimento especializado nos temas tratados pela Eris. Solicitamos aos especialistas que listassem as perguntas mais frequentes sobre os dois temas do escopo: remoção e PGD; e suas respectivas respostas de referência. Para simularmos diferentes formas de se fazer uma determinada pergunta, para cada pergunta da lista geramos 9 variações da mesma, totalizando 10 formas por pergunta, sendo uma original, a que foi fornecida pelo especialista do tema, e nove variações da pergunta original. As variações das perguntas foram geradas via ChatGPT.

Tabela 4 – Exemplo de variações de pergunta frequente geradas via ChatGPT

| PERGUNTA FREQUENTE                                   | VARIAÇÃO DA PERGUNTA                                                          | RESPOSTA DE REFERÊNCIA                                   |
|------------------------------------------------------|-------------------------------------------------------------------------------|----------------------------------------------------------|
| “A remoção por permuta pode ocorrer entre docentes?” | “É possível que a remoção por permuta aconteça entre os professores?”         | “Não há previsão de remoção por permuta entre docentes.” |
|                                                      | “Entre os docentes, é viável a remoção por permuta?”                          |                                                          |
|                                                      | “Existe a possibilidade de ocorrer remoção por permuta entre os professores?” |                                                          |
|                                                      | “Pode haver remoção por permuta entre os docentes?”                           |                                                          |
|                                                      | “Os docentes podem ser removidos por permuta?”                                |                                                          |
|                                                      | “Entre os professores, a remoção por permuta é uma opção?”                    |                                                          |
|                                                      | “É permitida a remoção por permuta entre os docentes?”                        |                                                          |
|                                                      | “Há a chance de a remoção por permuta acontecer com os professores?”          |                                                          |
|                                                      | “É factível a remoção por permuta envolvendo os docentes?”                    |                                                          |

Fonte: autor (2023)

Para o terceiro tema do escopo, PDP, utilizamos as 66 perguntas e respostas disponibilizadas pelo Governo Federal no Portal do Servidor, disponíveis em: <<https://www.gov.br/servidor/pt-br/aceso-a-informacao/gestao-de-pessoas/pndp/paginas/duvidas>> (BRASIL, 2020).

Similar ao realizado por Perez-Pino et al. (2023), adaptamos uma escala *Likert* de 5 pontos para realizar a pontuação da avaliação humana. Cada um dos cinco pontos dessa escala tem um significado, onde a pontuação máxima é o 5, que representa uma qualidade de resposta tão boa quanto a resposta de referência; e a pontuação mínima é o 1, que representa a situação em que a Eris não sequer conseguiu entender a pergunta recebida.

As respostas de referência fornecidas pelos especialistas foram definidas como padrão máximo de qualidade de resposta, por isso receberam a pontuação máxima, ou seja, 5. Desta forma, comparando as respostas geradas pela Eris com a sua respectiva resposta de referência, cada uma das respostas recebeu uma pontuação de acordo com as informações da Tabela 5.

Tabela 5 – Pontuação da escala *Likert* modificada utilizada na avaliação humana

| PONTUAÇÃO | SIGNIFICADO DA PONTUAÇÃO                                                                                                                                                                                                                                                                |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1         | Eris não foi capaz de entender a pergunta ou de gerar uma resposta para a pergunta realizada. A resposta gerada é a resposta padrão para a situação de não entendimento.                                                                                                                |
| 2         | Eris gerou uma resposta não satisfatória. A resposta é composta somente de informações incoerentes e/ou incorretas.                                                                                                                                                                     |
| 3         | Eris gerou uma resposta aceitável, mas incompleta. Apesar de aceitável, a resposta possui poucas informações relevantes ou algum trecho com informação incorreta que pode comprometer a coerência da resposta.                                                                          |
| 4         | Eris gerou uma resposta satisfatória. Apesar de satisfatória, a resposta não é tão boa quanto a resposta de referência, seja por possuir pequenos trechos confusos e/ou prolixos; ou por possuir algum trecho com informação incorreta, mas que NÃO compromete a coerência da resposta. |
| 5         | Eris gerou uma resposta tão boa quanto a resposta de referência                                                                                                                                                                                                                         |

Fonte: autor (2023)

Embora os resultados que serão apresentados neste projeto se concentrem exclusivamente na performance da assistente virtual Eris em relação às perguntas específicas usadas para avaliação, é importante ressaltar que uma variedade de outras perguntas foi direcionada à Eris durante o processo. Muitas dessas questões foram, propositalmente, sobre temas que estão além do escopo do conhecimento da assistente. Nestes casos, a resposta esperada seria a resposta padrão, indicando que a Eris não possui informação relevante para a pergunta. Este aspecto é crucial para uma avaliação abrangente, pois destaca as limitações do escopo da assistente e reconhece sua habilidade de discernir e comunicar a falta de conhecimento em determinados temas. Em outras palavras, caso seja feita uma pergunta sobre férias ou desconto na folha de pagamento, a Eris deve responder com sua resposta padrão, já que a pergunta recebida está fora do seu escopo de funcionamento.

### 3.4 Metodologia de Avaliação do Tempo e Custo Médio para Geração de Resposta

Além da qualidade das respostas, os outros dois aspectos avaliados neste projeto foram: tempo de geração das respostas e custo médio por geração de resposta. O tempo de geração das respostas é uma métrica que verifica a rapidez com que a Eris pode produzir e exibir uma resposta. Este tempo é contabilizado do momento em que uma pergunta é enviada até o momento em que a resposta é exibida em tela para o servidor. Quanto ao custo médio por geração de resposta, é uma métrica importante que estima o custo, em reais, necessário para Eris gerar uma resposta.

#### 3.4.1 Avaliação do tempo de geração de resposta

Para avaliar o tempo de geração de resposta, utilizamos a informação mostrada na interface da própria assistente virtual. Para Nah (2003) o tempo de espera tolerável dos usuários que estão realizando atividades de recuperação de informações é de aproximadamente 2 segundos.

No contexto da internet, quando o sistema utilizado não fornece qualquer indicação ou *feedback* sobre progresso da atividade em questão, o máximo de tempo que um usuário comum, que utiliza a internet para navegação de modo geral, está disposto a esperar é de 15 segundos (NAH, 2003). Isso significa que a utilização de elementos visuais como forma de indicação de progresso em uma atividade gera uma percepção de tempo de espera que é mais importante do que o próprio tempo de espera real.

Trabalhos como o de Li e Chen (2019), investigam quais os efeitos dos *designs* de *feedback* visual com relação ao tempo de espera que um usuário está disposto a aguardar. Neste projeto foi utilizado o *spinner*, um elemento visual de status do *Streamlit*, como forma de indicar ao servidor que a Eris está analisando a pergunta recebida e gerando uma resposta adequada.

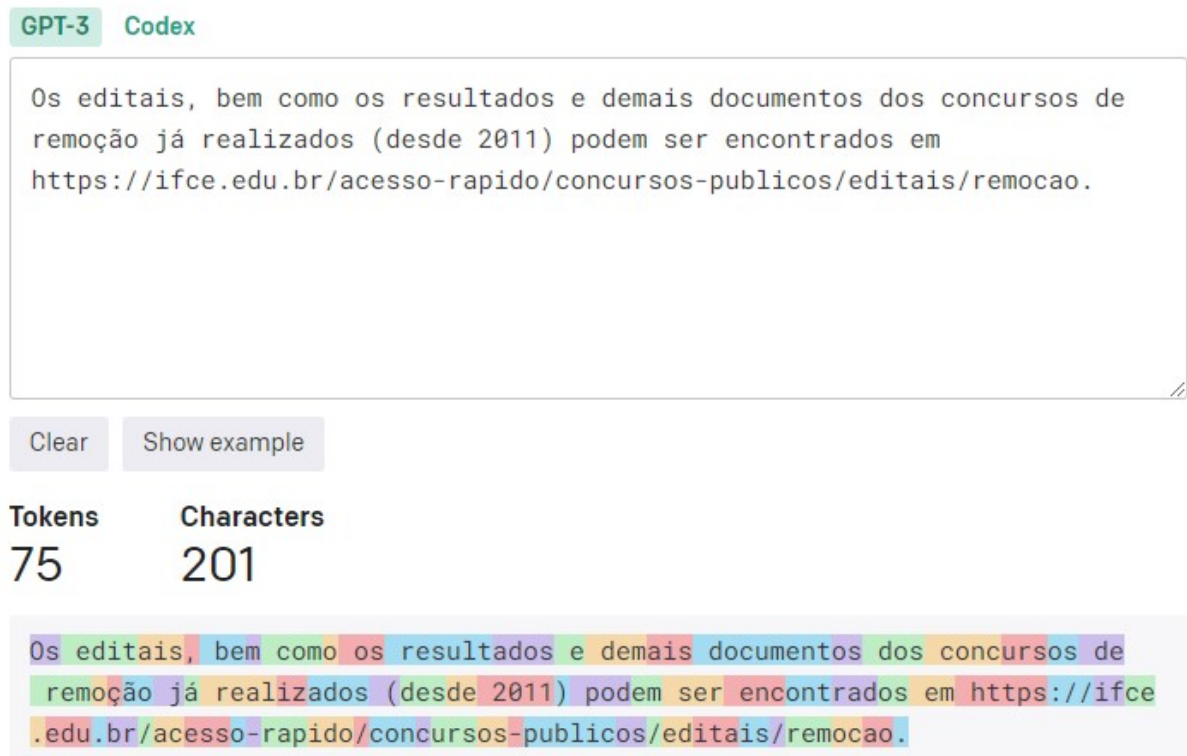
No total a Eris foi testada com 246 perguntas sobre os três temas do escopo. Para cada uma das 246 perguntas feitas para a Eris, o tempo de geração de resposta foi registrado e tabelado. Considerando os trabalhos citados, foi utilizado como limite aceitável para geração de resposta o tempo de 10 segundos. “Para os usuários da web, um tempo de espera superior a 10 segundos parece ser inaceitável, insatisfatório e tem maior probabilidade de criar experiências negativas” (LI; CHEN, 2019).

### 3.4.2 Avaliação do custo médio por geração de resposta

Para avaliar o custo médio, em reais, por geração de respostas da Eris, foi necessário utilizar o preço tabelado de processamento de *tokens* disponibilizado pela *OpenAI*. Esse preço varia dependendo da capacidade do modelo utilizado e é cobrado a cada 1000 *tokens* processados. Além da definição mostrada no Capítulo 2 deste trabalho, a *OpenAI* usualmente define *tokens* como sendo pedaços de palavras, onde 1000 *tokens* equivaleria aproximadamente 750 palavras, considerando a língua inglesa.

Neste trabalho utilizamos o modelo de linguagem *text-davinci-003*, que tem um custo de 2 centavos de dólar (\$0,02) por 1000 *tokens* processados. Considerando uma taxa de conversão de aproximadamente 1 para 5, entre a moeda americana e a moeda brasileira, o custo em reais para o *text-davinci-003* é de 10 centavos (R\$0,1). A utilização da ferramenta *Tokenizer* (OPENAI, 2022) foi fundamental para contabilizar a quantidade de *tokens* necessários para gerar cada resposta. Ao combinar esses dois fatores, foi possível calcular o custo médio para o uso da assistente virtual, proporcionando uma estimativa realista dos gastos de utilização da Eris.

Figura 18 – Utilização da ferramenta *Tokenizer* para verificar a quantidade de *tokens*



Fonte: autor (2023)

O custo associado à geração de respostas depende fundamentalmente da quantidade de *tokens* que são processados durante a interação. Esses *tokens* podem ser divididos em duas categorias principais: *tokens* de entrada e *tokens* de saída.

Os *tokens* de entrada consistem em todos os elementos necessários para que a Eris compreenda e responda adequadamente à pergunta do servidor. Eles incluem não apenas a pergunta em si, mas também as orientações do *prompt* que ajudam a direcionar a geração da resposta e as informações relevantes trazidas pelo *LangChain*, que auxiliam na compreensão do contexto. Portanto, os *tokens* de entrada representam a base do entendimento da Eris e podem variar significativamente de uma interação para outra, dependendo da complexidade da pergunta e das informações relevantes que servirão de contexto.

Por outro lado, os *tokens* de saída são essencialmente a resposta gerada pela Eris. A quantidade de *tokens* de saída depende diretamente da extensão da resposta gerada pela assistente e pode variar conforme a complexidade da informação a ser comunicada.

É importante notar que para a Eris, apenas as orientações do *prompt* possuem uma quantidade fixa de tokens, uma vez que as orientações sempre serão as mesmas independente das perguntas e respostas. Foram utilizados 446 *tokens* para as orientações no *prompt* da Eris. Todos os outros *tokens*, tanto de entrada quanto de saída, podem variar.

Desta forma, para realizar o cálculo do custo médio, foram selecionadas aleatoriamente 49 das 246 perguntas que foram utilizadas para testar a Eris, ou seja, aproximadamente 20% das perguntas. Todos os *tokens* de cada uma das perguntas selecionadas foram contabilizados e posteriormente foi calculado a média aritmética de *tokens*. Com a quantidade média de *tokens*, os valores foram substituídos na Equação (3.1):

$$C_mGR = (T_mP + T_mIR + TO + T_mRG) \times 10^{-4} \quad (3.1)$$

Onde:  $C_mGR$  é o custo médio por geração de resposta, dado em reais;  $T_mP$  é a quantidade média de *tokens* por pergunta,  $T_mIR$  é a quantidade média de *tokens* por informações relevantes retornadas pelo *LangChain*,  $TO$  é a quantidade de *tokens* para orientação, que nesse projeto é de 446;  $T_mRG$  é a quantidade média de *tokens* por resposta gerada e o  $10^{-4}$  corresponde ao custo tabelado da *OpenAI*, em real, por 1000 *tokens* processados para o modelo *text-davinci-003*.

## 4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados os resultados da avaliação de desempenho da assistente virtual Eris. A avaliação levou em consideração três aspectos principais: qualidade das respostas geradas, tempo de geração de resposta e custo médio por geração de resposta. Os resultados são apresentados em forma de gráficos e tabelas.

### 4.1 Qualidade das Respostas Geradas

Utilizando as respostas de referência como padrão máximo de qualidade e a escala *Likert* modificada de 5 pontos, onde a pontuação 5 representa o mesmo nível de qualidade da resposta de referência, cada uma das 246 perguntas recebeu uma pontuação, de acordo com a configuração mostrada na Tabela 5.

#### 4.1.1 Qualidade das respostas sobre Remoção

No tema Remoção a assistente virtual Eris foi testada com um total de 100 perguntas. A Figura 19 apresenta um gráfico do tipo radar que compara a qualidade das respostas geradas pela Eris com as respostas de referência. O centro do radar representa o valor 1, que é o pior desempenho de qualidade da escala Likert utilizada, e a borda do radar representa o valor 5, que é o melhor desempenho de qualidade.

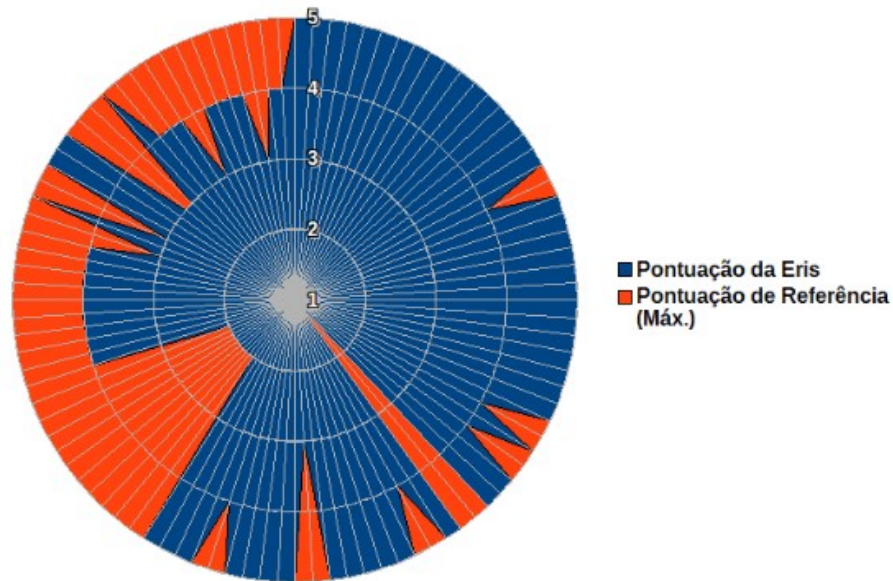
Nessa categoria de perguntas, 58% das respostas geradas pela Eris foram tão boas quanto as respostas de referência (pontuação 5), 24% foram de respostas satisfatórias (pontuação 4), 7% foram de respostas aceitáveis (pontuação 3) e 11% de respostas insatisfatórias e respostas padrão (pontuação 2 e 1, respectivamente). A Tabela 6 apresenta a análise estatística das pontuações das 100 respostas geradas, com o valor médio da pontuação, a variância entre os valores de pontuação de cada resposta, o desvio padrão da pontuação e a menor pontuação recebida.

Tabela 6 – Análise estatística da pontuação das respostas geradas pela Eris (Remoção)

| TIPO DE ANÁLISE ESTATÍSTICA    | RESPOSTAS GERADAS PELA ERIS |
|--------------------------------|-----------------------------|
| Pontuação média                | 4,28                        |
| Variância da pontuação         | 1,072323232323              |
| Desvio padrão da pontuação     | 1,03553041110497            |
| Pontuação mínima da amostragem | 1                           |

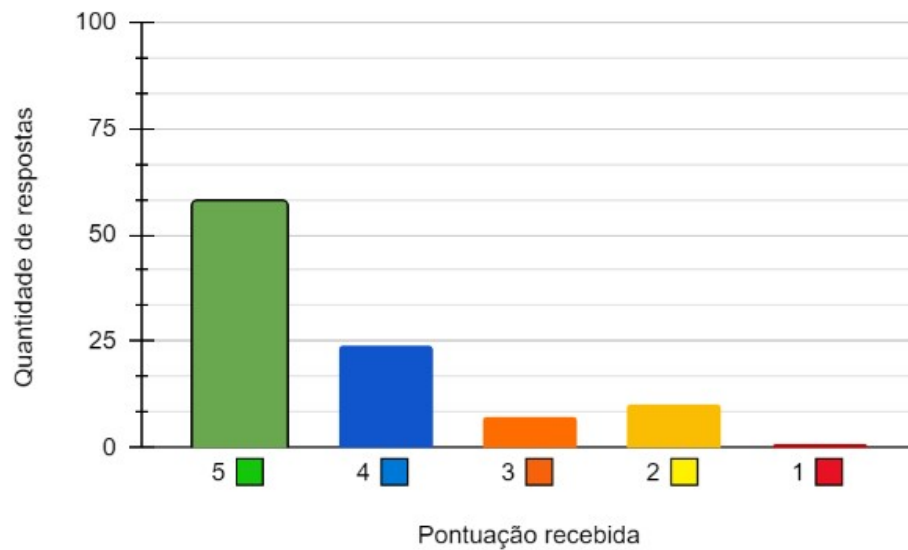
Fonte: autor (2023)

Figura 19 – Comparação entre a pontuação da Eris (Remoção) e a pontuação de referência (máxima qualidade)



Fonte: autor (2023)

Figura 20 – Respostas agrupadas pela pontuação recebida (Remoção)



Fonte: autor (2023)

Diante da análise apresentada, é possível constatar que para as perguntas sobre o tema Remoção, a Eris obteve uma pontuação média de 4,28 e apresentou 82% de respostas com qualidade de nível a partir do satisfatório, dos quais aproximadamente 60% são de qualidade máxima, ou seja, do mesmo nível de qualidade de uma resposta dada por um especialista humano.

#### 4.1.2 Qualidade das respostas das perguntas sobre (PGD)

No tema PGD a assistente virtual Eris foi testada com um total de 80 perguntas. Assim como no tópico anterior, a Figura 21 apresenta um gráfico do tipo radar que compara a qualidade das respostas geradas pela Eris com as respostas de referência. Nessa categoria de perguntas, a Eris apresentou uma performance superior quando comparada com a categoria de perguntas sobre Remoção.

Nessa categoria de perguntas, 90% das respostas geradas pela Eris foram tão boas quanto as respostas de referência (pontuação 5), 7,5% foram de respostas satisfatórias (pontuação 4), 1,25% foram de respostas aceitáveis (pontuação 3) e 1,25% de respostas insatisfatórias e respostas padrão (pontuação 2 e 1, respectivamente). A Tabela 7 apresenta a análise estatística das pontuações das 80 respostas geradas, com o valor médio da pontuação, a variância entre os valores de pontuação de cada resposta, o desvio padrão da pontuação e a menor pontuação recebida.

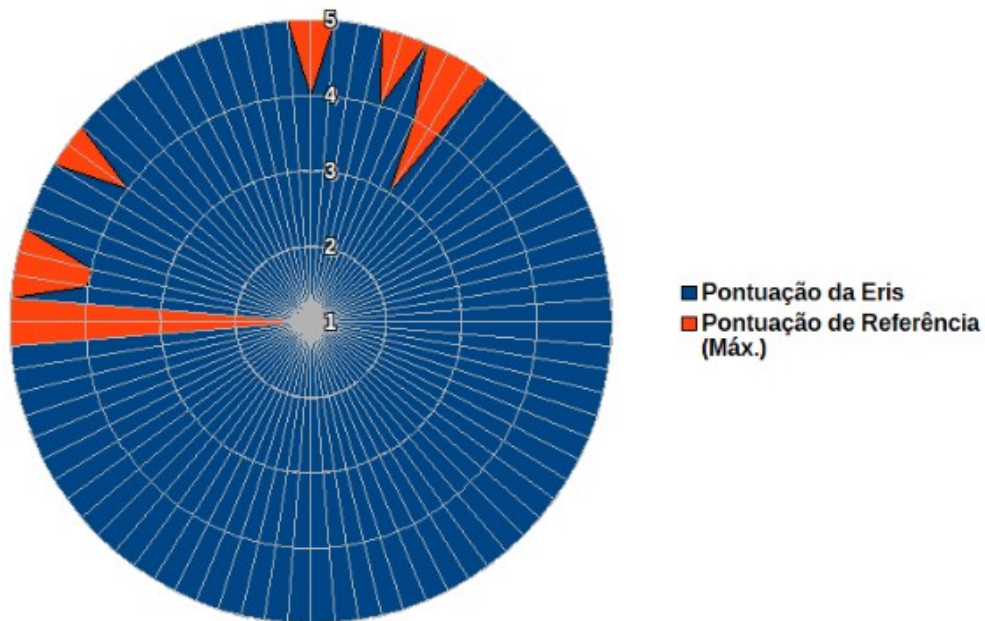
Assim como pode ser constatado nos gráficos das Figuras 21 e 22, os valores obtidos pela Eris na categoria PGD são superiores aos da categoria Remoção. Eris obteve uma pontuação média de 4,85 e apresentou 97,50% de respostas com qualidade de nível a partir do satisfatório, dos quais 90% possuem o mesmo nível de qualidade de uma resposta dada por um especialista humano.

Tabela 7 – Análise estatística da pontuação das respostas geradas pela Eris (PGD)

| TIPO DE ANÁLISE ESTATÍSTICA    | RESPOSTAS GERADAS PELA ERIS |
|--------------------------------|-----------------------------|
| Pontuação média                | 4,85                        |
| Variância da pontuação         | 0,306329113924051           |
| Desvio padrão da pontuação     | 0,553470065969291           |
| Pontuação mínima da amostragem | 1                           |

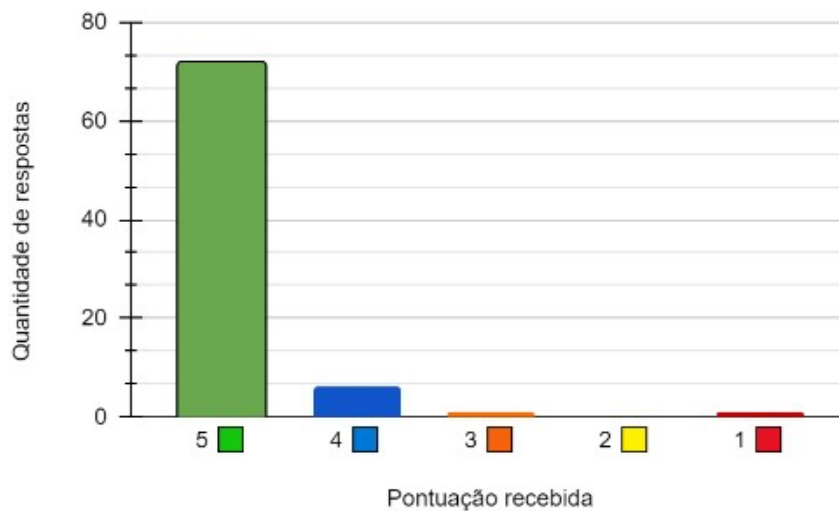
Fonte: autor (2023)

Figura 21 – Comparação entre a pontuação da Eris (PGD) e a pontuação de referência (máxima qualidade)



Fonte: autor (2023)

Figura 22 – Respostas agrupadas pela pontuação recebida (PGD)



Fonte: autor (2023)

#### 4.1.3 Qualidade das respostas das perguntas sobre PDP

No tema PDP a assistente virtual Eris foi testada com um total de 66 perguntas, que foram retiradas de uma lista de perguntas frequentes sobre Política Nacional de Desenvolvimento de Pessoas (PNDP), disponibilizada pelo Governo Federal. Como nessa lista existe uma quantidade

substancial de dúvidas, não houve necessidade de gerar variações para uma mesma pergunta. Isso significa que cada uma das 66 perguntas são de fato dúvidas diferentes. A Figura 23 apresenta um gráfico do tipo radar que compara a qualidade das respostas geradas pela Eris com as respostas de referência existentes na lista. Nessa categoria de perguntas, a Eris apresentou uma performance superior ao comparada com a categoria de perguntas sobre Remoção, mas um pouco inferior comparada com a categoria de perguntas sobre PGD.

Nessa categoria de perguntas, 86,36% das respostas geradas pela Eris foram tão boas quanto as respostas de referência (pontuação 5), 9,09% foram de respostas satisfatórias (pontuação 4), 3,03% foram de respostas aceitáveis (pontuação 3) e 1,52% de respostas insatisfatórias e respostas padrão (pontuação 2 e 1, respectivamente). A Tabela 8 apresenta a análise estatística das pontuações das 66 respostas geradas, com o valor médio da pontuação, a variância entre os valores de pontuação de cada resposta, o desvio padrão da pontuação e a menor pontuação recebida.

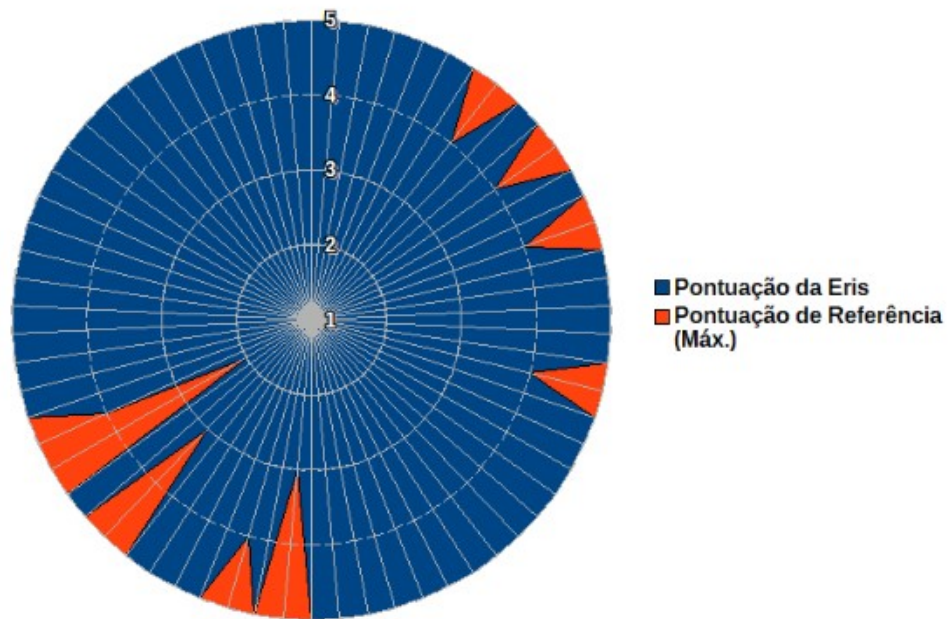
Os gráficos das Figuras 23 e 24 apontam que os valores obtidos pela Eris na categoria PDP são superiores aos da categoria Remoção, mas ficam um pouco abaixo quando comparado com a categoria PGD. Eris obteve uma pontuação média de 4,80 e apresentou 95,45% de respostas com qualidade de nível a partir do satisfatório, dos quais 86,36% possuem o mesmo nível de qualidade de uma resposta dada por um especialista humano.

Tabela 8 – Análise estatística da pontuação das respostas geradas pela Eris (PDP)

| <b>TIPO DE ANÁLISE ESTATÍSTICA</b> | <b>RESPOSTAS GERADAS PELA ERIS</b> |
|------------------------------------|------------------------------------|
| Pontuação média                    | 4,8030303030303                    |
| Variância da pontuação             | 0,314452214452215                  |
| Desvio padrão da pontuação         | 0,560760389517853                  |
| Pontuação mínima da amostragem     | 2                                  |

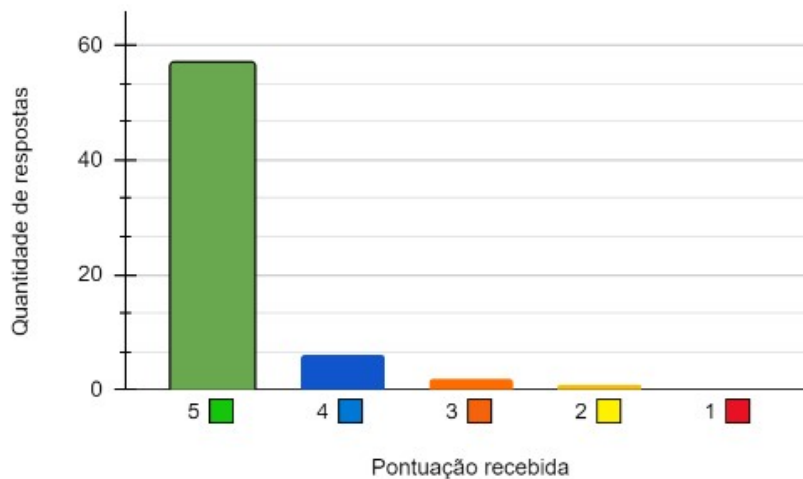
Fonte: autor (2023)

Figura 23 – Comparação entre a pontuação da Eris (PDP) e a pontuação de referência (máxima qualidade)



Fonte: autor (2023)

Figura 24 – Respostas agrupadas pela pontuação recebida (PDP)



Fonte: autor (2023)

#### 4.1.4 Qualidade geral das respostas geradas

Na análise geral das 246 perguntas, 76,02% das respostas geradas pela Eris foram tão boas quanto as respostas de referência (pontuação 5), 14,63% foram de respostas satisfatórias (pontuação 4), 4,07% foram de respostas aceitáveis (pontuação 3) e 5,28% de respostas insatisfatórias e respostas padrão (pontuação 2 e 1, respectivamente). A Tabela 9 apresenta a

análise estatística das pontuações das 246 respostas geradas, com o valor médio da pontuação, a variância entre os valores de pontuação de cada resposta, o desvio padrão da pontuação e a menor pontuação recebida.

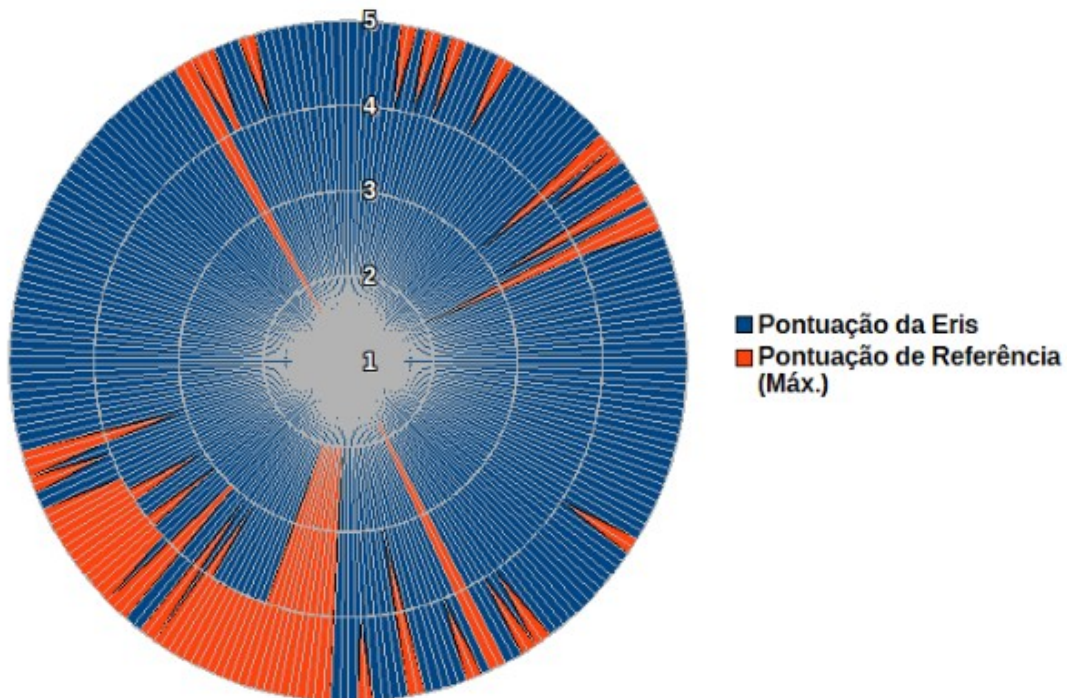
Os gráficos das Figuras 25 e 26 mostram que os valores gerais obtidos pela Eris no que se refere a qualidade das respostas geradas são satisfatórios. Eris obteve uma pontuação média de 4,60 e apresentou 90,65% de respostas com qualidade de nível a partir do satisfatório, dos quais 76,02% possuem o mesmo nível de qualidade de uma resposta dada por um especialista humano.

Tabela 9 – Análise estatística da pontuação das respostas geradas pela Eris(Geral)

| TIPO DE ANÁLISE ESTATÍSTICA    | RESPOSTAS GERADAS PELA ERIS |
|--------------------------------|-----------------------------|
| Pontuação média                | 4,60569105691057            |
| Variância da pontuação         | 0,688783806205413           |
| Desvio padrão da pontuação     | 0,82993000078646            |
| Pontuação mínima da amostragem | 1                           |

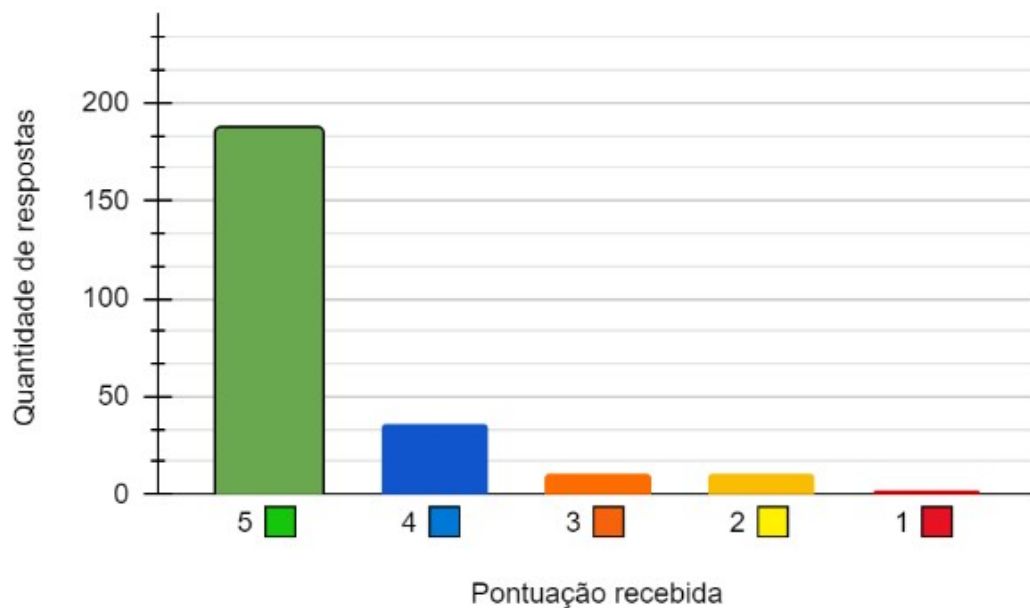
Fonte: autor (2023)

Figura 25 – Comparação entre a pontuação da Eris (Geral) e a pontuação de referência (máxima qualidade)



Fonte: autor (2023)

Figura 26 – Respostas agrupadas pela pontuação recebida (Geral)



Fonte: autor (2023)

## 4.2 Tempo de Geração de Resposta

Conforme descrito no Capítulo 3, para avaliar o tempo de geração das respostas foi estipulado um limite máximo de 10 segundos. Desta forma, para cada um das perguntas feitas à Eris foi realizado o registro do tempo necessário para geração da resposta. As respostas que foram geradas dentro do limite estipulado, ou seja, 10 segundos, estão com o tempo de geração de resposta satisfatório. Por outro lado, as respostas que levaram mais de 10 segundos para serem exibidas para o usuário estão com o tempo de geração de resposta insatisfatório.

Como neste projeto foi utilizado um LLM que recebe requisições e envia respostas por uma API via internet, é importante destacar que o tempo de geração de resposta é fortemente afetado pela velocidade do *link* de internet utilizado. Todos os testes de tempo de geração de respostas foram realizados em um *link* de internet que possui 500MB de *download* e 100MB de *upload*. Assim, considerando que a reitoria do IFCE possui um *link* de internet com a Rede Nacional de Pesquisa (RNP) com velocidade de 10GB, os valores alcançados nesses testes podem ser até melhores com Eris hospedada na infraestrutura do IFCE.

Durante os testes realizados, foi observado que o tempo de geração das respostas pela Eris era influenciado não apenas pelo comprimento das respostas, mas também pela complexidade da forma como as perguntas eram formuladas. Isso significa que a estrutura e o estilo da pergunta

desempenham um papel crucial no processo de geração de respostas. Um exemplo ilustrativo desse fenômeno é a comparação entre duas perguntas: "Vai ter edital de remoção esse ano?" e "É verdade que o edital de remoção não será publicado nesse ano?". Embora ambas as perguntas tratem da mesma dúvida, a segunda pergunta é formulada de uma maneira menos convencional e mais complexa. A presença de negação e a estrutura da frase tornam a segunda pergunta mais desafiadora para o entendimento do modelo de linguagem, o que pode resultar em um tempo de geração de resposta maior quando comparado à primeira pergunta, que é mais direta e simples. Isso ressalta a importância não apenas do conteúdo da pergunta, mas também de sua estrutura e complexidade na interação com a assistente virtual.

#### 4.2.1 Tempo de geração de respostas das perguntas sobre Remoção

Na Figura 27 é apresentado um gráfico de linhas, onde o eixo vertical representa o tempo em segundos e o eixo horizontal é a representação do tempo de geração de resposta de cada uma das 100 respostas. Nesse gráfico é possível verificar que das 100 respostas geradas, 93% apresentaram um tempo de geração de resposta satisfatório, ficando dentro do limite de 10 segundos. Somente 7% das respostas geradas ultrapassaram o limite de tempo de espera aceitável.

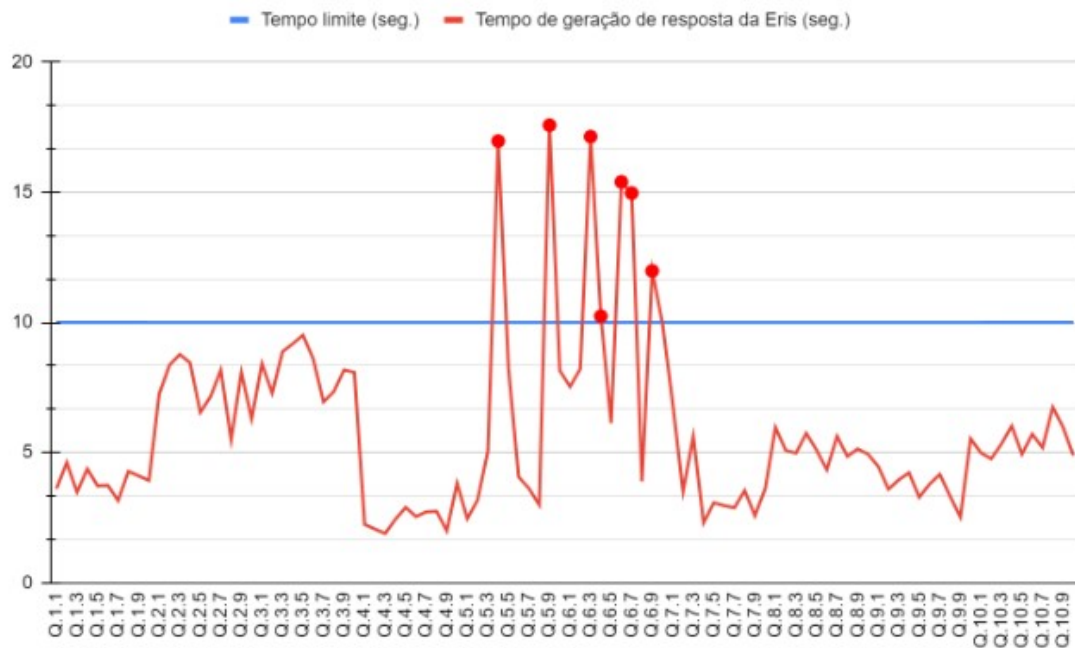
A partir das perguntas frequentes, foi utilizada a ferramenta *Tokenizer* para contabilizar a quantidade de tokens processados por pergunta. Na Tabela 10 é mostrado que para essa categoria de perguntas, o tempo médio de geração de resposta foi de 5,81 segundos, com a resposta mais rápida apresentada no tempo de 1,89 segundos, a mais lenta no tempo de 17,58 segundos e uma média de 376,94 *tokens* processados por segundo.

Tabela 10 – Análise do tempo de geração das respostas da Eris (Remoção)

| TIPO DE ANÁLISE ESTATÍSTICA                    | RESULTADO DAS RESPOSTAS GERADAS PELA ERIS |
|------------------------------------------------|-------------------------------------------|
| Tempo médio de geração de resposta             | 5,8109s                                   |
| Média de <i>tokens</i> por pergunta            | 2190,4 <i>tokens</i>                      |
| Média de <i>tokens</i> processados por segundo | 376,9467724 ( <i>tokens/s</i> )           |
| Menor tempo registrado                         | 1,89s                                     |
| Maior tempo registrado                         | 17,58s                                    |

Fonte: autor (2023)

Figura 27 – Tempo de geração de respostas (Remoção)



Fonte: autor (2023)

#### 4.2.2 Tempo de geração de respostas das perguntas sobre (PGD)

Na categoria de perguntas sobre o tema PGD é apresentado a Figura 28, que ilustra um gráfico de linhas, onde o eixo vertical representa o tempo em segundos e o eixo horizontal é a representação do tempo de geração de resposta de cada uma das 80 respostas. Nesse gráfico é possível verificar que das 80 respostas geradas, 93,75% apresentaram um tempo de geração de resposta satisfatório. Apenas 6,25% das respostas geradas ultrapassaram o limite de tempo de espera aceitável. Ao comparar com o desempenho de tempo de geração de resposta da categoria de perguntas sobre remoção, constatamos que ambas alcançaram uma performance muito similar.

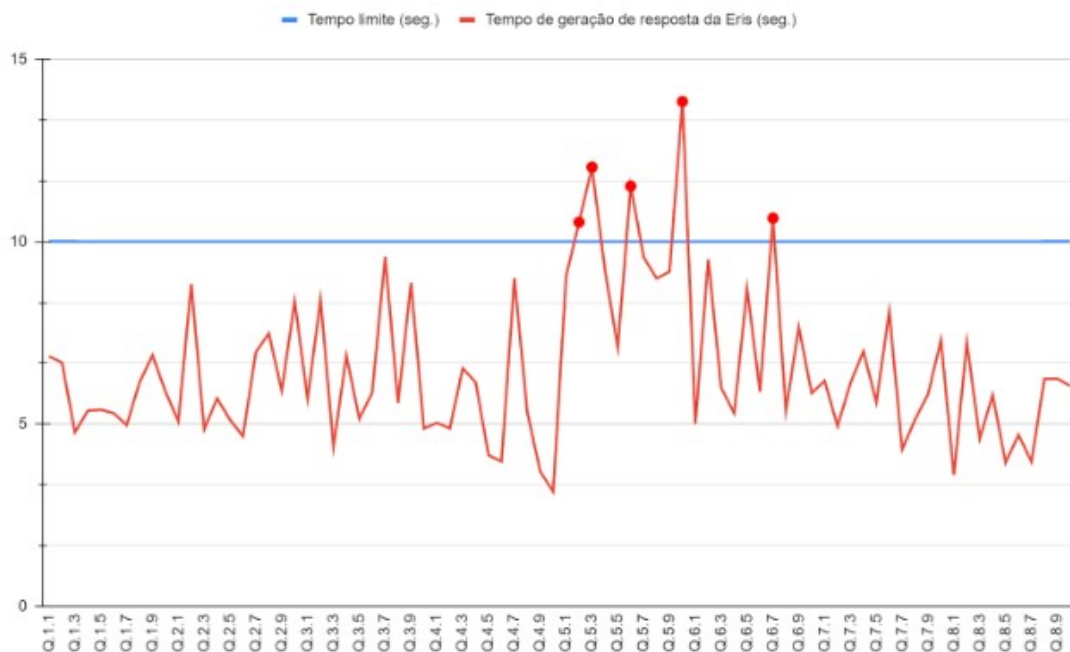
Novamente foi utilizada a ferramenta *Tokenizer* para contabilizar a quantidade de tokens processados a partir das perguntas frequentes. Na Tabela 11 é mostrado que o tempo médio de geração de resposta foi de 6,52 segundos, com a resposta mais rápida apresentada no tempo de 3,14 segundos, a mais lenta no tempo de 13,84 segundos e uma média de 349,22 *tokens* processados por segundo.

Tabela 11 – Análise do tempo de geração das respostas da Eris(PGD)

| TIPO DE ANÁLISE ESTATÍSTICA                    | RESULTADO DAS RESPOSTAS GERADAS PELA ERIS |
|------------------------------------------------|-------------------------------------------|
| Tempo médio de geração de resposta             | 6,523s                                    |
| Média de <i>tokens</i> por pergunta            | 2278 <i>tokens</i>                        |
| Média de <i>tokens</i> processados por segundo | 349,2258163 ( <i>tokens/s</i> )           |
| Menor tempo registrado                         | 3,14s                                     |
| Maior tempo registrado                         | 13,84s                                    |

Fonte: autor (2023)

Figura 28 – Tempo de geração de respostas (PGD)



Fonte: autor (2023)

#### 4.2.3 Tempo de geração de respostas das perguntas sobre (PDP)

A Figura 29 ilustra um gráfico de linhas para a categoria de perguntas sobre o tema PDP, onde o eixo vertical representa o tempo em segundos e o eixo horizontal é a representação do tempo de geração de resposta de cada uma das 66 respostas. Nesse gráfico é possível verificar que das 66 respostas geradas, 95,45% apresentaram um tempo de geração de resposta satisfatório. Apenas 4,54% das respostas geradas ultrapassaram o limite de tempo de espera aceitável. Performance similar a apresentada nas respostas geradas sobre Remoção e PGD.

Assim como nas outras categorias, foi utilizada a ferramenta *Tokenizer* para contabilizar a quantidade de tokens processados a partir das perguntas frequentes. Na Tabela 12 é mostrado

que o tempo médio de geração de resposta foi de 6,51 segundos, com a resposta mais rápida apresentada no tempo de 2,86 segundos, a mais lenta no tempo de 15,52 segundos e uma média de 352,86 *tokens* processados por segundo.

Tabela 12 – Análise do tempo de geração das respostas da Eris (PDP)

| TIPO DE ANÁLISE ESTATÍSTICA                    | RESULTADO DAS RESPOSTAS GERADAS PELA ERIS |
|------------------------------------------------|-------------------------------------------|
| Tempo médio de geração de resposta             | 6,51212121212121s                         |
| Média de <i>tokens</i> por pergunta            | 2297,896552 <i>tokens</i>                 |
| Média de <i>tokens</i> processados por segundo | 352,864524 ( <i>tokens/s</i> )            |
| Menor tempo registrado                         | 2,86s                                     |
| Maior tempo registrado                         | 15,62s                                    |

Fonte: autor (2023)

Figura 29 – Tempo de geração de respostas (PDP)



Fonte: autor (2023)

#### 4.2.4 Tempo de geração de respostas geral

Levando em conta a análise geral do tempo de geração de respostas das 246 perguntas, a Eris apresentou um ótimo resultado. Na Figura 30 há um gráfico de linhas, onde o eixo vertical representa o tempo em segundos e o eixo horizontal é a representação do tempo de geração de cada uma das 246 respostas. Nesse gráfico é possível verificar que das 246 respostas geradas,

somente 15 ultrapassaram o tempo limite estipulado de 10 segundos. Isso significa que 93,90% das respostas apresentaram um tempo de geração de resposta satisfatório.

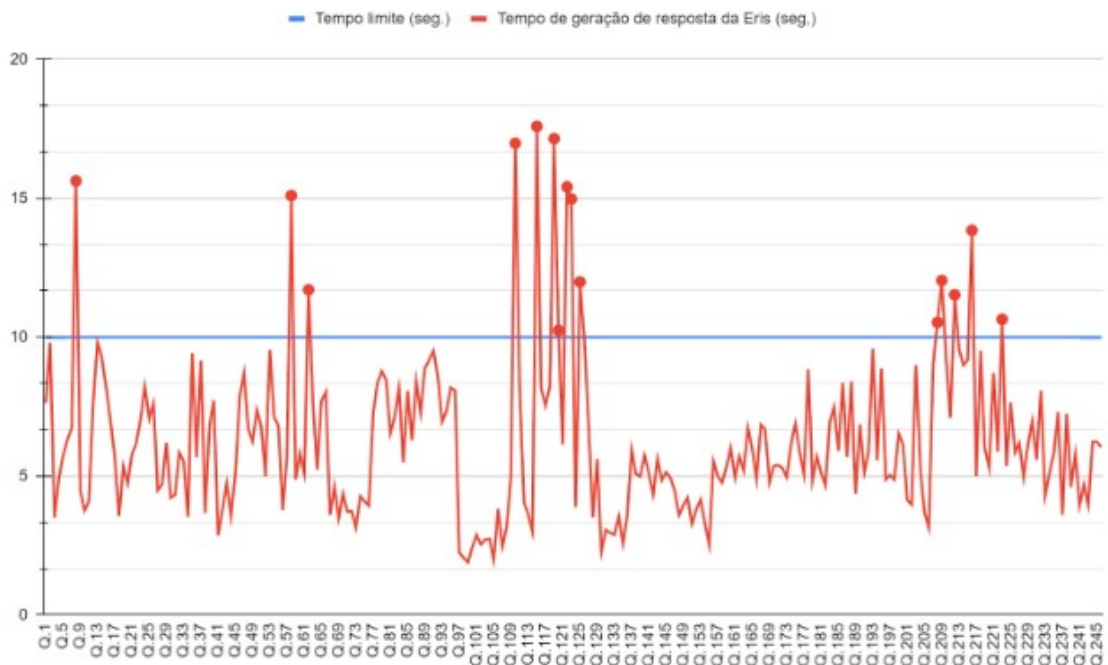
Com o uso da ferramenta *Tokenizer* foi contabilizado a quantidade de tokens processados a partir das perguntas frequentes. Na Tabela 13 é mostrado que o tempo médio de geração de resposta foi de 6,23 segundos, com a resposta mais rápida apresentada no tempo de 1,89 segundos, a mais lenta no tempo de 17,58 segundos e uma média de 364,63 *tokens* processados por segundo.

Tabela 13 – Análise geral do tempo de geração das respostas da Eris

| TIPO DE ANÁLISE ESTATÍSTICA                    | RESULTADO DAS RESPOSTAS GERADAS PELA ERIS |
|------------------------------------------------|-------------------------------------------|
| Tempo médio de geração de resposta             | 6,23060975609756s                         |
| Média de <i>tokens</i> por pergunta            | 2271,897959 <i>tokens</i>                 |
| Média de <i>tokens</i> processados por segundo | 364,6349311 ( <i>tokens/s</i> )           |
| Menor tempo registrado                         | 1,89s                                     |
| Maior tempo registrado                         | 17,58s                                    |

Fonte: autor (2023)

Figura 30 – Tempo de geração de respostas (Análise geral)



Fonte: autor (2023)

### 4.3 Custo Médio por Geração de Resposta

Para realizar o cálculo do custo médio por geração de resposta foi necessário a definição da quantidade média de *tokens* por pergunta. Assim como explicado no Capítulo 3, neste projeto apenas os *tokens* de orientação possuem valor fixo, todos os demais *tokens* variam de acordo com a pergunta feita. Logo, é necessário a definição da quantidade média de *tokens* utilizados na pergunta, nas informações relevantes recuperadas pelo *LangChain* e na resposta.

Do total das 246 perguntas utilizadas para testar a assistente virtual Eris, foram selecionadas aleatoriamente 20% dessas perguntas e suas respectivas respostas. Utilizando a ferramenta *Tokenizer* todos os *tokens* das 49 perguntas e respostas selecionadas foram contabilizados. Com essas informações foi possível determinar o valor médio de *tokens* e posteriormente utilizar esses valores na Equação 3.1.

Vale ressaltar que, uma vez definida a quantidade média de *tokens* por resposta, o total de *tokens* por resposta deve ser multiplicado pelo valor tabelado da *OpenAI* para o modelo de linguagem utilizado. No caso da Eris, o modelo utilizado foi o *text-davinci-003*, que possui um custo, em real, de 10 centavos por 1000 *tokens* processados.

A Tabela 14 mostra a quantidade média de *tokens* obtidos a partir dos tokens que foram contabilizados das 49 perguntas e respostas selecionadas.

Tabela 14 – Quantidade de média de *tokens* contabilizados utilizando a ferramenta *Tokenizer*

| <b>TOKENS DE ORIENTAÇÃO</b> | <b>TOKENS POR PERGUNTA</b> | <b>TOKENS POR INF. RELEVANTE</b> | <b>TOKENS POR RESPOSTA</b> |
|-----------------------------|----------------------------|----------------------------------|----------------------------|
| 446                         | 39,51020408                | 1565,285714                      | 221,1020408                |

Fonte: autor (2023)

Após análise do quantitativo de *tokens*, foi obtido um valor médio de **2271,89 *tokens*** por pergunta e resposta. Esse valor é obtido fazendo a soma de todos os valores médios de *tokens* de entrada: pergunta, orientação e informações relevantes; e saída: resposta. A partir desse valor, para se chegar no custo médio por pergunta, é necessário multiplicar pelo custo tabelado de utilização do modelo *text-davinci-003*, que é de 0,02 centavos de dólar, ou 10 centavos de real. Sendo assim, o custo médio, em real, por geração de resposta da Eris é de aproximadamente de R\$ 0,2271, ou seja, 23 centavos.

## 5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento e avaliação de desempenho da Eris, uma assistente virtual desenvolvida para a PROGEP do IFCE com o objetivo responder às dúvidas dos servidores. A avaliação de desempenho levou em consideração três aspectos principais: qualidade das respostas geradas, tempo de geração das respostas e custo médio por geração de resposta. Os resultados obtidos foram extremamente promissores, demonstrando que a Eris é uma ferramenta eficiente, com uma sólida relação de custo-benefício.

A qualidade das respostas geradas pela Eris foi medida a partir de uma avaliação humana, que utilizou como padrão de comparação de qualidade as respostas dadas por especialistas humanos. Eris demonstrou em suas respostas precisão nas informações fornecidas e clareza na linguagem utilizada. Os resultados mostraram que 90,65% das respostas da Eris foram classificadas como de qualidade satisfatória ou superior. Destaca-se também que 76,02% das respostas foram consideradas de qualidade equivalente à de uma resposta dada por um especialista humano. Esses resultados indicam que a Eris é capaz de fornecer respostas de alta qualidade, tornando-a uma ferramenta valiosa para atendimento ao público da PROGEP.

No que se refere ao tempo de geração de resposta, a Eris demonstrou um desempenho muito satisfatório. O tempo médio de geração de resposta foi de 6,23 segundos, ficando dentro do limite máximo de 10 segundos de espera estipulado neste estudo. Além disso, a Eris foi capaz de processar uma média de 364,63 *tokens* por segundo, demonstrando sua capacidade de lidar com perguntas e respostas extensas.

Por fim, após análise de 20% das perguntas utilizadas para testar o desempenho da Eris, foi possível calcular o custo médio por geração de resposta, onde o valor alcançado foi de R\$ 0,2271 por resposta, aproximadamente 23 centavos. A partir desse valor e de uma estimativa média de perguntas realizadas mensalmente, é possível definir o custo mensal de utilização da nova ferramenta desenvolvida.

Esses resultados indicam que a Eris é uma assistente virtual que está pronta para ser utilizada em um cenário real, estando apta a ser colocada em produção. Ela não apenas fornece respostas de alta qualidade, mas também o faz de maneira rápida e eficiente. Isso sugere que a Eris tem o potencial de ser uma ferramenta valiosa para a PROGEP, economizando tempo e esforço na pesquisa e formulação de respostas.

Vale ressaltar que a utilização de uma assistente virtual proporciona um acesso rápido e ininterrupto a informações essenciais, eliminando a necessidade de aguardar por respostas ou consultas demoradas. Além disso, promove a padronização das informações, garantindo que todos os servidores recebam orientações consistentes. Outro ponto positivo é que essa abordagem também alivia a carga de trabalho dos profissionais de Gestão de Pessoas, permitindo que se concentrem em tarefas mais estratégicas, ao passo que a assistente virtual lida com perguntas rotineiras.

Durante este estudo identificaram-se duas limitações importantes que merecem destaque. Primeiramente, a Eris requer uma conexão com a internet para operar, visto que utiliza o LLM GPT-3.5, que obtém respostas por meio de sua API online. Isso significa que a disponibilidade e a qualidade do serviço da Eris estão diretamente dependentes da conectividade à internet, o que pode representar um desafio em situações de conectividade intermitente ou ausente. Além disso, a privacidade das informações é uma preocupação, uma vez que todas as consultas e interações com a Eris envolvem o envio de dados para a *OpenAI* através da internet, incluindo todas as informações contidas na base de conhecimento. Isso exige um cuidado especial na gestão da privacidade e na conformidade com regulamentações de proteção de dados.

No caso deste projeto, como as informações utilizadas são de domínio público, a questão da privacidade não foi um problema, entretanto, dependendo da situação, essa pode ser uma questão de extrema atenção. As duas limitações apresentadas podem ser contornadas utilizando um LLM que permita ser executado localmente, o que é completamente viável, uma vez que o arquétipo apresentado neste trabalho permite a utilização de qualquer tipo de LLM. Porém, é importante considerar que para executar um LLM localmente é necessário acesso a *hardwares* específicos e robustos, o que pode trazer altos custos ao projeto.

Ao analisar a implementação da assistente virtual Eris no contexto do suporte da PROGEP ao servidor do IFCE, torna-se evidente a natureza dual de sua inovação. Por um lado, a introdução da Eris pode ser classificada como uma inovação sustentável, uma vez que aprimora um serviço já existente, o suporte ao servidor. Ao integrar a assistente virtual neste ambiente, busca-se a otimização contínua do atendimento ao servidor, proporcionando melhorias graduais e eficiência operacional. Por outro lado, a inovação também se revela radical, visto que a Eris habilita um novo paradigma de suporte ininterrupto, operando 24 horas por dia e 7 dias por semana. Esta capacidade, impulsionada pelo uso da IA, representa uma mudança substancial em relação

ao padrão anterior de atendimento humano, abrindo portas para a criação de novos serviços e redefinindo as expectativas quanto à disponibilidade e eficácia do suporte prestado pela PROGEP ao servidor.

Como trabalhos futuros, podem-se listar as seguintes sugestões de atividades:

- Implementar a assistente virtual utilizando outros LLMs, comparando os resultados de desempenho entre os modelos de linguagem utilizados, os benefícios e limitações de utilização de cada tipo de LLM testado;
- Adicionar novas camadas ao arquétipo de assistente virtual apresentado, como uma camada de memória cache para as respostas geradas. Desta forma, as respostas de perguntas já feitas poderiam ser recuperadas mais rapidamente sem a necessidade do LLM gerar novamente uma resposta, ou seja, sem custo adicional. Comparar o desempenho no que se refere a tempo de geração de resposta e custo por geração de resposta com a implementação das novas camadas à assistente virtual;
- Adicionar novas métricas ao avaliar o desempenho da assistente virtual, tais como métricas de engajamento que avaliam o envolvimento dos usuários com a assistente, incluindo a frequência de uso e o tempo gasto nas interações;
- Criação de base de dados para realização de refinamento (*fine tuning*) de LLMs em atividades específicas, tais como: base de dados para refinamento em atendimento à suporte técnico de TIC, base de dados para refinamento em atendimento à usuários de bibliotecas, base de dados para refinamento de atendimento à usuários do sistema SUAP;
- Comparar o desempenho do arquétipo apresentado neste trabalho com o desempenho de uma assistente virtual implementada a partir da utilização de modelos de linguagem refinados para casos específicos, ou seja, LLMs que foram treinados em base de dados específicas (*fine tuning*).
- Implementar o arquétipo apresentado neste trabalho como forma de promover o *Mobile Learning*, por meio do desenvolvimento de recursos interativos e engajadores, como *quizzes* e *microlearning*, para viabilizar uma experiência de aprendizado mais dinâmica e personalizada.
- Expansão das funcionalidades da Eris para oferecer suporte não apenas as dúvidas dos servidores sobre assuntos da PROGEP, mas também fornecer informações relevantes, orientações e recursos para professores, pesquisadores e colaboradores envolvidos nas atividades de ensino, pesquisa e projetos de extensão.

## REFERÊNCIAS

- ALHARBI, F.; VAKANSKI, A. Machine learning methods for cancer classification using gene expression data: A review. **Bioengineering**, v. 10, n. 2, 2023. ISSN 2306-5354.
- ALI, M. et al. **Tokenizer Choice For LLM Training: Negligible or Crucial?** 2023.
- ARIF, T. M. **Introduction to Deep Learning for Engineers**. Cham: Springer Cham, 2020. ISBN 978-3-031-79665-4.
- BANERJEE, S.; LAVIE, A. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: GOLDSTEIN, J. et al. (Ed.). **Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization**. Ann Arbor, Michigan: Association for Computational Linguistics, 2005. p. 65–72.
- BARBIERO, P.; SQUILLERO, G.; TONDA, A. **Modeling Generalization in Machine Learning: A Methodological and Computational Study**. 2020.
- BENDER, E. M. et al. On the dangers of stochastic parrots: Can language models be too big? In: **Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency**. New York, NY, USA: Association for Computing Machinery, 2021. (FAccT '21), p. 610–623. ISBN 9781450383097.
- BRASIL. Ministério da gestão e da inovação em serviços públicos. **Dúvidas**, 2020. Disponível em: <<https://www.gov.br/servidor/pt-br/acao-a-informacao/gestao-de-pessoas/pndp/paginas/duvidas>>. Acesso em: 23 set. 2023.
- BROWN, T. B. et al. **Language Models are Few-Shot Learners**. 2020.
- CAMPOS, S. L. B.; FIGUEIREDO, J. M. d. Aplicação de inteligência artificial no ciclo de políticas públicas. **Cadernos de Prospecção**, v. 15, n. 1, p. 196–214, jan. 2022.
- CHEN, X. et al. **Symbolic Discovery of Optimization Algorithms**. 2023.
- CLARK, P. et al. **Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge**. 2018.
- DAHIYA, M. A tool of conversation: Chatbot. **International Journal of Computer Sciences and Engineering**, v. 5, p. 158–161, 05 2017.
- DAYTON, F. Vector similarity search: A deeper dive. **CS168 Final Report**, 2023. Disponível em: <[https://www.finndayton.com/CS168\\_Final\\_Report.pdf](https://www.finndayton.com/CS168_Final_Report.pdf)>. Acesso em: 11 dez. 2023.
- DIBITONTO, M. et al. Chatbot in a campus environment: Design of lisa, a virtual assistant to help students in their university life. In: KUROSU, M. (Ed.). **Human-Computer Interaction. Interaction Technologies**. Cham: Springer International Publishing, 2018. p. 103–116. ISBN 978-3-319-91250-9.
- DOMINTE, I. Documenting the api. In: \_\_\_\_\_. **Web API Development for the Absolute Beginner**. [S.l.]: Apress, 2023. ISBN 978-1-4842-9348-5.
- GAO, L. et al. **A framework for few-shot language model evaluation**. Zenodo, 2021. Disponível em: <<https://doi.org/10.5281/zenodo.5371628>>.

- GIL, A. C. **Como Elaborar Projetos de Pesquisas**. 7. ed. São Paulo: Atlas, 2022.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.
- GUO, B. et al. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. 2023.
- HENDRYCKS, D. et al. **Measuring Massive Multitask Language Understanding**. 2021.
- IMAN, M.; ARABNIA, H. R.; RASHEED, K. A review of deep transfer learning and recent advancements. **Technologies**, v. 11, n. 2, 2023. ISSN 2227-7080.
- INCITTI, F.; URLI, F.; SNIDARO, L. Beyond word embeddings: A survey. **Information Fusion**, v. 89, p. 418–436, 2023. ISSN 1566-2535.
- JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. **Journal of Documentation**, v. 28, n. 1, p. 11–21, 01 1972.
- JURAFSKY, D.; MARTIN, J. **Speech and Language Processing An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. [S.l.: s.n.], 2020.
- KASNECI, E. et al. Chatgpt for good? on opportunities and challenges of large language models for education. **Learning and Individual Differences**, v. 103, p. 102274, 2023. ISSN 1041-6080.
- KOIVISTO, M. Tutoring postgraduate students with an ai-based chatbot. **International Journal of Advanced Corporate Learning (iJAC)**, v. 16, p. 41–54, 03 2023.
- KUFEL, J. et al. What is machine learning, artificial neural networks and deep learning? examples of practical applications in medicine. **Diagnostics**, v. 13, n. 15, 2023. ISSN 2075-4418.
- LESTER, B.; AL-RFOU, R.; CONSTANT, N. **The Power of Scale for Parameter-Efficient Prompt Tuning**. 2021.
- LI, S.; CHEN, C.-H. The effects of visual feedback designs on long wait time of mobile application user interface. **Interacting with Computers**, v. 31, n. 1, p. 1–12, 03 2019. ISSN 0953-5438.
- LIN, S.; HILTON, J.; EVANS, O. **TruthfulQA: Measuring How Models Mimic Human Falsehoods**. 2022.
- LIU, L.; LIU, J.; HAN, J. **Multi-head or Single-head? An Empirical Comparison for Transformer Training**. 2021.
- MAROENGSI, W. et al. A survey on evaluation methods for chatbots. In: **Proceedings of the 2019 7th International Conference on Information and Education Technology**. New York, NY, USA: Association for Computing Machinery, 2019. (ICIET 2019), p. 111–119. ISBN 9781450366397.
- MIKOLOV, T. et al. **Efficient Estimation of Word Representations in Vector Space**. 2013.
- MITCHELL, M.; KRAKAUER, D. C. The debate over understanding in ai's large language models. **Proceedings of the National Academy of Sciences**, Proceedings of the National Academy of Sciences, v. 120, n. 13, mar 2023. ISSN 1091-6490.

MURUGAN, S. et al. Autonomous vehicle assisted by heads up display (hud) with augmented reality based on machine learning techniques. In: \_\_\_\_\_. **Virtual and Augmented Reality for Automobile Industry: Innovation Vision and Applications**. Cham: Springer International Publishing, 2022. p. 45–64. ISBN 978-3-030-94102-4.

NAH, F. A study on tolerable waiting time: How long are web users willing to wait? In: . [S.l.: s.n.], 2003. v. 23, p. 285.

NAWAZ, M. et al. A study on application programming interface recommendation: state-of-the-art techniques, challenges and future directions. **Library Hi Tech**, v. 41, 08 2022.

OECD; EUROSTAT. **Oslo Manual 2018**. [s.n.], 2018. 256 p. Disponível em: <<https://www.oecd-ilibrary.org/content/publication/9789264304604-en>>. Acesso em: 21 out. 2023.

OMERAGIĆ, D. et al. The employment of a machine learning-based recommendation system to maximize netflix user satisfaction. In: ADEMOVIĆ, N.; KEVRIĆ, J.; AKŠAMIJA, Z. (Ed.). **Advanced Technologies, Systems, and Applications VIII**. Cham: Springer Nature Switzerland, 2023. p. 300–328. ISBN 978-3-031-43056-5.

OPENAI. **Tokenizer: Learn about language model tokenization**. OpenAI, 2022. Disponível em: <<https://platform.openai.com/tokenizer>>. Acesso em: 15 maio 2023.

PAPINENI, K. et al. Bleu: A method for automatic evaluation of machine translation. In: **Proceedings of the 40th Annual Meeting on Association for Computational Linguistics**. USA: Association for Computational Linguistics, 2002. (ACL '02), p. 311–318.

PARREIRAS, M. et al. Inteligência artificial aplicada para o aumento da produtividade no atendimento de intimações. In: **Anais do X Workshop de Computação Aplicada em Governo Eletrônico**. Porto Alegre, RS, Brasil: SBC, 2022. p. 180–191. ISSN 2763-8723.

PATIL, S.; LOKESHA, V. Live twitter sentiment analysis using streamlit framework. In: **Proceedings of the International Conference on Innovative Computing and Communication**. [S.l.: s.n.], 2022. (ICICC 2022).

PATTANAYAK, S. Introduction to deep-learning concepts and tensorflow. In: \_\_\_\_\_. **Pro Deep Learning with TensorFlow 2.0: A Mathematical Approach to Advanced Artificial Intelligence in Python**. Berkeley, CA: Apress, 2023. p. 109–197. ISBN 978-1-4842-8931-0.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543.

PEREZ-PINO, A. et al. The accuracy of artificial intelligence-based virtual assistants in responding to routinely asked questions about orthodontics. **The Angle orthodontist**, v. 93, n. 4, p. 427–432, 2023.

PESARU, A.; GILL, T. S.; TANGELLA, A. R. Ai assistant for document management using lang chain and pinecone. **International Research Journal of Modernization in Engineering Technology and Science**, v. 41, 06 2023.

RADFORD, A.; NARASIMHAN, K. Improving language understanding by generative pre-training. In: . [s.n.], 2018. Disponível em: <<https://api.semanticscholar.org/CorpusID:49313245>>. Acesso em: 8 jan. 2023.

RADFORD, A. et al. Language models are unsupervised multitask learners. In: . [s.n.], 2019. Disponível em: <<https://api.semanticscholar.org/CorpusID:160025533>>. Acesso em: 8 jan. 2023.

RADZIWILL, N. M.; BENTON, M. C. **Evaluating Quality of Chatbots and Intelligent Conversational Agents**. 2017.

REDDY, S. Evaluating large language models for use in healthcare: A framework for translational value assessment. **Informatics in Medicine Unlocked**, v. 41, p. 101304, 2023. ISSN 2352-9148.

RIBEIRO, R. L. O. et al. Modelo conceitual de inovação e transformação digital. **Revista de Ciências Empresariais da UNIPAR**, v. 24, n. 1, p. 138–161, ago. 2023.

RICCIO, V. et al. Testing machine learning based systems: a systematic mapping. **Empirical Software Engineering**, v. 25, p. 5193–5254, 2020.

RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier, 2013.

SALEM, N.; HUSSEIN, S. Data dimensional reduction and principal components analysis. **Procedia Computer Science**, v. 163, p. 292–299, 2019. ISSN 1877-0509. 16th Learning and Technology Conference 2019 Artificial Intelligence and Machine Learning: Embedding the Intelligence.

SARKAR, T. How to use python data science packages more productively. In: \_\_\_\_\_. **Productive and Efficient Data Science with Python: With Modularizing, Memory profiles, and Parallel/GPU Processing**. Berkeley, CA: Apress, 2022. p. 47–84. ISBN 978-1-4842-8121-5.

SHARIFANI, K.; AMINI, M. Machine learning and deep learning: A review of methods and applications. **World Information Technology and Engineering Journal**, v. 10, p. 3897–3904, 2023.

SHARKAWY, A.-N. Principle of neural network and its main types: Review. **Journal of Advances in Applied Computational Mathematics**, v. 7, p. 8–19, Aug. 2020.

SLECHTEN, K. A deep dive into transformers. **Medium**, 2023. Disponível em: <<https://kristofsl.medium.com/a-deep-dive-into-transformers-45687c374749>>. Acesso em: 3 maio 2023.

SOYGAZI, F.; OGUZ, D. An analysis of large language models and langchain in mathematics education. In: . [S.l.: s.n.], 2023.

THIRUNAVUKARASU, A. J. et al. Large language models in medicine. **Nature Medicine**, v. 29, p. 1930–1940, 2023. ISSN 1041-6080.

TOPSAKAL, O.; AKINCI, T. C. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. **International Conference on Applied Engineering and Natural Sciences**, v. 1, n. 1, p. 1050–1056, Jul. 2023.

TURING, A. Computing machinery and intelligence. **Mind**, LIX, n. 236, p. 433–460, 1950.

VASWANI, A. et al. **Attention Is All You Need**. 2017.

WARDAT, Y. et al. Chatgpt: A revolutionary tool for teaching and learning mathematics. **Eurasia Journal of Mathematics, Science and Technology Education**, v. 19, n. 7, 2023.

ZELLERS, R. et al. **HellaSwag: Can a Machine Really Finish Your Sentence?** 2019.

ZHANG, S. et al. Opt: Open pre-trained transformer language models. 2022.

ZHU, M. et al. A review of the application of machine learning in water quality evaluation. **Eco-Environment and Health**, v. 1, n. 2, p. 107–116, 2022. ISSN 2772-9850.

ZHUANG, F. et al. **A Comprehensive Survey on Transfer Learning**. 2020.