



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO LUCAS DE MOURA PIMENTA

TEXTFLOWREDUCE: UTILIZANDO THREADS E EXPRESSÕES REGULARES
PARA ANÁLISE DE QUESTÕES

RUSSAS

2026

JOÃO LUCAS DE MOURA PIMENTA

TEXTFLOWREDUCE: UTILIZANDO THREADS E EXPRESSÕES REGULARES PARA
ANÁLISE DE QUESTÕES

Trabalho de Conclusão de Curso de Ciência da Computação do Campus de Russas da Universidade Federal do Ceará, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Orientador : Prof. Dr. Cenez Araujo de Rezende

RUSSAS

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

P697t Pimenta, João Lucas De Moura.
TextFlowReduce: Utilizando Threads e Expressões Regulares para Análise de Questões / João Lucas De Moura Pimenta. – 2026.
34 f. : il.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2026.
Orientação: Prof. Dr. Cenez Araujo de Rezende.

1. Avaliação automática. 2. Processamento paralelo. 3. C#. 4. TPL. 5. Regex. I. Título.

CDD 005

JOÃO LUCAS DE MOURA PIMENTA

TEXTFLOWREDUCE: UTILIZANDO THREADS E EXPRESSÕES REGULARES PARA
ANÁLISE DE QUESTÕES

Trabalho de Conclusão de Curso de Ciência da Computação do Campus de Russas da Universidade Federal do Ceará, como requisito parcial para a obtenção do Grau de Bacharel em Ciência da Computação.

Aprovado em: 19/01/2026

BANCA EXAMINADORA

Prof. Dr. Cenez Araujo de Rezende (Advisor)
Universidade Federal do Ceará (UFC)

Prof. Dr. Reuber Regis De Melo
Universidade Federal do Ceará (UFC)

Prof. Ms. Pitagoras Graça Martins
Universidade Federal do Ceará (UFC)

À minha mãe, À minha família e minha esposa que sempre me apoiaram, ao Bigode e a Banguela, meus filhos (gatos) que perdi e que sempre levarei comigo, vocês estarão pra sempre no meu coração, e aos filhos, a quem sempre amarei.

AGRADECIMENTOS

Ao Prof. Dr. Cenez Araujo de Rezende por orientar meu Trabalho de Conclusão de Curso e me inspirar com a publicação de seu artigo, o qual foi tão importante para a construção do meu trabalho.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

À Minha mãe, Aos meus familiares e minha companheira, os quais sempre estiveram presentes em momento difíceis, e sempre me motivaram a continuar seguindo para um futuro melhor!

“O insucesso é apenas uma oportunidade para
recomeçar com mais inteligência.”

(Henry Ford)

RESUMO

O presente trabalho tem como objetivo apresentar o desenvolvimento e validação da biblioteca TextFlowReduce, uma solução automatizada para análise de respostas dissertativas curtas, baseada em critérios semânticos configuráveis, e processamento paralelo. A biblioteca foi projetada para reduzir a subjetividade e o tempo de correção manual, permitindo aos docentes definirem os critérios, que deverão ser formulados utilizando palavras-chave, frases e pesos, a depender da metodologia e do critério escolhido. A metodologia adotada neste trabalho, envolveu a implementação em C, utilizando Expressões Regulares (Regex) para busca precisa de padrões e a Task Parallel Library (TPL) para análise concorrente dos critérios. O experimento foi realizado com 30 participantes, e demonstrou que a ferramenta é capaz de quantificar objetivamente o conhecimento técnico, fornecendo relatórios detalhados de desempenho e identificando lacunas conceituais. Os resultados evidenciaram forte correlação entre o uso de termos técnicos e a pontuação final, validando a abordagem determinística e transparente. Desta feita, conclui-se que a ferramenta TextFlowReduce é eficaz para automatizar a avaliação de respostas discursivas, otimizando o processo avaliativo e oferecendo feedback imediato e detalhado.

Palavras-chave: Avaliação automática; Processamento paralelo; C#; TPL; Regex; Critérios configuráveis.

ABSTRACT

The purpose of this paper is to present the development and validation of the TextFlowReduce library, an automated solution for analyzing short essay responses based on configurable semantic criteria and parallel processing. The library was designed to reduce subjectivity and manual correction time, allowing teachers to define criteria, which should be formulated using keywords, phrases, and weights, depending on the methodology and criteria chosen. The methodology adopted in this work involved implementation in C, using Regular Expressions (Regex) for accurate pattern searching and the Task Parallel Library (TPL) for concurrent analysis of the criteria. The experiment was conducted with 30 participants and demonstrated that the tool is capable of objectively quantifying technical knowledge, providing detailed performance reports and identifying conceptual gaps. The results showed a strong correlation between the use of technical terms and the final score, validating the deterministic and transparent approach. Thus, it is concluded that the TextFlowReduce tool is effective for automating the evaluation of discursive responses, optimizing the evaluation process and providing immediate and detailed feedback.

Keywords: Automatic grading; Parallel processing; C; TPL; Regex; Configurable criteria.

LISTA DE TABELAS

Tabela 1 – Estatísticas descritivas das pontuações gerais	26
Tabela 2 – Distribuição de pontuações em faixas	27
Tabela 3 – Estatísticas por área de conhecimento	27
Tabela 4 – Desempenho por questão	28
Tabela 5 – Cronograma de Atividades TCC	33

LISTA DE ABREVIACOES E ACRNIMOS

ASAG	Automatic Short Answer Grading
ConcurrentBag	Uma implementao de coleo thread-safe no .NET Framework, otimizada para cenrios onde a mesma thread produz e consome dados, mas suportando acesso concorrente
IAS	Inteligncias Artificiais
LLM	Large Language Model
LSA	Latent Semantic Analysis
PLN	Processamento de Linguagem Natural
Regex	Abreviao para Expresses Regulares. Uma sequncia de caracteres que forma um padro de pesquisa, utilizada para correspondncia de strings
Thread-Safe	Conceito de programao aplicvel a cdigo multi-thread, garantindo que as estruturas de dados compartilhadas sejam acessadas por mltiplas threads simultaneamente sem causar condies de corrida ou inconsistncias
TPL	Task Parallel Library
Word Boundary	Em expresses regulares,  uma ncora que corresponde  posio onde uma palavra comea ou termina, garantindo que a busca case com a palavra inteira e no parte dela

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Problema e Justificativa	13
1.2	Objetivos	14
1.2.1	<i>Objetivo Geral</i>	14
1.2.2	<i>Objetivos Específicos</i>	14
1.3	Organização do Trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	O Desafio da Avaliação de Respostas Abertas	16
2.2	Processamento de Linguagem Natural e Normalização	16
2.3	Concorrência e Paralelismo de Dados	17
2.4	Correspondência de Padrões e Expressões Regulares	17
2.5	Modularidade e Estrutura de Software	18
3	TRABALHOS RELACIONADOS	19
3.1	Avaliação Automática Baseada em Aprendizado de Máquina	19
3.2	Sistemas de Avaliação em Língua Portuguesa	19
3.3	Ferramentas de Análise Qualitativa vs. Avaliação Objetiva	20
3.4	Desempenho e Processamento Paralelo em Análise Textual	20
4	METODOLOGIA	21
4.1	Definição dos Requisitos	21
4.2	Análise e Projeto do Sistema	22
4.3	Desenvolvimento	23
4.4	Disponibilização e Manutenção	24
4.5	Procedimentos de Coleta de Dados e Perfil dos Participantes	24
5	RESULTADOS E DISCUSSÕES	26
5.1	Análise Geral do Desempenho dos Alunos	26
5.1.1	<i>Estatísticas Descritivas Gerais</i>	26
5.1.2	<i>Distribuição de Pontuações</i>	27
5.2	Análise por Área de Conhecimento e por Questão	27
5.3	Precisão da Análise e Avaliação Qualitativa	28
5.4	Discussão dos Resultados	28

6	CONCLUSÃO	30
6.1	Limitações e Trabalhos Futuros	30
6.2	Considerações Finais	31
	REFERÊNCIAS	32
	APÊNDICES	33
	APÊNDICES – CRONOGRAMA DE ATIVIDADES	33

1 INTRODUÇÃO

O cenário educacional brasileiro tem enfrentado desafios crescentes com a digitalização e o aumento do volume de dados gerados em processos avaliativos. Em 2023, o Brasil manteve sua posição de destaque na produção científica global, com quase 157 mil artigos publicados (CGCOM/CAPES, 2023), o que evidencia uma tendência de expansão no consumo e geração de informações textuais. No entanto, essa escala de dados impõe um problema crítico às instituições de ensino: a dificuldade de realizar um processo de avaliações qualitativas cuja análise das questões dissertativas, ocorra de forma rápida, certa e objetiva.

Diferente de questões de múltipla escolha, que permitem correção binária e instantânea, as respostas dissertativas exigem que o professor identifique conceitos semânticos fundamentais em meio à variação da escrita do aluno. Segundo Nery (2017), o processo de correção manual é inerentemente subjetivo, sendo influenciado por fatores como a fadiga do avaliador e vieses cognitivos. Essa subjetividade, como aponta Kuckartz (2014), compromete a padronização e a reprodutibilidade dos critérios de avaliação, especialmente em turmas numerosas.

No campo da tecnologia, o desenvolvimento de sistemas para análise textual em larga escala exige uma arquitetura de software eficiente. Caseli and Nunes (2024) ressaltam que o Processamento de Linguagem Natural (Processamento de Linguagem Natural (PLN)) demanda etapas de normalização, como a remoção de marcas de acentuação e padronização de escrita (Caixa Alta), para assegurar que variações ortográficas simples não invalidem conceitos corretos. Para suportar esse fluxo de análise de maneira performática, o uso de processamento paralelo permite a verificação simultânea de diversas categorias e critérios, otimizando o tempo de resposta e garantindo a escalabilidade da solução.

1.1 Problema e Justificativa

Atualmente, o uso de Inteligências Artificiais (IAS) generativas para a avaliação de textos tem se tornado comum (GU *et al.*, 2025). Contudo, esses modelos muitas vezes operam como "caixas-pretas", onde o professor possui pouco controle sobre as heurísticas exatas de pontuação. A ausência de ferramentas que possibilitem a configuração explícita de palavras-chave e frases obrigatórias, com pesos definidos pelo docente, faz com que haja uma lacuna tecnológica entre a automação e o rigor pedagógico.

Além da sobrecarga cognitiva, a avaliação manual está suscetível ao viés implícito, onde a subjetividade do docente pode inadvertidamente privilegiar determinados estilos de escrita. Conforme Nery (2017), a avaliação humana carrega marcas de subjetividade que dificultam a padronização. A automação via critérios explícitos promove a equidade (isonomia) avaliativa, garantindo que todos os alunos sejam medidos pela mesma régua técnica.

A justificativa para o projeto **TextFlowReduce** reside na criação de uma biblioteca de análise de alta performance, baseado em funções bem declaradas e processamento paralelo. O sistema busca "reduzir"(daí o termo *Reduce*) a complexidade da correção manual ao consolidar múltiplos critérios em um escore objetivo único. Ao invés de utilizarmos de uma verificação linear exaustiva, o sistema utiliza o potencial de *threads* para analisar concorrentemente categorias distintas (palavras obrigatórias, frases e termos opcionais), garantindo que a resposta ao aluno seja precisa e imediata. é fundamentada em princípios de Engenharia de Software, buscando um índice maior de coesão entre classes de modelos existentes, e a análise resultante das consultas, de modo que assim haja a manutenção e evolução do sistema Pressman and Maxim (2016).

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver o sistema **TextFlowReduce**, uma ferramenta automatizada para análise de respostas dissertativas fundamentada em critérios semânticos configuráveis e processamento paralelo via *threads*, visando otimizar a correção docente e fornecer *feedbacks* detalhados aos discentes.

1.2.2 Objetivos Específicos

- a) Implementar um módulo de normalização textual capaz de realizar a limpeza de caracteres e a decomposição de acentos, seguindo padrões de PLN Caseli and Nunes (2024).
- b) Desenvolver um motor de análise baseado em *multithreading*, permitindo a verificação independente e simultânea de listas de palavras e frases.
- c) Estruturar uma lógica de validação de pesos (cuja soma totaliza 1.0) para diferentes categorias de critérios, permitindo a customização pedagógica total pelo professor.
- d) Gerar relatórios de análise detalhados através de um objeto de resultado estruturado, informando o escore final e os elementos ausentes.

1.3 Organização do Trabalho

Este trabalho divide-se em cinco capítulos: esta introdução; fundamentação teórica sobre PLN e concorrência (Cap. 2); metodologia e implementação técnica (Cap. 4); discussão de resultados e testes (Cap. 5); e as considerações finais (Cap. 6).

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste trabalho visa estabelecer as bases conceituais que sustentam o desenvolvimento do *TextFlowReduce*. A proposta transita entre a subjetividade da avaliação educacional, as técnicas de Processamento de Linguagem Natural (PLN) para normalização de dados e a eficiência computacional por meio de algoritmos de busca otimizados e uma arquitetura de software modular.

2.1 O Desafio da Avaliação de Respostas Abertas

A correção de questões dissertativas é uma tarefa complexa que exige do avaliador a identificação de conceitos semânticos em meio à variabilidade linguística do aluno. Segundo Nery (2017), a subjetividade do corretor pode ser influenciada por fatores externos e pelo estilo de escrita do discente, o que gera o risco de inconsistências na atribuição de notas e dificulta a padronização avaliativa.

No campo da tecnologia educacional, a área de Automatic Short Answer Grading (ASAG) estuda métodos para automatizar esse processo. Burrows *et al.* (2015) destacam que, diferentemente de redações longas, as respostas curtas focam na presença de conceitos específicos e objetivos. Por isso, sistemas que permitem ao professor definir "gabaritos de conceitos", compostos por palavras e frases obrigatórias, são essenciais para garantir que a avaliação reflita fielmente o conteúdo esperado, minimizando o viés interpretativo destacado pela hermenêutica de Ricoeur (1991).

Essa abordagem se alinha à Taxonomia de Bloom, especificamente nos níveis de 'Conhecimento' e 'Compreensão', onde a precisão terminológica é fundamental para demonstrar o domínio do conteúdo. Ademais, a automação possibilita o feedback imediato, um pilar central da avaliação formativa, permitindo que o aluno reconheça e aprenda com o erro no momento exato do teste, transformando a avaliação em uma etapa ativa de aprendizagem.

2.2 Processamento de Linguagem Natural e Normalização

Para que um sistema automatizado compare a resposta de um aluno com um gabarito docente, é necessário reduzir as variações superficiais da língua que não alteram o sentido conceitual. Conforme Caseli and Nunes (2024), o pré-processamento é uma etapa vital no PLN. No contexto do *TextFlowReduce*, as técnicas de normalização aplicadas incluem:

- **Conversão de Caixa (*Lowercasing*):** Garante que variações entre maiúsculas e minúsculas não impeçam a correspondência de termos idênticos.
- **Remoção de Acentuação:** Evita que erros ortográficos simples ou falta de acentuação invalidem um conceito semanticamente correto, focando na essência da resposta.

Essa abordagem encontra respaldo em Kuckartz (2014), que afirma que a análise qualitativa mediada por software deve focar na categorização sistemática de unidades de sentido, removendo ruídos textuais que não alteram o significado central do conteúdo analisado.

2.3 Concorrência e Paralelismo de Dados

O processamento de múltiplos critérios de avaliação, como a verificação simultânea de palavras-chave obrigatórias, frases específicas e termos opcionais, pode tornar-se computacionalmente oneroso se executado de forma puramente linear, especialmente em cenários de alta demanda. A utilização de paralelismo de dados permite que diferentes partes da análise sejam executadas simultaneamente, aproveitando a arquitetura *multi-core* dos processadores modernos.

Segundo Albahari (2012), o uso de *multithreading* é essencial para otimizar tarefas de processamento intensivo de dados, permitindo a execução concorrente de fluxos de trabalho independentes. No *TextFlowReduce*, o motor de análise utiliza a Task Parallel Library (TPL) do .NET para distribuir a carga de verificação entre diferentes *threads*. Essa estratégia, combinada com o uso de coleções seguras para concorrência (coleções Thread-Safe), como *ConcurrentBag*, garante que o sistema mantenha a integridade dos dados enquanto maximiza o desempenho, acelerando o retorno do *feedback* ao professor.

2.4 Correspondência de Padrões e Expressões Regulares

A identificação precisa de termos em um texto livre exige métodos robustos de busca que vão além da simples comparação de *strings*. O uso de Expressões Regulares (Regex) permite a definição de padrões formais de busca, possibilitando a detecção de palavras inteiras e frases exatas, respeitando as fronteiras das palavras (Word Boundary).

No desenvolvimento de ferramentas de análise textual, a eficiência na recuperação de informações é crucial. O ferramenta de análise do *TextFlowReduce* utiliza algoritmos de correspondência de padrões dentro de laços paralelos para verificar a presença dos critérios definidos pelo professor. Essa abordagem híbrida, que une a precisão determinística das ex-

pressões regulares com a velocidade do processamento paralelo, assegura que a pontuação seja atribuída de forma consistente e rápida. A aplicação de regras estritas de validação permite que o sistema processe respostas com alta precisão, identificando tanto a presença quanto a ausência dos elementos chaves exigidos no gabarito.

2.5 Modularidade e Estrutura de Software

A robustez de uma ferramenta de análise textual depende de uma arquitetura de software clara e bem definida. A utilização de funções bem declaradas e o encapsulamento de responsabilidades são princípios fundamentais da Engenharia de Software que visam a manutenibilidade e a testabilidade do sistema Pressman and Maxim (2016).

Na Ferramenta *TextFlowReduce*, essa modularidade é expressa através da separação entre o modelo de dados e a lógica de processamento:

- **AnswerCriteria:** Classe responsável pelo armazenamento dos parâmetros e pesos definidos pelo docente.
- **AnswerAnalyzer:** Classe que centraliza a lógica de normalização e os algoritmos de busca textual, garantindo que o fluxo de processamento seja independente da interface.
- **AnswerAnalysisResult:** Objeto que encapsula o resultado final, promovendo uma comunicação clara entre o motor de análise e o relatório de saída.

Essa estrutura funcional permite que cada etapa da análise seja isolada e validada individualmente, garantindo que o sistema seja extensível para novos critérios sem comprometer a integridade do núcleo de processamento.

Em suma, a integração entre a objetividade pedagógica, o pré-processamento textual e algoritmos eficientes de busca, formam o alicerce necessário para que o *TextFlowReduce* atue como uma ferramenta confiável no auxílio à avaliação docente.

3 TRABALHOS RELACIONADOS

A área de avaliação automática de respostas curtas (ASAG) tem sido objeto de estudo tanto na Ciência da Computação quanto na Educação, buscando soluções que equilibrem precisão, desempenho e interpretabilidade. Nesta seção, são apresentados trabalhos que se relacionam com a proposta do *TextFlowReduce*, organizados em abordagens baseadas em aprendizado de máquina, ferramentas de análise qualitativa e sistemas focados em correspondência de padrões.

3.1 Avaliação Automática Baseada em Aprendizado de Máquina

A abordagem predominante na literatura recente envolve o uso de técnicas avançadas de Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina. Mohler *et al.* (2011) propuseram métodos baseados em similaridade semântica e alinhamento de grafos de dependência para graduar respostas curtas. Embora seus resultados tenham superado métodos baseados apenas em sobreposição lexical (como *bag-of-words*), a complexidade computacional e a necessidade de grandes *corpora* de treinamento limitam sua aplicação em cenários onde o professor deseja configurar rapidamente uma nova questão sem pré-treinamento de modelos.

Mais recentemente, o uso de Large Language Models (LLMs) tem ganhado destaque. Gu *et al.* (2025) exploraram o uso de modelos generativos para avaliação de estratégias de tutoria. Apesar da alta capacidade de generalização, essas soluções operam frequentemente como "caixas-pretas", dificultando a explicação exata do motivo de uma nota específica, o que contrasta com a necessidade de transparência em avaliações somativas formais.

3.2 Sistemas de Avaliação em Língua Portuguesa

No contexto brasileiro, Rodrigues and Araujo (2012) desenvolveram um sistema para avaliação automática de respostas curtas em português utilizando Latent Semantic Analysis (LSA). O trabalho demonstrou não só a viabilidade de processar o idioma português considerando suas especificidades, mas também evidenciou os desafios de lidar com a polissemia e a sinonímia sem uma base de conhecimento robusta.

Galhardi and Brancher (2018) realizaram uma revisão sistemática sobre a avaliação automática de respostas dissertativas, concluindo que, embora existam muitas iniciativas, há uma carência de ferramentas que sejam leves, de fácil configuração pelo docente e que ofereçam *feedback* imediato e detalhado, lacuna que o presente trabalho busca preencher.

3.3 Ferramentas de Análise Qualitativa vs. Avaliação Objetiva

Ferramentas como *IRaMuTeQ* e *NVivo* são referências na análise qualitativa exploratória. Camargo and Justo (2013) destacam o poder do *IRaMuTeQ* para análises lexicais e de similitude em grandes volumes de texto. Entretanto, o foco dessas ferramentas é a descoberta de padrões e temas emergentes em pesquisas, e não a verificação da presença de critérios pré-definidos para fins de atribuição de nota. O *TextFlowReduce* diferencia-se por inverter essa lógica: ao invés de descobrir o que está no texto, ele verifica se o texto contém o que foi solicitado pelo professor.

3.4 Desempenho e Processamento Paralelo em Análise Textual

A eficiência no processamento de texto é um requisito crítico para sistemas de *feedback* em tempo real. Enquanto frameworks de *Big Data* como o Apache Spark (Cecchini (2023)) focam em processamento distribuído em *clusters* para volumes massivos de dados, a abordagem do *TextFlowReduce* foca no paralelismo local (*multithreading*) para otimizar a análise de respostas individuais em máquinas convencionais. Essa estratégia permite que a ferramenta seja leve e executável em computadores pessoais de docentes, sem a necessidade de infraestrutura de servidor complexa, mantendo, contudo, a alta performance através do uso eficiente dos núcleos do processador para a verificação simultânea de múltiplos critérios (palavras-chave e frases).

Em síntese, enquanto a literatura avança em direção a modelos probabilísticos complexos e pesados, o *TextFlowReduce* posiciona-se como uma solução determinística, transparente e performática, focada na autonomia do professor para definir critérios explícitos de correção.

4 METODOLOGIA

A metodologia adotada para o desenvolvimento da biblioteca *TextFlowReduce*, proposta neste trabalho, baseia-se no modelo cascata, uma abordagem sequencial clássica para engenharia de software. Segundo Pressman and Maxim (2016), o modelo cascata é dividido em fases bem definidas: levantamento de requisitos, análise, projeto, codificação, testes e manutenção. Essa estrutura linear foi escolhida por facilitar o controle e o planejamento detalhado de cada fase do desenvolvimento da biblioteca.

A *TextFlowReduce* foi desenvolvida em C#, visando desempenho e precisão na análise textual. A biblioteca implementa uma arquitetura modular que combina o poder das Expressões Regulares (Regex) para correspondência de padrões com a eficiência da TPL para processamento concorrente de múltiplos critérios de avaliação.

4.1 Definição dos Requisitos

Na primeira etapa, foram identificados os requisitos funcionais e não funcionais da biblioteca. Entre os requisitos funcionais, destacam-se:

- Permitir a configuração de listas de palavras-chave obrigatórias, opcionais e frases exatas.
- Realizar a normalização automática do texto (remoção de acentos e conversão de caixa).
- Calcular um escore final baseado em pesos ponderados definidos pelo usuário.
- Gerar um relatório detalhado contendo os elementos encontrados e ausentes na resposta.

Como requisitos não funcionais, foram definidos:

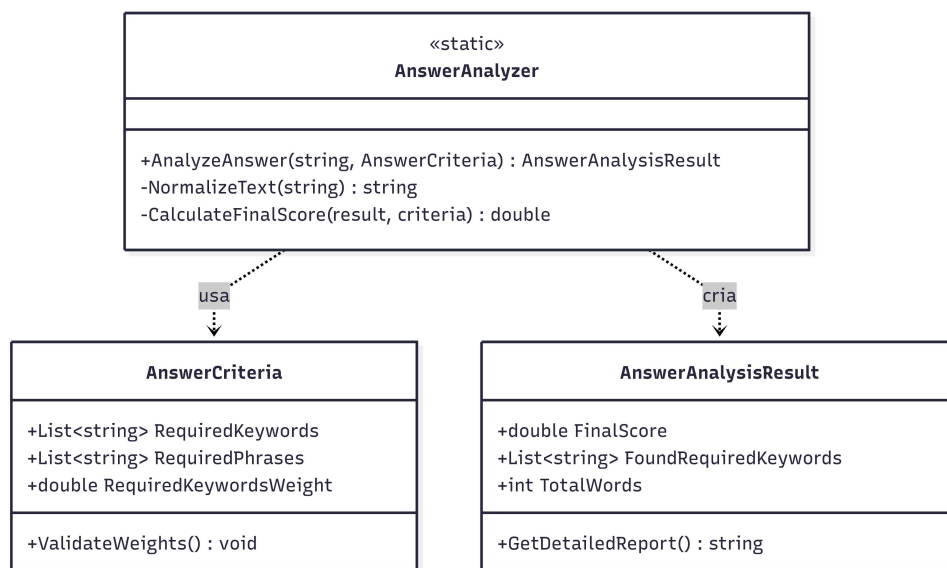
- Desempenho otimizado através do uso de paralelismo de dados na verificação de critérios.
- Segurança em ambientes concorrentes (Thread-Safe) durante a análise.
- Código limpo e tipado, utilizando recursos modernos da linguagem C#.

O modelo de dados deve incorporar o conceito pedagógico de 'Rubrica de Avaliação'. O objeto *AnswerCriteria* (detalhado na seção seguinte) funciona como uma rubrica digital, onde os critérios de correção são estruturados previamente. Para mitigar vieses interpretativos, a definição dessas rubricas deve ser fundamentada em bibliografia padrão da área, evitando que a avaliação reflita apenas a preferência lexical isolada de um único docente.

4.2 Análise e Projeto do Sistema

A arquitetura da biblioteca *TextFlowReduce* foi organizada em módulos independentes, priorizando a separação de responsabilidades entre os dados de configuração e a lógica de execução. O projeto segue rigorosamente os princípios de orientação a objetos para garantir alta coesão e baixo acoplamento entre os componentes. A estrutura fundamental do sistema e as interações entre seus componentes estão detalhadas no diagrama de classes apresentado na Figura 1.

Figura 1 – Diagrama de Classes da Biblioteca TextFlowReduce.



Fonte: O autor (2026).

A estrutura do sistema fundamenta-se em três pilares principais:

A classe `AnswerCriteria.cs` atua como o modelo de configuração do sistema. Ela é responsável por armazenar as regras de validação que definem o gabarito de correção, incluindo as listas de palavras-chave (obrigatórias e opcionais) e frases técnicas. Além disso, esta classe gerencia os pesos ponderados atribuídos a cada categoria, possuindo métodos internos para validar se a soma das frações de pontuação totaliza a unidade.

O núcleo da biblioteca reside na classe `AnswerAnalyzer.cs`. Diferente das demais, esta classe concentra a inteligência do sistema e a orquestração do fluxo de análise. Ela implementa métodos estáticos que realizam a normalização do texto, a execução paralela das verificações via TPL e o processamento das expressões regulares. Sua principal função é transformar uma entrada bruta de texto e um conjunto de critérios em um objeto de resultado processado.

Por fim, a classe `AnswerAnalysisResult.cs` funciona como um Objeto de Transferência de Dados (DTO). Sua função é encapsular de forma organizada todos os dados gerados após o processamento, como os scores parciais e finais, contagem de elementos gramaticais e as listas de termos encontrados ou ausentes. Esta classe facilita a exportação de relatórios detalhados para o usuário final.

O fluxo de execução, conforme ilustrado pelas associações no diagrama, inicia-se com a chamada ao método `AnalyzeAnswer`. O sistema valida as entradas contra as regras de `AnswerCriteria`, processa os dados de forma concorrente para otimizar o desempenho e, ao término, consolida toda a inteligência da análise em uma instância de `AnswerAnalysisResult`.

4.3 Desenvolvimento

A implementação da biblioteca utilizou recursos avançados do ecossistema .NET para atender aos requisitos de performance e precisão:

- **Paralelismo de Dados:** A verificação das listas de palavras-chave e frases é realizada utilizando `Parallel.ForEach`. Isso permite que o sistema verifique múltiplos termos simultaneamente, aproveitando os múltiplos núcleos do processador, o que é especialmente vantajoso quando o gabarito de correção é extenso.
- **Coleções Thread-Safe:** Para garantir a integridade dos dados durante a execução paralela, foram utilizadas coleções do tipo `ConcurrentBag<string>`. Essas estruturas permitem a inserção segura de itens por múltiplas *threads* sem a necessidade de bloqueios manuais (*locks*), evitando condições de corrida.
- **Expressões Regulares (Regex):** A busca por palavras-chave utiliza a classe `Regex` com o padrão de Word Boundary (`\b`). Isso assegura que termos como "pato" não sejam incorretamente identificados dentro de palavras como "sapato", garantindo a precisão semântica da análise.
- **Normalização Textual:** Antes da análise, o texto é submetido a um processo de normalização que converte todos os caracteres para minúsculas e remove diacríticos (acentos), utilizando a normalização `Unicode FormD`.

O cálculo do escore final é realizado através de uma média ponderada, onde os pesos definidos pelo usuário para palavras obrigatórias, frases e palavras opcionais são aplicados sobre a porcentagem de acertos em cada categoria. O resultado é arredondado para duas casas decimais, fornecendo uma métrica clara de aderência da resposta aos critérios esperados.

4.4 Disponibilização e Manutenção

A biblioteca *TextFlowReduce* foi projetada para ser facilmente integrada em outros sistemas educacionais. A manutenção do código é facilitada pela sua estrutura modular, permitindo:

- A adição de novos tipos de analisadores (ex: análise sintática) sem alterar a lógica existente.
- O ajuste fino dos algoritmos de normalização para suportar outros idiomas.
- A expansão das métricas de relatório conforme a necessidade pedagógica.

A solução entrega uma ferramenta robusta que combina a objetividade algorítmica com a flexibilidade necessária para a avaliação de respostas discursivas.

4.5 Procedimentos de Coleta de Dados e Perfil dos Participantes

Para validar a eficácia da biblioteca *TextFlowReduce* na quantificação objetiva de conhecimento técnico, foi realizado um experimento prático envolvendo a aplicação de um questionário dissertativo. O instrumento de coleta foi composto por 10 questões fundamentais de computação, selecionadas para abranger diferentes áreas de conhecimento:

1. O que é uma classe em POO?
2. Explique o conceito de Herança.
3. O que é uma Chave Primária?
4. O que caracteriza o Acoplamento?
5. Qual a função do protocolo HTTP?
6. Qual o papel da Memória RAM?
7. Defina Coesão em um código.
8. O que é Polimorfismo?
9. O que é o Encapsulamento em Java ou C#?
10. Explique o conceito de Recursividade.

A condução do experimento seguiu os seguintes critérios:

- **Perfil dos Participantes:** Foram consultados 30 voluntários com diferentes níveis de expertise. A amostra foi composta por: 5 profissionais de TI, 2 de Medicina, 3 de Engenharia Mecânica, 3 de Engenharia de Produção, 2 de Psicologia, 1 de Direito, 1 de Enfermagem e 13 participantes sem área de atuação específica definida.
- **Crterios de Seleção:** A seleção focou intencionalmente em pessoas de áreas externas à

Tecnologia da Informação, visando testar a precisão da ferramenta em identificar lacunas conceituais e separar o conhecimento técnico especializado do senso comum.

- **Coleta e Digitalização:** O processo de coleta ocorreu de forma híbrida: 17 testes foram realizados manualmente (manuscritos) e posteriormente digitalizados para processamento. Os 13 testes restantes foram coletados em formato digital. Todas as respostas foram submetidas ao motor de análise da biblioteca para a extração automática de pontuação baseada nos critérios configurados.

5 RESULTADOS E DISCUSSÕES

Nesta seção, são apresentados e discutidos os resultados obtidos a partir da aplicação da ferramenta *TextFlowReduce* sobre um conjunto de respostas discursivas. O experimento envolveu 30 participantes com diferentes níveis de escolaridade e áreas de formação, não restritos à computação. Esses participantes, denominados "estudantes" para fins deste trabalho, responderam a um questionário com 10 questões sobre conhecimentos gerais em computação. O baixo nível de acerto geral serviu para demonstrar a capacidade da ferramenta em quantificar objetivamente o nível de conhecimento técnico em um grupo heterogêneo, processando as respostas com base na presença rigorosa de palavras-chave e frases relevantes.

5.1 Análise Geral do Desempenho dos Alunos

A análise inicial focou em obter uma visão macro do desempenho do grupo. A média geral da turma foi de **4,93/100**, com o melhor desempenho individual atingindo **24,33/100**. A taxa de aprovação, considerando uma nota de corte de 70, foi de 0%, indicando uma dificuldade técnica generalizada nas respostas.

5.1.1 Estatísticas Descritivas Gerais

As principais métricas descritivas das pontuações finais dos 30 estudantes estão consolidadas na Tabela 1. A mediana (2,67) e o primeiro quartil (0,00) muito baixos indicam que a maioria dos estudantes obteve pontuações próximas de zero. O desvio padrão de 6,63 aponta para uma variabilidade considerável entre os participantes, embora a maior parte das notas esteja concentrada na faixa inferior.

Tabela 1 – Estatísticas descritivas das pontuações gerais

Métrica	Valor
Média (μ)	4,93
Desvio Padrão (s)	6,63
Primeiro Quartil (Q_1)	0,00
Terceiro Quartil (Q_3)	6,08
Mínimo	0,00
Máximo	24,33
Intervalo Interquartil (IQR)	6,08

Fonte: O autor (2026).

5.1.2 Distribuição de Pontuações

A distribuição das pontuações (Tabela 2) confirma a concentração de notas na faixa mais baixa. Cerca de 93,3% dos estudantes (28 de 30) obtiveram pontuação inferior a 20. A ausência de pontuações acima de 40 reforça a complexidade do teste e a precisão da ferramenta em não atribuir pontos a respostas que não atingiram os critérios técnicos mínimos definidos.

Tabela 2 – Distribuição de pontuações em faixas

Faixa de Score	Nº de Estudantes	Percentual
0-20	28	93,3%
20-40	2	6,7%
40-60	0	0,0%
60-70	0	0,0%
70-85	0	0,0%
85-100	0	0,0%

Fonte: O autor (2026).

5.2 Análise por Área de Conhecimento e por Questão

A Tabela 3 detalha o desempenho médio por área de conhecimento. As áreas de **Banco de Dados** (7,33) e **Programação** (5,29) apresentaram as maiores médias, enquanto **Redes** (2,44) foi a área com o menor desempenho médio. O alto desvio padrão em todas as áreas indica que a performance variou significativamente mesmo dentro de um mesmo tópico.

Tabela 3 – Estatísticas por área de conhecimento

Área	Média	Mediana	Desvio Padrão
B. Dados	7,33	0,00	12,78
Hardware	5,33	0,00	11,47
Progr.	5,29	0,00	9,60
Eng. Soft.	3,89	0,00	10,61
Redes	2,44	0,00	4,94

Fonte: O autor (2026).

A análise individual por questão (Tabela 4) revela que a questão 08 ("O que é Polimorfismo?") teve o melhor desempenho relativo (média 9,11), enquanto a questão 02 ("Conceito de Herança") teve o pior (média 0,89). Isso evidencia como a ferramenta identifica com precisão quais conceitos técnicos foram efetivamente mobilizados pelos alunos.

Tabela 4 – Desempenho por questão

Q	Área	Média Score	Média Chaves	Média Frases	Taxa Acima Média
01	Progr.	3,11/100	0,13	0,00	13,3%
02	Progr.	0,89/100	0,07	0,00	6,7%
03	B. Dados	7,33/100	0,40	0,07	33,3%
04	Eng. Soft.	6,44/100	0,33	0,07	20,0%
05	Redes	2,44/100	0,13	0,00	13,3%
06	Hardware	5,33/100	0,20	0,13	20,0%
07	Eng. Soft.	1,33/100	0,00	0,07	6,7%
08	Progr.	9,11/100	0,33	0,13	40,0%
09	Progr.	6,22/100	0,33	0,00	20,0%
10	Progr.	7,11/100	0,33	0,07	26,7%

Fonte: O autor (2026).

5.3 Precisão da Análise e Avaliação Qualitativa

A eficácia da ferramenta foi validada pela sua capacidade de identificar termos técnicos e frases específicas com alto grau de detalhamento. A análise demonstrou que a pontuação final está diretamente vinculada à precisão terminológica do estudante. O uso de frases exatas e palavras-chave obrigatórias permitiu uma avaliação determinística, onde a pontuação dependeu da mobilização correta do vocabulário técnico especializado.

A análise qualitativa dos "pontos de atenção" gerados revelou lacunas conceituais específicas. Por exemplo, na questão sobre "Herança"(Q02), a maioria dos estudantes falhou ao não utilizar os termos essenciais "classe base" e "derivada". Na questão sobre "Acoplamento"(Q04), a palavra "dependência" foi frequentemente omitida. O rigor na exigência desses termos tornou a análise extremamente precisa, funcionando como um diagnóstico fidedigno das dificuldades do grupo.

5.4 Discussão dos Resultados

Os resultados demonstram que a ferramenta proposta realiza uma análise técnica robusta. A baixa performance geral dos participantes não invalida o sistema; pelo contrário, reforça sua utilidade como um instrumento de diagnóstico capaz de separar o senso comum do conhecimento técnico especializado.

O baixo desempenho médio (4,93/100) levanta a discussão sobre o equilíbrio entre rigorosidade e viés de 'palavra-chave exata'. Embora a precisão técnica seja desejável, é possível

que o sistema gere 'falsos negativos' ao não reconhecer sinônimos válidos. Nesse contexto, propõe-se que a ferramenta seja utilizada como um diagnóstico inicial, onde o professor atue revisando os resultados para ajustar a sensibilidade pedagógica do motor de busca, calibrando-o para aceitar variações linguísticas legítimas.

A metodologia de avaliação mostra-se precisa ao focar em critérios altamente específicos. Ao exigir palavras e frases obrigatórias, a ferramenta elimina ambiguidades e fornece um mapeamento exato de quais conceitos foram compreendidos. A capacidade de detalhar o desempenho por estudante e por área de conhecimento oferece um feedback valioso que permite direcionar o processo de ensino-aprendizagem com base em dados objetivos.

Conclui-se que o experimento valida a eficácia da ferramenta para automatizar a análise de respostas discursivas, fornecendo dados quantitativos e qualitativos precisos que seriam difíceis de obter por meio de uma correção manual subjetiva.

6 CONCLUSÃO

Este trabalho se propôs a desenvolver e validar a biblioteca *TextFlowReduce*, capaz de automatizar a análise de respostas textuais discursivas, com o objetivo de reduzir a subjetividade e o tempo despendido em correções manuais. A metodologia fundamentou-se na avaliação quantitativa de textos a partir de critérios rigorosos e customizáveis, como a presença de palavras-chave e frases técnicas específicas.

Para validar a eficácia da solução, conduziu-se um experimento no qual a ferramenta analisou as respostas de 30 participantes com perfis diversos, submetidos a um questionário de conceitos fundamentais de computação. Os resultados obtidos, embora demonstrem um baixo desempenho geral do grupo (média de 4,93/100), foram fundamentais para atestar a principal capacidade da ferramenta: a de quantificar objetivamente o nível de conhecimento técnico e diagnosticar com precisão as lacunas conceituais por meio de uma análise terminológica detalhada.

6.1 Limitações e Trabalhos Futuros

A principal limitação do método atualmente reside na sua dependência da qualidade dos critérios pré-definidos. A eficácia da análise está diretamente ligada à escolha cuidadosa das palavras-chave e frases, tendo em vista que a ferramenta não possui a capacidade de interpretar o contexto semântico em que um vocábulo é utilizado.

Adicionalmente, surge o desafio de analisar respostas criativas, onde o respondente explica conceitos corretamente utilizando analogias ou linguagem coloquial, sem recorrer estritamente à terminologia técnica. Como trabalhos futuros, sugere-se a integração de técnicas de Processamento de Linguagem Natural (PLN) e conexão com dicionários de sinônimos ou APIs de Inteligência Artificial. Essa evolução permitiria que a ferramenta reconhecesse a validade semântica de uma explicação criativa mesmo na ausência de termos técnicos pré-cadastrados.

Outra possibilidade de expansão seria o desenvolvimento de uma interface gráfica (UI) intuitiva, permitindo que profissionais de áreas não-técnicas gerenciassem seus próprios critérios de avaliação de forma simplificada.

Para democratizar a criação de critérios e reduzir o viés humano na configuração do gabarito, trabalhos futuros devem incluir a integração com dicionários de sinônimos e mecanismos de validação por pares. Isso permitiria que múltiplos especialistas colaborassem na

definição das rubricas, garantindo uma avaliação mais plural e semanticamente abrangente.

6.2 Considerações Finais

Os resultados demonstram que a ferramenta proposta realiza uma análise técnica robusta. A metodologia de avaliação mostra-se precisa ao focar em critérios altamente específicos, mas também flexível o suficiente para mensurar o entendimento parcial. Através do sistema de pesos, a ferramenta identifica quando um aluno possui parte do contexto correto para a resposta, não penalizando de forma binária (certo ou errado), mas graduando o escore conforme os elementos identificados.

A capacidade de detalhar o desempenho por estudante e por área de conhecimento oferece um feedback valioso que permite direcionar o processo de ensino-aprendizagem com base em dados objetivos. Conclui-se que o experimento valida a eficácia da *TextFlowReduce* para automatizar a análise de respostas discursivas, fornecendo dados precisos que seriam difíceis de obter por meio de uma correção manual subjetiva.

REFERÊNCIAS

- ALBAHARI, J. **Threading in C#**. [S.l.]: O'Reilly Media, Inc., 2012. Disponível em: <http://www.albahari.com/threading/>.
- BURROWS, S.; GUREVYCH, I.; STEIN, B. The eras and trends of automatic short answer grading. **International Journal of Artificial Intelligence in Education**, v. 25, n. 1, p. 60–117, 2015.
- CAMARGO, B. V.; JUSTO, A. M. Iramuteq: um software gratuito para análise de dados textuais. **Temas em Psicologia**, v. 21, n. 2, p. 513–518, 2013. Available at: <https://pepsic.bvsalud.org/scielo.php?pid=S1413-389X2013000200016&script=sci_abstract>.
- CASELI, H.; NUNES, M. o. **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português**. 2. ed. BPLN, 2024. Disponível em: <https://brasileiraspln.com/livro-pln/2a-edicao>. Available at: <<https://brasileiraspln.com/livro-pln/2a-edicao>>.
- CECCHINI, D. **Processing Text at Scale with Spark NLP: Real Examples**. 2023. <<https://www.johnsnowlabs.com/spark-nlp-processing-examples/>>.
- CGCOM/CAPES. **Brasil publicou quase 157 mil artigos em 2023**. 2023. <<https://www.gov.br/capes/pt-br/assuntos/noticias/brasil-publicou-quase-157-mil-artigos-em-2023>>.
- GALHARDI, L. B.; BRANCHER, J. D. Avaliação automática de respostas dissertativas: uma revisão sistemática. **Revista Brasileira de Informática na Educação**, v. 26, n. 01, p. 1, 2018.
- GU, X.; ZHANG, W.; SUN, Y. **Automatic Evaluation of Tutorial Strategies Using GPT-3.5**. 2025. <<https://arxiv.org/abs/2503.01234>>. Preprint arXiv:2503.01234.
- KUCKARTZ, U. **Qualitative Text Analysis: A Guide to Methods, Practice & Using Software**. SAGE Publications Ltd, 2014. Available at: <<https://doi.org/10.4135/9781446288719>>.
- MOHLER, M.; BUNESCU, R.; MIHALCEA, R. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: LIN, D.; MATSUMOTO, Y.; MIHALCEA, R. (Ed.). **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**. Portland, Oregon, USA: Association for Computational Linguistics, 2011. p. 752–762. Available at: <<https://aclanthology.org/P11-1076/>>.
- NERY, F. As marcas da subjetividade no texto da redação do enem. **Revista da ANPOLL**, n. 42, p. 53–67, 2017.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software - 8ª Edição**. [s.n.], 2016. ISBN 9788580555349. Available at: <<https://books.google.com.br/books?id=wexzCwAAQBAJ>>.
- RICOEUR, P. **From Text to Action: Essays in Hermeneutics II**. [S.l.]: Northwestern University Press, 1991.
- RODRIGUES, F.; ARAUJO, L. Um sistema para avaliação automática de respostas curtas em português. In: **Anais do IX Encontro Nacional de Inteligência Artificial**. [S.l.: s.n.], 2012.

Apêndice A – CRONOGRAMA DE ATIVIDADES

O planejamento e a execução das atividades constituem um pilar essencial para a viabilização de projetos fundamentados de software, permitindo a governança sobre o ciclo de vida de desenvolvimento. Diferente de uma abordagem puramente prospectiva, o cronograma aqui apresentado assume um caráter descritivo e documental, registrando o itinerário técnico percorrido durante as etapas de desenvolvimento do trabalho.

A organização reflete a sequência lógica de construção do **TextFlowReduce**, partindo da elicitación de requisitos e análise do problema até a implementação do motor de processamento paralelo e validação das pontuações.

Ainda sim, o planejamento contempla o período destinado ao refinamento pós-defesa, etapa crítica para a incorporação de contribuições da banca examinadora. Tal previsão garante que o rigor científico e a coesão técnica do sistema sejam preservados na versão definitiva do documento. A Tabela 5 sintetiza a distribuição das atividades e o esforço empregado em cada fase do projeto.

Tabela 5 – Cronograma de Atividades TCC

Atividades	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ	JAN
Elicitación de Requisitos	x									
Análise e Projeto	x									
Desenvolvimento da Biblioteca		x	x				x	x		
Validação (Parcial/Final)				x					x	
Refinamento de Requisitos					x					
Análise e Projeto Final						x				
Defesa do TCC										x
Ajustes Pós Banca										x

Fonte: O autor (2026).