

TCC-Review: Uma Plataforma Integrada para Revisão e Aprimoramento de Artigos de TCC

**Trabalho de Conclusão do Curso de
Tecnologia em Sistemas Para Internet**

**Rodrigo Leite de Oliveira
Orientador(a): Rodrigo Prestes Machado**

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre
Av Cel Vicente, 281, Porto Alegre – RS – Brasil

rodrigo.oliveira1998@outlook.com, rodrigo.prestes@poa.ifrs.edu.br

Resumo. *A escrita é um método de registro do conhecimento e comumente usado como critério avaliativo. Como parte fundamental de seu processo, a revisão é uma etapa que visa o aprimoramento na qualidade da produção, porém pode se mostrar custosa. A fim de oferecer uma ferramenta de auxílio na revisão de Trabalho de Conclusão de Curso cuja escrita já se encontra em um nível avançado de desenvolvimento, buscou-se construir uma plataforma web que combine o uso do Google Docs com serviços ou bibliotecas de análise de texto baseados em inteligência artificial. Após o desenvolvimento, realizou-se um benchmark que validou a capacidade de avaliação da ferramenta.*

1. Introdução

A escrita é uma das formas mais antigas e essenciais de expressão e registro de ideias na história da humanidade, e sua associação com o conhecimento é inegável. Além disso, a redação desempenha um papel central em processos seletivos, como o Exame Nacional do Ensino Médio (ENEM) e vestibulares, onde a qualidade da escrita é avaliada criteriosamente.

Menegassi (1998) elabora um roteiro de etapas no processo de produção textual, que inclui a definição do tema, a leitura do material de apoio, a reflexão e o planejamento antes da escrita propriamente dita. Garcez (1998) define as etapas da produção de texto como um procedimento que envolve o planejamento e a produção, ou seja, a reformulação.

No modelo tradicional de professor e aluno, como descrito por Menegassi (1998), fica evidente a viabilidade da extensão do processo, adicionando mais etapas através da inclusão de rodadas de revisão. No entanto, a realização de um número considerável de repetições de leitura, reflexão e reescrita pode ser um trabalho cansativo devido ao tempo que isso demanda, seja o revisor o próprio redator ou um leitor terceiro, como um professor. Conforme Pinho et al. (2022), a intensificação de atividades de escrita tem gerado uma carga de trabalho adicional para os professores encarregados da correção, com custos crescentes para dar suporte a essas atividades.

Neste artigo, apresenta-se uma solução desenvolvida para atender a uma necessidade premente: um auxílio no processo de redação e revisão de Trabalhos de

Conclusão de Curso (TCCs) no ensino superior. Essa solução é um protótipo de aplicação que integra a tecnologia do *Google Docs* com análise de texto baseada em Inteligência Artificial (IA). Seu foco principal está na demonstração do funcionamento desta plataforma. Explica-se como ela pode ser utilizada para aprimorar a qualidade da escrita e revisão de TCCs, economizando tempo, reduzindo custos e fornecendo avaliações padronizadas e objetivas.

Para atingir esse objetivo, adotou-se uma abordagem de pesquisa exploratória, que visa entender como os estudantes lidam com a escrita de seus TCCs, identificando seus principais obstáculos. Paralelamente, foi conduzida uma pesquisa bibliográfica para examinar os recursos disponíveis relacionados à melhoria da escrita acadêmica.

A metodologia deste estudo inclui a coleta de dados de TCCs produzidos por estudantes universitários. Posteriormente, realizou-se uma análise comparativa abrangente, avaliando a qualidade da escrita, coesão, coerência dos textos, e outros aspectos relevantes. Essa análise permitiu verificar a eficácia do modelo de inteligência artificial selecionado para este propósito. Além disso, este artigo descreve a metodologia usada para avaliar como essa tecnologia atende às necessidades dos graduandos em seus trabalhos de conclusão.

2. Fundamentação teórica

Com base em Pinho et al (2020), a aplicação de técnicas de Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina (AM) tornou-se desejável para lidar com a demanda gerada pelos processos de escrita e reescrita. Essas técnicas, que abrangem análise léxica, morfológica, semântica, sintática e estrutural, visam identificar deficiências e inconsistências em textos educacionais. Do reconhecimento da forma das palavras à interpretação de significados, essa abordagem pode atuar como corretor gramatical e também identificar falhas de coesão, contribuindo para melhorar a qualidade dos textos (BURROWS et al., 2015).

Embora haja métodos estabelecidos e adaptados de forma especializada a critérios específicos de avaliação, como, por exemplo, a redação do ENEM, é importante destacar que esses métodos podem, em algumas situações, apresentar eficácia reduzida ao serem aplicados a critérios diferentes, conforme enfatizado por Bittencourt Júnior (2020):

Assim, tornou-se evidente a lacuna que justifica a implementação de uma plataforma dedicada ao texto científico do Ensino Superior. Isso ocorre porque o uso de plataformas cujo foco principal é a revisão de textos de outras categorias pode oferecer recursos limitados, especialmente quando considera uma série de requisitos que são essenciais para a diferenciação da escrita. Aliado a esta demanda, considerou-se a crescente acessibilidade a recursos de modelos de IA abertos, como os empregados no *ChatGPT*, e nas vantagens em desenvolver uma aplicação capaz de consumir tais recursos.

2.1. Trabalhos/Sistemas relacionados

A partir de uma pesquisa exploratória usando como base de dados a busca do Google Acadêmico e *strings* de busca similares à “texto* OR redaç* AND corrig*” foram localizados pelo menos três trabalhos que se relacionam pelo estudo da aplicação de ferramentas de análise, revisão e correção em prol da qualidade de escrita.

Destes trabalhos, destaca-se o RevisãoOnline (BIGOLIN, 2022), uma plataforma on-line que usa a revisão por pares guiada por recursos de inteligência artificial. As características descritas neste projeto têm como enfoque o desenvolvimento das competências cobradas durante a correção da redação do ENEM. Seu fluxo de utilização envolve o envio de textos próprios e a revisão de terceiros, sendo o uso colaborativo parte do seu funcionamento.

Quadro 1 - Comparativo em relação ao RevisãoOnline

Critérios	RevisãoOnline	TCC-Review
Plataforma web	Sim	Sim
Gratuito	Sim	Sim
Público-alvo (tipo de texto revisado pela plataforma)	Redações do ENEM	Artigos de TCC
Modo de envio do texto	Editor próprio	Integração com API do <i>Google Docs</i>
Método de revisão	Por pares (revisão do texto de terceiros)	Individual (revisão do próprio texto)

Fonte: Elaborado pelo autor (2023).

Relacionados ao referencial bibliográfico de um artigo, estão-se os gerenciadores de referências, os quais Yamakawa *et al* (2014) elaborou um comparativo de softwares que atuam com este propósito. Contudo, seu uso mais comum acontece paralelamente à escrita do artigo, isso é, sendo configurados de modo que o texto esteja em um estado inicial de desenvolvimento. Em cenários em que o pesquisador já possui seu texto em uma etapa avançada ou finalizado e tem como necessidade apenas realizar uma revisão, ou possui um prazo relativamente curto para a execução das etapas de pesquisa, ainda será acrescido à lista de tarefas, a assimilação das funcionalidades do gerenciadores de referências, sendo mais uma responsabilidade a se considerar em um período que muitas vezes se mostra escasso.

Em uma etapa posterior da pesquisa, que ocorreu simultaneamente ao desenvolvimento da aplicação, tomou-se conhecimento da ferramenta chamada *ChatPDF*. Essa ferramenta utiliza inteligência artificial para resumir documentos em formato PDF por meio de uma interface de chatbot. Inicialmente, o *ChatPDF* não havia

sido considerado na pesquisa, pois não havia sido mencionado na literatura acadêmica até que Fernandes (2023) fez uma breve referência a ele. No entanto, a inclusão tardia dessa ferramenta em nosso contexto desempenha um papel fundamental como ponto de referência e comparação para avaliar as características adotadas por nossa aplicação.

Quadro 2 - Comparativo em relação ao *ChatPDF*

Sistema	Formato de documento analisado	Resume o conteúdo do texto	Faz observações quanto à linguagem, estrutura, objetividade e fluidez	Sugere melhorias no texto
<i>TCC Review</i>	Textos do Google Docs	Comenta trechos durante a análise	Fornecer pareceres conforme a solicitação enviada ao modelo de IA	Sugere simplificações e substituições de frases
<i>ChatPDF</i>	Arquivos PDF	Resume e um parágrafo e cria atalhos para os principais tópicos	Apenas se questionado diretamente pela interface de <i>chat</i>	Não

Fonte: Elaborado pelo autor (2023).

Conforme o Quadro 2, observaram-se funcionalidades semelhantes no *ChatPDF*, como o envio de textos finalizados. Com o intuito de realizar a validação do uso do sistema, foram submetidos os mesmos trabalhos com enunciados semelhantes, buscando traçar comparativos entre as respostas fornecidas.

2.2. Tecnologias

Para atender aos requisitos essenciais do projeto, foi fundamental considerar a integração do sistema com duas principais fontes de funcionalidade: a *API* do *Google Docs* e a *API* da *OpenAI*. Estas integrações foram essenciais para cumprir os objetivos e garantir uma experiência eficaz aos usuários.

2.2.1. Integração com a *API* do *Google Docs*

Um dos requisitos essenciais foi a capacidade de permitir aos usuários importar e analisar textos de forma eficiente, sem as limitações de um editor de texto interno. Observou-se que muitos softwares similares optam por desenvolver seus próprios editores de texto, que frequentemente oferecem recursos de formatação limitados. Isso pode ser inadequado para textos acadêmicos que precisam aderir a normas específicas de formatação ou que contenham um grande número de elementos, como divisões de seções, quadros, tabelas e figuras. Portanto, decidiu-se integrar a plataforma à *API* do *Google Docs*, aproveitando um editor de texto robusto e amplamente adotado.

Para garantir um processo de análise adequadamente, é essencial que o documento no *Google Docs*, que será importado e analisado, seja configurado como público. Isso significa que ele deve ter a permissão de leitura para qualquer usuário que possua o link. A integração foi realizada por meio da biblioteca *Google APIs Node.js Client*, facilitando a autenticação usando credenciais do *Google Cloud* específicas para o nosso sistema.

Após a configuração das credenciais no ambiente do sistema, foi gerado um token *JWT*, o qual permitiu explorar as funcionalidades fornecidas pela biblioteca e integrá-las às rotas da *API*. A abordagem permitiu que os usuários fornecessem a *URL* completa de seu texto na interface da aplicação. Em seguida, processou-se essa *URL* para isolar o *ID* do documento, que seria essencial para a requisição adequada à *API* do *Google Docs*.

O resultado foi um conjunto de dados formatados em *JSON*, que incluem informações sobre a formatação do texto, título e conteúdo. Esses dados foram organizados e depurados, permitindo o isolamento de parágrafos com base na propriedade "*content*" na resposta da requisição. Além disso, foi dividido o conteúdo extraído em partes menores, prevenindo possíveis limitações de tamanho nas mensagens enviadas ao serviço de análise. Esta funcionalidade de integração com o *Google Docs* foi um requisito fundamental para o funcionamento da plataforma. Ela permitiu que os usuários importassem textos de forma rápida e eficaz, mantendo a formatação e estrutura originais.

2.2.2. Integração com a API da OpenAI

Outro requisito crucial para o nosso projeto foi a capacidade de analisar textos de forma inteligente e automatizada. Para isso, foi integrada a plataforma à *API* da *OpenAI*, que oferece modelos de linguagem avançados para análise textual.

Cada tipo de análise específica, como validar resumos, identificar palavras-chave ou gerar sugestões, é indicado por meio de uma requisição ao endpoint da *API* da *OpenAI*. Cada solicitação inclui parâmetros configuráveis, como:

messages: Um *array* de mensagens que estabelece a conversa entre o usuário e o assistente de IA. Isso inclui uma mensagem de sistema que descreve o papel do assistente e uma mensagem do usuário contendo o texto a ser analisado.

model: A escolha do modelo de linguagem da *OpenAI*, com configuração para o uso do modelo "gpt-3.5-turbo".

temperature: Um parâmetro que controla o grau de criatividade das respostas geradas pela IA.

max_tokens: Que define o número máximo de *tokens* permitidos na resposta gerada, controlando o tamanho máximo da saída.

A função encarregada dessas solicitações aguarda a resposta da *API OpenAI* e extrai o conteúdo da mensagem escolhida pela IA na resposta. Em caso de falha na solicitação, a função captura e registra o erro no console e o repassa para tratamento em níveis superiores.

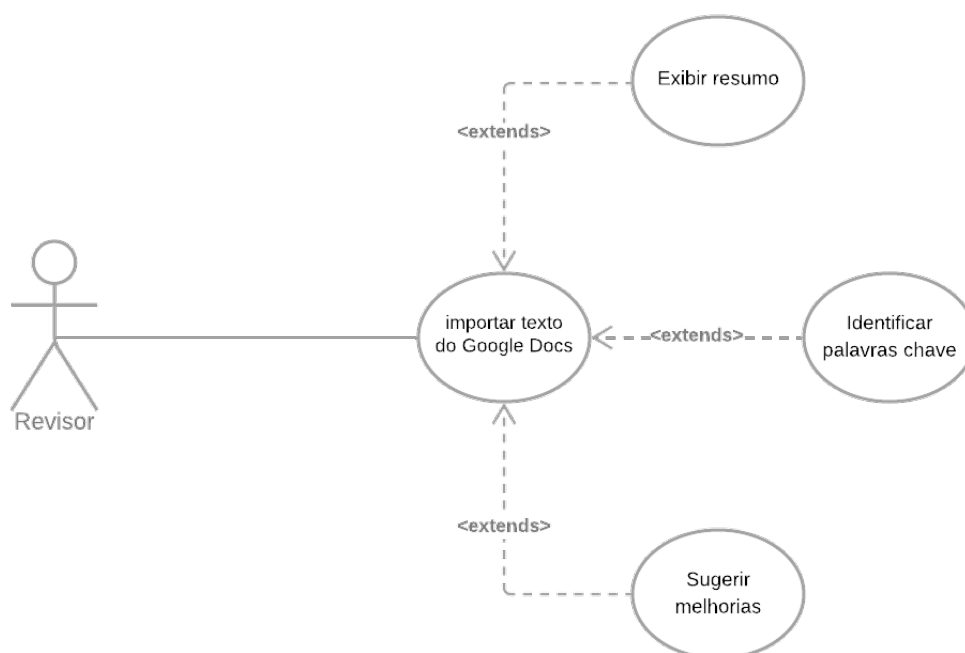
Essas integrações com o *Google Docs* e a *OpenAI* foram elementos centrais no cumprimento dos requisitos do nosso projeto, permitindo que os usuários importem e analisem textos de forma eficiente e inteligente, abrindo um leque de possibilidades para análises de texto avançadas.

3. Metodologia

3.1. Sistema Proposto

O Diagrama de Caso de Uso apresentado lista as funcionalidades disponíveis na plataforma, com destaque para os recursos essenciais e opcionais. Isso proporciona flexibilidade aos usuários, que podem combinar e ajustar as funcionalidades de acordo com suas necessidades específicas.

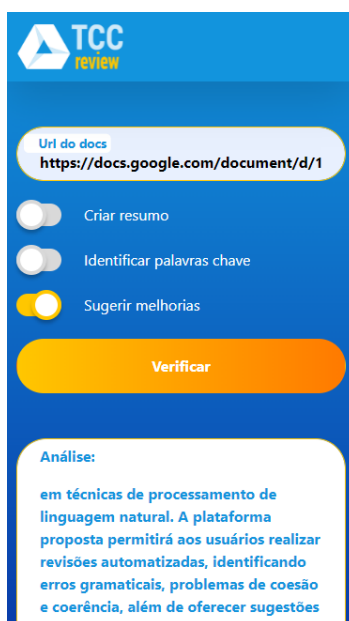
Figura 1. Diagrama de Caso de Uso



Fonte: Elaborado pelo autor (2023).

Para proporcionar uma experiência eficiente e intuitiva aos usuários, optamos por um *design* que integra todas as etapas da ferramenta em uma única tela, eliminando a necessidade de recarregar a página. Essa abordagem foi escolhida para enfatizar a simplicidade e a usabilidade, mantendo-se alinhada com as funcionalidades oferecidas por ferramentas semelhantes.

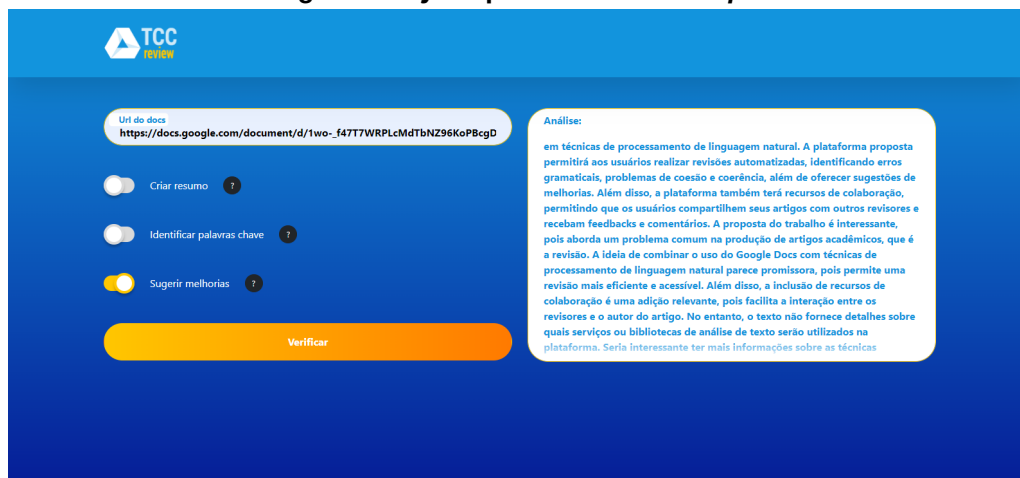
Figura 2. Layout para versão *mobile*



Fonte: Elaborado pelo autor (2023).

A Figura 2 ilustra o layout projetado para a versão móvel da plataforma. Neste esboço, é possível identificar um campo de entrada dedicado à *URL* do documento que será analisado, um requisito obrigatório para o funcionamento da plataforma. Além disso, notamos a presença de elementos do tipo *checkbox* ou *switch*, cuja finalidade é permitir que o usuário controle o uso ou desligue o uso de cada biblioteca, serviço ou *API* de forma independente.

Figura 3. Layout para versão desktop



Fonte: Elaborado pelo autor (2023).

Por outro lado, a Figura 3 apresenta o *layout* planejado para a versão *desktop* da plataforma. Novamente, enfatizamos a presença do campo de entrada para a *URL* do documento e dos elementos de controle para as bibliotecas e serviços. Este *design* foi otimizado para proporcionar uma experiência de usuário em um ambiente de tela maior.

Ambas as figuras têm o papel de servir como referência visual tanto para os desenvolvedores quanto para os usuários. Elas oferecem uma visão antecipada do *layout* da plataforma, o que auxilia na definição de expectativas e na garantia de que o *design* atenda às necessidades dos usuários, resultando em uma interface intuitiva e de fácil utilização.

A plataforma foi desenvolvida com a premissa de ser simples e direta para os usuários, eliminando a necessidade de um sistema de cadastro ou autenticação. No âmbito da persistência de dados, optamos por usar o banco de dados não relacional *MongoDB*. Essa escolha foi baseada na necessidade de simplificar o processo de armazenamento e recuperação de informações, alinhada à nossa abordagem de evitar a complexidade associada a uma modelagem de banco de dados relacional (ER - Entidade-Relacionamento).

O *MongoDB* oferece flexibilidade de esquema, o que significa que não foi necessário definir uma estrutura de esquema rígida antecipadamente. Isso foi útil, pois lida-se com análises de texto que podem variar em termos de formato e estrutura, com diferentes campos e propriedades.

Em relação à recuperação de dados, o *MongoDB* oferece recursos de consulta e índices, permitindo a recuperação eficiente de análises anteriores com base em critérios como ID de texto, tamanho e opções de análise. Por fim, a manutenção do *MongoDB* é geralmente mais simples do que em bancos de dados relacionais, o que facilita a administração do banco de dados.

Assim, a escolha do *MongoDB* como banco de dados foi fundamental para garantir a flexibilidade, escalabilidade e eficiência do armazenamento de dados da plataforma, ao mesmo tempo em que simplificou o desenvolvimento e a manutenção

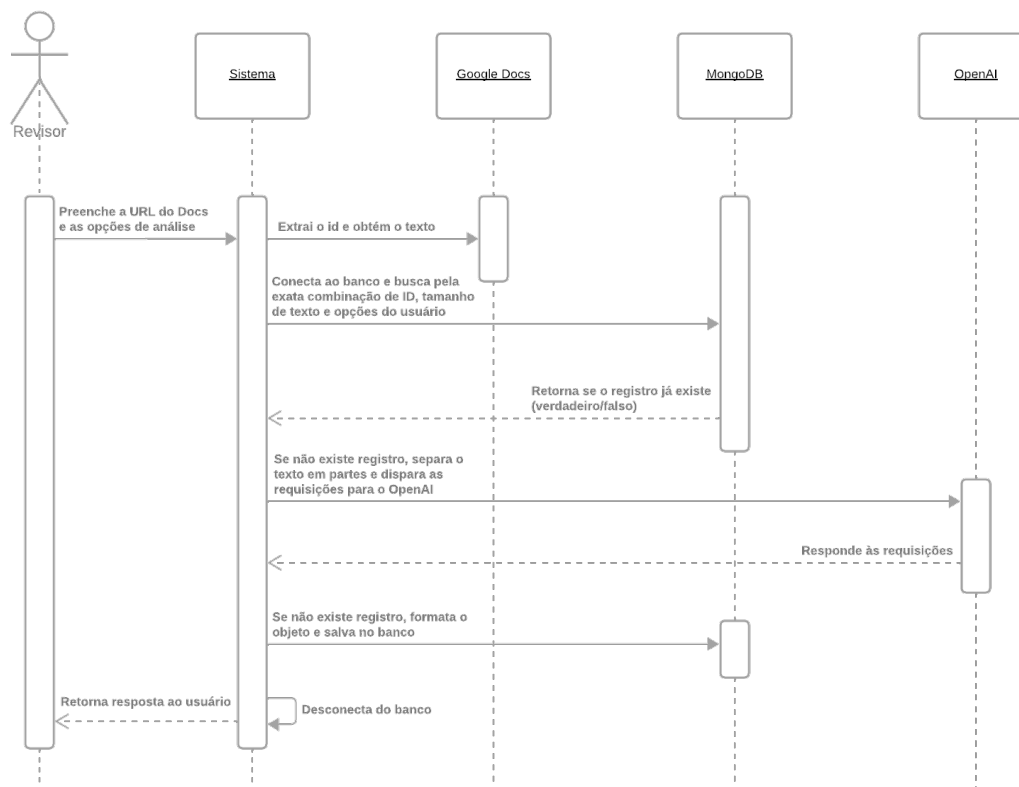
Quadro 3 - Coleção armazenada em banco de dados

Nome do campo no banco de dados	Informação armazenada
<i>_id</i>	ID do objeto gerado automaticamente
<i>analysis</i>	Conteúdo da análise devolvido pelo serviço do OpenAI
<i>documentId</i>	ID do Documentos Google extraído da URL enviada pelo usuário
<i>contentLength</i>	O tamanho da <i>string</i> do texto obtido
<i>options.validateSummary</i>	Valor <i>booleano</i> para o caso de uso de criar resumo.
<i>options.findKeywords</i>	Valor <i>booleano</i> para o caso de uso de identificar palavras-chave.
<i>options.improvementSuggestions</i>	Valor <i>booleano</i> para o caso de uso de sugerir melhorias.

Fonte: Elaborado pelo autor (2023).

O Quadro 3 detalha o objeto que compõe a coleção persistida na base de dados. Durante cada execução da API desenvolvida, uma verificação é realizada nesta coleção. Com base no resultado dessa consulta, a API pode adotar dois comportamentos para fornecer uma análise ao usuário, conforme descrito no diagrama de sequência representado pela Figura 4. Esta estratégia foi implementada para evitar requisições repetidas à OpenAI, resultando em tempos de resposta mais rápidos e uma utilização mais eficiente dos recursos.

Figura 4. Diagrama de Sequência



Fonte: Elaborado pelo autor (2023).

A interface da plataforma foi desenvolvida usando o *framework* *Vue.js*, que proporciona uma manipulação e consumo estruturado de dados. Isso permite a criação de uma aplicação de página única (*Single-Page Application*) que oferece uma experiência fluida ao usuário. Essa escolha de tecnologia foi feita para atender aos requisitos de eficiência e usabilidade da plataforma.

O projeto foi estruturado com base em um Mínimo Produto Viável (*MVP*) que define o escopo inicial do protótipo a ser entregue neste Trabalho de Conclusão de Curso (TCC). O *MVP* concentra-se nas funcionalidades essenciais que compõem o núcleo da plataforma, permitindo ao usuário escolher quais análises deseja aplicar ao seu texto. A lista inicial de casos de uso no *MVP* inclui:

- **Exibir Resumo:** Gera um breve resumo automaticamente. Esse resumo automático busca capturar de forma concisa os principais pontos e informações essenciais do texto.
- **Identificar Palavras-chave:** Essa funcionalidade determina quais termos têm a maior ocorrência ao longo do texto, auxiliando na identificação das palavras-chave mais relevantes dentro do documento.
- **Sugerir melhorias:** Esta funcionalidade permite solicitar sugestões gerais de aprimoramento no texto. Ao utilizá-la, são recebidas recomendações para melhorar diferentes aspectos do documento, como estrutura, clareza, coesão e

estilo. Isso ajuda a elevar a qualidade do seu trabalho de forma abrangente, tornando-o mais eficaz e fluido para os leitores.

Estes casos de uso representaram as funcionalidades iniciais do protótipo, proporcionando flexibilidade aos usuários para escolher quais análises desejam aplicar aos seus textos. Esta abordagem permitiu que o projeto mantivesse um escopo gerenciável e eficiente, ao mesmo tempo em que oferece um conjunto de recursos para análise de texto. Além disso, pode-se ressaltar que mais casos de uso podem ser adicionados posteriormente, expandindo as capacidades da plataforma conforme as necessidades dos usuários evoluem.

3.2. Verificação da solução

Com o propósito de avaliar a eficácia do modelo de inteligência artificial adotado para o protótipo, foi conduzido um benchmark a partir de uma amostra representativa de seis trabalhos de conclusão do curso superior de Tecnologia em Sistemas para Internet, do Instituto Federal do Rio Grande do Sul (IFRS). Para compor a amostra de testes foram selecionados da Biblioteca Clóvis Vergara Marques, disponível no repositório Atom do IFRS, os seguintes trabalhos:

- Um *Framework* para criação de jogos *towerdefense* baseado no motor ORX (MORAIS; MACHADO; 2016)
- *DashBoard* para Internet das Coisas com Dados Abertos (ALMEIDA; PERES; 2016)
- Gerador de Formulários (KORNDORFER; SCHMITT; 2016)
- Hidra: uma Inteligência Artificial Solucionadora de Labirintos (DAUDT; OKUYAMA; 2016)
- Integração de repositórios de objetos de aprendizagem com ambientes virtuais de aprendizagem (VIDEIRA; SCHMITT; 2016)
- Um estudo sobre as recomendações de acessibilidade em *tablets Android* para deficientes visuais (MACHADO; MACHADO; 2016)

Os artigos selecionados foram individualmente submetidos a testes de execução no sistema proposto. O conteúdo de cada artigo foi copiado para um documento próprio do Google Docs de permissão pública para leitura. Para cada requisição foi utilizada a instrução padrão da plataforma, buscando situar o modelo de IA que o texto em questão é composto por partes, seguindo a requisição de tentar identificar quaisquer inconsistências no texto, analisando seu conteúdo, linguagem, clareza e fluidez.

Após cada execução, foi tomada nota dos principais aspectos observados pelo modelo de IA, buscando sintetizar um resumo geral das suas considerações em relação à legibilidade e objetividade do texto. Os resultados foram comparados com testes análogos realizados na plataforma *ChatPDF*, software relacionado que foi identificado durante o processo de desenvolvimento, objetivando estabelecer uma base coerente para a validação e comparação do desempenho dos interpretadores de texto. Essa comparação com o *ChatPDF* visa contextualizar os resultados no panorama atual de tecnologias similares.

4. Resultados e discussão

As avaliações feitas pelo *TCC Review* e *ChatPDF* forneceram insights sobre distintos aspectos das amostras analisadas. Com a necessidade de trabalhar com o texto fragmentado em blocos, a fim de comportar a limitação do tamanho de respostas, o *TCC Review* retornou um grande volume de observações, muitas vezes se repetindo devido a reincidências ou limitado pelo escopo do contexto durante a análise do bloco.

Ainda assim, o *TCC Review* ressaltou aspectos específicos, como a falta de um nome para o software proposto no trabalho “Um *Framework* para criação de jogos *towerdefense* baseado no motor ORX”. Também sugeriu melhorias quanto à estrutura e fluidez, bem como uma transição mais suave de ideias e a simplificação em “Gerador de Formulários” e “Integração de repositórios de objetos de aprendizagem com ambientes virtuais de aprendizagem”, visando fornecer uma leitura mais acessível a um público não tão familiarizado com a área especificamente de conhecimento. Observações similares em relação ao uso de termos técnicos foram feitas sobre “Hidra”.

Por sua vez, o *ChatPDF* focou-se em resumir os temas de cada, fornecendo atalhos para o detalhamento de tópicos centrais. Somente ao ser ativamente acionado através da interface de chat a plataforma executou a tarefa de fornecer pareceres sobre o aspecto textual, observando a complexidade dos termos técnicos abordados ao longo do artigo. Observou-se que, devido ao seu propósito primário de resumir conteúdo de arquivos, o *ChatPDF* é condicionado a não avaliar o texto de maneira absoluta, sempre declarando que seus insights dependem do contexto do leitor.

Em síntese, a eficiência do *TCC Review* equiparou-se ao *ChatPDF* no que diz respeito à capacidade de fornecer apontamentos sobre o texto. A principal distinção reside no fato de que o *TCC Review* fornece respostas de forma proativa, enquanto o *ChatPDF* demanda que o usuário questione cada aspecto a ser analisado individualmente, com a possibilidade de o sistema se considerar incapaz de avaliar a qualidade da escrita.

5. Conclusão

Com base nos resultados desta análise, considera-se que com um refinamento futuro no algoritmo de integração como uma etapa para o aprimoramento deste projeto e com um desempenho adequado em termos de tempo de resposta, o sistema pode desempenhar o papel de assistente de revisão, constituindo uma ferramenta para autores, orientadores e avaliadores.

Durante o desenvolvimento da aplicação, deparou-se com uma limitação de recursos na API do *OpenAI* relacionada à extensão do texto que poderia ser enviado. Observou-se que a coleção de mensagens usada para definir o contexto da conversa pelo modelo não suportava o envio do artigo em partes suficientemente grandes. Para resolver esse desafio, foi necessário um processo de pré-processamento do conteúdo, segmentando-o em partes menores e enviando requisições separadas para cada bloco de

texto. Como resultado, o sistema passou a apresentar tempos de resposta consideravelmente maiores, uma vez que tinha que aguardar múltiplas respostas externas. Além disso, devido à necessidade de análises distintas, foi ajustado o escopo dos casos de uso para garantir que sua precisão e confiabilidade não fossem comprometidas pela complexidade da interpretação do modelo.

Apesar dessas complexidades, o serviço de análise de texto foi capaz de identificar os termos necessários e construir a linha de raciocínio conforme descrito no documento, cumprindo assim com sucesso o que foi proposto como protótipo de aplicação. Observa-se que o resultado das respostas fornecido pelo modelo IA está sujeito aos enunciados disparados pelo sistema, bem como contextualização dada pelo texto. Desta forma, pareceres equivocados podem ocorrer durante a análise, sendo fortemente recomendo o uso do *TCC Review* como ferramenta auxiliar e não como substituição da revisão feita por um leitor humano.

Referências

- ALMEIDA, Gutierre Bessauer; PERES, André. *DashBoard* para Internet das Coisas com Dados Abertos. 2016. 15 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Instituto Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: <http://atom.poa.ifrs.edu.br/uploads/r/biblioteca-clovis-vergara-marques-4/8/7/a/87ada6afc512a9800b8470540807550d42a18b00df3d777ee90363657df12495/Artigo-Final-Gutierre.pdf>. Acesso em: 21 nov. 2023.
- BIGOLIN, M. et al. RevisãoOnline - Ferramenta para melhora da escrita utilizando revisão por pares. [s.l.: s.n.]. Disponível em: http://www.tise.cl/Volumen16/Software%20Educativo/TISE_2022_paper_54.pdf. Acesso em: 16 abr. 2023.
- BITTENCOURT JÚNIOR, José Adenaldo Santos. Avaliação automática de redação em língua portuguesa empregando redes neurais profundas. 2020. 100 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Goiás, Goiânia, 2020.
- BURROWS, S.; GUREVYCH, I.; STEIN, B. *The eras and trends of automatic short answer grading. International Journal of Artificial Intelligence in Education*, [S.l.], v. 25, n. 1, p. 60-117, 2015.
- DAUDT, Guilherme NM; OKUYAMA, Fábio Y. Hidra: uma Inteligência Artificial Solucionadora de Labirintos. 10 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Instituto Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: <http://documentos.poa.ifrs.edu.br/uploads/r/biblioteca-clovis-vergara-marques-4/c/7/5/c75523e3f1e160cb2e55a94ae76df77870efa65a59eb5c31e367d9a4ed8a95e5/Artigo-Guilherme.pdf>. Acesso em: 21 nov. 2023.

- FERNANDES, Afonso Fonseca. INTELIGÊNCIA ARTIFICIAL E EDUCAÇÃO. BIUS-Boletim Informativo Unimotrisaúde em Sociogerontologia, v. 39, n. 33, p. 1-3, 2023. Disponível em: <https://periodicos.ufam.edu.br/index.php/BIUS/article/view/12646>. Acesso em: 13 de setembro de 2023
- GARCEZ, L.H.C. 1998. A escrita e o outro: os modos de participação na construção do texto. Brasília, Editora Universidade de Brasília, 173 p.
- KORNDORFER, Jonas Henrique M.; SCHMITT, Marcelo Augusto R. Gerador de Formulários. 2016. 12 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Instituto Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: <http://documentos.poa.ifrs.edu.br/uploads/r/biblioteca-clovis-vergara-marques-4/d/c/c/dccb7dbf79e41ceb60dbf5c5b39832224afe2acb8427e99b23a749206b8167ee/Artigo-Final-Jonas.pdf>. Acesso em: 21 nov. 2023.
- MACHADO, Douglas Ritter; MACHADO, Rodrigo Prestes. Um estudo sobre as recomendações de acessibilidade em *tablets Android* para deficientes visuais. 15 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Instituto Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: http://documentos.poa.ifrs.edu.br/uploads/r/biblioteca-clovis-vergara-marques-4/9/4/2/942a07c69949db8691d30c867fde1c538d3e5479f9ccea78776a6ce02566ec92/TCC_DOUGLAS_FINAL.pdf. Acesso em: 21 nov. 2023.
- MENEGASSI, R.J. 1998. Da revisão à reescrita: operações e níveis linguísticos na construção do texto. Assis, SP. Tese de Doutorado. Faculdade de Ciências e Letras de Assis, Universidade Estadual Paulista, 228 p. (sic)
- MORAIS, Gabriel Maciel; MACHADO, Rodrigo Prestes. Um *Framework* para criação de jogos *towerdefense* baseado no motor ORX. 2016. 11 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Instituto Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: <http://atom.poa.ifrs.edu.br/uploads/r/biblioteca-clovis-vergara-marques-4/e/b/1/eb196e04d1855cf4a3f3c34a814972d733c1c3fefb84c495b412c6ddaaafb9de/Artigo-Final-Gabriel.pdf>. Acesso em: 21 nov. 2023.
- PINHO, C. M. DE A. *et al.* Identificação de deficiências em textos educacionais com a aplicação de processamento de linguagem natural e aprendizado de máquina. ETD: Educação Temática Digital, v. 24, n. 2, p. 350–372, 2022.
- VIDEIRA, Jonathan Alba; SCHMITT, Marcelo Augusto Rauh. Integração de repositórios de objetos de aprendizagem com ambientes virtuais de aprendizagem. 10 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Instituto Federal do Rio Grande do Sul, Porto Alegre, 2016. Disponível em: <http://atom.poa.ifrs.edu.br/uploads/r/biblioteca-clovis-vergara-marques-4/1/a/7/1a75b8f9362f56c25a3d60c8718de0a1181f5637a04748437b36f16d82fb7673/Artigo-Jonathan.pdf>. Acesso em: 21 nov. 2023.

YAMAKAWA, E. K. *et al.*. Comparativo dos softwares de gerenciamento de referências bibliográficas: *Mendeley*, *EndNote* e *Zotero*. *Transinformação*, v. 26, n. 2, p. 167–176, maio 2014.