aurum
Revista Multidisciplinar

ISSN: 3085-7783

PROCESSAMENTO DE LINGUAGEM NATURAL PARA SISTEMAS DE GERENCIAMENTO DE DEMANDAS

NATURAL LANGUAGE PROCESSING FOR CORPORATE MANAGEMENT SYSTEMS

https://doi.org/10.63330/armv1n10-004

Submetido em: 08/12/2025 e Publicado em: 12/12/2025

Victor Hugo de Sá Reis
Graduando em Engenharia de Computação
Universidade do Norte (UNINORTE)
E-mail: victorsa.br@gmail.com

Guilherme Elias Marinho
Graduando em Engenharia de Computação
Universidade do Norte (UNINORTE)
E-mail: gmarinho2002@hotmail.com

Wanderlan Carvalho de Albuquerque
Prof. Me. Orientador
Universidade do Norte (UNINORTE)
E-mail: wcacarvalho@gmail.com

Roneuane Grazielle da Gama Araujo
Especialista em Engenharia e Mineração de Dados
Centro Universitário do Norte – Uninorte
E-mail: roneuanegrazielle@gmail.com

RESUMO
O gerenciamento de demandas em corporações é frequentemente prejudicado pela falta de objetividade na descrição das solicitações, resultando em perda de produtividade. Este trabalho objetiva explorar as oportunidades em Processamento de Linguagem Natural (PLN) em sua intersecção com tal problemática. Após um passeio bibliográfico por técnicas de PLN e ferramentas de gerenciamento, propõe-se uma arquitetura de sistema integrável objetivando categorizar demandas e sumarizar seus objetivos. Como resultado esperado, estima-se trazer luz à problemática da comunicação corporativa através da inteligência artificial com enfoque especial em demonstrar seu alto potencial para mitigar os ruídos da comunicação natural, acelerar a triagem de tarefas e melhorar a alocação de recursos nas empresas.

Palavras-chave: Processamento de Linguagem Natural; Gerenciamento de Demandas; Sumarização Automática; Categorização Automática; Otimização de Processos.

ABSTRACT
The management of requests within corporations is often hindered by a lack of objectivity in the description of demands, resulting in productivity losses. This study aims to explore opportunities in Natural Language Processing (NLP) at its intersection with this issue. Following a bibliographic review of NLP techniques and management tools, a system architecture is proposed, designed for integration and intended to categorize requests and summarize their objectives. As an expected outcome, the research seeks to shed light on the challenges of corporate communication through artificial intelligence, with a particular focus

on demonstrating its high potential to mitigate noise in natural communication, accelerate task triage, and improve resource allocation within organizations.

Keywords: Natural Language Processing; Demand Management; Automatic Summarization; Automatic Categorization; Process Optimization.

1 INTRODUÇÃO

A comunicação eficiente é um dos pilares do sucesso empresarial, uma vez que "é essencial para alinhar os objetivos da organização com as ações dos seus membros" (Leone et al., 2024). A forma como as demandas internas são descritas e organizadas impacta diretamente a produtividade, pois "a comunicação clara e eficaz pode reduzir mal-entendidos, melhorar a disposição dos funcionários e promover um ambiente de trabalho mais colaborativo" (Leone et al., 2024).

Atualmente, inúmeras empresas dependem de processos manuais onde os colaboradores descrevem suas necessidades em texto livre, método que abre margem para ambiguidades, visto que "grande parte dos problemas organizacionais têm origem na falta de comunicação, ou nos ruídos que impedem com que a mensagem seja devidamente compreendida" (Willemann et al., 2023).

A delimitação do problema de pesquisa reside na constatação de que, frequentemente, os colaboradores não possuem o vocabulário técnico ou o tempo necessário para formalizar uma demanda de forma clara. Essa falha pode criar "barreiras invisíveis que fragmentam a unidade organizacional, levando a uma deterioração do comprometimento e do desempenho" (Leone et al., 2024). Isso acarreta um ciclo de retrabalho para gestores, que precisam dedicar tempo para interpretar cada solicitação. Essa "falta de comunicação clara e consistente pode levar a erros de produção, um ambiente de trabalho tóxico, atrasos e resistência à mudança" (Leone et al., 2024), gerando gargalos significativos.

Nesse contexto, este trabalho busca demonstrar como é possível utilizar de tecnologias de PLN para a facilitação da comunicação em gestão de demandas, reduzindo o tempo gasto na triagem, minimizando erros e liberando os profissionais para atividades de maior valor agregado. Como objetivos específicos, busca-se: a) investigar algoritmos de classificação e sumarização de texto; b) pesquisar as principais ferramentas de gerenciamento de demandas do mercado; c) propor uma arquitetura de sistema para processamento de demandas; d) implementar um protótipo de alta fidelidade.

2 METODOLOGIA

Quanto à sua natureza, este trabalho classifica-se como uma pesquisa aplicada, pois "visa gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos" (Gil, 2017, p. 26), neste caso, o gargalo comunicacional na gestão de demandas empresariais.

Adota-se como procedimento técnico principal a Design Science Research (DSR), ou Pesquisa Baseada em Design. Esta abordagem é amplamente utilizada na ciência da computação e sistemas de informação, pois foca na "criação e avaliação de artefatos de TI (constructos, modelos, métodos, instâncias) com o objetivo de resolver problemas organizacionais" (Hevner et al., 2004). O artefato central desenvolvido nesta pesquisa é o protótipo funcional de PLN.

Quanto aos objetivos, a pesquisa incorpora elementos da pesquisa exploratória (na investigação de

algoritmos) e da pesquisa descritiva (na proposta e projeto da arquitetura). A abordagem do problema é predominantemente qualitativa. Para atender aos objetivos propostos, esta pesquisa adota uma abordagem metodológica estruturada em quatro principais fases, alinhadas aos objetivos específicos.

2.1 INVESTIGAÇÃO DE ALGORITMOS

A primeira fase consistiu em uma revisão bibliográfica estruturada sobre o estado da arte em algoritmos de classificação e sumarização de texto. O instrumento de coleta foi a pesquisa em bases de dados científicas, como IEEE Xplore, ACM Digital Library e Google Scholar, utilizando palavras-chave como "classificação de texto PLN", "sumarização automática de texto", "NLP for task management" e "BERT text classification". A técnica de análise foi a síntese qualitativa dos artigos, focando em comparar a adequação de modelos clássicos (ex: TF-IDF com SVM) e modelos baseados em deep learning (ex: Transformers, BERT) para o problema de demandas curtas e, por vezes, ambíguas.

2.2 PESQUISA DE FERRAMENTAS DE GERENCIAMENTO DE DEMANDAS

Esta fase teve como objetivo mapear e comparar ferramentas de mercado voltadas ao gerenciamento de demandas corporativas, especialmente quanto a APIs, webhooks, automações e suporte à integração com softwares terceiros. A técnica de coleta utilizada foi a análise documental, baseando-se principalmente na documentação oficial de cada ferramenta bem como em manuais e artigos técnicos. O foco da análise foi identificar funcionalidades, limitações e oportunidades de integração com a arquitetura a ser proposta.

2.3 PROPOSTA DE ARQUITETURA

Com base nos requisitos e informações levantados nas fases precedentes, esta etapa focou no design do artefato. A técnica utilizada foi a modelagem de sistemas. Foi projetada a arquitetura de um sistema capaz de integrar-se com as ferramentas de gerenciamento de demandas e oferecer controle e eficiência aos seus clientes. O principal instrumento de formalização foi o uso do diagrama sequencial para elucidar simultaneamente o caso de uso principal e seus os elementos arquiteturais constituintes.

2.4 IMPLEMENTAÇÃO DO PROTÓTIPO

Finalmente, a esta fase compete instanciar, em menor escala, o artefato (Hevner et al., 2004). A técnica foi a prototipagem rápida. Como viabilizadores, optou-se pela linguagem Python (v. 3.9+) utilizando modelos pré-treinados para sumarização e classificação do texto, *RabbitMQ* para implementação da fila de mensagens e *Flask* para a implementação da interface de API.

3 RESULTADOS E DISCUSSÃO

Os resultados da metodologia de Pesquisa Baseada em Design (DSR) foram aprofundadas em fases mapeadas a partir dos objetivos específicos.

3.1 INVESTIGAÇÃO DE ALGORITMOS DE CLASSIFICAÇÃO E SUMARIZAÇÃO DE TEXTO

A área de Processamento de Linguagem Natural (PLN) utiliza algoritmos de classificação para organizar textos em categorias predefinidas, os quais se dividem em duas grandes famílias: a clássica e a moderna. As técnicas clássicas, conforme exploradas no survey de Aggarwal e Zhai (2012), dependem fundamentalmente de uma etapa explícita de engenharia de features para converter texto em representações numéricas. A abordagem mais difundida para isso é o TF-IDF (Term Frequency-Inverse Document Frequency). Este método calcula um peso para cada palavra (token) com base em dois fatores: sua frequência no documento específico (Term Frequency) e sua raridade no conjunto total de documentos (Inverse Document Frequency), permitindo assim que palavras mais relevantes e distintivas recebam maior peso.

Entre os algoritmos clássicos (Aggarwal, 2012) que consomem essas features, o Naive Bayes é um dos mais tradicionais. Ele funciona como um algoritmo probabilístico baseado no Teorema de Bayes, calculando a probabilidade de um texto pertencer a uma categoria com base na ocorrência das palavras, mas assume uma "ingênua" independência entre elas. Embora muito rápido e surpreendentemente eficaz com poucos dados (sendo um ótimo baseline), sua principal desvantagem é ignorar o contexto e a ordem das palavras. Outra abordagem clássica robusta (Aggarwal, 2012) é a das Máquinas de Vetor de Suporte (SVM). O SVM opera encontrando um hiperplano ideal que melhor separa as diferentes classes de dados em um espaço de alta dimensão, como o criado pelas features TF-IDF. Ele demonstra alto desempenho, especialmente com vocabulários extensos, e é bom em prevenir overfitting, mas pode ser computacionalmente caro para treinar com grandes volumes de dados.

Em contraste, as técnicas modernas baseadas em deep learning revolucionaram a classificação ao aprenderem automaticamente as features relevantes diretamente dos dados brutos. Isso elimina a necessidade de métodos manuais como o TF-IDF, permitindo a captura de contexto e nuances semânticas complexas. As Redes Neurais Recorrentes (RNN), e suas variantes mais sofisticadas como LSTM e GRU, foram as primeiras a se destacar. Elas processam o texto sequencialmente, palavra por palavra, mantendo um estado de memória que captura informações de palavras anteriores. Embora capazes de entender a ordem sequencial, seu treinamento é inerentemente lento (pois não pode ser paralelizado) e podem ter dificuldade com textos muito longos.

O estado da arte, no entanto, foi redefinido pelos Modelos Transformers, notavelmente o BERT (Devlin, 2018). Diferente das RNNs, os Transformers utilizam um mecanismo de "auto-atenção" para

analisar todas as palavras de um documento de uma vez, pesando a importância de cada palavra em relação a todas as outras no contexto geral. Modelos como o BERT são “pré-treinados em grandes volumes de texto e depois ajustados” (Devlin, 2018) para tarefas específicas. Suas vantagens são o desempenho de ponta na compreensão de contexto profundo e ambiguidades, mas seu custo computacional para treinamento e execução é muito intenso.

No campo da sumarização de texto, o objetivo é criar uma versão concisa de um documento. Conforme levantado no survey de Allahyari et al. (2017), as técnicas dividem-se em extrativas e abstratas.

A sumarização extrativa "seleciona e extrai as sentenças mais importantes do texto original" (Allahyari, 2017) para compor o resumo. Uma abordagem extrativa simples (Allahyari, 2017) é a baseada em TF-IDF: cada sentença é pontuada somando os pesos TF-IDF de suas palavras, e as sentenças com maior pontuação são selecionadas. É um método rápido, mas pode resultar em resumos redundantes ou incoerentes. Uma evolução significativa é o TextRank (Mihalcea, 2004), um algoritmo baseado em grafos onde as sentenças são os nós, e as arestas representam a similaridade entre elas. Conforme proposto por Mihalcea e Tarau (2004), o TextRank identifica as sentenças mais "centrais" (ou seja, mais similares às outras). Isso tende a criar resumos mais relevantes e menos redundantes que o TF-IDF, mas ainda sofre de problemas de coerência, pois apenas "cola" sentenças originais.

Em oposição, a sumarização abstrata busca imitar o processo humano, "compreendendo o significado do texto original e gerando um novo resumo com palavras próprias" (Allahyari, 2017). Os pioneiros nessa área foram os modelos Sequence-to-Sequence (Seq2Seq), que usam um codificador (encoder) para "ler" o texto de entrada e um decodificador (decoder) para gerar o resumo palavra por palavra. Embora tenham sido a primeira abordagem a realmente "escrever" resumos, eles tendem a esquecer detalhes do início do texto se o documento for longo. Atualmente, os Modelos Transformers representam o estado da arte. Modelos como o BART (Lewis, 2019) e o T5 (Raffel, 2020) são projetados especificamente para tarefas de reescrita e geração. O BART, por exemplo, é otimizado através de "denoising sequence-to-sequence pre-training" (Lewis, 2019), tornando-o excelente em reformular textos. O T5 (Raffel, 2020) unifica diversas tarefas de PLN sob um mesmo framework "text-to-text". Esses modelos produzem os resumos mais fluentes e coerentes, mas são computacionalmente muito pesados e podem, ocasionalmente, "alucinar" fatos que não estavam no texto original.

O que se confirma é que modelos de deep learning, como o BERT (Devlin, 2018), são mais adequados para textos curtos e ambíguos, superando as abordagens clássicas (Aggarwal, 2012) em contextos corporativos complexos.

3.2 PESQUISA DE FERRAMENTAS DE GERENCIAMENTO DE DEMANDAS

Das plataformas analisadas, observou-se que **Jira**, **Asana**, **monday.com** e **Trello** são amplamente adotadas e oferecem APIs e mecanismos de automação que permitem integração com pipelines externas de PLN. Ferramentas como **ClickUp**, **ServiceNow** e **Zendesk** também apresentam APIs robustas e são frequentes em cenários corporativos (enterprise), sendo especialmente indicadas quando se exige alta governança, controle de fluxos de trabalho complexos e conformidade. Estas plataformas, porém, **não** costumam incorporar, nativamente, modelos avançados de PLN (classificação ou sumarização automática) — a integração típica é feita via API / webhooks / automações que disparam serviços externos responsáveis pelo processamento de texto.

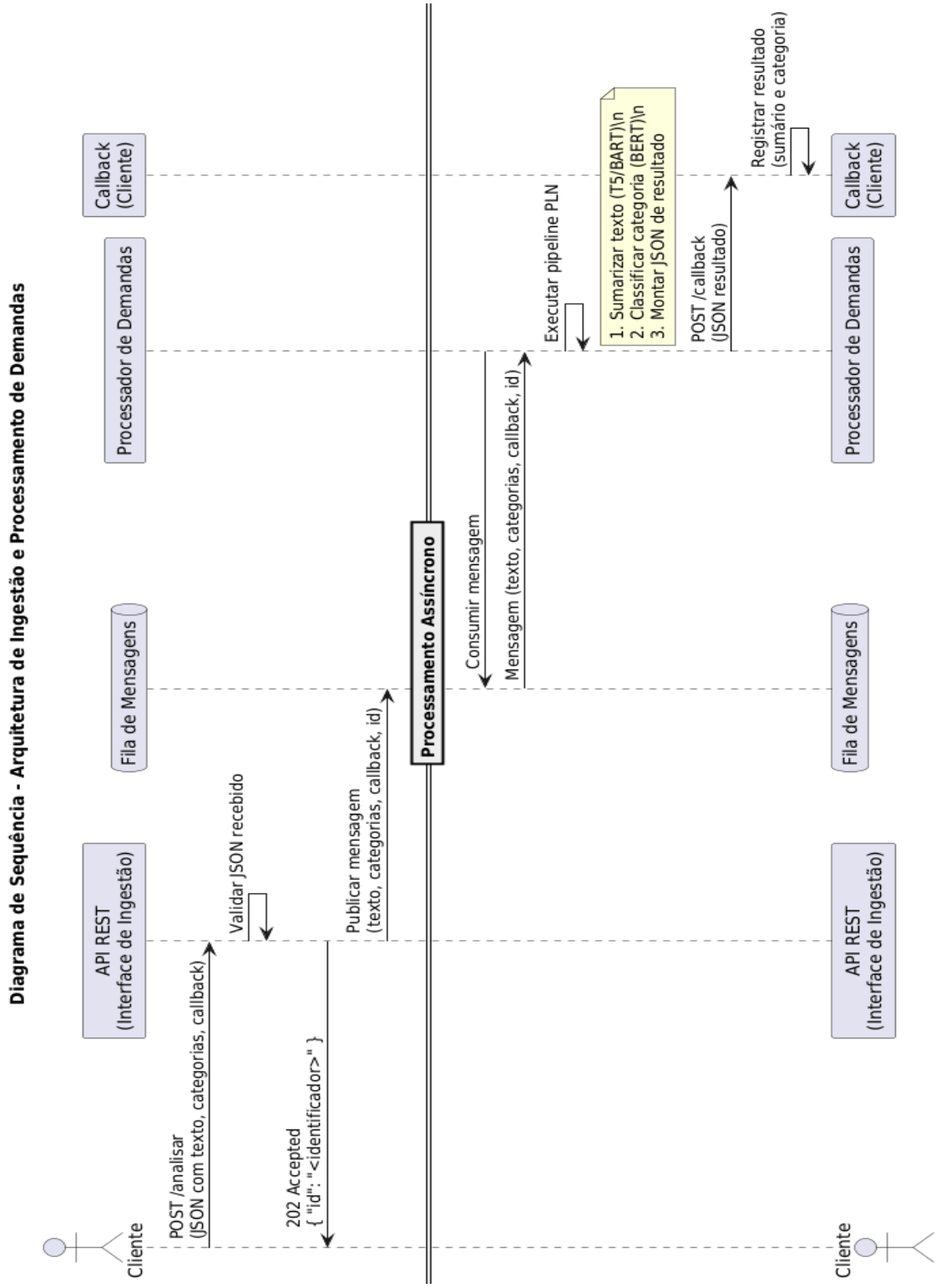
Tabela 1 - Resultados de pesquisa ferramental

Ferramenta	Tipo / Público	API / Webhooks	Automação / Workflows	Integração com Arquitetura Proposta
Jira (Atlassian)	Equipes de software / enterprise	REST API madura; suporte a apps e webhooks.	Workflows configuráveis; automações internas e apps.	Excelente para integração em times que já usam issue tracking; bom para log de eventos e atribuição automática.
Trello (Atlassian)	Projetos simples / kanban	REST API e webhooks fáceis de consumir.	Power-Ups e automações (Butler).	Ótimo para MVPs e protótipos; API simples facilita ingestão/atualização de cards.
Asana	Times cross-functional / mid-large	API e SDKs; automações e integrações.	Regras, templates e integrações nativas.	Bom equilíbrio entre usabilidade e poder; indicado para POCs com workflows mais complexos.
monday.com	Business teams / personalizável	GraphQL API; automações; Webhooks.	Altamente personalizável; automações visuais e integração com apps.	Muito bom para criar colunas customizadas (ex.: campo "categoria sugerida") e automações que acionam seu serviço de PLN.
ClickUp	Equipes que querem tudo-em-um	API com recursos de automação; foco em customização.	Templates, automações; scripts via API.	Útil para automações internas; boa documentação para integrações.
ServiceNow	Grandes corporações / ITSM	Plataforma ITSM com APIs e grande foco em fluxo corporativo.	Workflows complexos, conformidade e governança.	Indicada quando o ambiente exige ITSM e integração com CMDB/processos corporativos.
Zendesk	Atendimento / tickets / suporte	APIs para tickets, omnichannel e webhooks.	Regras, gatilhos e automações; foco em atendimento.	Ótimo quando as demandas nascem de atendimento ao cliente; permite chamadas a serviços de PLN para classificação/sumarização.

Fonte: Elaboração própria

3.3 PROPOSTA DE ARQUITETURA DE SISTEMA

Figura 1 - Diagrama de sequência: Arquitetura de Ingestão e Processamento de Demandas



Fonte: Elaboração própria.

Objetivando atender os requisitos do projeto, propomos uma arquitetura composta por dois módulos principais: a Interface de Ingestão e o Processador de Demandas. O fluxo de comunicação será assíncrono, utilizando uma fila de mensagens para mediar a comunicação entre os módulos.

A "Interface de Ingestão" será uma API REST pública, servindo como único entry point para o cliente. Ela será responsável por validar, enfileirar e responder imediatamente à requisição. Ao receber uma requisição, a API primeiro valida o JSON recebido (conforme a figura 1.1). Se for válido, a API não invocará o modelo de processamento imediatamente, ela gerará um identificador único para essa tarefa e publicará a solicitação completa (o texto, as categorias e um link de callback) na fila de mensagens.

O "Processador de Demandas" é um serviço privado que opera independentemente da API responsável apenas pelo consumo das mensagens na fila. Assim que uma nova demanda aparece na fila, o Processador a consome. Ele então executa o pipeline de PLN: primeiro, aplica o modelo de sumarização (como o T5 ou BART) ao texto original e, em seguida, alimenta o texto original (ou o sumário, dependendo da estratégia) no modelo de classificação (como o BERT) para determinar a categoria mais provável. Após concluir a análise e gerar o JSON de resultado (contendo o sumário e a categoria), utiliza-se a url de callback (fornecida na requisição original) para enviá-lo ao cliente.

Figura 2 - Exemplo de JSON utilizado na requisição

```json
{
  "id_demanda": "TICKET-1234",

  "texto_original": "texto original do cliente aqui...",

  "categorias_disponiveis": [
    "Demanda de Ti",
    "Financeiro",
    "Dúvidas Comerciais",
    "Cancelamento"
  ],

  "url_callback": "https://api.sistema-cliente.com/resultado_analise/{id_processamento}"
}
```

Fonte: Elaboração própria.

Figura 3 - Exemplo de JSON utilizado na resposta imediata

```json
{
  "id_processamento": "a1b2c3d4-e5f6g7h8-i9j1k2l3m4n5",

  "id_demanda_cliente": "TICKET-1234",

  "status": "EM_FILA",
  "mensagem": "Demanda recebida e registrada para processamento."
}
```

Fonte: Elaboração própria.

Figura 4 - Exemplo de JSON utilizado na resposta definitiva

```json
{
  "id_demanda_cliente": "TICKET-1234",

  "id_processamento": "a1b2c3d4-e5f6g7h8-i9j1k2l3m4n5",

  "sumario": "Texto original sumarizado aqui...",

  "classificacao": {
    "categoria_sugerida": "Financeiro",
    "confianca": 0.931
  },

  "status_processamento": "CONCLUIDO"
}
```

Fonte: Elaboração própria.

3.4 IMPLEMENTAÇÃO DE PROTÓTIPO DE ALTA FIDELIDADE

A Fase 4 resultou na implementação de um protótipo de alta fidelidade da arquitetura de sistema proposto utilizando Flask para implementação da API REST, RabbitMQ para implementação da fila de mensagens, para implementação do classificador e sumarizador, utilizou-se modelos pré-treinados. O código fonte integral do protótipo pode ser encontrado em nosso repositório (GMAR02, 2025).

4 CONCLUSÃO

Este trabalho teve como objetivo central trazer luz à problemática da comunicação corporativa no gerenciamento de demandas aos olhos da inteligência artificial, bem como propor uma arquitetura de sistema integrável com as principais plataformas do mercado. O problema investigado foi a perda de produtividade causada por descrições de solicitações ambíguas ou pouco objetivas, que geram gargalos e retrabalho. Os objetivos específicos foram alcançados. A pesquisa investigou algoritmos de classificação e sumarização bem como as principais ferramentas de mercado, suas oportunidades, características e impeditivos, propôs uma arquitetura de sistema terceiro para integração em ambiente corporativo e a instanciar através de um protótipo de alta fidelidade. A principal contribuição desta pesquisa é a materialização de uma solução de *Design Science* que aplica técnicas avançadas de PLN para resolver um problema organizacional prático e recorrente. Conclui-se que o uso de PLN, conforme proposto, possui alto potencial para mitigar os ruídos da comunicação natural, reduzir o tempo gasto na triagem de demandas e, consequentemente, otimizar a alocação de recursos nas empresas. Para pesquisas futuras, sugere-se a aplicação do protótipo em um estudo de caso dentro de uma organização específica, utilizando dados

proprietários para treinar o modelo, o que permitiria aferir seu desempenho com o jargão e os processos internos da empresa. Adicionalmente, a expansão do sistema para incluir novas funcionalidades, como a sugestão automática de prioridade da demanda.

REFERÊNCIAS

AGGARWAL, C. C.; ZHAI, C. A survey of text classification algorithms. In: MINING TEXT DATA. Boston, MA: Springer, 2012. p. 163–222.

ALLAHYARI, M. et al. Text summarization techniques: a brief survey. *arXiv preprint*, arXiv:1707.02268, 2017.

ASANA. Asana Developers — API, Documentation & Community. Disponível em: https://asana.com/developers. Acesso em: 20 set. 2025.

ATLASSIAN. Jira Cloud platform — REST API. Disponível em: https://developer.atlassian.com/cloud/jira/platform/rest/. Acesso em: 23 set. 2025.

ATLASSIAN. Trello REST API — Developer Documentation. Disponível em: https://developer.atlassian.com/cloud/trello/rest/. Acesso em: 25 set. 2025.

CLICKUP. ClickUp API Documentation. Disponível em: https://developer.clickup.com/. Acesso em: 28 set. 2025.

CRESWELL, J. W. Projeto de pesquisa: métodos qualitativo, quantitativo e misto. 3. ed. Porto Alegre: Artmed, 2010.

DEVLIN, J. et al. BERT: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*, arXiv:1810.04805, 2018.

GIL, A. C. Como elaborar projetos de pesquisa. 6. ed. São Paulo: Atlas, 2017.

GMAR02. Tccprot. GitHub, 2025. Disponível em: https://github.com/gmar02/tccprot. Acesso em: 18 nov. 2025.

GROWTH MARKETING PRO. Monday vs ClickUp vs Asana vs Trello [2025]. Disponível em: https://www.growthmarketingpro.com/monday-vs-clickup-vs-asana-vs-trello-project-management-tool-comparison/. Acesso em: 10 out. 2025.

HEVNER, A. R. et al. Design science in information systems research. *MIS Quarterly*, v. 28, n. 1, p. 75–105, 2004.

LEONE, E. L.; BORGES, T. F.; SANTOS, W. S. dos. Comunicação corporativa: desafios da comunicação para a melhoria da produção. *Ciência & Tecnologia: FATEC-JB*, Jaboticabal (SP), v. 15, n. 1, e16113, 2024.

LEWIS, M. et al. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint*, arXiv:1910.13461, 2019.

MIHALCEA, R.; TARAU, P. TextRank: bringing order into texts. In: *PROCEEDINGS OF THE 2004 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP)*, 2004.

MONDAY.COM. monday.com Platform API — GraphQL API Reference. Disponível em: https://developer.monday.com/api-reference/. Acesso em: 15 out. 2025.

RAFFEL, C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, v. 21, n. 140, p. 1–67, 2020.

SERVICENOW. IT Service Management — Product Overview / Docs. Disponível em: https://www.servicenow.com/products/itsm.html. Acesso em: 20 out. 2025.

SILVA, J. da. O impacto da inteligência artificial na educação moderna. 2. ed. São Paulo: Editora Futuro, 2023. 245 p.

SOFTWARE FINDER. Trello vs Asana vs Monday vs ClickUp: Ultimate Guide. Disponível em: https://softwarefinder.com/resources/trello-vs-asana-vs-monday-vs-clickup. Acesso em: 25 out. 2025.

WIERINGA, R. J. Design science methodology for information systems and software engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

WILLEMANN, I. M. et al. A relevância da comunicação eficaz no ambiente organizacional. In: *ANAIS DO XX CONGRESSO VIRTUAL DE ADMINISTRAÇÃO*, 2023.

ZENDESK. Zendesk Developer Docs — API Reference. Disponível em: https://developer.zendesk.com/api-reference/. Acesso em: 05 nov. 2025.