



UNIVERSIDADE DO ESTADO DO AMAZONAS -UEA
ESCOLA SUPERIOR DE TECNOLOGIA - EST
CURSO DE ENGENHARIA DA COMPUTAÇÃO



**USO DE TÉCNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL E
MACHINE LEARNING NA CRIAÇÃO DE UM MODELO DE RECOMENDAÇÃO DE
VAGAS DE EMPREGABILIDADE: UM ESTUDO DE CASO NA PLATAFORMA MAIS
ACESSO CONECTA**

ERIC DIAS PERIN

Manaus – AM

Junho de 2026

ERIC DIAS PERIN

**USO DE TÉCNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL E
MACHINE LEARNING NA CRIAÇÃO DE UM MODELO DE RECOMENDAÇÃO DE
VAGAS DE EMPREGABILIDADE: UM ESTUDO DE CASO NA PLATAFORMA MAIS
ACESSO CONECTA**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de Engenharia de Computação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Computação.

Orientador(a): *Prof^a.Dr^a.* Áurea Hiléia da Silva Melo

Manaus – AM

Junho de 2026

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

P445u	Perin, Eric Dias Uso de técnicas de Processamento de Linguagem Natural e Machine Learning na criação de um modelo de recomendação de vagas de empregabilidade: Um estudo de caso na plataforma Mais Acesso Conecta / Eric Dias Perin. Manaus : [s.n], 2026. 79 f.: color.; 21.0 cm. TCC - Graduação em Engenharia de Computação- Universidade do Estado do Amazonas, Manaus, 2026. Inclui Bibliografia. Inclui Apêndice. Inclui Anexo. Orientador: Áure Hileia da Silva Melo. 1. Sistemas de Recomendação. 2. Sentence-BERT. 3. Zero-Shot Learning. 4. Empregabilidade PCD. 5. Processamento de Linguagem Natural. I. Áure Hileia da Silva Melo (Orient.) II. Universidade do Estado do Amazonas. III. Título CDU(1997)62:004
-------	---



FOLHA DE APROVAÇÃO

USO DE TECNICAS DE PROCESSAMENTO DE LINGUAGEM NATURAL E MACHINE LEARNING NA CRIAÇÃO DE UM MODELO DE RECOMENDAÇÃO DE VAGAS DE EMPREGABILIDADE: UM ESTUDO DE CASO NA PLATAFORMA MAIS ACESSO CONECTA

ERIC DIAS PERIN

Trabalho de Conclusão de Curso defendido e aprovado pela banca avaliadora constituída pelos professores:

Profª Dra. Áurea Hiléia da S. Melo
Presidente

Prof. Dra. Elloá Guedes
Avaliador

Profª. Dr. Sergio Cleger Tamayo
Avaliador

Manaus, 23 de junho de 2026

Resumo

A inserção de Pessoas com Deficiência (PCD) no mercado de trabalho enfrenta barreiras que vão além da disponibilidade de vagas, esbarrando na dificuldade de conexão eficiente entre perfis e oportunidades. A plataforma *Mais Acesso Conecta* centraliza essas vagas, porém o processo de busca dependia de navegação e triagem manuais, sendo muito trabalhoso e ineficiente tanto para os candidatos quanto para os recrutadores. Diante dessa limitação, tornou-se necessário o desenvolvimento de um mecanismo que recomendasse oportunidades mais parecidas e alinhadas com as reais competências de cada perfil de forma automatizada. Este trabalho apresenta o desenvolvimento de um sistema de recomendação *baseado em conteúdo*, utilizando técnicas de Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina. A abordagem adotada emprega o modelo de *embeddings* contextuais Sentence-BERT em uma estratégia de *Zero-Shot Learning*, eliminando a necessidade de treinamento supervisionado. A validação, realizada por meio de comparação com *baseline* estatístico (TF-IDF) e análise de cenários, demonstrou que a abordagem vetorial contextual é superior na distinção semântica, apresentando uma distinção significativa de *score* em relação a vagas de ruído, validando a eficácia da solução proposta para ampliar a empregabilidade de forma inteligente.

Palavras-chave: Sistemas de Recomendação; Sentence-BERT; Zero-Shot Learning; Empregabilidade PCD; Processamento de Linguagem Natural.

Abstract

The inclusion of People with Disabilities (PwD) in the labor market faces barriers that go beyond job availability, struggling with the efficient connection between profiles and opportunities. The *Mais Acesso Conecta* platform centralizes these job openings, but the search process relied on manual navigation and screening, being labor-intensive and inefficient for both candidates and recruiters. Given this limitation, it became necessary to develop a mechanism to automatically recommend similar opportunities aligned with the actual competencies of each profile. This work presents the development of a *content-based* recommendation system utilizing Natural Language Processing (NLP) and Machine Learning techniques. The adopted approach employs the Sentence-BERT contextual *embeddings* model in a *Zero-Shot Learning* strategy, eliminating the need for supervised training. The validation, performed through comparison with a statistical *baseline* (TF-IDF) and scenario analysis, demonstrated that the contextual vector approach is superior in semantic distinction, presenting a significant *score* distinction compared to noise vacancies, validating the efficacy of the proposed solution to intelligently expand employability.

Keywords: Recommender Systems; Sentence-BERT; Zero-Shot Learning; PwD Employability; Natural Language Processing.

Sumário

Lista de Tabelas	v
Lista de Quadros	vii
Lista de Figuras	viii
1 Introdução	1
1.1 Justificativa	3
1.2 Objetivo Geral	4
1.3 Objetivos Específicos	4
1.4 Metodologia	5
1.4.1 TCC1	6
1.4.2 TCC 2	9
1.4.3 Ferramentas e Tecnologias	10
2 Fundamentação Teórica	12
2.1 Inteligência Artificial e Aprendizado de Máquina	12
2.2 Processamento de Linguagem Natural (PLN)	14
2.3 Sistemas de Recomendação	16
2.3.1 O Paradigma do Recrutamento Inteligente (<i>E-recruitment</i>)	18
2.3.2 O Desafio do <i>Cold-Start</i> e a Escassez de Dados	18
2.4 Aprendizado de Representações (<i>Embeddings</i>)	19
2.4.1 A Evolução dos Embeddings	20

2.4.2	Sentence-BERT (SBERT) e Redes Siamesas	20
2.5	Métricas de Similaridade e Estratégia de Validação	22
2.5.1	Métricas de Avaliação de Ranqueamento e Recuperação de Informação	23
2.5.2	Desafios de Implantação e <i>AI Engineering</i>	24
2.5.3	Fundamentação Estatística, Ética e Testes Não Paramétricos	25
2.6	Metodologia CRISP-DM	26
2.7	Trabalhos Relacionados	28
3	Desenvolvimento do modelo de recomendação	32
3.1	Compreensão: Revisão Bibliográfica e Definição de Requisitos	32
3.1.1	Requisitos Funcionais do Algoritmo	33
3.2	Preparação dos Dados: Agregação e Pipeline de PLN	34
3.2.1	Exploração e Engenharia de Atributos	35
3.2.2	Engenharia de Atributos: Criação do <i>Corpus</i> Textual	35
3.2.3	Amostra de Dados Brutos	35
3.2.4	Pipeline de Processamento de Linguagem Natural (PLN)	36
3.2.5	Normalização e Filtragem de <i>Stopwords</i>	37
3.2.6	Filtragem de Domínio (Limpeza Agressiva)	37
3.3	Modelagem: Representação Vetorial e Algoritmo de Recomendação	39
3.3.1	Seleção do Modelo Base	40
3.3.2	Engenharia de Contexto e Priorização de Tokens	42
3.3.3	Algoritmo de Recomendação e Cálculo de Similaridade	45
3.3.4	Aplicação da Métrica	46
3.3.5	Cálculo de Score e Ranqueamento	46
3.4	Implantação: Arquitetura de Integração	47
3.4.1	Processamento Assíncrono com Mensageria e <i>Worker</i> de Indexação	49
3.4.2	Recomendação Bidirecional e Fluxos de Busca	50
3.4.3	Integração de Filtros Rígidos (<i>Hard Filters</i>)	50
3.4.4	Composição do Modelo Híbrido	50

4	Resultados e Discussão	52
4.1	Validação Global: Matriz de Confusão Semântica (<i>Heatmap</i>)	52
4.2	Validação Comparativa Controlada e o Gap Semântico	54
4.3	Avaliação Quantitativa com Interações Reais (<i>Ground Truth</i>)	54
4.4	Validação das Técnicas de Preparação de Dados	56
4.4.1	Análise de Viés e Ética Algorítmica (PCD vs. Não-PCD)	57
4.4.2	Estudo de Caso: Correspondência Funcional e Técnica	57
4.5	Estudo de Caso: Visão do Candidato e Priorização Afirmativa	58
4.6	Estudo de Caso: Busca Reversa e Visão do Recrutador	59
4.7	Considerações Finais	62

Lista de Tabelas

3.1	Benchmark Empírico de Modelos SBERT sobre Dados Reais da Plataforma . . .	41
4.1	Avaliação Baseada em Candidaturas Reais: Comparação de Modelos (N=105) . .	55

Lista de Quadros

- 2.1 Comparação entre os Trabalhos Relacionados e as Contribuições para a Pesquisa 31
- 3.1 Amostra da Base de Dados de Candidatos 36
- 3.2 Amostra da Base de Dados de Vagas 36
- 3.3 Amostra da Lista de Exclusão Personalizada (*Custom Stoplist*) 39

Lista de Figuras

1.1	Fluxograma da metodologia	6
3.1	Fluxo do Pipeline de Processamento de Linguagem Natural: transformação do texto bruto em representação semântica limpa.	38
3.2	Impacto da priorização de <i>tokens</i> sob truncamento: similaridade do perfil sinté- tico extenso com a vaga-alvo nas ordens convencional e priorizada.	44
3.3	Efeito da Engenharia de Contexto no <i>ranking</i> de recomendação sob truncamento.	45
3.4	Diagrama da arquitetura de integração da recomendação baseada em microserviços.	48
4.1	Matriz de Confusão Semântica: Média de Similaridade entre Perfis e Áreas de Vagas.	53
4.2	Estudo de Caso: Ranqueamento das Top 4 Vagas para o Candidato com Forma- ção em Mecânica.	58
4.3	Visão da a: Ranqueamento de vagas para perfil Administrativo PCD.	60
4.4	Interface do Recrutador: Detalhes da vaga de Analista de Dados.	60
4.5	Resultado da Busca Reversa: Ranqueamento de candidatos por Match Score.	61
4.6	Visão do Recrutador: Exemplo de ranqueamento para perfis com baixa aderência técnica.	63

Introdução

A inclusão de pessoas com deficiência (PCDs) no mercado de trabalho é um tema de grande relevância social e econômica. Segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE) (2023), a taxa de desemprego entre pessoas com deficiência é de 9,1%, enquanto para pessoas sem deficiência é de 8,7%. Essa diferença reflete as barreiras estruturais e sociais que dificultam a participação efetiva das PCDs no mercado de trabalho.

A partir de 1991, no Brasil, por meio da Lei de Cotas, determinou-se que empresas com 100 ou mais funcionários devem reservar entre 2% e 5% de seus postos de trabalho para pessoas com deficiência. No entanto, apesar da legislação e de avanços nas políticas de inclusão, os números de contratação ainda não alcançam as quantidades esperadas, evidenciando a necessidade de estratégias que promovam acessibilidade em processos seletivos e no ambiente corporativo.

Atualmente, plataformas como o LinkedIn, oferecem diversas ofertas e oportunidades no mercado de trabalho, porém não possuem um foco e exclusividade para atender pessoas com deficiência, o que pode resultar em uma falta de personalização a este público. Em busca de atingir esse foco de vagas inclusivas, destacam-se alternativas como o site Meu Emprego Inclusivo, que conta com 17 polos de Empregabilidade Inclusivos, no entanto todos centrados no estado de São Paulo. Outro exemplo relevante é a plataforma Inklua, de abrangência nacional, especializada em recrutamento e seleção de PCDs, oferecendo mais uma importante plataforma de vagas para esse público.

Sendo assim, na região Norte, a plataforma *Mais Acesso Conecta* (MAIS ACESSO, 2025) foi desenvolvida em parceria com o Tribunal Regional do Trabalho, procurando atender especificamente pessoas com deficiência no estado do Amazonas. A ferramenta conecta candidatos

a oportunidades de emprego, e também busca reduzir desigualdades e estimular a inclusão profissional. Diante disso, embora a plataforma atualmente já possua um expressivo cadastro de candidatos e de empresas, com suas respectivas vagas, ainda há espaço para aprimoramentos que aumentem sua eficácia. O desafio central reside na ausência de um mecanismo inteligente capaz de realizar o *match* (cruzamento) assertivo entre os candidatos e as empresas. Atualmente, há uma dificuldade evidente tanto para o profissional, que precisa encontrar as oportunidades mais alinhadas às suas competências, quanto para o recrutador, que enfrenta um processo trabalhoso e demorado para triar e identificar o talento ideal em meio a um grande volume de perfis.

Apesar da existência de ferramentas com alcance nacional ou concentradas na região Sudeste, as características próprias de cada mercado evidenciam a importância de plataformas adaptadas às realidades regionais. Nesse contexto, voltando o foco para a região Norte, a plataforma *Mais Acesso Conecta* (MAIS ACESSO, 2025) foi desenvolvida em parceria com o Tribunal Regional do Trabalho, com foco prioritário no atendimento a pessoas com deficiência no estado do Amazonas. A ferramenta conecta candidatos a oportunidades de emprego, e também busca reduzir desigualdades e estimular a inclusão profissional. Diante disso, embora a plataforma atualmente já possua um expressivo cadastro de candidatos e de empresas, com suas respectivas vagas, ainda há espaço para aprimoramentos que aumentem sua eficácia. O desafio central reside na ausência de um mecanismo inteligente capaz de realizar o *match* (cruzamento) assertivo entre os candidatos e as empresas. Atualmente, há uma dificuldade evidente tanto para o profissional, que precisa encontrar as oportunidades mais alinhadas às suas competências, quanto para o recrutador, que enfrenta um processo trabalhoso e demorado para triar e identificar o talento ideal em meio a um grande volume de perfis.

Nesse contexto, a tecnologia se mostra uma aliada essencial para ampliar a acessibilidade e conectar pessoas com deficiência a vagas de emprego. Dessa forma, este trabalho apresenta um modelo de recomendação incorporado à plataforma *Mais Acesso Conecta*, a fim de aprimorar a experiência dos usuários, aumentar a eficiência na correspondência entre candidatos e vagas e, consequentemente, contribuir para a redução da desigualdade no acesso ao mercado de trabalho.

1.1 Justificativa

A plataforma *Mais Acesso* é uma alternativa que atua com o propósito de aumentar as ofertas no ambiente de trabalho. Em seu site oficial, além de disponibilizar vagas, a plataforma apresenta uma seção de *Valores*, onde um de seus pilares é destacado na subseção “Oportunidades”. Nela, afirma-se o compromisso de “criar oportunidades transformadoras para pessoas com deficiência ou mobilidade reduzida, promovendo a inclusão e criando um ambiente mais igualitário e acessível para todos”. (MAIS ACESSO, 2025). Sendo assim, apesar da plataforma permitir a aplicação de vagas, no entanto, não possui recomendações personalizadas de vagas para os usuários, baseadas no perfil e em suas habilidades.

A plataforma possibilita que as empresas cadastradas publiquem suas vagas de emprego, especificando requisitos e o perfil do candidato desejado. Por sua vez, os usuários têm acesso a essas vagas e podem escolher aquelas de seu interesse para se candidatar. No entanto, como o processo de busca depende da navegação manual por entre um grande número de ofertas, os candidatos precisam dedicar tempo significativo para localizar, entre várias vagas existentes, aquelas mais alinhadas às suas qualificações. Do lado das empresas, a análise de um grande volume de candidaturas exige maior esforço e, em muitos casos, a alocação de mais profissionais para o processo de recrutamento, o que torna a filtragem de perfis adequados mais trabalhosa e com menor assertividade.

Diante desse cenário, o uso de Inteligência Artificial, por meio de sistemas de recomendação, mostra-se efetivo para otimizar o processo de intermediação entre candidatos e empresas. Essa abordagem permite relacionar de forma mais precisa as ofertas de emprego disponíveis na plataforma com a demanda apresentada pelos usuários, considerando seus perfis e habilidades. Nesse contexto, este trabalho se insere com o objetivo de desenvolver um modelo baseado em técnicas de *Machine Learning* e Processamento de Linguagem Natural (PLN), a fim de facilitar a vinculação entre candidatos e vagas, tornando a busca mais ágil e assertiva. Assim, justifica-se tanto pela contribuição prática de potencializar a plataforma *Mais Acesso Conecta*, ampliando seu impacto social, quanto pela contribuição acadêmica no uso da Inteligência Artificial aplicada

a contextos de acessibilidade e inclusão.

1.2 Objetivo Geral

O objetivo geral deste trabalho é desenvolver e validar um modelo de recomendação baseado em Inteligência Artificial e Processamento de Linguagem Natural utilizando modelos de linguagem pré-treinados baseados em *Transformers* e abordagens estatísticas de Processamento de Linguagem Natural para otimizar a conexão entre perfis profissionais e vagas na plataforma Mais Acesso Conecta.

1.3 Objetivos Específicos

Para alcançar o objetivo geral proposto, este trabalho apresenta os seguintes objetivos específicos:

- Levantar, organizar e pré-processar os dados de candidatos e vagas da plataforma Mais Acesso Conecta, para estruturar um *corpus* textual unificado para análise.
- Investigar e aplicar técnicas avançadas de Processamento de Linguagem Natural, com ênfase em modelos de *Embeddings* contextuais, para representação vetorial de competências profissionais.
- Implementar um algoritmo de recomendação baseado em conteúdo para ranquear vagas de acordo com a compatibilidade do perfil.
- Validar a eficácia do modelo por meio de estudos de caso e análise de cenários controlados.
- Analisar o potencial de aplicação em plataformas de empregabilidade inclusiva, propondo uma solução escalável e fundamentada em técnicas avançadas de Processamento de Linguagem Natural e modelos de *embeddings* contextuais.
- Integrar o modelo de recomendação à plataforma Mais Acesso Conecta, por meio de uma API para recomendar vagas e candidatos.

1.4 Metodologia

A metodologia deste trabalho adota uma abordagem híbrida, integrando a macroestrutura de gestão de projetos da CRISP-DM (*Cross Industry Standard Process for Data Mining*) (CHAPMAN et al., 2000) com a microestrutura técnica detalhada no *Recommender Systems Handbook* (RICCI; ROKACH; SHAPIRA, 2022). Essa combinação justifica-se pela complementaridade das referências: enquanto a primeira define o ciclo de vida do projeto (fases de Compreensão, Preparação, Modelagem, Avaliação e Implantação); a segunda fundamenta as escolhas algorítmicas de filtragem baseada em conteúdo e o Modelo de Espaço Vetorial. Essa integração assegura, simultaneamente, rigor processual e embasamento matemático na construção do motor de recomendação.

Na execução prática deste projeto, as etapas foram conduzidas de forma estruturada, com iterações de refinamento concentradas especificamente na fase de Avaliação, onde os resultados das interações reais (*ground truth*) retroalimentaram os parâmetros do modelo híbrido por meio do *grid search* de pesos.

Dessa integração estrutural e técnica, extraíram-se as etapas de desenvolvimento ilustradas no diagrama da Figura 1.1:

Dessa combinação estrutural e técnica, extraiu-se o seguinte para a formulação deste trabalho:

- **Da metodologia CRISP-DM (A Macroestrutura):** Adotou-se o ciclo de vida iterativo do projeto. Este *framework* orientou a divisão macro do trabalho nas fases de Compreensão do Negócio (Revisão e Definição de Requisitos), Preparação dos Dados (Coleta, Agregação e PLN), Modelagem (Geração de *Embeddings*), Avaliação (Validação Comparativa e Gap Semântico) e Implantação (Integração via API e Refinamento de Produção).
- **Do *Recommender Systems Handbook* (A Microestrutura Técnica):** Extraiu-se a base algorítmica e teórica do motor. Isso inclui os preceitos clássicos de Sistemas de Recomendação Baseados em Conteúdo (*Content-Based Filtering*), a utilização do Modelo de Espaço Vetorial para o cálculo de similaridade (Similaridade do Cosseno), as estratégias

de representação textual de itens/usuários e a concepção da arquitetura (unindo a IA semântica com *Hard Filters* de regras de negócio).

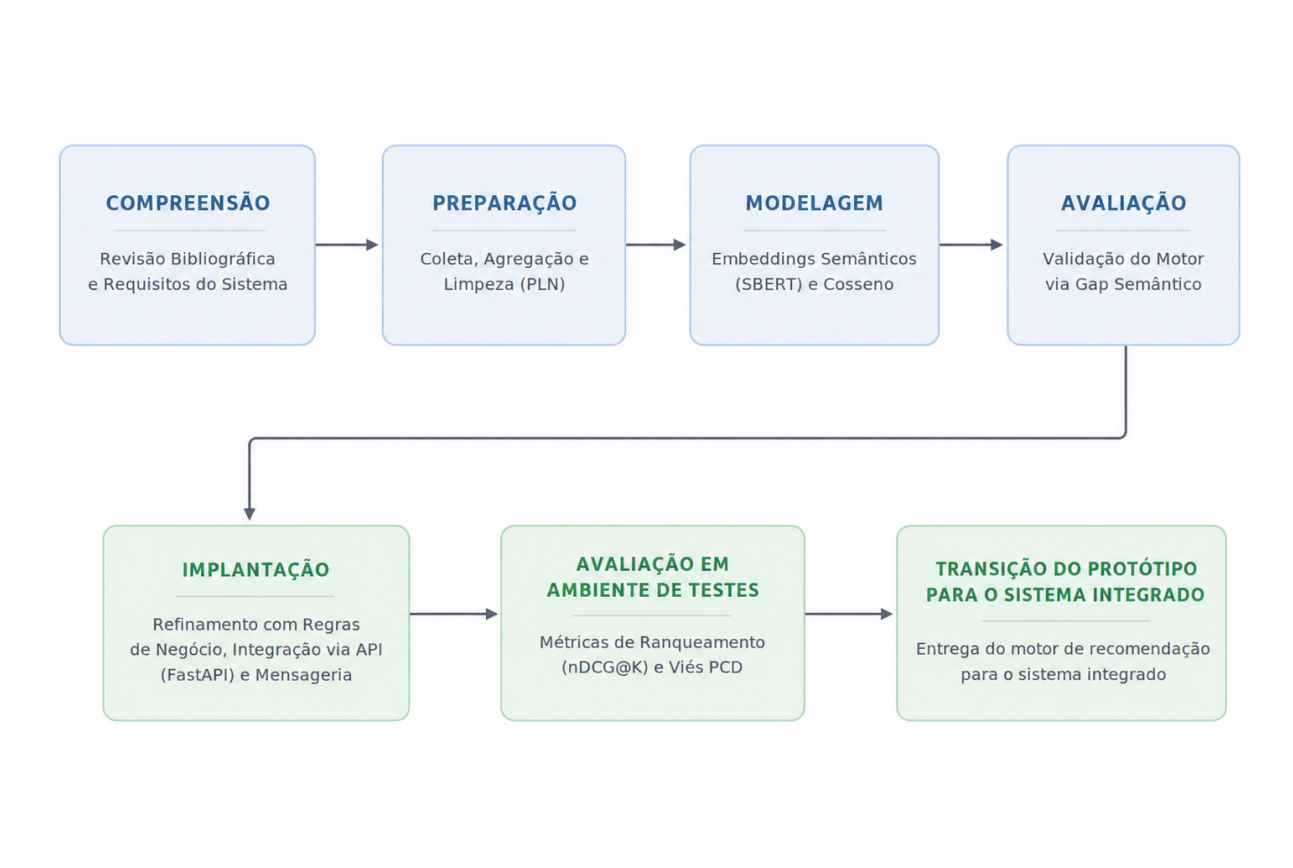


Figura 1.1: Fluxograma da metodologia
Elaborado pelo autor (2026) com o auxílio da inteligência artificial Claude.

A seguir, detalham-se as etapas metodológicas de cada fase, descrevendo os procedimentos técnicos adotados, as ferramentas empregadas e os artefatos produzidos em cada etapa da metodologia.

1.4.1 TCC1

Revisão Bibliográfica e Definição de Requisitos

- **Revisão da Literatura:** Essa atividade corresponde ao levantamento bibliográfico focado em sistemas de recomendação baseados em conteúdo, com ênfase no uso de *Deep Learning* para Processamento de Linguagem Natural (PLN), tendo como objetivo a investigação d a

arquitetura Transformer e modelos de Embeddings pré-treinados, com foco na capacidade de generalização Zero-Shot em domínios específicos.

- **Definição de Requisitos:** Esta etapa representa o mapeamento dos requisitos para o modelo de recomendação proposto, de forma que seja capaz de priorizar a competência técnica (*hard skills*) e atender às especificidades das vagas, objetivando garantir que a recomendação não seja enviesada por termos genéricos ou ruído semântico.

Coleta, Agregação e Pré-processamento dos Dados

- **Coleta e Agregação:** Etapa onde os dados são extraídos de um banco de dados contendo múltiplas tabelas (candidatos, formação, experiências, vagas). A partir de várias tabelas, é feito o desenvolvimento de uma rotina de agregação para consolidar essas informações dispersas em documentos textuais únicos. Nesta etapa, é implementada uma estratégia de Injeção de Contexto Prioritário, onde informações críticas (como Habilidades Técnicas e Título do Cargo) são posicionadas no início da sequência textual para evitar o truncamento devido ao limite de *tokens* dos modelos BERT.
- **Pré-processamento e Limpeza Avançada (PLN):** Diferente de abordagens tradicionais que removem indiscriminadamente certas classes de palavras, propõe-se uma limpeza textual estratégica para isolar competências técnicas sem perder o contexto hierárquico. As técnicas aplicadas incluem:
 - Normalização de texto e remoção de caracteres especiais;
 - Criação de *stoplists* personalizadas para remover *soft skills* (ex: “proatividade”, “comunicação”) e termos genéricos de recrutamento (ex: “vaga”, “requisitos”), visando reduzir o ruído na similaridade;
 - Filtragem de termos de localização, preservando-se, contudo, marcadores de senioridade (ex: “Júnior”, “Sênior”, “Estágio”), visto que estes termos são determinantes para a correta interpretação do nível de complexidade da vaga pelo modelo contextual.

Modelagem Vetorial e Estratégia de Representação

Esta etapa representa a substituição do uso de classificadores clássicos ou treinamentos supervisionados (que requerem grandes volumes de dados rotulados) por uma abordagem focada em similaridade semântica vetorial utilizando modelos de estado da arte.

- **Seleção do Modelo Base:** Utiliza-se o Sentence-BERT (modelo `paraphrase-multilingual-mpnet-base-v2`), escolhido por sua robustez multilíngue e capacidade de gerar representações vetoriais densas para sentenças completas. Optou-se por uma abordagem *Zero-Shot Learning*, aproveitando o vasto pré-treinamento do modelo em bilhões de pares de texto para capturar similaridades semânticas intrínsecas sem a necessidade de *fine-tuning* específico, evitando assim a introdução de vieses através de dados sintéticos.
- **Engenharia de Contexto (Estruturação dos Dados):** Em substituição ao treinamento do modelo, o foco metodológico recaiu sobre a otimização da estrutura de entrada dos dados. A concatenação dos campos textuais foi desenhada para maximizar a densidade de informação relevante dentro da janela de atenção do modelo, garantindo que a comparação vetorial priorize *hard skills* e experiência prática sobre informações acessórias.

Validação e Análise de Similaridade

- **Cálculo de Similaridade:** A compatibilidade entre candidatos e vagas é calculada utilizando a Similaridade do Cosseno entre os vetores gerados pelo modelo SBERT.
- **Validação Comparativa (Baseline):** Etapa responsável pelos experimentos comparando a abordagem proposta (SBERT Zero-Shot) com métodos estatísticos tradicionais (TF-IDF). Compara-se a abordagem proposta (SBERT Zero-Shot) com métodos estatísticos tradicionais (TF-IDF), utilizando o GAP Semântico como métrica de qualidade em cenários controlados.
- **Análise de Cenários e Gap Semântico:** A validação é conduzida submetendo perfis de controle (ex: “Desenvolvedor Full Stack”) ao sistema e verificando se o *ranking* de vagas

prioriza oportunidades da mesma área técnica (Nicho) em detrimento de vagas operacionais com terminologia similar (Ruído), medindo-se a magnitude da diferença de *score* (Gap Semântico) entre estes grupos.

1.4.2 TCC 2

Integração e Avaliação em Produção (Fase Final)

Nesta fase, o foco metodológico recai sobre a operacionalização e integração do modelo validado. As etapas compreendem:

Integração na Plataforma e Arquitetura de Microserviços

O motor de recomendação é encapsulado em uma API REST independente construída em FastAPI (*recomendacao_api*), consumida de forma assíncrona pelo *backend* principal da plataforma *Mais Acesso Conecta*.

Processamento Assíncrono e Indexação Pontual

Arquitetura orientada a eventos com mensageria (RabbitMQ) e *Worker* especializado, garantindo a geração e atualização contínua de *embeddings* de candidatos e vagas sem impactar a latência transacional da plataforma.

Filtros Rígidos de Negócio (*Hard Filters*)

Camadas lógicas complementares ao modelo semântico, garantindo o respeito a restrições de negócio como vagas exclusivas PCD e requisitos obrigatórios por meio de filtros aplicados diretamente na consulta ao banco de dados.

Recomendação Bidirecional

O sistema disponibiliza dois fluxos de consulta: recomendação de vagas para candidatos e ranqueamento de candidatos para recrutadores, utilizando o mesmo núcleo de similaridade semântica.

Avaliação com *Ground Truth* Real e *Grid Search* de Pesos

A eficácia do modelo é mensurada sobre candidatos reais com histórico de candidaturas na plataforma, utilizando métricas de Recuperação de Informação (*Precision@K*, *Recall@K* e *nDCG@K*). Os pesos do modelo híbrido são determinados empiricamente por *grid search*, eliminando a subjetividade da escolha manual de parâmetros.

Validação das Contribuições Metodológicas (*Ablation Studies*)

Experimentos de *ablation* mensuram o impacto isolado de cada contribuição metodológica, removendo individualmente a *Custom Stoplist* e a Engenharia de Contexto e avaliando a variação no GAP Semântico do sistema.

1.4.3 Ferramentas e Tecnologias

Para a execução e o desenvolvimento do motor de recomendação, emprega-se um *stack* de tecnologias e bibliotecas amplamente utilizadas na área de Ciência de Dados e Processamento de Linguagem Natural (PLN).

- **Linguagem: Python.** A linguagem Python foi escolhida devido ao seu vasto ecossistema de bibliotecas especializadas em Machine Learning e PLN, além de sua sintaxe clara que facilita desde a prototipação e o desenvolvimento de rotinas complexas até a construção de APIs e microsserviços.
- **Bibliotecas de IA/PLN:** Sentence-Transformers (SBERT), Scikit-learn, NLTK, Pandas/NumPy. O SBERT é o núcleo do sistema, sendo crucial para a geração de Embeddings vetoriais densos. O NLTK é utilizado nas rotinas iniciais de pré-processamento e limpeza do texto. Pandas e NumPy são empregados para manipulação eficiente dos dados, enquanto Scikit-learn fornece as funcionalidades necessárias para o cálculo da Similaridade do Cosseno e métricas de comparação.
- **Framework de Microsserviços: FastAPI.** Framework web moderno e de alto desempenho utilizado na linguagem Python para a construção do microsserviço isolado da inteligência

artificial (*recomendacao_api*), responsável por disponibilizar os pontos de acesso HTTP bidirecionais da plataforma.

- **Mensageria e Processamento Assíncrono: RabbitMQ.** Atuando como *message broker*, foi implementado para gerenciar a fila de indexação pontual, orquestrando as tarefas pesadas do modelo SBERT em segundo plano (*Worker*) e evitando o bloqueio das operações transacionais do usuário no *frontend*.
- **Bancos de Dados: PostgreSQL.** Sistema de gerenciamento de banco de dados relacional adotado para sustentar a arquitetura de domínios separados. É utilizado tanto para o armazenamento dos dados mestres no backend principal (*Postgres Principal*) quanto para o armazenamento isolado das representações vetoriais da inteligência artificial (*Postgres Embeddings*).
- **Containerização e Empacotamento: Docker.** Empregado para o empacotamento do microserviço de recomendação e de todas as suas dependências em um contêiner isolado. O uso dessa tecnologia garantiu a padronização e a paridade de configuração entre os ambientes de desenvolvimento e produção, facilitando a entrega (*handoff*) segura do artefato de software para a equipe responsável pela infraestrutura e implantação (*deploy*) final na plataforma *Mais Acesso Conecta*.
- **Ambientes de Execução: Google Colab e Ambiente Local de Testes.** O Google Colab foi o ambiente utilizado para a fase de experimentação, validação matemática e geração das métricas do modelo; operando como precursor do ambiente final de produção construído para a implantação da arquitetura de integração no sistema *Mais Acesso Conecta*. Foram realizados também experimentos com o servidor local, utilizando o Docker Desktop.

A combinação dessas ferramentas e tecnologias fundamenta a viabilidade técnica e a escalabilidade do sistema proposto. Dessa forma, o conjunto tecnológico adotado garante não apenas o rigor na modelagem matemática e vetorial, mas também a eficiência, a resiliência e a integração fluida do motor de recomendação à arquitetura da plataforma *Mais Acesso Conecta*.

Fundamentação Teórica

Neste capítulo, são apresentados os principais conceitos teóricos que embasam o desenvolvimento deste trabalho. As seções a seguir apresentam as bases da Inteligência Artificial (IA) e do Aprendizado de Máquina (Machine Learning), os fundamentos do Processamento de Linguagem Natural (PLN), bem como os princípios de funcionamento dos Sistemas de Recomendação e sua aplicação em contextos de empregabilidade. A compreensão desses tópicos é essencial para a construção de uma solução capaz de relacionar perfis de candidatos a vagas de emprego de forma automatizada e inteligente.

2.1 Inteligência Artificial e Aprendizado de Máquina

A Inteligência Artificial (IA) é uma área da ciência da computação dedicada ao desenvolvimento de sistemas capazes de realizar tarefas que, tipicamente, exigem inteligência humana, como raciocínio, percepção, aprendizado e tomada de decisão. A IA busca reproduzir ou simular a capacidade humana de pensar e agir de maneira inteligente, aplicando algoritmos e modelos computacionais para resolver problemas complexos de forma autônoma (RUSSELL; NORVIG, 2021).

No centro da IA moderna está o Aprendizado de Máquina (*Machine Learning* - ML). De acordo com Alpaydin (2020), o ML tem como objetivo programar computadores para utilizar dados de exemplo ou experiências passadas para resolver problemas específicos. Em contraste com sistemas tradicionais, que seguem regras explícitas, os modelos de ML aprendem padrões a partir dos dados, o que lhes permite fazer previsões, classificações ou decisões automáticas de forma mais adaptativa e eficiente.

O aprendizado de máquina é geralmente categorizado em três vertentes principais (ALPAY-DIN, 2020):

- **Supervisionado:** O modelo é treinado com exemplos de entrada e saída rotulados, buscando aprender uma função que relacione os dois (ex.: classificar currículos como compatíveis ou não com uma vaga).
- **Não Supervisionado:** O modelo trabalha com dados sem rótulos, identificando automaticamente agrupamentos, similaridades ou padrões ocultos (ex.: segmentação de perfis profissionais).
- **Por Reforço:** Envolve um agente que interage com um ambiente e aprende a partir de recompensas ou punições, aprimorando suas ações ao longo do tempo.

As aplicações de IA e ML se expandiram significativamente nos últimos anos, especialmente em Sistemas de Recomendação. Plataformas amplamente utilizadas, como Netflix, Amazon e LinkedIn, empregam algoritmos de ML para compreender preferências dos usuários e oferecer recomendações personalizadas. No contexto do recrutamento inteligente, esses algoritmos são aplicados para analisar descrições de vagas e perfis profissionais, identificando a compatibilidade entre candidatos e oportunidades de forma automatizada.

Para sistemas de recomendação baseados em texto, duas subáreas da IA desempenham papéis cruciais: o Processamento de Linguagem Natural (PLN) e o Aprendizado de Representações. O PLN é responsável pela compreensão textual de currículos e descrições de vagas, extraindo competências e experiências. O Aprendizado de Representações (*Representation Learning*), por sua vez, utiliza técnicas de *embeddings* (como Sentence-BERT), que transformam o texto em vetores numéricos de alta dimensão, preservando o significado semântico. Essa conversão possibilita a medição precisa da similaridade entre candidatos e vagas.

Dessa forma, a Inteligência Artificial, ancorada no Aprendizado de Máquina e em suas subáreas de PLN e *Embeddings*, constitui a base para o desenvolvimento de um sistema de recomendação voltado à empregabilidade. O uso dessas tecnologias resulta em processos de seleção mais precisos, acessíveis e que promovem uma análise justa dos perfis profissionais.

2.2 Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural (PLN), ou *Natural Language Processing* (NLP), é uma subárea da Inteligência Artificial que visa possibilitar que os computadores compreendam, interpretem e gerem a linguagem humana de forma significativa (JURAFSKY; MARTIN, 2023). Essa área é fundamental em aplicações que envolvem texto, como é o caso de sistemas de recomendação baseados em descrições textuais, além de traduções automáticas, análise de sentimentos, correções ortográficas e geração de texto.

Em sistemas de recomendação de emprego, o PLN desempenha um papel essencial, pois tanto as descrições de vagas quanto os currículos de candidatos são predominantemente textuais. Por meio de suas técnicas, é possível extrair informações relevantes (competências, experiências e formações), compreender o contexto semântico e, o mais importante para este trabalho, avaliar a similaridade entre perfis e requisitos das vagas, viabilizando “*matches*” mais inteligentes e precisos.

A estruturação de dados textuais para o aprendizado de máquina geralmente segue um fluxo de trabalho metodológico e sequencial conhecido como *Pipeline* de Processamento de Linguagem Natural (*NLP Pipeline*) (ELOV; KHAMROEVA; XUSAINOVA, 2023). No contexto de recrutamento, a aplicação rigorosa dessas etapas de pré-processamento é essencial para reduzir o ruído semântico e extrair características relevantes de currículos e descrições de vagas (LIU; WANG; ZHANG, 2021). Entre as principais etapas desse *pipeline* aplicáveis a este trabalho, destacam-se:

- **Pré-processamento de texto:** Consiste na limpeza e padronização dos textos, incluindo tokenização (divisão em termos), remoção de *stopwords* (palavras de pouca relevância semântica) e lematização. Essas etapas são cruciais para reduzir ruídos e garantir a qualidade das análises subsequentes.
- **Extração e Representação de Características:** Tem como objetivo converter o texto em uma forma numérica, ou vetorial, compreensível para os algoritmos de ML. Técnicas como

o *TF-IDF* (*Term Frequency–Inverse Document Frequency*) são utilizadas para ponderar a relevância de cada palavra no documento em relação ao corpus total.

- **Modelagem Semântica:** Busca representar o significado de palavras e frases considerando seu contexto. Modelos avançados baseados em *embeddings* (como *Word2Vec*, *GloVe* e *BERT*) geram vetores densos que capturam relações de similaridade, permitindo que o sistema compreenda nuances semânticas (ex.: “desenvolvedor back-end” e “engenheiro de software Java” são semanticamente próximos).
- **Análise de Similaridade:** A partir das representações vetoriais obtidas (seja via TF-IDF ou Embeddings), é possível medir o grau de compatibilidade entre textos. A Similaridade do Cosseno é a métrica padrão para comparar a orientação desses vetores, indicando quão próximos semanticamente estão um currículo e uma descrição de vaga.

Essas etapas permitem que sistemas de recomendação baseados em PLN superem a simples correspondência de palavras-chave, alcançando uma compreensão mais profunda do significado. Isso viabiliza uma análise mais justa e contextualizada dos perfis.

Neste trabalho, a técnica TF-IDF foi utilizada inicialmente para a representação vetorial, servindo como uma linha de base (*baseline*) para o sistema. Essa escolha ofereceu um modelo funcional, de baixa complexidade computacional e alta interpretabilidade para a fase inicial de validação.

No entanto, para aprimorar drasticamente a precisão e a capacidade de compreensão semântica do sistema *Mais Acesso Conecta*, o modelo principal adotado é baseado em *embeddings* contextuais, como o Sentence-BERT. A transição e a comparação de desempenho entre a abordagem clássica (TF-IDF) e a abordagem contextual (Embeddings) são fundamentais para validar a melhoria na qualidade semântica e na precisão das correspondências, conforme detalhado na próxima seção.

2.3 Sistemas de Recomendação

Os sistemas de recomendação são amplamente utilizados na área da Inteligência Artificial, com o objetivo principal de sugerir itens relevantes a um usuário com base em seus interesses, histórico de comportamento ou em características compartilhadas com outros usuários. Esses sistemas estão presentes em diversas plataformas digitais, como serviços de streaming, comércio eletrônico e redes sociais. Mais recentemente, têm sido aplicados também ao contexto de recrutamento e seleção, recomendando vagas de emprego adequadas a candidatos e vice-versa (RICCI; ROKACH; SHAPIRA, 2022).

De forma geral, um sistema de recomendação busca estimar o grau de relevância de um item para um determinado usuário. No caso de plataformas de empregabilidade, essa estimativa corresponde ao nível de compatibilidade entre o perfil profissional de um candidato e os requisitos de uma vaga. Diversas abordagens podem ser utilizadas para realizar essa correspondência, dependendo do tipo de dado disponível e da natureza do problema.

Os principais tipos de sistemas de recomendação são:

- **Baseados em conteúdo (content-based filtering):** analisam as características intrínsecas dos itens (neste caso, descrições de vagas) e dos usuários (currículos e perfis profissionais), buscando similaridades entre eles. Por exemplo, um candidato que possui habilidades em “Python” e “machine learning” pode receber recomendações de vagas que exigem competências semelhantes. Essa abordagem depende fortemente de técnicas de Processamento de Linguagem Natural para representar o conteúdo textual em formato vetorial e mensurar similaridades semânticas.
- **Baseados em colaboração (collaborative filtering):** utilizam o comportamento coletivo de vários usuários para gerar recomendações. Nesse modelo, candidatos com perfis semelhantes ou que se candidataram a vagas parecidas tendem a receber recomendações em comum. Essa técnica é amplamente empregada em cenários com grande volume de interações históricas, mas pode sofrer com o chamado problema do *cold start*, isto é, a

difficuldade de recomendar para novos usuários ou vagas que ainda não possuem histórico suficiente (SCHAFER et al., 2007).

- **Híbridos:** combinam elementos das duas abordagens anteriores, unindo o conhecimento baseado em conteúdo com a inferência colaborativa. Essa combinação tende a oferecer resultados mais robustos e personalizados, especialmente em domínios complexos como o de empregabilidade, onde tanto as características textuais quanto o comportamento dos usuários são relevantes (BURKE, 2002).

Além das abordagens tradicionais, avanços recentes incorporam técnicas de aprendizado profundo (*Deep Learning*) e modelos de representação semântica para aprimorar a qualidade das recomendações. Modelos de *embeddings* como BERT e Sentence Transformers permitem representar perfis e descrições de vagas em um mesmo espaço vetorial, possibilitando comparações mais precisas e contextualizadas. Assim, o sistema é capaz de compreender que termos diferentes, como “engenheiro de software” e “desenvolvedor full-stack”, podem representar competências semelhantes, melhorando o *matching* entre candidatos e oportunidades.

No contexto deste trabalho, adotou-se a abordagem baseada em conteúdo, que analisa as características dos candidatos e das vagas por meio de representações textuais obtidas com técnicas de Processamento de Linguagem Natural e aprendizado de máquina. Essa escolha se deveu, inicialmente, à ausência de um histórico robusto de interações entre usuários e vagas, o que inviabilizaria o uso efetivo do filtro colaborativo neste estágio.

O sistema implementado utiliza uma arquitetura híbrida que combina representações vetoriais TF-IDF com *embeddings* contextuais gerados pelo modelo Sentence-BERT, ponderados por pesos otimizados empiricamente (60% SBERT e 40% TF-IDF). Em trabalhos futuros, pretende-se evoluir o modelo incorporando informações comportamentais (como histórico de candidaturas e avaliações de vagas), ampliando a capacidade adaptativa das recomendações por meio de filtragem colaborativa.

2.3.1 O Paradigma do Recrutamento Inteligente (*E-recruitment*)

Sistemas de recomendação clássicos operam com base na predição de interesse unilateral, como ocorre em plataformas de *streaming* e comércio eletrônico, onde algoritmos sugerem produtos que o usuário tem maior probabilidade de consumir com base em seu histórico. Contudo, a aplicação dessas tecnologias no domínio de Recursos Humanos introduz um cenário mais complexo conhecido como recomendação recíproca. No recrutamento inteligente, não basta apenas o interesse do candidato na oportunidade de emprego. É fundamental que as qualificações do profissional também atendam aos requisitos técnicos exigidos pela empresa.

O recrutamento enfrenta ainda o desafio do vocabulário assimétrico, pois os currículos e as descrições de vagas raramente utilizam as mesmas palavras para abordar as mesmas competências. Na verdade, os documentos se complementam. Algoritmos antigos que se baseiam em correspondências exatas de palavras sofrem de miopia semântica e falham ao tentar conectar termos diferentes que possuem o mesmo significado. Por esse motivo, a utilização de modelos de linguagem vetoriais é essencial para traduzir as intenções reais e conectar as qualificações corretas, superando assim as barreiras da polissemia no mercado de trabalho.

2.3.2 O Desafio do *Cold-Start* e a Escassez de Dados

Uma limitação frequente na criação de sistemas de recomendação em plataformas recentes é o problema do *cold-start* ou início frio. Esse fenômeno ocorre quando o sistema não possui dados históricos suficientes de interações dos usuários, como cliques e candidaturas anteriores, limitando o uso de algoritmos baseados em comportamento.

Em ecossistemas focados em inclusão, a exemplo da plataforma *Mais Acesso Conecta*, a escassez de dados é mais acentuada porque muitos usuários recém-cadastrados possuem perfis curtos. A falta de dados pode prejudicar a eficácia do sistema, visto que recomendações estritamente dependentes de histórico podem falhar em conectar novos usuários a oportunidades e acabar marginalizando esses perfis.

Para solucionar esse cenário, a literatura científica sugere a adoção de sistemas híbridos

combinados com a filtragem baseada em conteúdo. Ao focar na análise textual das habilidades descritas nos currículos e exigidas nas vagas por meio de modelos vetoriais como o SBERT, o sistema consegue sugerir oportunidades com alta precisão para o candidato logo em seu primeiro acesso, dependendo apenas da qualidade do texto que ele cadastrou.

2.4 Aprendizado de Representações (*Embeddings*)

O Aprendizado de Representações, também conhecido como *Representation Learning*, é uma subárea fundamental do Aprendizado de Máquina que busca transformar dados brutos em representações numéricas (vetores densos) capazes de capturar padrões e relações semânticas relevantes (BENGIO; COURVILLE; VINCENT, 2013; DEVLIN et al., 2019). No contexto do Processamento de Linguagem Natural (PLN), essas representações, chamadas de *embeddings*, são cruciais para que textos e sentenças sejam compreendidos computacionalmente, preservando seu significado e contexto.

Em sistemas de recomendação baseados em conteúdo, o uso de embeddings permite a comparação de descrições de vagas e currículos em um mesmo espaço vetorial. Isso viabiliza medir a similaridade semântica entre eles, identificando correspondências que superam as limitações de abordagens baseadas apenas em palavras-chave. Dessa forma, o sistema é capaz de inferir que termos diferentes, como "desenvolvedor front-end" e "especialista em React", são semanticamente correlacionados.

As principais técnicas utilizadas para gerar essas representações incluem:

- **Word Embeddings (*Word2Vec*, *GloVe*):** Representam palavras isoladamente em vetores contínuos, capturando relações de significado. No entanto, não conseguem diferenciar o sentido de uma palavra em contextos distintos (polissemia).
- **Contextual Embeddings (*BERT*, *Sentence-BERT*):** Baseiam-se em arquiteturas de *transformers* e analisam a sequência da frase para gerar vetores que refletem o significado em contexto. Essa capacidade é vital para compreender nuances complexas. O Sentence-BERT (*S-BERT*) é uma variação otimizada para gerar vetores de alta qualidade para

sentenças e documentos inteiros de forma eficiente, sendo a escolha ideal para o cálculo de similaridade entre um perfil de candidato e uma vaga.

- **Doc2Vec:** Extensão do *Word2Vec* voltada para gerar uma representação vetorial única de documentos completos.

2.4.1 A Evolução dos *Embeddings*

As abordagens tradicionais de representação textual, como *Bag-of-Words* (BoW) e TF-IDF, tratam as palavras como unidades atômicas e independentes, resultando em vetores esparsos de alta dimensionalidade que falham em capturar a semântica e a similaridade entre termos (por exemplo, não conseguem inferir que “carro” e “automóvel” são sinônimos).

Para superar essa limitação, surgiram os modelos de *Word Embeddings* estáticos, como o Word2Vec e o GloVe. Estes modelos baseiam-se na hipótese distributiva, onde palavras que aparecem em contextos similares tendem a ter significados semelhantes. No entanto, eles atribuem um vetor fixo a cada palavra, independentemente do contexto, falhando em casos de polissemia (ex: a palavra “banco” teria a mesma representação em “banco de dados” e “banco da praça”).

A introdução dos *Embeddings* Contextuais, baseados na arquitetura Transformer, trouxe avanços significativos para o estado da arte, impulsionados pela arquitetura *Transformer* e pelo modelo BERT (*Bidirectional Encoder Representations from Transformers*) (DEVLIN et al., 2019). Diferente de seus antecessores, o BERT gera representações vetoriais dinâmicas baseadas em toda a sentença, permitindo que o sistema compreenda o significado de uma palavra com base nas palavras que a cercam.

2.4.2 Sentence-BERT (SBERT) e Redes Siamesas

Embora o BERT tenha estabelecido novos padrões de desempenho em PLN, ele apresenta limitações computacionais significativas para tarefas de similaridade textual. O BERT original utiliza uma estrutura de *cross-encoder*, exigindo que ambos os textos sejam inseridos simul-

taneamente na rede para comparação. Isso torna a busca por similaridade em grandes bases de dados computacionalmente proibitiva, pois exigiria comparar o candidato com cada vaga individualmente em tempo real.

Para solucionar esse gargalo, (REIMERS; GUREVYCH, 2019) propuseram o Sentence-BERT (SBERT), que é uma modificação da arquitetura BERT que utiliza Redes Siamesas (*Siamese Networks*) para derivar *embeddings* de sentenças semanticamente significativos. Nesta arquitetura, duas redes BERT idênticas (com pesos compartilhados) processam as sentenças de entrada separadamente e geram vetores independentes (embeddings de sentença). Esses vetores podem então ser comparados utilizando métricas de distância leves, como a Similaridade do Cosseno.

Essa característica torna o SBERT ideal para o problema de recomendação de vagas, pois permite:

1. **Eficiência:** Os vetores das vagas podem ser pré-calculados e armazenados.
2. **Zero-Shot Learning:** Devido ao vasto pré-treinamento em bilhões de pares de frases, modelos SBERT robustos (como o `mpnet-base`) já possuem conhecimento intrínseco sobre a relação semântica entre termos (ex: sabem que “Python” é uma linguagem de programação relacionada a “Desenvolvimento de Software”), eliminando a necessidade de treinamento supervisionado específico para domínios com vocabulário padrão.

Diante disso, o modelo principal da plataforma *Mais Acesso Conecta* é baseado em Contextual Embeddings. O Sentence-BERT foi escolhido e validado para converter as descrições de vagas e os perfis de candidatos em vetores de características comparáveis. O uso desta técnica de *representation learning* é o diferencial da solução proposta, pois aproxima a interpretação computacional da semântica humana e fornece uma base robusta e inteligente para gerar recomendações personalizadas por meio do cálculo de Similaridade do Cosseno.

2.5 Métricas de Similaridade e Estratégia de Validação

A avaliação de sistemas de recomendação em cenários de *cold start* (onde não há histórico prévio de interações de usuários reais) exige uma abordagem focada na qualidade semântica das sugestões, em vez de métricas puramente estatísticas de conversão. Para validar a eficácia do modelo proposto na plataforma *Mais Acesso Conecta*, foram adotadas métricas que mensuram a proximidade vetorial e a coerência das recomendações em cenários controlados.

As principais métricas e métodos de avaliação definidos para este trabalho são:

- **Similaridade do Cosseno (*Cosine Similarity*):** Esta é a métrica matemática central do algoritmo. Ela mede o cosseno do ângulo entre dois vetores em um espaço multidimensional (o vetor do candidato e o vetor da vaga). O resultado varia de 0 a 1, onde valores próximos a 1 indicam alta similaridade semântica e valores próximos a 0 indicam dissemelhança.

$$\text{Sim}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

No contexto do trabalho, essa métrica é utilizada para quantificar o quão "próximo" o perfil técnico de um candidato está dos requisitos de uma vaga, servindo como base para o ranqueamento.

- **Gap de Relevância (Validação Cruzada):** Para validar a capacidade de distinção do modelo, utiliza-se a análise do *Gap* (diferença) entre os scores de similaridade de pares "Candidato-Vaga Ideal" e "Candidato-Vaga Irrelevante". Um modelo é considerado eficaz quando maximiza essa diferença, garantindo que vagas pertinentes recebam pontuações significativamente superiores às vagas de ruído.

$$\text{Gap} = \text{Score}_{\text{Match Ideal}} - \text{Score}_{\text{Ruído}}$$

- **Análise de Cenários (Estudo de Caso):** Devido à subjetividade da escolha profissional, a validação qualitativa é fundamental. Utiliza-se a técnica de *Top-N Recommendation*

Analysis, onde perfis sintéticos de candidatos com competências conhecidas são submetidos ao sistema, e as N primeiras recomendações são analisadas manualmente para verificar a coerência semântica (ex.: verificar se um "Desenvolvedor Full Stack" recebe vagas de tecnologia no topo da lista e não vagas operacionais).

Essa abordagem mista (matemática e qualitativa) permite demonstrar que o sistema, impulsionado pela Engenharia de Contexto aplicada sobre o modelo *Zero-Shot* SBERT, é capaz de superar ambiguidades linguísticas e priorizar competências técnicas (*Hard Skills*), cumprindo o objetivo de conectar candidatos com vagas que realmente correspondam à sua qualificação profissional.

2.5.1 Métricas de Avaliação de Ranqueamento e Recuperação de Informação

Embora a *Similaridade do Cosseno* atue como o mecanismo matemático interno para quantificar a proximidade vetorial entre os perfis, a avaliação da eficácia de um sistema de recomendação operando em cenários reais exige métricas clássicas de Recuperação de Informação (*Information Retrieval*). Essas métricas são essenciais para atestar a utilidade prática das listas geradas para o usuário final (RICCI; ROKACH; SHAPIRA, 2022). Neste trabalho, o desempenho do motor foi mensurado através das seguintes métricas de ranqueamento:

Precisão@K (*Precision at K*) e Revocação (*Recall@K*): A Precisão@K avalia a exatidão pontual das recomendações, consistindo na proporção de itens (vagas ou candidatos) verdadeiramente relevantes que aparecem nas K primeiras posições da lista gerada pelo algoritmo em relação ao total de K itens recomendados (RICCI; ROKACH; SHAPIRA, 2022). A Revocação@K, por sua vez, mensura a capacidade de cobertura do modelo, calculando a fração de todos os itens relevantes existentes na base de dados que o algoritmo conseguiu recuperar e ranquear dentro desse limite K .

Em ecossistemas de recrutamento, o balanceamento entre essas duas métricas é fundamental: uma alta Precisão garante que a tela do usuário não seja poluída com falsos-positivos (recomendações irrelevantes), enquanto uma alta Revocação assegura que oportunidades altamente compatíveis ou talentos ocultos não sejam omitidos pelo sistema.

Normalized Discounted Cumulative Gain (nDCG@K): Enquanto a Precisão e a Revocação avaliam a presença de itens relevantes de forma binária, o nDCG é considerado a métrica padrão-ouro para sistemas de recomendação modernos porque avalia a qualidade global da ordenação do ranqueamento. A premissa do algoritmo é que resultados altamente compatíveis devem aparecer no topo absoluto da lista. Caso uma recomendação excelente seja rebaixada para posições inferiores, o nDCG aplica um fator de desconto logarítmico, penalizando a nota final do modelo. O *Discounted Cumulative Gain* (DCG) na posição p é calculado pela soma da relevância dos itens dividida pelo logaritmo de suas respectivas posições:

$$DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)} \quad (2.1)$$

Para permitir uma comparação justa entre diferentes consultas de busca (uma vez que o número total de vagas ideais pode variar consideravelmente para cada candidato), o valor do DCG é dividido pelo *Ideal DCG* (IDCG), resultando no valor normalizado (nDCG). No contexto prático da plataforma *Mais Acesso Conecta*, alcançar um nDCG alto atesta matematicamente que candidatos e recrutadores estão encontrando as melhores correspondências logo nas primeiras posições do sistema, otimizando o tempo de triagem e poupando o esforço de analisar páginas extensas de resultados irrelevantes.

2.5.2 Desafios de Implantação e *AI Engineering*

Apesar da alta precisão oferecida por modelos de linguagem como o Sentence-BERT na análise de similaridade semântica, a implantação dessas arquiteturas em ambientes de produção impõe desafios significativos de infraestrutura. A geração de representações vetoriais densas exige alto processamento computacional, o que pode causar gargalos de latência se executado de forma síncrona junto às operações transacionais de um sistema *web* (HUYEN, 2025).

Para mitigar esses impactos, a disciplina emergente de *AI Engineering* propõe o uso de abordagens de Engenharia de Software modernas, como o padrão de Microsserviços (NEWMAN, 2015) e o processamento assíncrono orientado a eventos (HOHPE; WOOLF, 2004). A adoção de mensageria (como *message brokers* e filas) atua como um amortecedor de requisições, isolando o

peso do processamento de inteligência artificial do sistema principal e garantindo que a aplicação permaneça responsiva e escalável, premissas essenciais para plataformas de recrutamento com alto volume de acesso.

2.5.3 Fundamentação Estatística, Ética e Testes Não Paramétricos

Avaliar um sistema de recomendação vai muito além de apenas olhar para a sua precisão. Nesse sentido, é preciso usar testes de hipóteses para garantir que as melhorias observadas no modelo são reais, e não apenas coincidências estatísticas. Além disso, no mundo da Inteligência Artificial, as notas de similaridade e as métricas de ranqueamento quase nunca seguem uma distribuição normal. Por conta dessa assimetria, como os dados costumam fugir desse padrão, a literatura recomenda o uso de testes estatísticos não paramétricos, pois eles funcionam adequadamente mesmo quando as amostras não seguem um formato específico.

Desse modo, pensando na avaliação do algoritmo e, principalmente, na ética da Inteligência Artificial (*fairness*), dois testes ganham destaque neste trabalho:

- **Teste de Wilcoxon *Signed-Rank*:** Trata-se de um teste voltado para amostras pareadas. Na prática, ele se apresenta como a ferramenta matemática ideal para comparar dois algoritmos diferentes (como o método tradicional e o modelo de Inteligência Artificial proposto) rodando exatamente sobre os mesmos dados. Consequentemente, ele serve para provar que a melhora no desempenho do sistema foi um ganho estatístico real, e não apenas uma variação aleatória do algoritmo.
- **Teste U de Mann-Whitney:** Diferente do método anterior, este teste compara duas amostras independentes. No contexto atual, é fundamental garantir que a Inteligência Artificial não repita preconceitos históricos ou crie barreiras contra grupos minoritários (NOBRE; LEITE, 2025). Por essa razão, o Teste de Mann-Whitney é utilizado para checar se o sistema trata dois grupos distintos (por exemplo, candidatos com e sem deficiência) de maneira igualitária. Assim, ao calcular o *p-value*, o teste consegue provar matematicamente se o algoritmo possui ou não algum viés discriminatório, ajudando a assegurar uma IA mais justa e inclusiva.

A preocupação com a equidade (*fairness*), que justifica o uso do Teste de Mann-Whitney, está diretamente ligada aos riscos do aprendizado de máquina. De acordo com Nobre e Leite (NOBRE; LEITE, 2025), os sistemas de IA correm o risco de internalizar e amplificar desigualdades sociais se forem desenvolvidos sem mecanismos adequados de auditoria. Nesse cenário, a literatura destaca dois tipos de vieses prejudiciais a grupos minoritários: o viés comportamental e o viés de agregação.

O viés comportamental ocorre quando o sistema captura e replica padrões e preconceitos humanos inconscientes contidos nos dados históricos. Já o viés de agregação surge quando dados heterogêneos são combinados de forma inadequada para criar um modelo único de candidato "ideal", falhando em capturar as variações e especificidades de minorias (NOBRE; LEITE, 2025).

No contexto da inclusão de Pessoas com Deficiência (PCDs) no mercado de trabalho, o viés de agregação representa um risco crítico. Se o motor de recomendação for modelado considerando apenas os padrões de currículo da maioria (pessoas sem deficiência), ele tenderá a penalizar e ocultar os perfis de candidatos PCDs, que frequentemente apresentam descrições e trajetórias atípicas. Portanto, a aplicação estatística dos testes não paramétricos consolida-se como a ferramenta de auditoria que garante matematicamente a ausência do viés de agregação no motor da plataforma *Mais Acesso Conecta*.

2.6 Metodologia CRISP-DM

O desenvolvimento de soluções baseadas em Inteligência Artificial e mineração de dados exige um processo estruturado que alinhe os objetivos técnicos às necessidades reais do negócio. Para suprir essa demanda, a indústria consolidou o *framework* CRISP-DM (*Cross Industry Standard Process for Data Mining*) (CHAPMAN et al., 2000). Trata-se de um modelo de processo abrangente e iterativo, amplamente adotado como padrão para o ciclo de vida de projetos de *Machine Learning* e Ciência de Dados.

A teoria do CRISP-DM organiza o ciclo de vida do projeto em seis fases interdependentes e iterativas (CHAPMAN et al., 2000):

1. **Compreensão do Negócio (*Business Understanding*):** Fase inicial focada em compreender os objetivos e requisitos do projeto sob a ótica do negócio, convertendo esse conhecimento em uma definição de problema de mineração de dados e um plano de projeto preliminar.
2. **Compreensão dos Dados (*Data Understanding*):** Etapa de coleta inicial e familiarização com os dados, com o objetivo de identificar problemas de qualidade, descobrir *insights* preliminares ou detectar subconjuntos interessantes.
3. **Preparação dos Dados (*Data Preparation*):** Abrange todas as atividades necessárias para construir o conjunto de dados final que será inserido na ferramenta de modelagem. Inclui tarefas críticas de Processamento de Linguagem Natural (PLN), como limpeza de texto, seleção de tabelas, formatação e transformação de variáveis.
4. **Modelagem (*Modeling*):** Fase em que diversas técnicas de modelagem (como a geração de *embeddings* semânticos) são selecionadas e aplicadas. Seus parâmetros são calibrados para otimizar os resultados.
5. **Avaliação (*Evaluation*):** Antes de proceder com a implantação, o modelo construído é avaliado rigorosamente para garantir que ele atinge os objetivos de negócio estabelecidos na primeira fase. Verifica-se se há algum viés, falha ou lacuna na solução técnica proposta.
6. **Implantação (*Deployment*):** A fase final onde o conhecimento adquirido e o modelo validado são organizados e apresentados de forma que o cliente possa utilizá-los. Na engenharia de software moderna, isso frequentemente envolve o encapsulamento do modelo em microsserviços e APIs para integração em tempo real com plataformas existentes.

A adoção teórica da macroestrutura do CRISP-DM justifica-se por sua flexibilidade. Como o *framework* não prescreve algoritmos específicos, ele permite que engenheiros de dados apliquem arquiteturas de ponta, como o Sentence-BERT para sistemas de recomendação, enquanto mantêm o controle rigoroso sobre as fases de validação, integração e entrega de valor de engenharia (CHAPMAN et al., 2000).

2.7 Trabalhos Relacionados

Nesta seção são apresentados estudos que abordam o desenvolvimento de sistemas de recomendação de vagas de emprego, destacando metodologias, técnicas e contribuições relevantes que fundamentam a proposta deste trabalho.

O artigo de Liu, Wang e Zhang (2021), “*An Intelligent Job Recommendation System Based on Semantic Embeddings and Machine Learning*”, propõe um sistema que combina técnicas de Processamento de Linguagem Natural (PLN), embeddings semânticos e algoritmos de aprendizado de máquina para otimizar a correspondência entre perfis de candidatos e descrições de vagas. Os autores estruturam sua metodologia em etapas bem definidas: coleta de dados, pré-processamento textual, representação vetorial (Bag of Words, TF-IDF e embeddings), aplicação de modelos de classificação (SVM e Random Forest) e avaliação com métricas como precisão, revocação e F1-score. Este estudo fornece uma referência metodológica clara, sendo especialmente útil para a definição das etapas de pré-processamento e representação textual adotadas no presente trabalho. É importante destacar que os resultados reportados por Liu, Wang e Zhang (2021) foram obtidos sobre um *dataset* com *labels* supervisionados e controlados, o que difere do cenário de avaliação adotado neste trabalho, onde as métricas são calculadas sobre interações reais e esparsas de usuários da plataforma, condição metodologicamente mais rigorosa, porém que tende a produzir valores absolutos menores nas métricas de ranqueamento.

Complementando essa abordagem prática, Yap, Haw e Al-Juboori (2025), em “*A Comprehensive Review on Machine Learning-Based Job Recommendation Systems*”, realiza uma revisão abrangente das técnicas empregadas em sistemas de recomendação de empregos baseados em aprendizado de máquina. O estudo classifica as abordagens em três categorias: baseadas em conteúdo, colaborativas e híbridas. Nesse trabalho, demonstra-se que os modelos híbridos apresentam melhor desempenho, sobretudo quando combinam embeddings semânticos e redes neurais profundas. Entretanto, também ressalta que abordagens baseadas em conteúdo, como TF-IDF e modelos supervisionados mais simples, são eficazes em bases de dados menores, o que se alinha ao escopo do presente trabalho. Assim, o artigo orienta a escolha por representa-

ções textuais leves e metodologias de menor complexidade computacional, adequadas ao estágio inicial de desenvolvimento da plataforma *Mais Acesso Conecta*.

Direcionando o foco para a acessibilidade e a ética algorítmica, o estudo de Shishehchi e Banihashem (2019), “*JRDP: A Job Recommender System Based on Ontology for Disabled People*”, apresenta o sistema JRDP, voltado à inclusão de pessoas com deficiência no mercado de trabalho. Seu diferencial é o uso de ontologias para representar habilidades, restrições e requisitos das vagas, permitindo recomendações mais justas e explicáveis. Embora a implementação de ontologias formais não seja o foco deste projeto, os conceitos de acessibilidade e explicabilidade extraídos desse trabalho são diretamente aplicáveis à proposta da plataforma *Mais Acesso Conecta*, que também visa promover recomendações mais inclusivas e transparentes. A representação semântica sugerida pelo JRDP pode ser adaptada para métodos textuais como TF-IDF ou embeddings, mantendo a viabilidade técnica e a coerência com os objetivos deste estudo.

Aprofundando a literatura sobre arquiteturas vetoriais aplicadas ao recrutamento, o estudo de Lavi, Medentsiy e Graus (2021a) aborda diretamente o desafio de conectar currículos ruidosos a vagas de emprego. Os autores destacam um paradigma central na recuperação de informação de Recursos Humanos: no contexto de recrutamento, os documentos de candidatos e vagas raramente utilizam o mesmo vocabulário exato; em vez disso, eles se complementam como peças de um quebra-cabeça. Para resolver isso, propõem uma arquitetura baseada em Redes Siamesas (SBERT) otimizada pela função de Similaridade do Cosseno. Esse trabalho corrobora diretamente a arquitetura desenvolvida para a plataforma *Mais Acesso Conecta*, validando cientificamente que modelos de *Embeddings* contextuais são ideais para capturar o contexto de competências que se complementam, superando o problema de vocabulários incompatíveis entre a vaga e o candidato (LAVI; MEDENTSIY; GRAUS, 2021a).

Por fim, no que tange às limitações inerentes à implantação de sistemas de recomendação em cenários reais, o trabalho de Ribeiro (2025) oferece uma fundamentação teórica essencial. O autor investiga como a escassez de dados prejudica a eficácia e a adoção de sistemas automatizados, propondo a integração de algoritmos híbridos para superar a ausência de histórico do

usuário e perfis incompletos (*cold-start*). Esta pesquisa sustenta a solução arquitetural adotada neste TCC. Ela evidencia que a combinação de diferentes métodos de busca (como a fusão do SBERT com o TF-IDF) é o caminho recomendado pela literatura contemporânea para gerar recomendações precisas em plataformas onde uma grande parcela dos usuários ainda não possui perfis ricos em dados (RIBEIRO, 2025).

De forma integrada, os cinco trabalhos analisados fornecem um forte referencial teórico para a construção da metodologia deste TCC. O estudo de Liu, Wang e Zhang (2021) contribui com a estrutura prática de pré-processamento e modelagem; Yap, Haw e Al-Juboori (2025) fornece um panorama teórico das abordagens e métricas; e Shishehchi e Banihashem (2019) traz a dimensão ética e inclusiva da recomendação. Somando-se a eles, as pesquisas de Lavi, Medentsiy e Graus (2021a) e Ribeiro (2025) validam as escolhas mais avançadas da arquitetura: o uso de Redes Siamesas (SBERT) para mitigar a incompatibilidade de vocabulário e a adoção de métodos híbridos para combater o *cold-start*. Assim, os elementos centrais dessas pesquisas orientam a escolha das técnicas que fundamentam o desenvolvimento do sistema proposto.

Para consolidar a análise da literatura, o Quadro 2.1 sintetiza a comparação estrutural entre os trabalhos correlatos e o sistema desenvolvido para a plataforma *Mais Acesso Conecta*, evidenciando principalmente como cada pesquisa contribuiu teoricamente ou metodologicamente para a fundamentação deste trabalho.

Estudo	Foco e Tecnologia	Contribuição para este TCC
Liu, Wang e Zhang (2021)	Recomendação inteligente de vagas de emprego utilizando <i>Embeddings</i> Semânticos (BoW/TF-IDF) e Aprendizado de Máquina clássico (SVM/Random Forest).	Forneceu a base teórica e a referência metodológica para estruturar as etapas de pré-processamento e representação vetorial dos textos.
Yap, Haw e Al-Juboori (2025)	Revisão sistemática sobre técnicas de Aprendizado de Máquina aplicadas ao recrutamento, focando em modelos híbridos e baseados em conteúdo.	Orientou a escolha conservadora por modelos baseados em conteúdo e metodologias leves, adequadas ao estágio inicial da plataforma.
Shishehchi e Banhashem (2019)	Inclusão e empregabilidade de Pessoas com Deficiência (PCD) através de um sistema (JRDP) fundamentado na construção de Ontologias Formais.	Inspirou a dimensão ética e inclusiva do projeto, evidenciando a importância de alinhar a Inteligência Artificial ao respeito às necessidades dos candidatos PCD.
Lavi, Medentsiy e Graus (2021)	Resolução do vocabulário assimétrico em Recursos Humanos utilizando Redes Siamesas (SBERT) e Similaridade do Cosseno.	Validou cientificamente a adoção da arquitetura SBERT como a tecnologia ideal para solucionar as diferenças de vocabulário entre empresas e candidatos.
Ribeiro (2025)	Resolução do problema de <i>cold-start</i> (escassez de dados em usuários novos) através da integração de algoritmos híbridos de recomendação.	Embasou e justificou a decisão arquitetural de construir um modelo híbrido (SBERT + TF-IDF) para combater a escassez de dados de novos usuários.

Fonte: Elaborado pelo autor.

Quadro 2.1: Comparação entre os Trabalhos Relacionados e as Contribuições para a Pesquisa

Desenvolvimento do modelo de recomendação

Este capítulo apresenta o desenvolvimento prático do modelo de recomendação, detalhando desde a concepção dos requisitos técnicos até a consolidação da arquitetura de integração. Para garantir a coerência entre a metodologia proposta e a execução prática, a estrutura deste capítulo reflete as fases do *framework* CRISP-DM. A seguir, detalham-se as etapas de Compreensão, Preparação, Modelagem, Implantação e Avaliação que fundamentaram a construção do sistema.

3.1 Compreensão: Revisão Bibliográfica e Definição de Requisitos

A fase inicial de Compreensão, segundo as diretrizes do *framework* CRISP-DM, englobou a fundamentação teórica e o mapeamento das necessidades do sistema. Inicialmente, o levantamento bibliográfico focou em sistemas de recomendação baseados em conteúdo, com ênfase no uso de *Deep Learning* para Processamento de Linguagem Natural (PLN). A investigação priorizou o estudo da arquitetura *Transformer* e a adoção de modelos de *Embeddings* pré-treinados, explorando a sua capacidade de generalização *Zero-Shot* aplicada a domínios específicos de recrutamento.

A partir desse embasamento técnico, procedeu-se à definição dos requisitos centrais para o modelo de recomendação proposto. Estabeleceu-se que a solução deveria ser capaz de priorizar as competências técnicas (*hard skills*) do candidato e atender com precisão às especificidades

de cada vaga. Desse modo, o principal requisito técnico mapeado foi a necessidade de garantir que a recomendação não fosse enviesada por termos genéricos, jargões corporativos ou ruído semântico, assegurando um *match* baseado na real qualificação profissional.

3.1.1 Requisitos Funcionais do Algoritmo

Para que a integração na plataforma *Mais Acesso Conecta* fosse efetiva, o modelo precisou ser estruturado para atuar de forma bidirecional. Desse modo, os requisitos funcionais foram divididos em duas frentes complementares: uma focada na jornada do candidato e outra voltada à necessidade do recrutador, conforme detalhado a seguir.

Recomendação de Vagas (Candidato $\rightarrow$ Vaga)

O principal requisito do motor de recomendação é maximizar o engajamento e a satisfação do usuário ao oferecer uma experiência altamente personalizada. Em um mercado de trabalho digital saturado, a assertividade da recomendação é crucial. O algoritmo foi, portanto, desenhado para ir além da simples busca por palavras-chave, processando o perfil completo do candidato (incluindo formação, experiências e habilidades) e

traduzindo-o em um vetor semântico unificado. Para que isso fosse possível, o sistema utiliza uma rotina de Engenharia de Contexto para estruturar os dados do candidato e, em seguida, aplica o modelo *paraphrase-multilingual-mpnet-base-v2* para converter esse texto em representações matemáticas comparáveis.

A justificativa para este modelo é que ele deve capturar o contexto e o “potencial” do candidato, e não apenas o seu histórico. A saída deve ser um ranqueamento que priorize vagas com alta aderência técnica (as *hard skills* isoladas pelo pipeline de PLN).

- **Entrada:** Perfil consolidado do candidato (formação, experiências, idiomas, deficiência, etc.).
- **Processamento:** Comparação semântica vetorial (Similaridade do Cosseno) entre o perfil vetorizado e o banco de vagas ativas.

- **Saída:** Lista ordenada de vagas por *score* de similaridade, priorizando aquelas com maior aderência técnica e comportamental.

Triagem de Talentos (Vaga → Candidato)

De forma análoga, o modelo deve oferecer uma solução eficiente para os recrutadores, otimizando o processo de triagem manual, que é notavelmente demorado e suscetível a vieses humanos. Ao aplicar o mesmo núcleo de similaridade semântica utilizado para o candidato, o sistema permite objetividade e consistência no *matching*. A triagem deve ser vista como uma pré-seleção inteligente que ranqueia os candidatos com maior compatibilidade técnica e funcional. Adicionalmente, este requisito exige a implementação de regras de negócio mandatórias no nível da saída, garantindo que a lista de candidatos gerada já atenda a filtros críticos, como requisitos de certificações ou vagas exclusivas para Pessoas com Deficiência (PCD).

- **Entrada:** Descrição agregada da vaga, incluindo requisitos e habilidades.
- **Processamento:** Comparação semântica vetorial entre a descrição da vaga e o banco de perfis de candidatos.
- **Regras de Negócio e Saída:** Lista de candidatos ordenada por *score* de similaridade, com a aplicação de filtros para garantir a aderência a requisitos obrigatórios (por exemplo vagas exclusivas para PCD).

3.2 Preparação dos Dados: Agregação e Pipeline de PLN

A fase de Preparação dos Dados, segunda etapa do *framework* CRISP-DM, tem como objetivo transformar os registros brutos da plataforma em um formato estruturado e otimizado para o consumo do modelo de Inteligência Artificial. Para garantir o rigor técnico, as atividades desta fase foram divididas em duas frentes sequenciais: a consolidação relacional dos atributos textuais e a limpeza de ruídos por meio de técnicas linguísticas.

3.2.1 Exploração e Engenharia de Atributos

O desenvolvimento do modelo de recomendação é um processo dependente da qualidade dos dados de entrada. Esta etapa iniciou-se com a ingestão e tratamento dos dados brutos da plataforma *Mais Acesso Conecta*, composta por 525 perfis de candidatos e 199 vagas, obtidos no dia 27/10/2025. Esta etapa foi realizado em ambiente Python, utilizando a biblioteca Pandas para manipulação estrutural dos dados relacionais.

3.2.2 Engenharia de Atributos: Criação do *Corpus* Textual

O primeiro passo para criar o novo texto a ser usado para os candidatos e as vagas foi a agregação de dados para construir um *corpus* textual coeso. Dados dispersos em múltiplas tabelas (ex.: experiências, formações, idiomas) foram concatenados em um único campo de texto denso, denominado `texto_agregado`, tanto para candidatos quanto para vagas.

Essa consolidação foi essencial para que o modelo de *Embeddings* contextuais analisasse o perfil do usuário como um “documento unificado”, capturando o contexto global de suas competências e experiências, conforme pode ser observado nos exemplos dos Quadros 3.1 e 3.2 a seguir.

3.2.3 Amostra de Dados Brutos

Os Quadros 3.1 e 3.2 apresentam uma amostra da estrutura dos dados de entrada utilizados no processo, após a engenharia de atributos e antes de qualquer etapa de limpeza. É possível notar a natureza não estruturada do campo *texto_agregado*, que contém múltiplos atributos textuais, incluindo localização, formação e experiências, o que justifica a necessidade do *pipeline* de Processamento de Linguagem Natural (PLN).

Em consonância com as diretrizes da Lei Geral de Proteção de Dados (LGPD - Lei nº 13.709/2018), e visando a proteção da privacidade dos usuários da plataforma *Mais Acesso Conecta*, optou-se por utilizar dados fictícios (*mock data*) nesta etapa de demonstração. Os nomes de candidatos foram substituídos por pseudônimos genéricos, e as informações sensíveis,

como nomes de instituições de ensino e empresas privadas, foram explicitamente censuradas no corpo do texto para fins de ilustração metodológica, mantendo a complexidade semântica dos dados originais.

Nome (Amostra)	É PCD	Texto Agregado Bruto
Candidato 1	Não	Manaus AM Formações: Bacharelado em Física ([censurado]); Mestrado em Física... Idiomas: Inglês (Intermediário)
Candidato 2	Não	Manaus AM Formações: Técnico em Mecânica ([censurado]) Experiências: Operador Técnico na empresa FCC do Brasil
Candidato 3	Sim	Manaus AM PCD: Pessoa com perda auditiva parcial
Candidato 4	Não	Manaus AM Formações: Psicologia Organizacional (Faculdade [censurado])... Experiências: Secretária na empresa [censurado]

Quadro 3.1: Amostra da Base de Dados de Candidatos

Título da Vaga	Exclusiva PCD	Descrição/Habilidades Agregadas Brutas
Auxiliar de Depto Pessoal	Não	Função: Auxiliar de Depto Pessoal... Descrição: Cadastrar admissões; Conferir documentos... Habilidades: Noções de legislação trabalhista...
Montador Naval	Não	Função: Montador Naval... Descrição: Executar montagem de caldeiraria no canteiro de obras... Habilidades: Básico nos procedimentos da montagem...
Auxiliar de Manutenção Veicular	Não	Função: Auxiliar de Manutenção Veicular... Descrição: Auxiliar no serviço de elétrica... Habilidades: Treinamento em sistema de freios e válvulas...
Oportunidade PCD Auxiliar de Manutenção Predial	Sim	Função: Auxiliar de manutenção predial... Descrição: Apoio na manutenção preventiva e corretiva predial... Habilidades: Conhecimentos básicos em elétrica ou hidráulica...

Quadro 3.2: Amostra da Base de Dados de Vagas

3.2.4 Pipeline de Processamento de Linguagem Natural (PLN)

A qualidade de um sistema de recomendação baseado em conteúdo depende diretamente da qualidade dos dados textuais processados. Conforme o trabalho de Liu, Wang e Zhang (2021),

a correta aplicação do PLN é crucial para reduzir a dimensionalidade do vocabulário e garantir que a posterior representação vetorial seja baseada em termos de alta relevância. Diante disso, aplicaram-se técnicas de normalização e filtragem para reduzir o ruído semântico do corpus textual.

3.2.5 Normalização e Filtragem de *Stopwords*

O processamento inicial utilizou a biblioteca NLTK para a remoção de *stopwords* (palavras de parada) padrão da língua portuguesa (artigos, preposições). Palavras como “de”, “a”, “o”, “e”, “para”, “com”, entre outras (conforme lista de *stopwords* padrão do NLTK para português), foram removidas para reduzir o ruído e focar nos termos com maior peso semântico.

Em seguida, aplicou-se a normalização de texto, convertendo todos os caracteres para minúsculas e removendo pontuação e caracteres especiais, preservando apenas a estrutura alfanumérica essencial. Por exemplo:

- **Texto Bruto:** "Gestor de Projetos Sênior. Experiência em AWS."
- **Após Normalização e Filtragem:** ["gestor", "projetos", "sênior", "experiência", "aws"]

Este processo garantiu que algumas dessas variações não prejudicassem a similaridade vetorial a ser comparada.

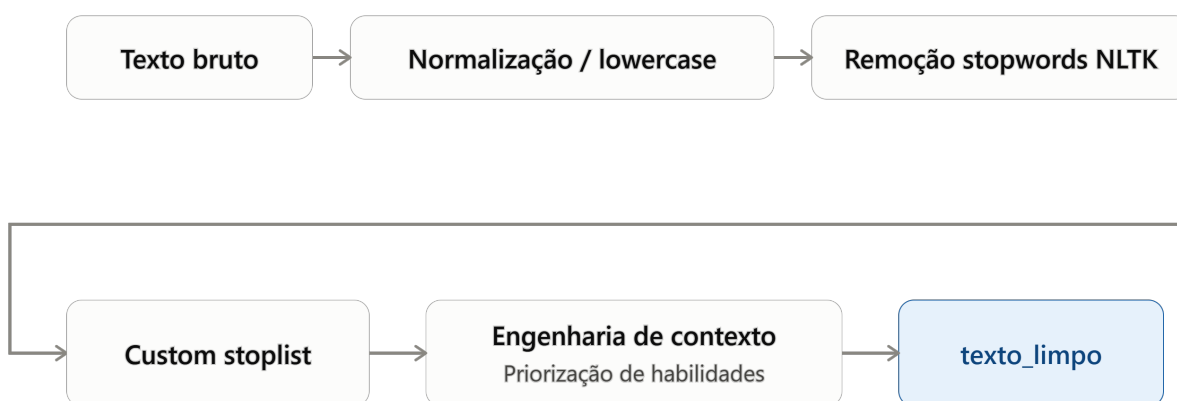
3.2.6 Filtragem de Domínio (Limpeza Agressiva)

A principal contribuição metodológica neste pré-processamento consistiu em uma abordagem de limpeza em duas frentes. Primeiramente, utilizou-se Expressões Regulares (Regex) para a normalização morfológica do texto, removendo caracteres especiais e pontuações indesejadas. Em seguida, para a filtragem semântica, criou-se uma lista de exclusão personalizada (*Custom Stoplist*). Esta lista foi construída iterativamente para ir além das *stopwords* comuns, focando na remoção intencional de termos que introduzem ruído semântico. Os principais alvos de remoção foram:

- **Termos de Localização:** Nomes de cidades e estados que enviesavam a similaridade geográfica.
- **Termos Genéricos de Vagas:** Palavras como “vaga”, “requisitos”, “benefício”, “empresa”.
- **Soft Skills:** Termos comportamentais que, como demonstrado em testes preliminares, introduziam ruído semântico e causavam falsos positivos.

A Figura 3.1 ilustra o fluxo completo de transformação aplicado a cada texto da plataforma, desde o dado bruto até o `texto_limpo` utilizado pelo modelo SBERT.

Figura 3.1: Fluxo do Pipeline de Processamento de Linguagem Natural: transformação do texto bruto em representação semântica limpa.



A construção da *Custom Stoplist* foi fundamentada em três critérios técnicos objetivos. O primeiro critério estabelecido foi a frequência de documento. Conforme o princípio matemático subjacente ao algoritmo TF-IDF (JURAFSKY; MARTIN, 2023), notou-se que termos presentes em praticamente todos os documentos do *corpus* possuíam valor discriminativo próximo de zero. Exemplos disso foram palavras como “proatividade” e “comunicação”, que figuravam em quase todos os currículos e descrições de vagas analisados, independentemente da área de atuação.

O segundo critério adotado foi a relevância de domínio. Constatou-se que *soft skills* comportamentais e termos procedurais genéricos não diferenciavam as competências técnicas entre

os candidatos. A manutenção desse vocabulário introduziria falsos positivos ao aproximar vetorialmente perfis tecnicamente incompatíveis.

Por fim, o terceiro critério referiu-se ao viés semântico indesejado. Verificou-se que termos geográficos, como “Manaus” e “Amazonas”, apareciam de forma redundante nos registros da plataforma. Esse padrão aumentaria artificialmente a similaridade entre os documentos, independentemente da real compatibilidade técnica.

A aplicação dessa filtragem resultou no campo *texto_limpo*, que passou a conter exclusivamente as *hard skills* e os termos técnicos do candidato e da vaga. O Quadro 3.3 lista alguns dos termos removidos durante o processo:

Categoria de Remoção	Justificativa	Exemplos de Termos Removidos
<i>Soft Skills</i> Comportamentais	Introduzem ruído semântico.	proatividade, comunicação, aprendizado, dinamismo, liderança.
Termos Genéricos	Não agregam valor semântico técnico.	vaga, empresa, requisitos, benefícios, área, equipe, resultados.
Localização e Tempo	Viés geográfico/temporal.	manaus, amazonas, am, brasil, meses, anos, período, estágio.

Quadro 3.3: Amostra da Lista de Exclusão Personalizada (*Custom Stoplist*)

Esta metodologia teve o objetivo de isolar as *hard skills* (competências técnicas) para o cálculo vetorial. Ao garantir que o sistema de similaridade se concentre exclusivamente no conhecimento técnico, o modelo pode atingir uma correspondência robusta e transparente, evitando o ruído semântico que comprometeria a justiça das recomendações. O foco na aptidão técnica alinha o modelo aos princípios de inclusão e explicabilidade para perfis PCD, conforme defendido por Shishehchi e Banihashem (2019).

3.3 Modelagem: Representação Vetorial e Algoritmo de Recomendação

Após o pré-processamento, a etapa seguinte consistiu na conversão dos textos limpos em vetores numéricos (*Embeddings*). Para isso, optou-se pelo uso de redes neurais baseadas em *Transfor-*

mers, adotando uma abordagem de *Zero-Shot Learning* com modelos pré-treinados de estado da arte. Para estruturar esta fase de forma clara, as subseções a seguir detalham, respectivamente, a seleção do modelo base por meio de testes práticos, as estratégias de representação do contexto e o algoritmo matemático utilizado para o ranqueamento final.

3.3.1 Seleção do Modelo Base

A escolha do modelo de *embeddings* foi fundamentada em um *benchmark* empírico comparativo conduzido diretamente sobre os dados reais da plataforma *Mais Acesso Conecta*. Para garantir a viabilidade técnica e financeira da implantação no sistema, todos os modelos avaliados são de código aberto (*open-source*), disponibilizados na comunidade *Hugging Face*.

Foram avaliados quatro modelos candidatos: dois multilíngues (`paraphrase-multilingual-mpnet-base-v2` e `paraphrase-multilingual-MiniLM-L12-v2`) e dois monolíngues especializados em inglês (`all-mpnet-base-v2` e `JobBERT`). A avaliação considerou não apenas a eficácia de ranqueamento (medida pelo GAP Semântico e nDCG@5 sobre o *ground truth*), mas também a eficiência computacional, analisando o número de parâmetros da rede neural (que impacta o tempo de processamento) e as dimensões do vetor gerado (que impacta o custo de armazenamento no banco de dados).

O *benchmark* descartou empiricamente os modelos monolíngues em inglês: o `all-mpnet-base-v2` obteve nDCG@5 de 0,0726 e o `JobBERT` (`jjzha/jobbert-base-cased`), modelo especializado em descrições de cargos e habilidades (ZHANG; JENSEN; PLANK, 2022), apresentou GAP Semântico de apenas 0,0824, sendo incapaz de separar vagas relevantes de ruído no *corpus* em português. Esse resultado confirma empiricamente que modelos treinados exclusivamente em inglês não são transferíveis ao contexto brasileiro sem adaptação, independentemente de seu desempenho em *benchmarks* internacionais.

Tabela 3.1: Benchmark Empírico de Modelos SBERT sobre Dados Reais da Plataforma

Modelo	Idioma	Parâmetros	Dimensões	GAP Semântico
paraphrase-multilingual-mpnet-base-v2	pt-BR + EN	~ 278 Milhões	768	0,5988
paraphrase-multilingual-MiniLM-L12-v2	pt-BR + EN	~ 118 Milhões	384	0,5482
all-mpnet-base-v2	Inglês	~ 109 Milhões	768	0,5385
jjzha/jobbert-base-cased	Inglês	~ 110 Milhões	768	0,0824

Entre os modelos multilíngues avaliados, o `paraphrase-multilingual-mpnet-base-v2` superou a alternativa `MiniLM` ao demonstrar um GAP Semântico superior (0,5988). A seleção definitiva por este modelo foi baseada no melhor custo-benefício para a arquitetura do sistema, apoiada nos seguintes critérios:

- **Custo-Benefício e Eficiência Computacional:** Embora o modelo MPNet possua cerca de 278 milhões de parâmetros (sendo computacionalmente mais pesado e exigindo maior tempo de inferência do que o MiniLM, de 118 milhões), a arquitetura assíncrona orientada a eventos construída com o RabbitMQ suporta perfeitamente essa carga em segundo plano sem impactar a latência do usuário final. Em contrapartida, os vetores de 768 dimensões gerados por ele oferecem um nível de detalhamento semântico que justifica a sua escolha.
- **Superioridade na distinção entre vagas tecnicamente compatíveis e vagas de ruído:** métrica mais crítica para o contexto de inclusão PCD, pois recomendar uma vaga completamente inadequada tem impacto direto e negativo na experiência do candidato.
- **Robustez multilíngue nativa:** característica essencial para lidar com o vocabulário técnico misto em português brasileiro e com a terminologia em inglês presente nos perfis da plataforma.

O modelo opera em abordagem *Zero-Shot Learning*, aproveitando o pré-treinamento em bilhões de pares de texto em mais de 50 idiomas para converter cada perfil em um vetor denso de 768 dimensões. A decisão de não aplicar um *fine-tuning* imediato à rede neural foi uma escolha arquitetônica embasada no perfil dos dados da plataforma.

Devido à extrema esparsidade de dados rotulados e ao fenômeno de *cold-start* no domínio de recrutamento PCD, a aplicação de *fine-tuning* sobre uma amostra muito reduzida acarretaria

um alto risco de superajuste (*overfitting*) catastrófico, onde o modelo memorizaria os poucos currículos existentes em vez de aprender a linguagem real do domínio. Além disso, experimentos exploratórios preliminares (detalhados nas Considerações Finais) demonstraram que o modelo pré-treinado já possui uma capacidade de representação semântica suficiente para distinguir competências no idioma português. Portanto, justificou-se a adoção do *Zero-Shot* como a estratégia mais segura e de menor custo computacional para a fase atual do sistema.

3.3.2 Engenharia de Contexto e Priorização de Tokens

A utilização de modelos baseados na arquitetura *Transformer*, como o Sentence-BERT, esbarra em uma limitação técnica inerente à sua construção: a restrição da janela de atenção a um limite estrito de 512 *tokens*. Em cenários reais de recrutamento, o conteúdo somado de experiências, formações e habilidades de um candidato frequentemente excede esse limite, como também apontado pela literatura da área (LAVI; MEDENTSIY; GRAUS, 2021b). Caso a concatenação dos dados do candidato ocorresse na ordem convencional de um currículo (iniciando pela Formação, passando pelas Experiências e deixando as Habilidades para o final), as competências técnicas cruciais poderiam ser sumariamente ignoradas pela rede neural devido ao truncamento do texto (corte do texto que ultrapassa o limite de 512 *tokens*). Para contornar essa limitação arquitetônica de forma eficiente, este trabalho propôs uma estratégia de Engenharia de Contexto baseada na priorização da ordem das informações. A decisão de posicionar as competências técnicas no topo do documento gerado encontra forte respaldo na literatura de Processamento de Linguagem Natural aplicada a Recursos Humanos. Conforme demonstrado por Bocharova, Malakhov e Mezhuyev (2023) no desenvolvimento da arquitetura *VacancySBERT*, adota-se a premissa fundamental de que as habilidades (*skills*) constituem os componentes essenciais que definem um cargo ou trabalho. Como as descrições de experiências profissionais podem ser longas e suscetíveis a ruídos semânticos, a solução consistiu em reestruturar a montagem do perfil de entrada, garantindo que as informações de maior peso discriminatório para a Inteligência Artificial fossem lidas primeiro.

A nova estruturação do texto foi definida na seguinte ordem decrescente de importância:

1. **Habilidades e Competências:** Conhecimentos técnicos diretos (*hard skills*);
2. **Título do Cargo:** Função principal de atuação;
3. **Experiências Profissionais:** Histórico descritivo de empresas e cargos;
4. **Formação Acadêmica e Localização:** Informações de base.

Dessa forma, o vetor de texto gerado para a Inteligência Artificial obedece ao padrão [Habilidades + Cargo + Experiências + Formação]. Essa simples inversão da ordem atua como uma medida de segurança estrutural. Caso o resumo profissional de um candidato seja excepcionalmente extenso e ultrapasse o limite de processamento do modelo, o algoritmo descartará apenas as informações posicionadas no final da *string* (como cursos antigos ou detalhes de localização). Assim, o sistema garante a blindagem das *hard skills* do candidato, assegurando que suas reais competências técnicas sempre participem do cálculo de similaridade com a vaga.

Validação Empírica da Estratégia

Para verificar a aplicabilidade prática e a eficácia da estratégia proposta, conduziu-se uma avaliação empírica em duas frentes: a mensuração da incidência real de truncamento na base da plataforma e um teste de estresse controlado para isolar o efeito da reordenação.

A análise dos textos das vagas revelou que a maioria das descrições é curta e cabe no limite do modelo. Contudo, 13 das 199 vagas (cerca de 6,5%) são longas o bastante para ultrapassar os 512 *tokens*, tendo parte de seu conteúdo cortada no processamento. Por serem as descrições mais detalhadas e técnicas, confirma-se que o truncamento é um problema concreto, embora restrito a uma parcela do acervo.

Para isolar o impacto da ordem de concatenação sob a condição crítica de truncamento, o modelo foi submetido a um teste de estresse com um perfil sintético extenso (1.243 *tokens*), no qual as *hard skills* técnicas coexistem com descrições genéricas e volumosas de experiência e formação. O perfil foi vetorizado nas duas configurações e comparado a uma vaga-alvo da mesma área técnica. Na ordem convencional, a totalidade dos termos técnicos relevantes (*tensorflow*, *pytorch*, *machine learning*, *nlp*, *mlops*) foi descartada no truncamento; na ordem priorizada,

todos foram preservados dentro da janela de atenção do modelo. O impacto sobre a similaridade do cosseno com a vaga-alvo foi expressivo, conforme a Figura 3.2: o *score* elevou-se de 0,0889 na ordem convencional para 0,3411 na ordem priorizada, um aumento de aproximadamente 283%.

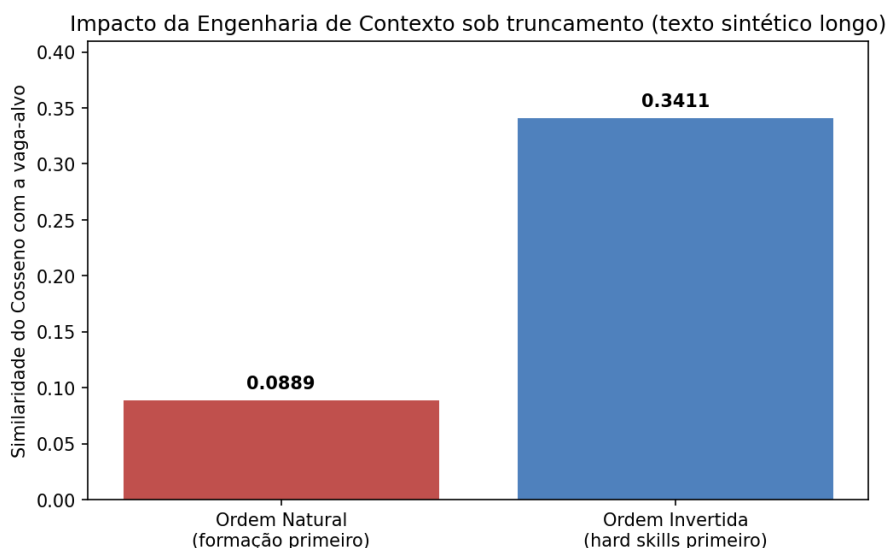


Figura 3.2: Impacto da priorização de *tokens* sob truncamento: similaridade do perfil sintético extenso com a vaga-alvo nas ordens convencional e priorizada.

O impacto dessa perda de informação manifesta-se de forma ainda mais evidente quando se observa o efeito sobre o *ranking* de recomendação, ou seja, a posição da vaga correta na lista apresentada ao usuário. Para ilustrar esse efeito, o mesmo perfil sintético foi comparado a um conjunto de seis vagas, composto pela vaga-alvo (da área técnica do perfil) e por cinco vagas de áreas distintas (administrativa, comercial, financeira, entre outras).

Os resultados, apresentados na Figura 3.3, revelam uma inversão drástica de comportamento. Na ordem convencional, com o perfil truncado e desprovido de suas competências técnicas, a vaga correta foi rebaixada à última posição (sexto lugar), sendo superada inclusive por vagas de áreas completamente alheias ao perfil do candidato. Em contraste, na ordem priorizada, a mesma vaga foi corretamente posicionada em primeiro lugar, resultando em uma subida de cinco posições no *ranking*.

Este resultado demonstra que o truncamento não representa apenas uma redução no *score* de similaridade, mas compromete diretamente a utilidade da recomendação: uma vaga altamente compatível pode ser ocultada do candidato, ao passo que oportunidades irrelevantes ocupam

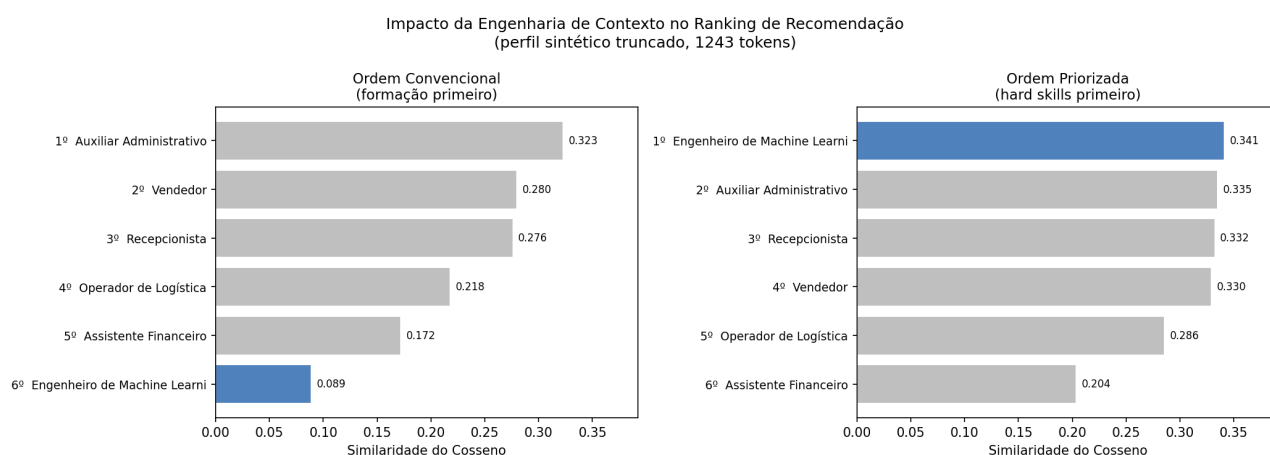


Figura 3.3: Efeito da Engenharia de Contexto no *ranking* de recomendação sob truncamento.

as primeiras posições. A priorização das competências técnicas, ao preservar as *hard skills* na janela de processamento do modelo, impede essa degradação e assegura que a correspondência seja determinada pelo conteúdo de maior relevância.

Conclui-se que a priorização da ordem atua como uma medida de segurança preventiva. Para os documentos curtos, que representam a maioria da base atual e não ultrapassam o limite de leitura do modelo, a mudança na ordem tem um impacto prático mínimo, visto que o algoritmo consegue ler todo o conteúdo. Contudo, para os currículos mais extensos, essa prática torna-se essencial, pois evita que as competências técnicas do candidato sejam cortadas no processamento. Portanto, o benefício dessa estratégia tenderá a crescer naturalmente à medida que a plataforma amadurecer e os usuários cadastrarem perfis mais detalhados, garantindo a precisão das recomendações a longo prazo de forma simples e eficiente.

3.3.3 Algoritmo de Recomendação e Cálculo de Similaridade

A etapa de recomendação consistiu no cálculo da Similaridade do Cosseno entre o vetor do candidato e os vetores de todas as vagas disponíveis no banco de dados. Para estruturar o entendimento desta fase, as subseções a seguir detalham a aplicação matemática da métrica aos dados textuais e o processo computacional de cálculo de *score* e ranqueamento das oportunidades.

3.3.4 Aplicação da Métrica

Conforme fundamentado na seção de referencial teórico (Seção 2.5), a Similaridade do Cosseno é a métrica matemática central do algoritmo. A sua aplicação direta neste modelo ocorreu por meio da seguinte equação:

$$\text{Sim}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

onde A representa o vetor denso (*embedding*) gerado a partir do perfil do candidato, e B representa o vetor correspondente à descrição da vaga.

A principal vantagem dessa abordagem no desenvolvimento do sistema foi considerar a orientação dos vetores (o conteúdo semântico) e não a sua magnitude (o tamanho do texto ou a frequência absoluta das palavras). No cenário de recrutamento, isso foi crucial, pois currículos e descrições de vagas possuíam tamanhos muito diferentes. Um currículo extenso e detalhado e uma descrição de vaga mais “limpa” podiam abordar as mesmas competências técnicas (*hard skills*). A Similaridade do Cosseno garantiu que, se ambos os documentos apontassem para a mesma direção semântica no espaço vetorial gerado pelo SBERT, eles seriam considerados altamente similares, independentemente da quantidade de palavras utilizadas.

3.3.5 Cálculo de Score e Ranqueamento

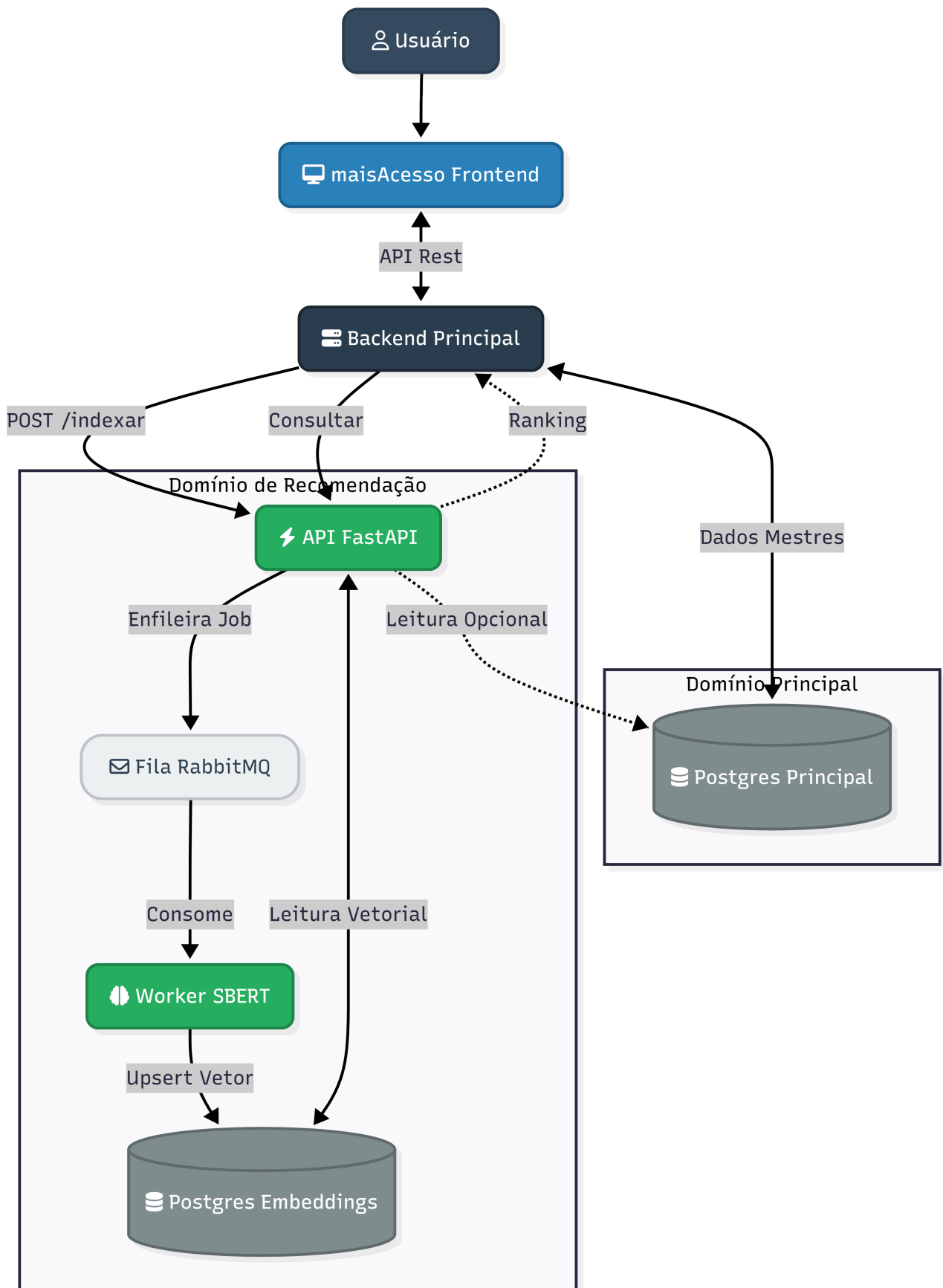
Após a geração dos valores vetoriais, foi computada uma matriz de similaridade, onde cada célula (i, j) representou o grau de compatibilidade entre o candidato i e a vaga j . Como esta etapa integrou a fase de experimentação do protótipo inicial do motor de recomendação (TCC 1), uma representação visual dos recortes dessa matriz será apresentada no capítulo de Resultados, por meio da Matriz de Confusão Semântica.

A partir desse cálculo, os resultados foram ordenados de forma decrescente, apresentando as vagas com *score* mais próximo de 1.0 no topo da lista. Para validar a eficácia do modelo ajustado, foram conduzidos estudos de caso comparativos utilizando perfis de controle, conforme será detalhado junto aos resultados práticos deste trabalho.

3.4 Implantação: Arquitetura de Integração

A integração do protótipo de recomendação ao ambiente de produção do sistema *Mais Acesso Conecta* exigiu o desenho de uma arquitetura escalável e resiliente. Modelos de inteligência artificial baseados em *Transformers*, como o Sentence-BERT (SBERT), caracterizam-se por um alto custo computacional, exigindo capacidade significativa de processamento e memória para a geração de representações vetoriais densas de alta dimensionalidade (geralmente matrizes de 768 ou 1024 dimensões) (REIMERS; GUREVYCH, 2019). A execução dessa carga de inferência matemática diretamente na infraestrutura do servidor principal causaria gargalos de processamento, resultando em latência inaceitável para a experiência do usuário durante operações cotidianas, como o cadastro de uma nova conta ou a atualização de um currículo. Para evitar o comprometimento da performance do sistema transacional, optou-se por uma arquitetura fortemente desacoplada baseada em microsserviços (NEWMAN, 2015).

Para sustentar esse desacoplamento, a solução estabeleceu uma separação rigorosa de domínios (EVANS, 2004). O banco de dados relacional primário (*Postgres Principal*) continuou encarregado de armazenar exclusivamente os dados mestres e transacionais de usuários, candidatos e vagas. Simultaneamente, foi provisionado um banco de dados isolado (*Postgres Embeddings*) dedicado apenas ao armazenamento das representações vetoriais pré-calculadas pela IA (*candidato_embeddings* e *vaga_embeddings*). Essa segregação estrutural representa uma necessidade arquitetônica fundamental: operações matemáticas complexas, como o cálculo da *Similaridade do Cosseno* comparando um único perfil contra centenas de vagas simultaneamente, demandam buscas intensivas. Ao isolar os vetores em um banco próprio, assegura-se que os cálculos de similaridade sejam realizados de forma otimizada e instantânea, protegendo as operações normais de leitura e escrita do *frontend* contra travamentos por concorrência de recursos (*locks*). Toda essa estruturação de domínios e microsserviços pode ser visualizada em detalhes na Figura 3.4.

Figura 3.4: Diagrama da arquitetura de integração da recomendação baseada em microserviços.

Fonte: Elaborado pelo autor utilizando a ferramenta Mermaid.

3.4.1 Processamento Assíncrono com Mensageria e *Worker* de Indexação

A integração do modelo SBERT ao ambiente de produção exigiu uma separação computacional em serviços independentes. Segundo a literatura de *AI Engineering* e os preceitos de Huyen (HUYEN, 2025), a implantação de modelos densos requer infraestruturas assíncronas e armazenamento vetorial isolado. Essa abordagem previne que o alto custo de processamento semântico interfira na estabilidade e no tempo de resposta das transações cotidianas do sistema.

Como a plataforma recebe atualizações simultâneas de diversos usuários, o cálculo síncrono dos vetores resultaria em bloqueios de recursos e lentidão na interface. Para contornar esse problema, adotou-se o *RabbitMQ* como ferramenta de mensageria. Ele atuou como um *buffer* (fila de espera), organizando as requisições de indexação de forma assíncrona.

A arquitetura foi desenhada para realizar indexações pontuais. Sempre que um candidato ou uma vaga era criado ou modificado no *backend* primário (*maisAcessoAPI*), o sistema disparava uma requisição HTTP não bloqueante (*fire-and-forget*) para o microserviço de recomendação, que inseria a tarefa na fila `recomendacao.indexacao`. Com o intuito de maximizar a assertividade dos *embeddings*, seguiu-se a recomendação de Liu, Wang e Zhang (2021) (LIU; WANG; ZHANG, 2021) sobre a necessidade de alta densidade semântica na entrada. Assim, o *payload* enviado já continha o agrupamento completo dos dados (habilidades, experiências, formações e indicativos de deficiência).

Por fim, um serviço isolado operando em segundo plano (o *Worker* de indexação) era encarregado de consumir essas mensagens. O ciclo de execução desse *Worker* consistia em montar o texto semântico agregado, processá-lo através do modelo *Sentence-BERT* e realizar a operação de *upsert* (inserção ou atualização do registro) diretamente no banco de dados vetorial. Esse fluxo de trabalho concentrou a carga de processamento apenas nas entidades que sofriam alterações, evitando cálculos demorados em lote (*batch*) e garantindo a atualização da plataforma em tempo real.

3.4.2 Recomendação Bidirecional e Fluxos de Busca

Consolidada a infraestrutura de inserção automática e assíncrona dos vetores, a API de recomendação passou a disponibilizar pontos de acesso dinâmicos operando de maneira bidirecional. Foram implementadas lógicas de consulta para cruzar os vetores espaciais tanto no sentido Candidato $\rightarrow$ Vagas, quanto no sentido Vaga $\rightarrow$ Candidatos.

3.4.3 Integração de Filtros Rígidos (*Hard Filters*)

Embora o ranqueamento gerado pela *Similaridade do Cosseno* ordene os vetores por proximidade contextual de forma eficaz, os resultados da inteligência artificial operam em conjunto com filtros rígidos fundamentados em regras de negócio. Essa arquitetura de integração assegura que o algoritmo não recomende vagas que ultrapassem barreiras restritivas inegociáveis, a exemplo da exclusividade de vagas para Pessoas com Deficiência (PCD) e limitações de modelo de contratação ou localização geográfica.

3.4.4 Composição do Modelo Híbrido

Apesar da superioridade dos *embeddings* contextuais (SBERT) na compreensão semântica profunda, a busca tradicional baseada em frequência de termos (TF-IDF) possui relevância operacional no recrutamento, atuando como uma âncora para a correspondência exata de siglas, ferramentas ou certificações específicas.

Para unir os benefícios de ambas as arquiteturas e mitigar as limitações individuais de cada uma, este trabalho propõe a construção de um Modelo Híbrido. O cálculo de ranqueamento final deste modelo consiste em uma combinação linear (soma ponderada) das similaridades calculadas pelos dois métodos. A pontuação final de compatibilidade (*Score Final*) entre um candidato e uma vaga é dada pela seguinte equação:

$$\text{Score}_{\text{Híbrido}} = (\alpha \times \text{Sim}_{\text{SBERT}}) + (\beta \times \text{Sim}_{\text{TF-IDF}}) \quad (3.1)$$

Onde $\text{Sim}_{\text{SBERT}}$ e $\text{Sim}_{\text{TF-IDF}}$ representam as similaridades do cosseno calculadas por cada

modelo, e os parâmetros α e β representam os pesos atribuídos a cada abordagem, definidos sob a restrição de que $\alpha + \beta = 1$. A otimização empírica para a descoberta dos pesos ideais que maximizam a assertividade do modelo é discutida na fase de avaliação de resultados.

Resultados e Discussão

Este capítulo corresponde à fase de Avaliação da metodologia CRISP-DM adotada na pesquisa. Com a arquitetura do sistema definida e o motor de recomendação plenamente integrado via microsserviços, fez-se necessário submeter a solução a testes para atestar sua eficácia. A validação deste trabalho foi estruturada em duas frentes principais: primeiramente, análises em cenários controlados, visando comprovar a capacidade matemática do modelo de compreender e isolar domínios semânticos; e, em um segundo momento, avaliações quantitativas sobre o histórico real de candidaturas da plataforma *Mais Acesso Conecta*, para mensurar o impacto direto da inteligência artificial sobre o comportamento prático dos usuários. As seções a seguir detalham os experimentos realizados e os ganhos obtidos em cada etapa.

4.1 Validação Global: Matriz de Confusão Semântica (*Heatmap*)

A validação global do modelo foi realizada através da construção de uma Matriz de Confusão Semântica, que cruza a média dos *scores* de similaridade entre perfis de candidatos de diferentes áreas de atuação (TI, Financeiro, Saúde, Vendas) e grupos de vagas correspondentes.

A Figura 4.1 apresenta o mapa de calor gerado, onde a intensidade da cor representa o grau de similaridade médio (0 a 1).

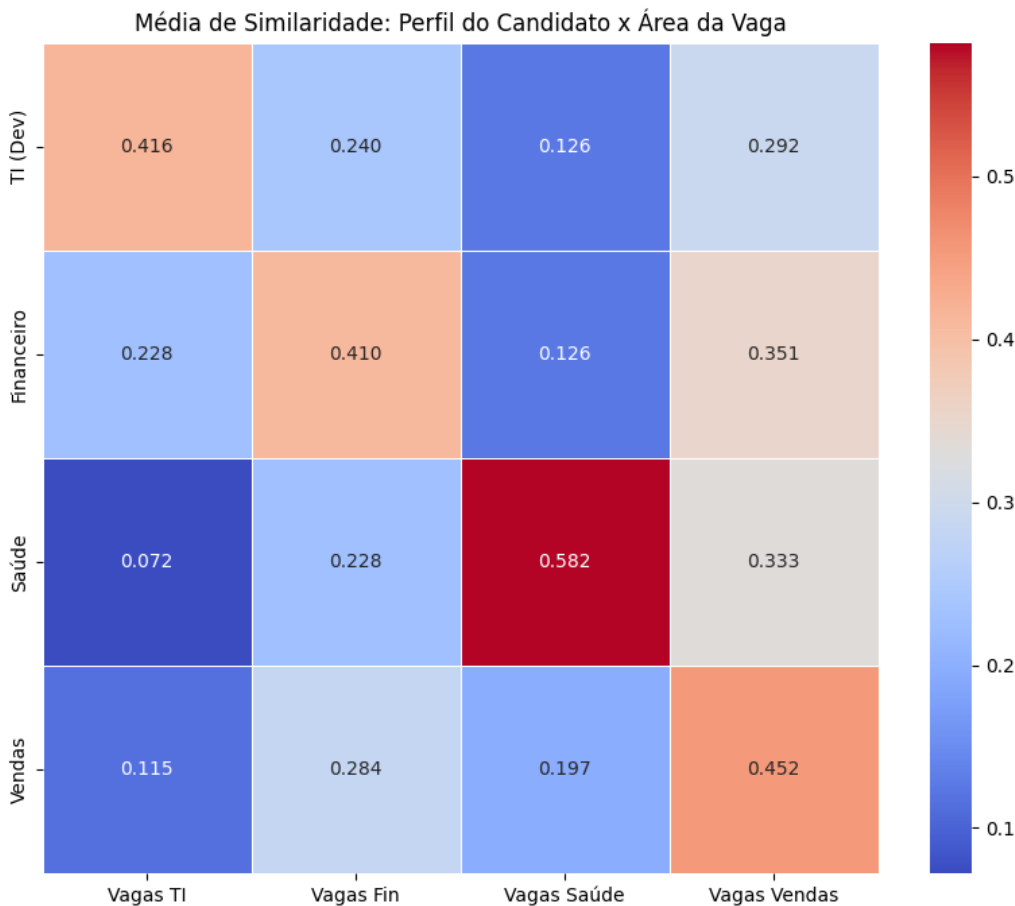


Figura 4.1: Matriz de Confusão Semântica: Média de Similaridade entre Perfis e Áreas de Vagas.

A análise da diagonal principal da matriz (destacada em tons mais quentes) demonstra a capacidade intrínseca do modelo *paraphrase-multilingual-mpnet-bas e-v2*, identificado e selecionado previamente no *benchmark* empírico da Seção 3.4.1, em identificar a área correta de atuação para cada perfil, mesmo sem treinamento supervisionado específico:

- **Alta Discriminação em Saúde:** O perfil de *Saúde* obteve um *score* médio significativamente superior com vagas da mesma área (**0.582**) em comparação com vagas de TI (**0.072**). Essa disparidade comprova que o modelo isolou com sucesso as competências técnicas específicas, validando a eficácia da limpeza textual seletiva.
- **Assertividade em Tecnologia:** O perfil de *Desenvolvedor (TI)* manteve uma média de **0.416** dentro de sua área de atuação, valor superior às médias obtidas com as demais categorias. Esse resultado **evidencia** a capacidade de generalização da **arquitetura SBERT**

ao aproximar **perfis e descrições que compartilham contextos correlatos** (como “Python”, “Java” e “Software”) na mesma região do espaço vetorial semântico.

- **Interseções Semânticas de Domínio:** Observou-se uma proximidade semântica natural entre as áreas *Financeira* e de *Vendas*, explicada pelo compartilhamento de vocabulário de negócios (termos como “metas”, “valores”, “gestão”).

4.2 Validação Comparativa Controlada e o Gap Semântico

Para validar a hipótese de que modelos contextuais superam métodos estatísticos clássicos no domínio de recrutamento, realizou-se inicialmente um experimento em cenários controlados, comparando o *score* de um perfil técnico (“Desenvolvedor *Full Stack*”) contra uma vaga pertinente ao seu nicho (“Cientista de Dados”) e uma vaga operacional irrelevante (“Recepcionista”).

Os resultados obtidos para este teste de controle foram:

- **Score de Nicho (Dev ↔ Engenheiro de Dados):** 0,5636
- **Score de Ruído (Dev ↔ Recepcionista):** 0,1874
- **GAP Semântico:** 0,3762

Como demonstrado neste estudo de caso ilustrativo, o modelo atribuiu um valor aproximadamente três vezes maior para a vaga de nicho em comparação à vaga de ruído. Este cenário exemplifica de forma qualitativa como a estratégia de estruturação dos dados direciona o modelo a priorizar a competência técnica, ilustrando na prática a distinção semântica alcançada pelo algoritmo.

4.3 Avaliação Quantitativa com Interações Reais (*Ground Truth*)

Embora as análises em cenários controlados comprovem a capacidade de distinção semântica do modelo, a eficácia de um sistema de recomendação deve ser atestada contra o comportamento

real dos usuários. Dessa forma, a avaliação quantitativa final foi realizada sobre o *ground truth* das candidaturas voluntárias registradas na plataforma *Mais Acesso Conecta*.

Tabela 4.1: Avaliação Baseada em Candidaturas Reais: Comparação de Modelos (N=105)

Modelo	K	Precision@K	Recall@K	nDCG@K
TF-IDF (<i>Baseline</i>)	3	0,0698	0,0322	0,0689
SBERT <i>Zero-Shot</i>	3	0,1270	0,0745	0,1276
Híbrido (40/60)	3	0,1302	0,0774	0,1242
TF-IDF (<i>Baseline</i>)	5	0,0724	0,0588	0,0751
SBERT <i>Zero-Shot</i>	5	0,1010	0,1045	0,1217
Híbrido (40/60)	5	0,1257	0,1184	0,1331
TF-IDF (<i>Baseline</i>)	10	0,0733	0,1360	0,1042
SBERT <i>Zero-Shot</i>	10	0,0971	0,1779	0,1504
Híbrido (40/60)	10	0,1143	0,2063	0,1655

Fonte: Elaborado pelo autor.

Ganho Relativo e Esparsidade de Dados

A análise dos resultados no cenário *Top-5* demonstra a superioridade da abordagem semântica. O modelo TF-IDF obteve nDCG@5 de 0,0751, enquanto o SBERT *Zero-Shot* atingiu 0,1217 e o modelo Híbrido alcançou 0,1331, representando ganhos relativos de 62,1% e 77,2%, respectivamente, sobre o *baseline*.

Para confirmar que essa melhoria representa um ganho real e não um artefato do acaso, aplicou-se o Teste de Wilcoxon *Signed-Rank*, um método não paramétrico adequado para amostras pareadas. Neste teste, estabeleceu-se a Hipótese Nula (H_0) de que não haveria diferença estatística no desempenho de ranqueamento (medido pelo nDCG@5) entre o modelo TF-IDF e o SBERT. Em contrapartida, a Hipótese Alternativa (H_1) postulava que o desempenho do modelo SBERT seria estatisticamente superior ao do método tradicional.

O teste retornou os valores $W = 829,50$ e $p = 0,0066$. Como o *p-value* obtido é estritamente inferior ao nível de significância adotado ($\alpha = 0,05$), rejeita-se a Hipótese Nula (H_0). Isso comprova matematicamente que a diferença de desempenho não é resultado de variação aleatória, validando a eficácia superior da abordagem semântica.

É importante contextualizar os valores absolutos obtidos. Conforme documentado na lite-

ratura de sistemas de recomendação com dados esparsos, em avaliações com interações reais, métricas como nDCG tendem a valores absolutos significativamente menores do que em ambientes controlados, pois o algoritmo é penalizado matematicamente por vagas compatíveis sugeridas com as quais o candidato, por fatores comportamentais, não interagiu. Esse efeito é agravado pelo *cold-start*: 64,5% dos candidatos da plataforma possuem perfis mínimos (5 *tokens* ou menos), limitando a extração de contexto. Contudo, os expressivos ganhos relativos comprovam que a busca semântica otimizou o poder de conexão da plataforma.

4.4 Validação das Técnicas de Preparação de Dados

Conforme estabelecido na metodologia, para avaliar o impacto prático das técnicas propostas na etapa de preparação de dados, foram conduzidos testes de isolamento de componentes. Nesse procedimento analítico, cada técnica foi temporariamente desativada do algoritmo para mensurar a sua contribuição direta e isolada no cálculo do GAP Semântico final:

- **Impacto da *Custom Stoplist*:** O experimento comparou o SBERT aplicado sobre o texto bruto contra o texto limpo pela lista de exclusão personalizada. A remoção da *custom stoplist* reduziu o GAP Semântico de 0,4759 para 0,3663 (queda de 29,9%). Isso valida que a remoção intencional de *soft skills* e termos genéricos concentra a representação vetorial nas *hard skills* relevantes, reduzindo o ruído semântico.
- **Impacto da Priorização de Contexto:** A comparação entre a ordem de concatenação estruturada neste projeto ([Habilidades + Cargo + Experiências + Formação]) e a ordem convencional de currículos ([Formação + Experiências + Habilidades]) resultou em um ganho marginal de +0,0077 no GAP Semântico (passando de 0,6885 para 0,6967). Embora o impacto quantitativo tenha sido discreto na amostra avaliada, a direção positiva justifica a manutenção da estratégia. Esse resultado indica que posicionar as competências técnicas no início do texto atua como uma medida de segurança arquitetônica preventiva: mesmo que o ganho médio seja baixo, a prática garante que as *hard skills* jamais sejam descartadas pelo limite de leitura de 512 *tokens* dos modelos BERT,

protegendo o ranqueamento de perfis excepcionalmente extensos.

4.4.1 Análise de Viés e Ética Algorítmica (PCD vs. Não-PCD)

O fundamento deste trabalho e o propósito principal da plataforma *Mais Acesso Conecta* exigem que a adoção da Inteligência Artificial não prejudique os candidatos que o sistema se propõe a incluir. Para comprovar a equidade do sistema, extraiu-se o *score* de similaridade das melhores vagas (*Top-1 Score*) recomendadas em um cenário de ampla concorrência para os 105 usuários que compõem a amostra de candidaturas reais extraída da própria plataforma (conforme estabelecido na Seção 4.1), sendo eles divididos em um grupo de 35 candidatos PCD e 70 candidatos não-PCD.

Aplicando-se o teste estatístico não paramétrico de Mann-Whitney, obteve-se um *p-value* de 0,2528. Sendo o valor *p* significativamente maior que o limiar de significância ($\alpha = 0,05$), não há evidências estatísticas suficientes para rejeitar a hipótese nula de igualdade entre as distribuições. Dessa forma, infere-se que não existe diferença significativa na qualidade das recomendações entre os dois grupos. A ausência de viés sistemático atesta que a estratégia de representação vetorial adotada trata candidatos PCD e não-PCD com equivalência semântica no mercado de trabalho real.

4.4.2 Estudo de Caso: Correspondência Funcional e Técnica

Para demonstrar a capacidade prática de recomendação do modelo, foi selecionado o caso de um candidato que destaca experiência prática como “Operador” e formação em “Mecânica”. A Figura 4.2 ilustra o ranqueamento das 4 vagas com maior similaridade para este perfil.

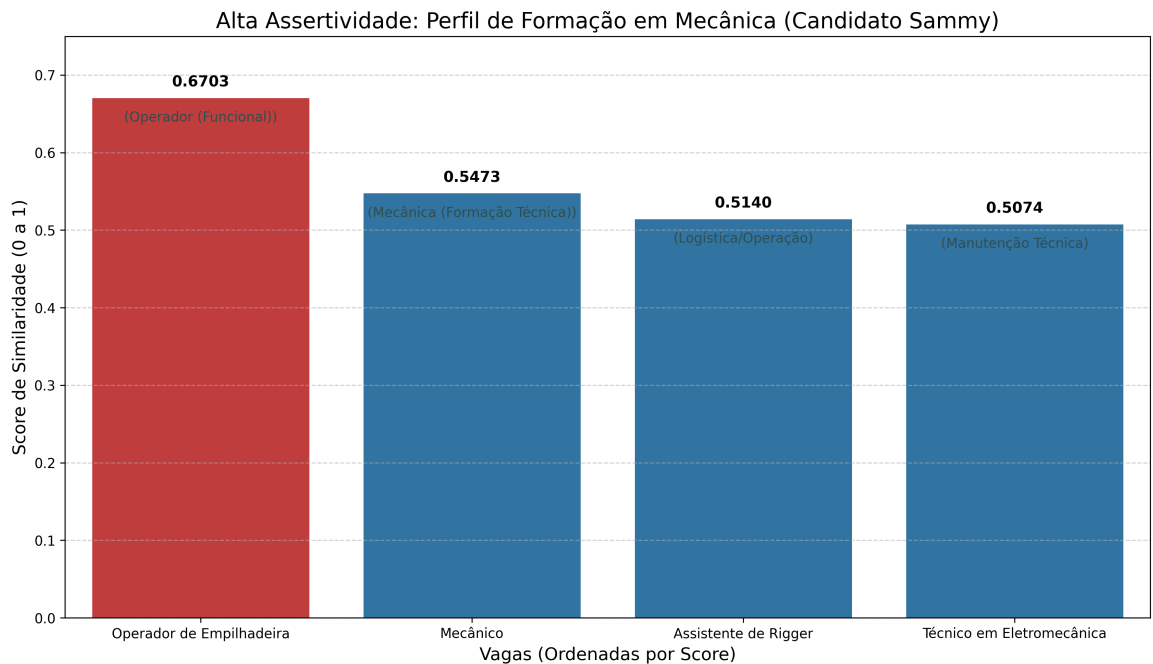


Figura 4.2: Estudo de Caso: Ranqueamento das Top 4 Vagas para o Candidato com Formação em Mecânica.

A análise da Figura 4.2 confirma a eficácia do modelo em ponderar experiência e formação de maneira unificada. O maior *score* (0,67) foi atribuído à vaga de *Operador de Empilhadeira*, identificando corretamente que a experiência prática (*Operador*) possuía alto peso semântico. A segunda vaga, *Mecânico* (0,54), correspondeu diretamente à formação técnica do candidato, demonstrando a compreensão profunda e multidimensional da Inteligência Artificial.

4.5 Estudo de Caso: Visão do Candidato e Priorização Afirmativa

Para validar o fluxo direto de recomendação (sentido Candidato → Vagas), simulou-se a experiência de uma usuária com deficiência (PCD) buscando oportunidades na plataforma. O escopo do perfil da candidata é composto por formação superior em Administração e histórico profissional atuando como Assistente Administrativa, com ênfase prática em controle de documentos, agendamentos e administração de eventos.

A Figura 4.3 ilustra a interface de usuário da plataforma com as respectivas oportunidades

ranqueadas pelo motor de inteligência artificial para este perfil.

A análise do ranqueamento evidencia o êxito do algoritmo em duas frentes fundamentais: a correspondência semântica e o alinhamento às regras de inclusão. Do ponto de vista técnico, o modelo SBERT correlacionou adequadamente as vivências da candidata (rotinas administrativas e organização de eventos) com posições pertinentes na área de gestão corporativa. O modelo foi capaz de mapear o escopo de atuação da formação em Administração, sugerindo com precisão cargos correlatos como *Assistente Administrativo*, *Assistente de Compras* e *Auxiliar Administrativo*.

Adicionalmente, o ranqueamento visualizado comprova o sucesso prático da integração entre o motor semântico e as regras de negócio (*hard filters*), conforme discutido na Seção 3.6.3. O sistema não apenas recuperou vagas com alta similaridade técnica, mas identificou a *flag* de deficiência no perfil da usuária e priorizou ativamente as oportunidades exclusivas e afirmativas para o público PCD no topo da lista (posicionando a vaga de *Assistente Administrativo Remoto PCD* na primeira posição).

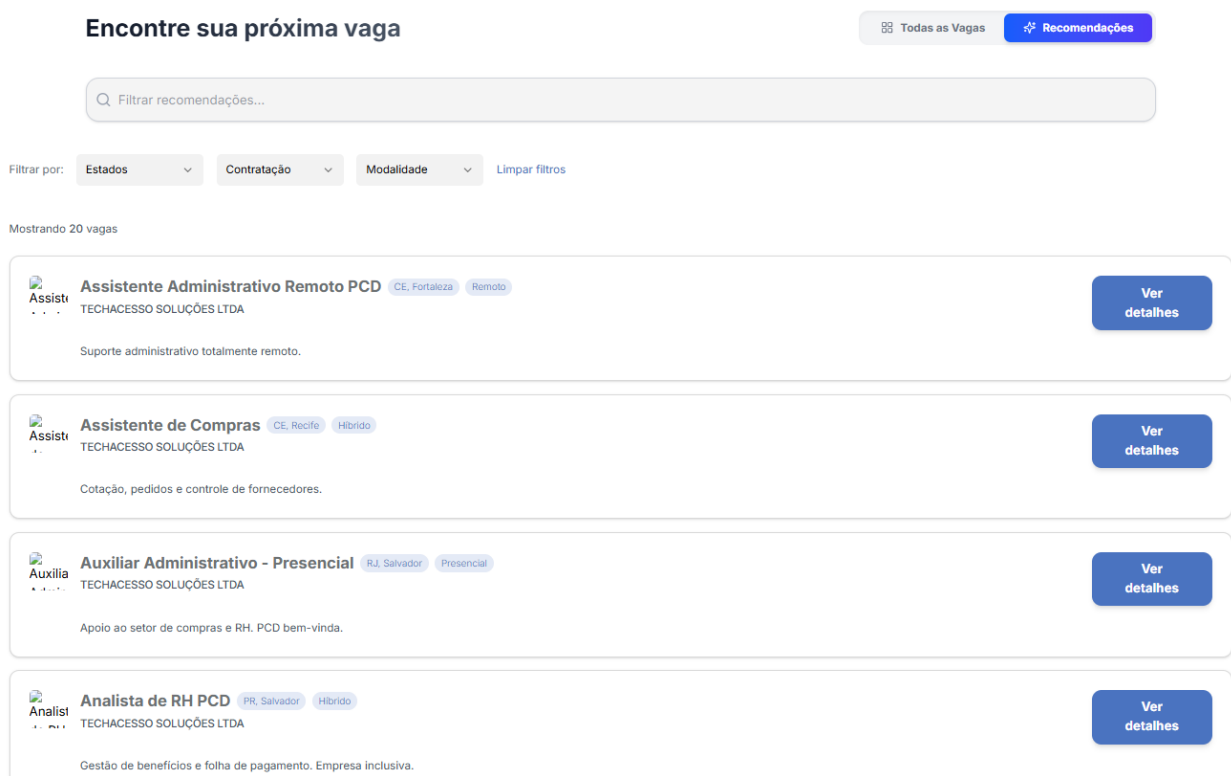
Esse comportamento atesta que a arquitetura híbrida desenvolvida atinge seu objetivo principal: aliar a alta precisão matemática do Processamento de Linguagem Natural à política de inclusão da plataforma, poupando a candidata de buscas manuais exaustivas e entregando, de imediato, um portfólio de vagas altamente compatíveis e acessíveis.

4.6 Estudo de Caso: Busca Reversa e Visão do Recrutador

Para comprovar a bidirecionalidade da arquitetura desenvolvida, simulou-se a experiência de uso sob a ótica do recrutador (fluxo Vaga $\rightarrow$ Candidatos). A plataforma *Mais Acesso Conecta* permite que a empresa, ao cadastrar uma nova oportunidade, acione a Inteligência Artificial para realizar a triagem automática do banco de talentos.

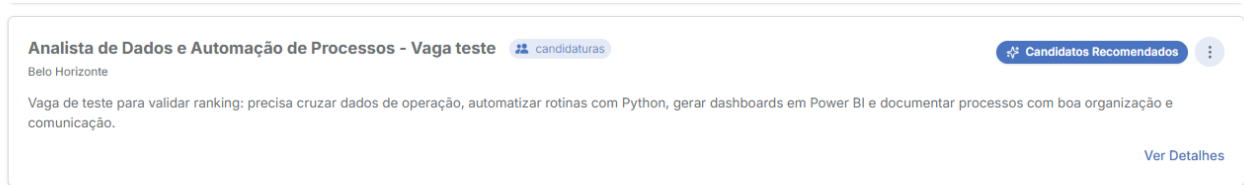
A Figura 4.4 ilustra a interface de uma oportunidade cadastrada para o cargo de “Analista de Dados e Automação de Processos”. O escopo da vaga exige o cruzamento de dados de operação, automação de rotinas utilizando Python e a construção de *dashboards* em Power BI, além de requisitar competências comportamentais como organização e comunicação.

Figura 4.3: Visão da a: Ranqueamento de vagas para perfil Administrativo PCD.



Fonte: Elaborado pelo autor.

Figura 4.4: Interface do Recrutador: Detalhes da vaga de Analista de Dados.



Fonte: Elaborado pelo autor.

Ao solicitar a recomendação de talentos, o microserviço de inteligência artificial calcula a Similaridade do Cosseno entre o vetor semântico desta vaga e os vetores de todos os candidatos da base. A Figura 4.5 apresenta o modal de resultados retornado pelo sistema.

A análise do ranqueamento evidencia a alta precisão técnica do modelo SBERT e o impacto direto do *pipeline* de Processamento de Linguagem Natural (PLN). O sistema atribuiu os maiores *Match Scores* (71% e 70%) aos candidatos Ricardo Almeida Torres e Paulo Henrique Nogueira. O algoritmo identificou com sucesso que as *hard skills* de ambos (Python, SQL e Power BI) e suas trajetórias funcionais (“Coordenador de Inteligência Operacional”, “Especia-

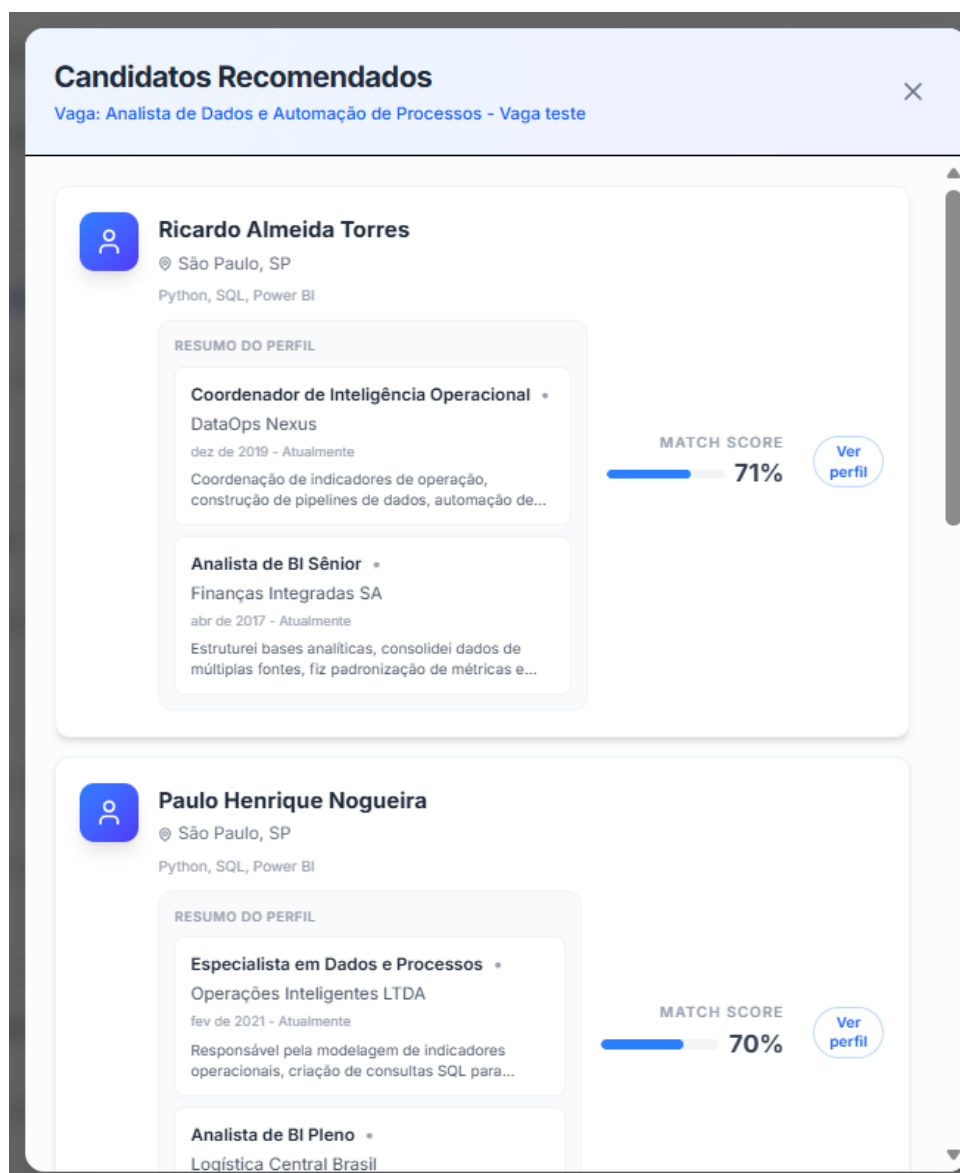


Figura 4.5: Resultado da Busca Reversa: Ranqueamento de candidatos por Match Score.

lista em Dados” e “Analista de BI”) estão perfeitamente alinhadas às necessidades de cruzamento e automação descritas na vaga.

Destaca-se, o aprimoramento com o uso da *Custom Stoplist* (detalhada na Seção 3.7.4). Embora o recrutador tenha incluído *soft skills* como “boa organização e comunicação” na descrição da vaga, a Inteligência Artificial filtrou esse ruído semântico. Em vez de retornar perfis generalistas com boa comunicação, o modelo isolou a essência técnica do texto e posicionou no topo da lista profissionais de dados altamente especializados. A análise deste resultado indica que a engenharia de contexto reduz o impacto de ambiguidades no sistema, poupando o tempo

de triagem manual e direcionando o *match* para a qualificação técnica do candidato.

Cabe ressaltar que, conforme fundamentado na metodologia, todos os nomes de candidatos exibidos nestas capturas de tela consistem em dados inteiramente fictícios (*mock data*), gerados exclusivamente para fins de homologação do sistema, em estrita conformidade com as diretrizes da Lei Geral de Proteção de Dados (LGPD).

Para evidenciar a capacidade discriminatória do sistema, a Figura 4.6 apresenta o comportamento do modelo ao avaliar perfis com baixa aderência técnica para a mesma oportunidade de Analista de Dados e Automação.

O ranqueamento demonstra que este candidato obteve um *Match Score* substancialmente inferior (51%). A análise de seu resumo profissional revela que sua trajetória é inteiramente focada no setor operacional e comercial, com experiências como “Auxiliar de Atendimento” e “Repositor”. O motor de recomendação penalizou adequadamente a pontuação devido ao enorme distanciamento semântico entre essas vivências e as *hard skills* fundamentais exigidas para o cargo.

Esse contraste reforça a robustez do algoritmo frente a perfis com trajetórias profissionais divergentes. Fica comprovado que o modelo SBERT compreende a profundidade do escopo de atuação da vaga, diferenciando claramente uma experiência em atendimento e reposição de estoque de uma formação puramente tecnológica. Na prática, essa precisão semântica blindou a plataforma contra candidatos fora do perfil, garantindo ao recrutador que profissionais sem a qualificação técnica pertinente permanecessem nas últimas posições da fila de triagem.

4.7 Considerações Finais

O desenvolvimento deste trabalho demonstrou que a aplicação de modelos semânticos no recrutamento requer a combinação de processamento de linguagem natural com regras de negócio. A arquitetura implementada uniu as representações vetoriais geradas pelo Sentence-BERT aos filtros de exclusividade para vagas PCD, garantindo que as recomendações fossem tecnicamente precisas e respeitassem as diretrizes da plataforma. Para viabilizar a execução do algoritmo em produção, adotou-se o processamento assíncrono via *RabbitMQ*. Essa decisão isolou a carga

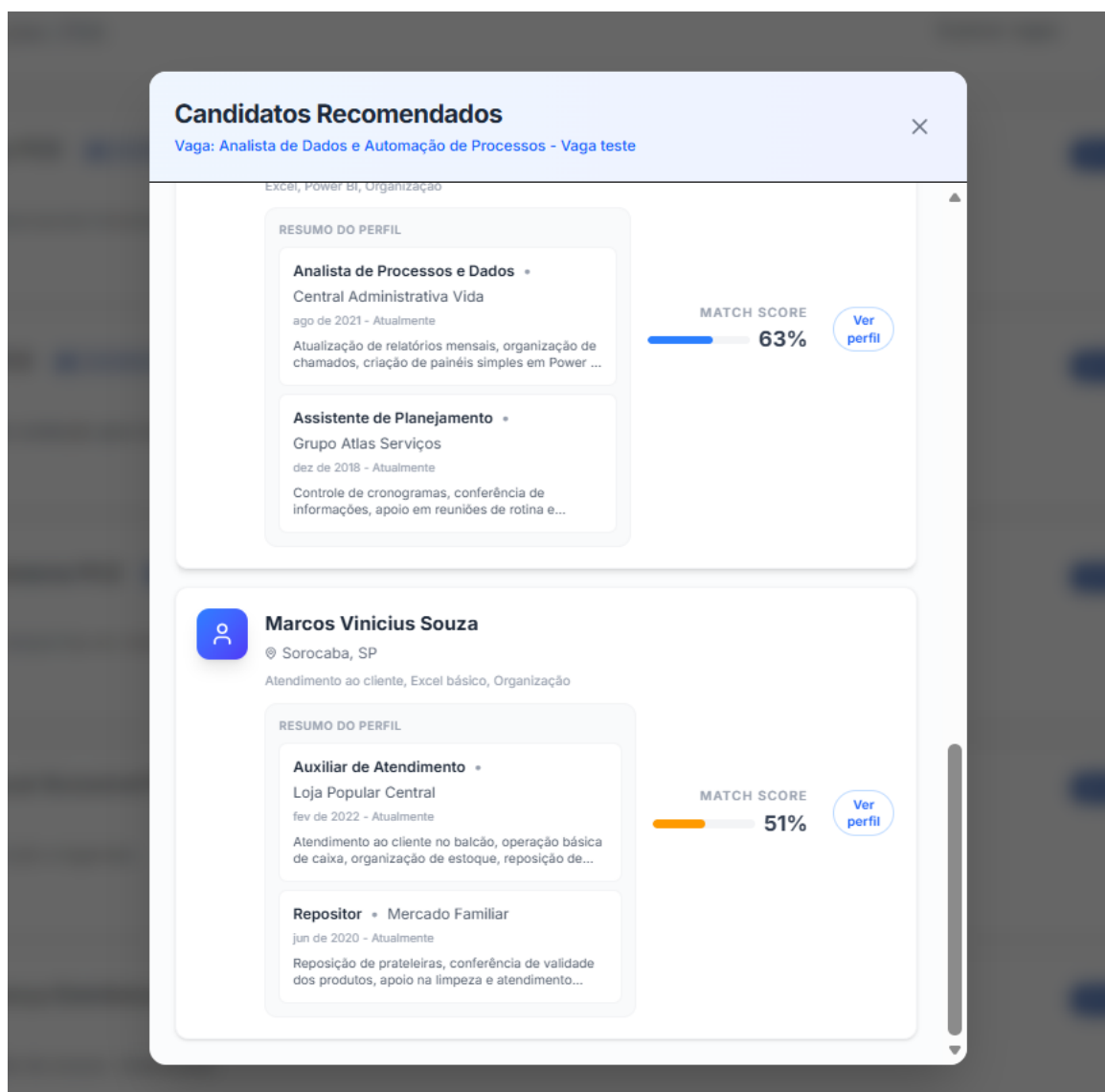


Figura 4.6: Visão do Recrutador: Exemplo de ranqueamento para perfis com baixa aderência técnica.

computacional da geração de *embeddings*, evitando lentidão no sistema principal e permitindo a atualização do ranqueamento de forma dinâmica.

Durante a fase de experimentação, os testes com *fine-tuning* do modelo SBERT (utilizando a base ASSIN2 e pares sintéticos em português) indicaram que o modelo pré-treinado *paraphrase-multilingual-mpnet-base-v2* já possuía capacidade de representação suficiente para o domínio de recrutamento. Sendo assim, a adoção da abordagem *Zero-Shot Learning* aliada à técnica de Engenharia de Contexto mostrou-se a solução mais adequada e de menor custo computacional para o estágio atual do projeto.

Apesar dos resultados positivos obtidos na avaliação das métricas, a validação do modelo evidenciou o impacto do problema de *cold-start*. A análise da base de dados revelou que 64,5% dos candidatos cadastrados possuíam perfis com cinco *tokens* ou menos após a etapa de pré-processamento. Como a eficácia do motor de recomendação baseado em conteúdo depende da densidade das informações, a capacidade preditiva do algoritmo sofre quedas substanciais para usuários com cadastros incompletos. Somado a isso, devido ao número restrito de candidatos com histórico robusto de interações na plataforma, os pesos do modelo híbrido foram calculados utilizando a mesma amostra do *ground truth* avaliado, configurando uma limitação metodológica que deve ser considerada na interpretação dos resultados absolutos.

Para mitigar essas limitações e evoluir o sistema nas próximas atualizações da plataforma *Mais Acesso Conecta*, sugere-se a implementação de técnicas de filtragem colaborativa. A utilização do histórico de cliques e candidaturas permitirá identificar padrões de interesse entre usuários semelhantes, complementando a análise de texto e contornando a ausência de dados em perfis recém-criados. Recomenda-se, também, a condução de testes A/B com recrutadores e candidatos para mensurar o ganho prático de usabilidade, além da futura aplicação de *fine-tuning* supervisionado utilizando dados orgânicos rotulados diretamente pela equipe de Recursos Humanos da plataforma.

Referências Bibliográficas

ALPAYDIN, E. *Introduction to Machine Learning*. 4. ed. Cambridge: MIT Press, 2020.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 35, n. 8, p. 1798–1828, 2013.

BOCHAROVA, M. Y.; MALAKHOV, E. V.; MEZHUYEV, V. I. Vacancysbert: the approach for representation of titles and skills for semantic similarity search in the recruitment domain. *Applied Aspects of Information Technology*, v. 6, n. 1, p. 52–59, 2023.

BURKE, R. Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction*, v. 12, n. 4, p. 331–370, 2002.

CHAPMAN, P. et al. *CRISP-DM 1.0: Step-by-step data mining guide*. [S.l.], 2000. Acesso em: mai. 2026. Disponível em: <<https://www.the-modeling-agency.com/crisp-dm.pdf>>.

DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*. [s.n.], 2019. p. 4171–4186. Disponível em: <<https://arxiv.org/abs/1810.04805>>.

ELOV, B. B.; KHAMROEVA, S. M.; XUSAINOVA, Z. Y. The pipeline processing of NLP. *E3S Web of Conferences*, EDP Sciences, v. 413, p. 03011, 2023.

EVANS, E. *Domain-Driven Design: tackling complexity in the heart of software*. Boston: Addison-Wesley Professional, 2004.

HOHPE, G.; WOOLF, B. *Enterprise Integration Patterns: designing, building, and deploying messaging solutions*. Boston: Addison-Wesley Professional, 2004.

HUYEN, C. *AI Engineering*. 1. ed. Sebastopol: O'Reilly Media, 2025.

Instituto Brasileiro de Geografia e Estatística (IBGE). *Pesquisa Nacional por Amostra de Domicílios Contínua: indicadores de mercado de trabalho*. Rio de Janeiro: [s.n.], 2023. Acesso em: 10 set. 2025. Disponível em: <<https://www.ibge.gov.br/>>.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: an introduction to natural language processing, computational linguistics, and speech recognition*. 3. ed. Stanford University, 2023. Versão preliminar (draft). Acesso em: set. 2025. Disponível em: <<https://web.stanford.edu/~jurafsky/slp3/>>.

- LAVI, D.; MEDENTSIY, V.; GRAUS, D. consultantbert: Fine-tuned siamese sentence-bert for matching jobs and job seekers. In: *RecSys in HR 2021: The 1st Workshop on Recommender Systems for Human Resources*. Amsterdam, the Netherlands: [s.n.], 2021.
- LAVI, D.; MEDENTSIY, V.; GRAUS, D. consultantbert: Fine-tuned siamese sentence-bert for matching jobs and job seekers. In: *RecSys in HR 2021: The 1st Workshop on Recommender Systems for Human Resources*. Amsterdam, the Netherlands: [s.n.], 2021. Disponível em: <<https://arxiv.org/abs/2109.06501>>.
- LIU, X.; WANG, L.; ZHANG, J. An intelligent job recommendation system based on semantic embeddings and machine learning. *Journal of Intelligent Information Systems*, v. 56, n. 2, p. 321–345, 2021.
- MAIS ACESSO. *Plataforma Mais Acesso Conecta*. 2025. Acesso em: 10 set. 2025. Disponível em: <<https://www.maisacesso.com.br/>>.
- NEWMAN, S. *Building Microservices: designing fine-grained systems*. 1. ed. Sebastopol: O'Reilly Media, 2015.
- NOBRE, A. J. P.; LEITE, A. C. R. *Desafios Éticos Associados aos Vieses em Algoritmos de Inteligência Artificial: um mapeamento sistemático da literatura*. Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2025.
- REIMERS, N.; GUREVYCH, I. Sentence-BERT: sentence embeddings using siamese BERT-networks. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, 2019. p. 3982–3992.
- RIBEIRO, B. F. d. S. *Resolver o problema de cold start e escassez de dados com algoritmos híbridos*. Dissertação (Mestrado) — Instituto Superior de Engenharia do Porto (ISEP), 2025. Disponível em: <<http://hdl.handle.net/10400.22/30790>>.
- RICCI, F.; ROKACH, L.; SHAPIRA, B. *Recommender Systems Handbook*. 3. ed. Cham: Springer, 2022.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 4. ed. New Jersey: Pearson, 2021.
- SCHAFER, J. B. et al. Collaborative filtering recommender systems. In: BRUSILOVSKY, P.; KOBSA, A.; NEJDL, W. (Ed.). *The Adaptive Web: methods and strategies of web personalization*. [S.l.]: Springer, 2007. p. 291–324.
- SHISHEHCHI, S.; BANIHASHEM, S. Y. JRDP: a job recommender system based on ontology for disabled people. *International Journal of Technology and Human Interaction*, v. 15, n. 1, p. 85–99, 2019.
- YAP, R.-E.; HAW, S.-C.; AL-JUBOORI, S. A comprehensive review on machine learning-based job recommendation systems. *International Journal on Robotics, Automation and Sciences*, v. 7, n. 2, p. 36–55, 2025.

ZHANG, M.; JENSEN, K. N.; PLANK, B. *JobBERT: understanding job titles through job advertisements*. 2022. ArXiv:2109.09605. Acesso em: mai. 2026. Disponível em: <<https://arxiv.org/abs/2109.09605>>.