INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

BIANCA ALMEIDA DE OLIVEIRA BEZERRA

OUVIDOR.IA: sistema de ouvidoria com agrupamento e roteamento por inteligência artificial aplicado a contexto escolar

TERESINA, PIAUÍ
2025

BIANCA ALMEIDA DE OLIVEIRA BEZERRA

OUVIDOR.IA: sistema de ouvidoria com agrupamento e roteamento por inteligência artificial aplicado a contexto escolar

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Orientador: Prof. M^{e}. Fernando Castelo Branco Gonçalves Santana

TERESINA, PIAUÍ
2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Bezerra, Bianca Almeida de Oliveira

B574o Ouvidor IA : sistema de ouvidoria com agrupamento e roteamento por inteligência artificial aplicado a contexto escolar / Bianca Almeida de Oliveira Bezerra. - 2025.

47 f.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central, 2025.

Orientador : Prof Me. Fernando Castelo Branco Gonçalves Santana .

1. Inteligência artificial. 2. Ouvidoria. 3. Escola. I.Título.

CDD - 004.21

Elaborado por Tanize Maria Sales CRB 3/1024

BIANCA ALMEIDA DE OLIVEIRA BEZERRA

OUVIDOR.IA: sistema de ouvidoria com agrupamento e roteamento por inteligência artificial aplicado a contexto escolar

Trabalho de Conclusão de Curso (monografia) apresentado como exigência parcial para obtenção do diploma do Curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Aprovada em 19/12/2025.

BANCA EXAMINADORA:

Prof. Me. Fernando Castelo Branco Gonçalves Santana
(Orientador)
Instituto Federal do Piauí (IFPI)

Profª. Esp. Sandra Elisa Veloso Aguiar
Instituto Federal do Piauí (IFPI)

Profª. Esp. Ana Caroline Rodrigues Santana
Colégio Lerote

Dedico este trabalho aos meus pais Maria das Graças e Rogério.

AGRADECIMENTOS

Agradeço primeiramente a minha família, meu maior símbolo de amor incondicional. Aos meus pais Rogério e Graça sou eternamente grata por todo o apoio e pelos sacrifícios feitos para que minha trajetória educacional fosse a mais completa e me trouxesse até aqui hoje. Vocês são meus maiores exemplos, espero honrá-los da melhor maneira possível.

Ao meu irmão Gabriel, de quem tive o privilégio de acompanhar de perto a evolução pessoal e profissional. Sua dedicação e paixão por sua profissão me inspiram diariamente a buscar ser a melhor profissional que posso ser.

Agradeço aos meus amigos do IFPI, com quem compartilhei o curso e momentos especiais que sempre lembrarei com muito carinho. Obrigado por estarem sempre comigo em absolutamente tudo.

Agradeço aos professores do curso por todo o aprendizado. Sou grata em especial pelo meu orientador, Fernando, por ter me acolhido tão prontamente e por todas as direções dadas para a construção deste trabalho.

Agradeço ao Diego, que sempre esteve ao meu lado durante cada etapa. Suas palavras de apoio e seus gestos de cuidado foram essenciais para me manter firme mesmo em momentos de dificuldade.

Agradeço a equipe da gerência de TI da SSP-PI, onde tive minha primeira oportunidade em desenvolvimento de sistemas. Obrigada por terem acreditado no meu potencial, guardo todos na memória com carinho.

Por fim, agradeço às minhas amigas Tandryanny, Alanne e Fernanda, com quem divido a vida desde a infância e que continuam comigo me apoiando em cada passo. Muito obrigada por tudo!

RESUMO

Este trabalho apresenta o desenvolvimento do Ouvidor.IA, uma plataforma de ouvidoria escolar que emprega modelos de linguagem e técnicas de processamento de linguagem natural. O sistema reúne módulos de recuperação semântica, sumarização textual e roteamento automático para identificar temas recorrentes, agrupar relatos relacionados e direcionar cada demanda ao setor responsável. A solução busca organizar o crescente volume de dados em ouvidorias, oferecendo aos operadores um processo de triagem mais claro e consistente, sem substituir a decisão humana. Com a modernização do fluxo de atendimento e a qualificação da leitura das manifestações da comunidade, o Ouvidor.IA contribui para atualizar as práticas de ouvidoria, fortalecer a gestão institucional e aprimorar a escuta no ambiente escolar.

Palavras-chaves: inteligência artificial; ouvidoria; escola.

ABSTRACT

This work presents the development of Ouvidor.IA, a school ombudsman platform that employs language models and natural language processing techniques. The system integrates semantic retrieval, text summarization, and automatic routing modules to identify recurring themes, group related reports, and direct each request to the responsible sector. The solution aims to organize the growing volume of data in ombudsman offices, providing operators with a clearer and more consistent triage process without replacing human decision-making. By modernizing the service workflow and improving the interpretation of community requests, Ouvidor.IA contributes to updating ombudsman practices, strengthening institutional management, and enhancing listening processes within the school environment.

Keywords: artificial intelligence; ombudsman; school.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

ACID	*Atomicity, Consistency, Isolation, Durability* (Atomicidade, Consistência, Isolamento, Durabilidade)
API	*Application Programming Interface* (Interface de Programação de Aplicações)
BAAI	*Beijing Academy of Artificial Intelligence* (Academia de Inteligência Artificial de Pequim)
BERT	*Bidirectional Encoder Representations from Transformers* (Representações Bidirecionais de Codificadores de Transformadores)
CSR	*Client-Side Rendering* (Renderização no Cliente)
DRF	*Django REST Framework* (Framework REST do Django)
FAISS	*Facebook AI Similarity Search* (Busca de Similaridade da IA do Facebook)
GPT	*Generative Pre-trained Transformer* (Transformador Generativo Pré-Treinado)
GPU	*Graphics Processing Unit* (Unidade de Processamento Gráfico)
IA	Inteligência Artificial
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
JSON	*JavaScript Object Notation* (Notação de Objetos JavaScript)
LLM	*Large Language Model* (Modelo de Linguagem de Grande Escala)
MoE	*Mixture of Experts* (Mistura de Especialistas)
PLN	Processamento de Linguagem Natural
RAG	*Retrieval-Augmented Generation* (Geração Aumentada por Recuperação)
SEO	*Search Engine Optimization* (Otimização para Mecanismos de Busca)
SQL	*Structured Query Language* (Linguagem de Consulta Estruturada)
SSG	*Static Site Generation* (Geração de Site Estático)

SSR	*Server-Side Rendering* (Renderização no Servidor)
UML	*Unified Modeling Language* (Linguagem de Modelagem Unificada)

SUMÁRIO

1 INTRODUÇÃO

As ouvidorias surgiram como espaços institucionais voltados à mediação entre cidadãos e organizações. Seu desenvolvimento ganhou força ao longo do século XX, principalmente em democracias que buscavam formalizar mecanismos de participação social e controle dos serviços públicos. No Brasil, a consolidação dessas estruturas acompanhou o processo de redemocratização, aumentando o reconhecimento da escuta como instrumento de fortalecimento institucional e de garantia de direitos (PICCINI; REIS, 2023).

Nas últimas décadas, o papel das ouvidorias cresceu, deixando de atuar apenas como instâncias de reclamação para se tornarem espaços estratégicos de diálogo e gestão da informação. Esse movimento resultou da evolução do perfil dos profissionais que atuam na área e do aumento expressivo no volume das manifestações recebidas. Desde a Constituição de 1988, a institucionalização desses canais se intensificou em órgãos públicos e privados, respondendo à crescente demanda social por participação e melhoria dos serviços prestados (SPILLEIR; SILVA; SOUZA, 2021; BRASIL. Conselho Nacional de Justiça, 2025; DISTRITO FEDERAL. Ouvidoria-Geral, 2024).

No contexto escolar, as ouvidorias são importantes para promover uma gestão democrática, criando canais diretos de comunicação entre alunos, famílias, professores e gestores. Esses canais fortalecem o diálogo e permitem identificar problemas recorrentes no ambiente educacional, como questões de infraestrutura, relações interpessoais e acesso a direitos estudantis. Além de resolverem questões práticas, as ouvidorias escolares funcionam como ferramenta pedagógica, desenvolvendo nos estudantes habilidades de cidadania ao incentivar a participação ativa. Dessa forma, contribuem não apenas para resolver problemas individuais dos alunos, mas também para construir uma cultura voltada à melhoria contínua da educação e da participação de todos os atores envolvidos no contexto escolar (MIRANDA; CRUZ, 2022).

Com o crescimento das manifestações recebidas pelas ouvidorias, surgem desafios relevantes. O alto volume de demandas exige mecanismos eficientes de triagem e organização das informações, sob a pressão de prazos cada vez mais curtos. A necessidade de equilibrar agilidade na resposta com a qualidade da solução apresentada torna difícil manter padrões consistentes conforme o fluxo de mensagens aumenta. Para evitar que problemas recorrentes fiquem invisíveis, é essencial adotar ferramentas que identifiquem padrões, priorizem temas sensíveis e proporcionem uma visão consolidada dos dados, permitindo ações rápidas sem comprometer a precisão e a responsabilidade na gestão das manifestações.

Nesse cenário, soluções de inteligência artificial são uma alternativa capaz de potencializar a atuação das ouvidorias. Técnicas de processamento de linguagem

natural permitem analisar textos de forma automatizada, identificar temas dominantes, agrupar manifestações similares e sugerir encaminhamentos adequados para cada setor responsável. Ferramentas baseadas em modelos de linguagem e busca vetorial possibilitam recuperar rapidamente casos semelhantes, gerar resumos coerentes e apoiar a classificação automática das manifestações, reduzindo o esforço humano em tarefas repetitivas.

1.1 OBJETIVO GERAL

Diante desse cenário, este trabalho tem como objetivo geral desenvolver uma plataforma de ouvidoria integrada à inteligência artificial aplicada ao contexto escolar, capaz de automatizar a sumarização, agrupamento e o encaminhamento das manifestações, oferecendo uma análise facilitada e uma visão organizada que facilite a tomada de decisões pelos operadores da ouvidoria.

1.2 OBJETIVOS ESPECÍFICOS

Para alcançar esse propósito, definem-se os seguintes objetivos específicos:

- Desenvolver um módulo de recuperação semântica capaz de identificar similaridades entre manifestações e agrupá-las, evidenciando padrões recorrentes que não seriam perceptíveis por meio de análises individuais.
- Implementar um módulo de encaminhamento automatizado que direcione cada manifestação ao setor responsável, considerando o conteúdo identificado e o escopo de atuação de cada área.
- Criar um módulo de sumarização capaz de sintetizar o conteúdo de cada grupo de manifestações, facilitando a compreensão das informações pelo operador.
- Projetar interfaces adequadas tanto para o público usuário quanto para o operador da ouvidoria, garantindo uma experiência de uso intuitiva e fácil.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a base teórica necessária para a compreensão do trabalho, abrangendo sistemas de ouvidoria, técnicas de processamento de linguagem natural, modelos de inteligência artificial, busca vetorial e engenharia de *prompt*. Esses fundamentos sustentam as escolhas metodológicas e técnicas que orientaram o desenvolvimento do software proposto.

2.1 OUVIDORIAS

As ouvidorias são canais formais de comunicação entre instituições e seus públicos de interesse, responsáveis pela recepção e tratamento de manifestações. No Brasil, a Lei Federal nº 13.460/2017 estabelece diretrizes para ouvidorias públicas, definindo prazos de atendimento e classificações de demandas. Embora voltadas ao setor público, suas práticas também são adotadas por instituições privadas e organizações do terceiro setor como referência de boa gestão do relacionamento com o público. A ouvidoria privada se inspira na pública ao valorizar as manifestações recebidas e utilizar essas informações para identificar pontos de melhoria e aperfeiçoar seus serviços.

As instituições privadas também se beneficiam de outros aspectos das ouvidorias públicas, como a utilização de categorias para organizar as manifestações. A Controladoria-Geral da União (CGU) desenvolveu metodologias de padronização, classificando as manifestações em seis tipos principais: denúncia (irregularidades), reclamação (insatisfação com serviços), solicitação (pedidos de providência), sugestão (propostas de melhoria), elogio (reconhecimento positivo de uma prática ou profissional) e informação (esclarecimentos sobre procedimentos). Além disso, as manifestações podem ser avaliadas quanto à gravidade, critério principal para priorizar o atendimento e alocar recursos (Controladoria-Geral da União (CGU), 2019)

Mesmo com uma série de padronizações e protocolos para facilitar a gestão, a administração de ouvidorias enfrenta desafios, especialmente em contextos com alto volume de demandas. A categorização das manifestações é um processo subjetivo e, muitas vezes, demorado. O roteamento para os setores apropriados exige conhecimento detalhado da instituição e a identificação de padrões recorrentes pode se perder na análise individualizada. Um estudo com 216.832 manifestações da Ouvidoria do SUS revelou que a classificação manual requer tempo e expertise consideráveis (SANTOS *et al.*, 2022). Esses desafios se tornam ainda mais complexos quando há a necessidade de equilibrar rapidez no atendimento com a análise cuidadosa, respeitando os prazos legais estabelecidos.

No contexto educacional, as ouvidorias escolares buscam atender de forma eficiente estudantes, pais, professores e funcionários. A Lei de Diretrizes e Bases da Educação (Lei nº 9.394/1996) garante a participação da comunidade escolar em decisões institucionais, e as ouvidorias desempenham um papel importante nesse diálogo (MIRANDA; CRUZ, 2022). As manifestações escolares envolvem questões pedagógicas, de infraestrutura, relações interpessoais e casos sensíveis que envolvem menores de idade, o que exige a adoção de protocolos específicos e respostas rápidas, mas fundamentadas.

Diante desse cenário, soluções baseadas em inteligência artificial surgem como alternativas para otimizar a triagem e a categorização das manifestações. Técnicas de processamento de linguagem natural possibilitam a análise automatizada de textos, o agrupamento de manifestações semelhantes e a classificação por temas ou grau de gravidade (MCTI, 2024).

2.2 PROCESSAMENTO DE LINGUAGEM NATURAL

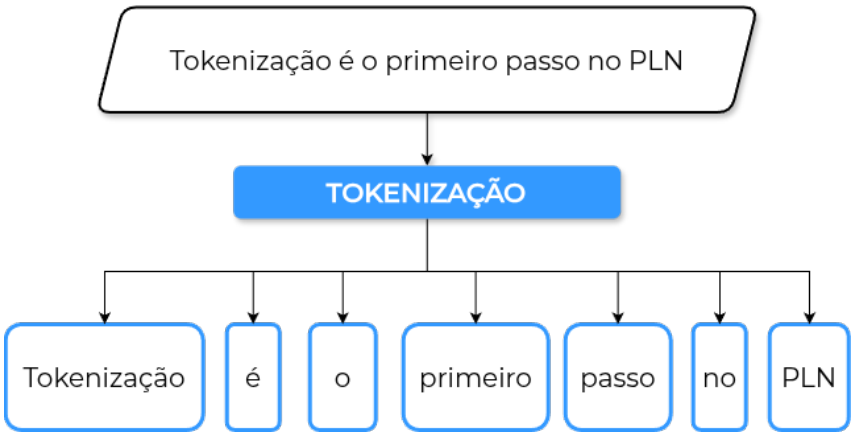
O Processamento de Linguagem Natural (PLN) é uma subárea da ciência da computação e inteligência artificial dedicada ao estudo da interação entre computadores e linguagem humana natural. Segundo (DANDE; PUND, 2023), o objetivo principal do PLN é capacitar sistemas computacionais a compreender, interpretar e gerar linguagem natural, possibilitando a execução de tarefas que tradicionalmente demandariam inteligência humana. O crescimento da área tem sido significativo nos últimos anos, provocado tanto pela crescente disponibilidade de conteúdo digital quanto pela necessidade de extração inteligente de informações dessas fontes.

Esse crescimento reflete na diversidade de aplicações de PLN, que incluem análise de sentimento, classificação de documentos, reconhecimento de entidades nomeadas (NER), detecção de spam, tradução, sistemas de pergunta-resposta e sumarização de textos (HOSSAIN *et al.*, 2024; ROUMELIOTIS; TSELIKAS; NASIOPOULOS, 2024; FALCãO; LOPES; SOUZA, 2022). Contudo, o processamento computacional de linguagem enfrenta desafios relacionados à ambiguidade linguística. Uma palavra pode apresentar diferentes significados dependendo do contexto em que é utilizada (YADAV; JANU, 2024), o que demanda o uso de técnicas capazes de representar e interpretar o texto de forma adequada.

Nesse sentido, destacam-se os métodos de pré-processamento e de análise textual empregados na atividade de processamento. A tokenização, prática de dividir o texto bruto em unidades menores chamadas de *tokens*, que geralmente correspondem a palavras individuais, sub-palavras ou até caracteres (a depender da abordagem utilizada), é frequentemente o primeiro passo em um pipeline de PLN. Esse procedimento é importante por segmentar as unidades semânticas relevantes e converter o texto em representações que possibilitam o processamento com maior eficiência por modelos

computacionais (ZOUHAR *et al.*, 2023). A Figura 1 demonstra um exemplo de tokenização de palavras, segmentando a frase em *tokens*.

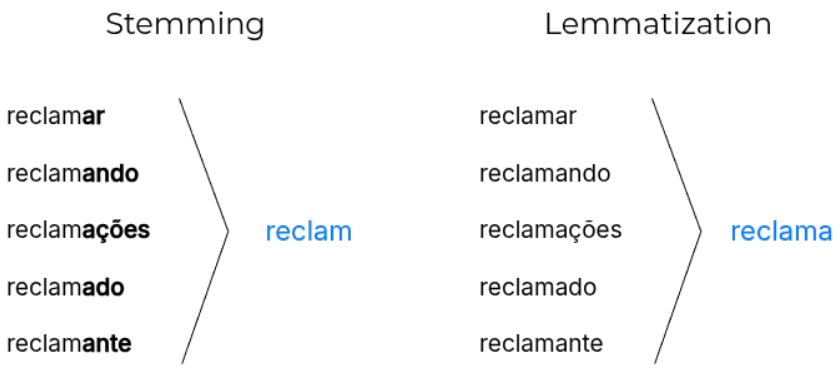
Figura 1 – Processo de Tokenização



Fonte: Elaborado pelo autor (2025)

No pré-processamento, inclui-se também a normalização textual, que converte letras maiúsculas em minúsculas e remove sinais de pontuação. Outro passo comum é a remoção de *stopwords*, que consiste em filtrar palavras que, apesar de muito frequentes no texto, agregam pouco significado ao texto, como artigos e preposições (ASGARI *et al.*, 2022). Além disso, são aplicadas técnicas de redução de palavras, como o *stemming*, que elimina sufixos e reduz os termos a sua forma radical, e o *lemmatization*, que gera o lema, isto é, a forma base e semanticamente válida da palavra. Diferentemente do lema, o radical produzido pelo *stemming* pode corresponder a uma palavra sem significado, o que pode impactar sua interpretação (TABASSUM; PATIL, 2020). A Figura 2 evidencia a diferença entre os dois processos de redução.

Figura 2 – Diferença entre Stemming e Lemmatization



Fonte: Elaborado pelo autor (2025)

Já as técnicas utilizadas para análise textual, evidencia-se a representação vetorial de texto. Nesse formato, conhecido como *embedding*, palavras ou documentos

são transformados em vetores numéricos de alta dimensionalidade. Estes vetores capturam propriedades semânticas de forma que palavras ou documentos com significados similares são representados por vetores próximos no espaço vetorial. As primeiras abordagens, como Word2Vec (MIKOLOV *et al.*, 2013) e GloVe (PENNINGTON; SOCHER; MANNING, 2014), geravam vetores estáticos, que atribuíam a cada palavra uma única representação, independentemente do contexto. Essa limitação provocou o desenvolvimento de modelos contextuais, nos quais o vetor gerado varia conforme o uso.

Nesse cenário, a arquitetura Transformer, proposta por Vaswani et al. (VASWANI *et al.*, 2017) no trabalho "Attention is All You Need", revolucionou o Processamento de Linguagem Natural. Ao introduzir o mecanismo de *self-attention*, o modelo permite que o sistema processe todas as palavras de uma frase simultaneamente e capture as relações entre elas, mesmo quando estão distantes entre si. Essa capacidade de modelar dependências de longo alcance tornou-se a base dos principais modelos atuais de PLN (WU; WANG; QUACH, 2025), como o BERT (Bidirectional Encoder Representations from Transformers), voltado para tarefas de compreensão textual, e o GPT (Generative Pre-trained Transformer), focado na geração de texto.

Dessa forma, os modelos baseados em Transformers aprimoraram o processamento do texto e ampliaram a capacidade dos sistemas de compreender relações e nuances presentes na linguagem. Ao lidar melhor com contexto, esses modelos permitem interpretações mais precisas das mensagens. Essa evolução é especialmente relevante em domínios que envolvem dados textuais complexos, como manifestações em ouvidorias, onde a compreensão do contexto e da intenção comunicativa é de extrema importância.

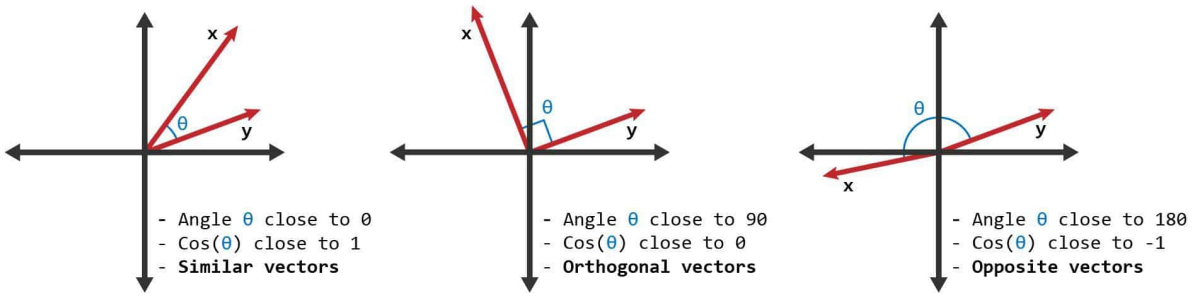
2.3 MODELO DE EMBEDDING

Modelos de embedding são algoritmos de aprendizado de máquina que transformam texto em representações númericas vetoriais. Essa conversão supera a limitação dos computadores de lidar diretamente com linguagem natural em sua forma textual. A ideia principal desses modelos é gerar embeddings que preservem relações semânticas no espaço vetorial (MUENNIGHOFF *et al.*, 2023), de modo que textos com significados similares possuam embeddings próximos nesse espaço, mesmo quando apresentam baixa sobreposição lexical (GAO; YAO; CHEN, 2021).

O funcionamento desses modelos é dividido em treinamento e uso prático, a inferência. No treinamento, o modelo é exposto a grandes volumes de texto, ajustando seus parâmetros para aprender a reconhecer padrões linguísticos e o significado das palavras em determinado contexto. Durante a inferência, ao receber um texto de entrada, o modelo gera um vetor de tamanho fixo que captura o significado semântico do conteúdo.

Para verificar se um texto é semelhante a outro, emprega-se comumente a similaridade de cosseno (GAHMAN; ELANGOVAN, 2023), que quantifica o ângulo entre os vetores gerados. Na Figura 3, nota-se que quanto menor o ângulo entre os vetores, maior a similaridade. Assim, a métrica produz valores entre -1 e 1, onde valores próximos a 1 indicam forte correspondência entre os textos comparados.

Figura 3 – Similaridade de Cosseno



Fonte: Elaborado pelo autor (2025)

2.4 MODELOS DE LINGUAGEM DE GRANDE ESCALA

Modelos de linguagem de larga escala (LLMs, do inglês *Large Language Models*) são modelos de inteligência artificial capazes de entender e gerar linguagem natural ou texto semelhante ao ser humano (MINAEE *et al.*, 2025). Eles são treinados com enormes conjuntos de dados, incluindo textos da internet, livros, artigos e códigos, para aprender padrões linguísticos, relações entre conceitos e contextos variados. Alguns exemplos conhecidos são o ChatGPT, da OpenAI, o Bard, do Google, e os modelos LLaMA, da Meta.

O funcionamento dos LLMs baseia-se em aprendizado profundo na arquitetura Transformer, que processa informações em camadas e aprende como palavras e frases se relacionam entre si (VASWANI *et al.*, 2017). Um mecanismo chamado autoatenção permite ao modelo compreender o contexto, identificando relações entre elementos de uma sequência de texto e melhorando a coerência das respostas. Essa arquitetura possibilita que LLMs realizem tarefas como tradução, sumarização, geração de texto, compreensão contextual e outras funções específicas quando ajustados por meio de técnicas de fine-tuning ou prompt engineering (FAN *et al.*, 2024; OUYANG *et al.*, 2022).

Atualmente, os LLMs são aplicados em diversas áreas para otimizar processos que antes dependiam exclusivamente de intervenção humana. Eles podem atuar como assistentes virtuais, chatbots, análise de sentimentos, auxílio à programação e até em pesquisas científicas (MINAEE *et al.*, 2025).

2.5 ENGENHARIA DE PROMPT

O avanço das LLMs provocou o surgimento da área chamada de engenharia de prompt. Ela envolve o uso de técnicas de elaboração e otimização das instruções, chamadas de *prompts*, fornecidas ao modelo, com o objetivo de esclarecer a intenção do usuário e obter respostas com maior qualidade em tarefas específicas (MARVIN *et al.*, 2024; WANG *et al.*, 2025). A área ganhou relevância porque, apesar da grande capacidade das LLMs em gerar retornos coerentes, os modelos apresentam forte sensibilidade à forma como as instruções são formuladas, com pequenas alterações na estrutura do prompt podendo gerar grandes variações no desempenho (SCLAR *et al.*, 2024).

Entre as técnicas de engenharia de prompt destacam-se a role prompting, que consiste na definição explícita do papel ou da especialidade que o modelo deve assumir, indicando no *prompt* a área de conhecimento em que deve atuar. Outra técnica é a especificação do formato da saída esperada, que orienta a estrutura das respostas ao indicar, por exemplo, o tamanho, o tipo de texto ou a estrutura dos dados, como JSON. Além disso, o uso de delimitadores e a organização visual do *prompt*, separando instruções do conteúdo a ser processado, ajuda a reduzir ambiguidades e melhora a interpretação das tarefas pelo modelo (CHEN *et al.*, 2025).

Essas técnicas são ainda mais eficazes em conjunto com instruções claras, que definem critérios e contextos relevantes para a realização da tarefa solicitada. Essas características presentes em um *prompt* são comuns no método *zero-shot prompting*, que possibilita o modelo realizar trabalhos complexos mesmo sem ter recebido um exemplo ou demonstração da resposta esperada, apoiando-se exclusivamente na clareza da instrução dada. Tais práticas mostram-se relevantes em aplicações que demandam o processamento sistemático e padronizado de conteúdo textual em contextos organizacionais específicos, como explorado neste trabalho.

2.6 BUSCA VETORIAL

Sistemas de busca tradicionais, baseados em palavras-chave, funcionam através da correspondência exata ou parcial entre termos, mostrando-se sensíveis a variações lexicais, sinônimos e estrutura textual (MATTIUZZI, 2025). Em contraste, a busca vetorial representa os documentos como vetores que capturam seu significado semântico. Assim, mesmo sem correspondência exata de palavras, é possível identificar documentos similares analisando a proximidade desses vetores no espaço multidimensional.

A eficiência da busca vetorial depende da organização dos embeddings em estruturas chamadas índices. São estruturas de dados que permitem localizar rapidamente os documentos mais semelhantes a uma consulta sem precisar comparar todos

os registros da base. Diversas bibliotecas fornecem suporte a essa tarefa, incluindo FAISS, Annoy e ChromaDB, que utilizam diferentes estratégias de indexação, como árvores de busca, grafos e índices invertidos, para acelerar a recuperação de vetores. Essas bibliotecas comumente usam algoritmos de busca aproximada de vizinhos mais próximos (ANN, do inglês Approximate Nearest Neighbors).

Para medir a similaridade entre os vetores, existem diferentes métricas, como distância euclidiana, produto escalar e similaridade de cosseno. Neste trabalho, optou-se pela similaridade de cosseno em conjunto com o FAISS, pois ela avalia a proximidade entre os vetores independente do comprimento do texto (KRANTZ; JONKER, 2024). Essa abordagem permite identificar e agrupar manifestações que tratam de temas semelhantes, mesmo que sejam expressas de formas diferentes, facilitando a análise e classificação das manifestações que chegam à ouvidoria.

Nesse cenário, a busca vetorial tem se mostrado útil em diversos outros contextos, como sistemas de recomendação, motores de busca semânticos, recuperação de artigos científicos, classificação de documentos jurídicos e análise de imagens (LIMA, 2025; MONIR *et al.*, 2024; SHUO; AFFENDEY; SIDI, 2024). Essa versatilidade se deve à capacidade de identificar itens próximos rapidamente em grandes volumes de dados.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta aplicações e pesquisas que possuem propósitos e desafios semelhantes aos do sistema proposto neste trabalho, a inserção de inteligência artificial nas ouvidorias com o fim de modernizar e facilitar a operação de análise das manifestações.

A participação da inteligência artificial beneficia uma série de setores e exemplos de êxito não faltam. Na área jurídica, acelerou a análise de processos e documentos. No transporte, reduziu tempos de manutenção através de análises preditivas e aprimorou a segurança dos sistemas. Na agricultura, otimizou o plantio e a irrigação, além de identificar pragas e doenças antecipadamente. Na saúde, contribuiu para a detecção precoce de doenças e a identificação de focos de contaminação (SPILLEIR; SILVA; SOUZA, 2021).

Na área de ouvidorias, ainda são poucos os casos de utilização de IA integrados ao processo ou aos softwares envolvidos na análise de manifestações. No entanto, mesmo em menor ocorrência, são significativos os resultados obtidos em tempo de resposta e eficiência das operações que a adotaram.

3.1 ROBÔ IZA

A Controladoria-Geral do Distrito Federal (CGDF) criou a robô IZA para ajudar no acolhimento e classificação de manifestações. Desde dezembro de 2022, a inteligência artificial analisa o texto escrito pelo cidadão e identifica automaticamente se trata-se de reclamação, sugestão, solicitação, elogio ou denúncia, além de sugerir o assunto mais adequado entre mais de 1.200 opções disponíveis.

Esse método de classificação automática gerou impactos nos dois lados do processo. Para a ouvidoria, houve redução de 70% na reclassificação manual de demandas, tirando a necessidade de alocar três servidores, que dedicavam cerca de cinco horas diárias, apenas nessa atividade. Já para os usuários, o tempo necessário para abrir uma chamada na ouvidoria diminuiu de 60 minutos para 20 minutos.

3.2 ROBÔ KAIRÓS

No Rio Grande do Norte, a ouvidoria do Tribunal de Contas do Estado (TCE-RN) passou a utilizar o robô Kairós, uma tecnologia desenvolvida pela Universidade Federal do Rio Grande do Norte (UFRN). Kairós utiliza inteligência artificial para monitorar todo o ciclo de vida das manifestações, desde a abertura até o cumprimento das respostas, controlando prazos e alertando automaticamente sobre pendências. Com

esse monitoramento o sistema reduziu o tempo médio de atendimento de cinco dias para 2,76 dias.

3.3 SISTEMA FARO

No âmbito federal, (PAIVA, 2023) desenvolveu o sistema FARO (Ferramenta de Análise de Risco em Ouvidoria) com foco na classificação de aptidão de denúncias, ou seja, verifica se as manifestações tem os elementos mínimos para ser investigadas. Essa solução une o processamento de linguagem natural com extração e enriquecimento de dados estruturados. Sua metodologia inclui a identificação de entidades nos textos das denúncias (como CPFs, CNPJs e números de contratos) e a busca de informações correlacionadas em bases de dados externas, gerando variáveis que são utilizadas em conjunto com o texto original para auxiliar na classificação. O modelo foi implementado na Ouvidoria-Geral da União (OGU/CGU), onde classifica cerca de 50% das denúncias recebidas pela ouvidoria federal desde 2020.

3.4 ANÁLISE COMPARATIVA

Todas essas soluções, incluindo a deste trabalho, adotam um modelo híbrido, no qual a IA automatiza tarefas operacionais enquanto humanos mantêm a supervisão e a tomada de decisões. O estudo feito por (RUSANOV, 2025) alerta que sistemas 100% automatizados podem apresentar viés algorítmico, ausentar de sensibilidade emocional e comprometer a confiança dos usuários, especialmente em casos sensíveis. A ideia é que a IA atue na classificação, triagem e monitoramento, mas a decisão final deve sempre permanecer com um profissional humano.

As soluções apresentadas demonstram aplicações de IA em diferentes aspectos de ouvidorias. A Tabela 1 compara as funcionalidades dos trabalhos relacionados com o Ouvidor.IA.

Tabela 1 – Comparação de funcionalidades entre trabalhos relacionados e o Ouvidor.IA

Funcionalidade	**IZA**	**Kairós**	**FARO**	**Ouvidor.IA**
Classificação de tipo	X			
Monitoramento de prazos		X		
Análise de aptidão			X	
Agrupamento semântico				X
Roteamento automático				X
Contexto escolar				X

Fonte: Elaborado pelo autor (2025)

Como observado na Tabela 1, este trabalho combina três funcionalidades não encontradas simultaneamente nas soluções existentes:

- **Aplicação ao contexto escolar**: Desenvolvido especificamente para ouvidorias escolares, considerando as particularidades desse ambiente, ao contrário das soluções voltadas para administração pública geral.
- **Agrupamento semântico**: Utiliza recuperação semântica para agrupar manifestações relacionadas ao mesmo evento ou situação, permitindo a detecção de padrões recorrentes que passariam despercebidos em análises individuais.
- **Roteamento automático**: Propõe encaminhamento para setores responsáveis baseado em análise de conteúdo da manifestação e atribuições institucionais de cada área.

Enquanto as soluções apresentadas focam isoladamente em classificação de tipo, monitoramento de prazos ou análise de aptidão, o Ouvidor.IA explora dimensões ainda não contempladas em conjunto pelas ferramentas existentes.

4 OUVIDOR.IA

4.1 TECNOLOGIAS DE IMPLEMENTAÇÃO

O Ouvidor.IA é um sistema web de ouvidoria que integra mecanismos de inteligência artificial. O back-end foi desenvolvido com Django Ninja para construção da API em Python, enquanto o front-end utiliza Next.js para a criação da interface do usuário. Para as funcionalidades de IA, foram utilizados o modelo de linguagem Qwen3-30B-A3B para processamento e geração de texto, o modelo BGE-M3 para gerar embeddings, a biblioteca FAISS para busca de similaridade, e o LangChain para facilitar a integração dos componentes de IA. O sistema utiliza PostgreSQL como banco de dados relacional, MinIO para gerenciamento e armazenamento de imagens e documentos, e Docker para orquestração de todos os serviços em containers.

4.1.1 Django Ninja

Django Ninja é um framework web moderno para construção de APIs em Python, construído sobre o Django e inspirado fortemente no FastAPI. Essa tecnologia combina a robustez do Django com validação automática de tipos usando Pydantic, oferecendo performance superior ao Django REST Framework. Gera documentação automática (OpenAPI/Swagger) e possui curva de aprendizado suave para desenvolvedores Django. Tem se tornado uma escolha popular para criar APIs rápidas mantendo a familiaridade do ecossistema Django (Django Ninja, 2024).

4.1.2 Next.js

Next.js é um framework React desenvolvido pela Vercel, lançado em 2016, que permite criar aplicações web full-stack com renderização híbrida (SSR, SSG e CSR). O framework destaca-se por oferecer suporte nativo ao Server-Side Rendering (SSR), uma abordagem em que as páginas são renderizadas no servidor a cada requisição, fazendo com que o conteúdo chegue ao navegador já processado. Isso favorece significativamente o SEO (Search Engine Optimization), já que os motores de busca conseguem indexar as páginas com maior facilidade. O Next.js também facilita a atualização dinâmica de dados e oferece um sistema de roteamento baseado na estrutura de pastas do projeto, o que torna o desenvolvimento mais ágil (Next.js Documentation, 2024).

4.1.3 FAISS

FAISS (Facebook AI Similarity Search) é uma biblioteca desenvolvida pela Meta AI, lançada em 2017, que permite encontrar rapidamente itens similares em grandes conjuntos de dados. Essa tecnologia foi criada para trabalhar com bilhões de registros transformados em vetores numéricos, utilizando GPUs para acelerar as buscas nesse cenário. A biblioteca é especialmente útil em sistemas de recomendação, busca semântica (aplicação dada neste trabalho) e aplicações de RAG (Geração Aumentada por Recuperação, do inglês *Retrieval-Augmented Generation*) (Meta AI, 2024).

4.1.4 LangChain

LangChain é um framework open-source criado em 2022 por Harrison Chase que simplifica a construção de aplicações com modelos de linguagem de grande escala (LLMs). Essa tecnologia funciona como um kit de módulos que permite conectar LLMs com outras funcionalidades, como busca na web, bancos de dados, leitura de documentos e memória de conversas, isso por meio de "correntes"(chains) de operações (CHASE, 2022; LangChain Inc., 2024).

4.1.5 Qwen/Qwen3-30B-A3B

Qwen3-30B-A3B é um modelo de linguagem de última geração desenvolvido pela Alibaba Cloud, parte da série Qwen3. Trata-se de um modelo Mixture-of-Experts (MoE) com 30,5 bilhões de parâmetros totais, dos quais apenas 3,3 bilhões são ativados durante a inferência, proporcionando um ótimo equilíbrio entre capacidade e eficiência computacional. O modelo destaca-se por suas capacidades em seguir instruções multilíngues, geração de código, raciocínio lógico e integração com ferramentas externas para tarefas agênticas (TEAM, 2025; Qwen Team, 2025).

4.1.6 BAAI/bge-m3

BGE-M3 é um modelo de embeddings de texto desenvolvido pela BAAI (Beijing Academy of Artificial Intelligence) que se destaca pela sua versatilidade. O termo "M3"de seu nome é em razão de três características principais: Multi-lingual (suporta mais de 100 idiomas), Multi-functionality (funciona com diferentes tipos de busca) e Multi-granularity (processa até documentos com até 8192 tokens). O modelo é capaz de transformar textos em representações numéricas que computadores compreendem, sendo especialmente eficiente em tarefas de recuperação de documentos e sistemas de perguntas e respostas (CHEN *et al.*, 2024; BAAI, 2024).

4.1.7 PostgreSQL

PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional de código aberto, poderoso e amplamente utilizado. Ele se baseia na linguagem SQL e é reconhecido por sua arquitetura comprovada, confiabilidade, integridade de dados e conjunto robusto de funcionalidades. Compatível com os principais sistemas operacionais e aderente aos princípios ACID, o PostgreSQL é altamente extensível e oferece recursos avançados para desenvolvedores e administradores. Além disso, conta com uma comunidade ativa que o mantém constantemente atualizado, tornando-o uma das opções favoritas para gerenciamento de dados de qualquer tamanho (PostgreSQL Global Development Group, 2024).

4.1.8 MinIO

O MinIO é um sistema de armazenamento de objetos de código aberto projetado para lidar com grandes volumes de dados não estruturados, como imagens, vídeos e documentos, compatível com o protocolo S3 da Amazon (MinIO, Inc., 2024).

4.1.9 Docker

O Docker é uma plataforma de software que permite criar, testar e implantar aplicações rapidamente. O Docker "empacota"o sistema em unidades padronizadas chamadas contêineres, que contêm tudo o que o software precisa para ser executado, incluindo bibliotecas, ferramentas de sistema, código e ambiente de execução. A conteinerização garante que a aplicação funcione de forma consistente em diferentes ambientes (Amazon Web Services, 2025). No sistema desenvolvido, o Docker é utilizado para orquestrar todos os serviços da aplicação, o que facilita o gerenciamento e a portabilidade da solução.

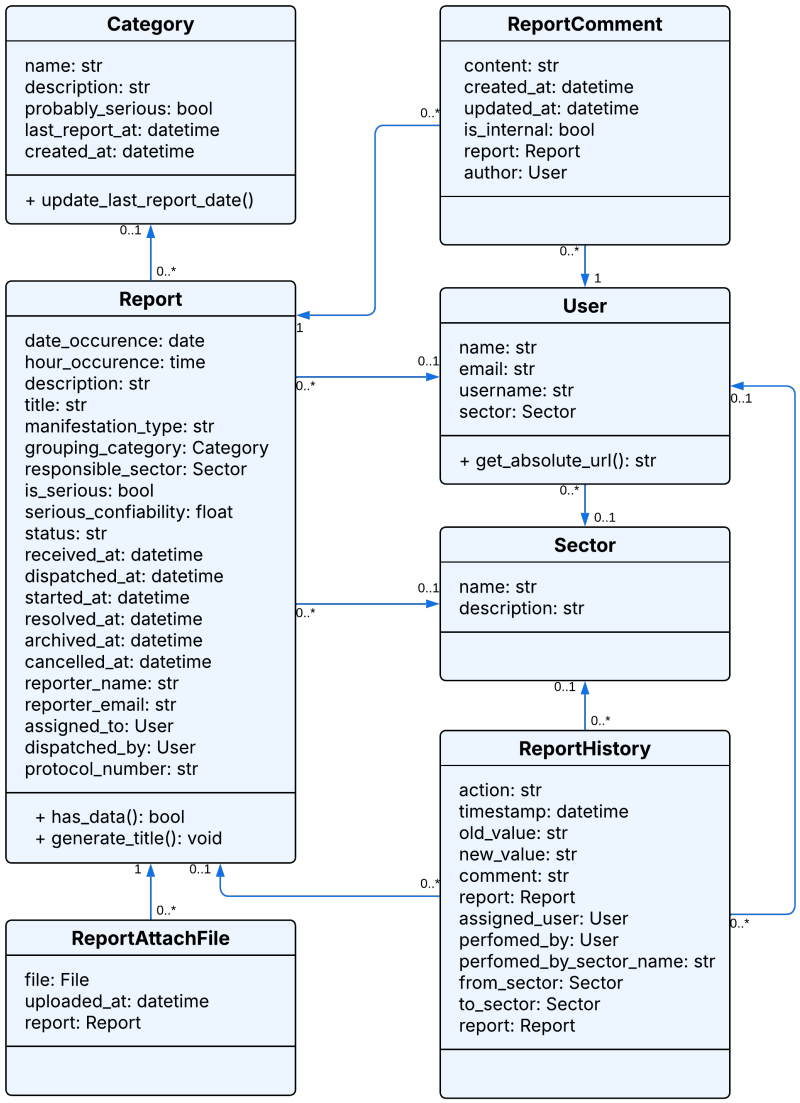
4.2 MODELAGEM DO SISTEMA

A modelagem de sistema consiste na representação visual e estruturada dos requisitos funcionais e da organização interna de um software. Por meio de diagramas padronizados pela UML (Unified Modeling Language), é possível documentar tanto as interações entre usuários e sistema (diagrama de casos de uso) quanto a estrutura de dados e relacionamentos entre as entidades (diagrama de classes). Esta etapa garante que todos os requisitos levantados sejam adequadamente representados, servindo como base para o desenvolvimento e como documentação técnica do projeto.

4.2.1 Diagrama de Classes

O diagrama de classes é uma representação visual da estrutura estática do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas.

Figura 4 – Diagrama de Classes



Fonte: Elaborado pelo autor (2025)

Na Figura 4, observa-se que a entidade Report está no centro da modelagem do Ouvidor.IA, sendo associada diretamente a diversas entidades que representam registros e ações relacionadas ao ciclo de vida de uma manifestação da ouvidoria.

As entidades ReportAttachFile, ReportComment e ReportHistory estão diretamente associadas a Report em relacionamentos do tipo um-para-muitos. Um relatório pode ter diversos arquivos anexados, múltiplos comentários e um histórico completo de todas as ações realizadas. ReportAttachFile armazena os arquivos enviados como

evidência junto com a data de upload. ReportComment permite que operadores registrem pareceres e observações sobre o caso, podendo ser marcados como internos ou externos. ReportHistory atua como log de auditoria, registrando cada ação executada, quem a realizou, quando ocorreu, valores anteriores e novos, e movimentações entre setores.

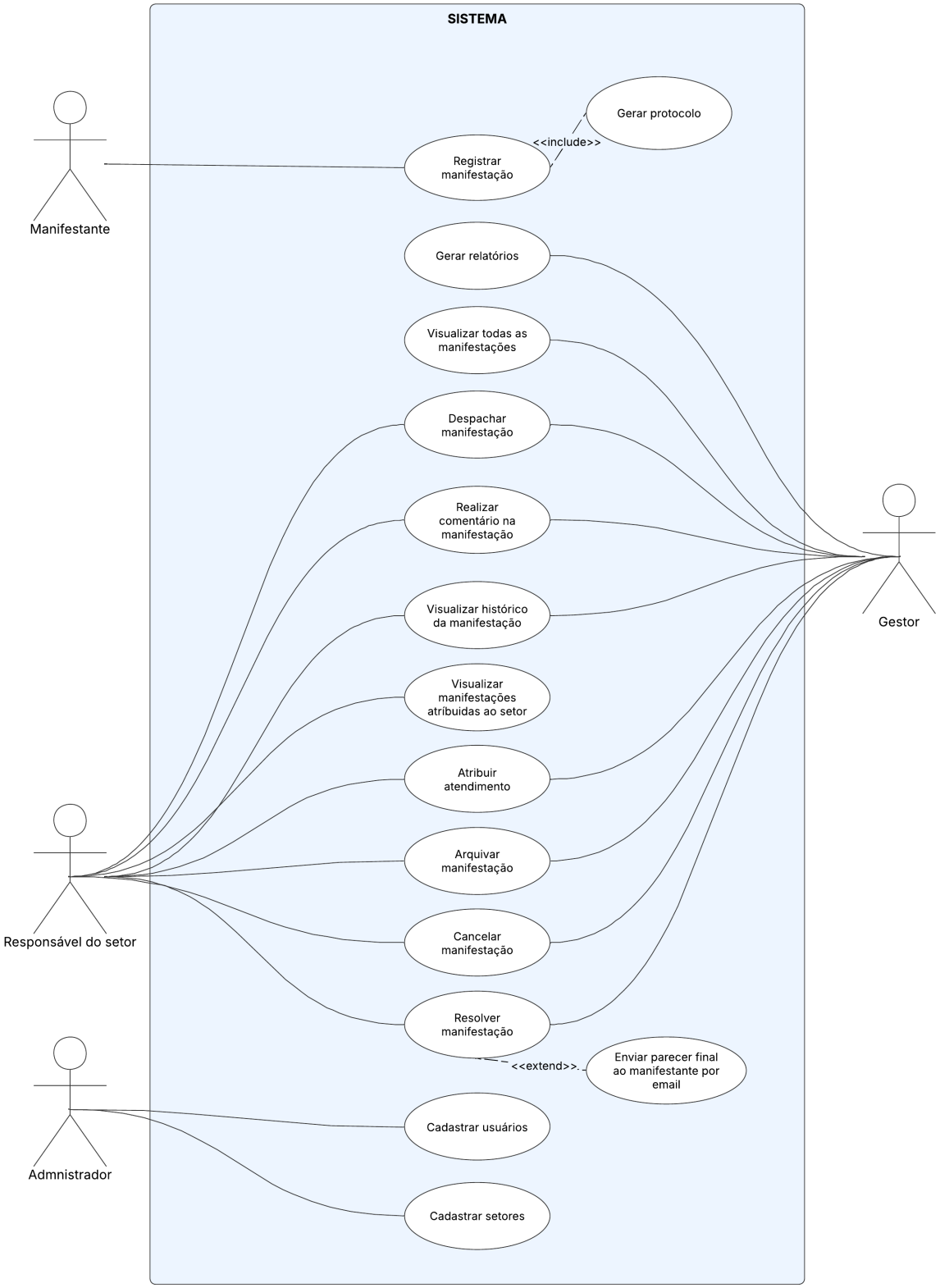
A modelagem também inclui as entidades User, Sector e Category. User representa os operadores do sistema, enquanto Sector define os departamentos responsáveis por processar as manifestações. O relacionamento entre User e Sector é de muitos-para-um, pois vários usuários podem pertencer a um mesmo setor. Category representa as categorias de agrupamento geradas pela IA para classificar manifestações similares. Cada categoria está vinculada a um setor responsável e pode ser marcada como "provavelmente grave", facilitando a priorização das manifestações.

4.2.2 Diagrama de Caso de Uso

O diagrama de casos de uso representa o sistema sob a perspectiva do usuário, destacando de forma clara as interações entre atores e funcionalidades. Ele é composto por quatro elementos principais:

- Sistema: delimitado por um retângulo que define a aplicação.
- Atores: representados por bonecos de palito, que simbolizam os usuários.
- Casos de uso: elipses que indicam as ações possíveis do sistema.
- Relacionamentos: linhas que conectam os atores aos casos de uso para mostrar as interações.

Figura 5 – Diagrama de Caso de Uso



Fonte: Elaborado pelo autor (2025)

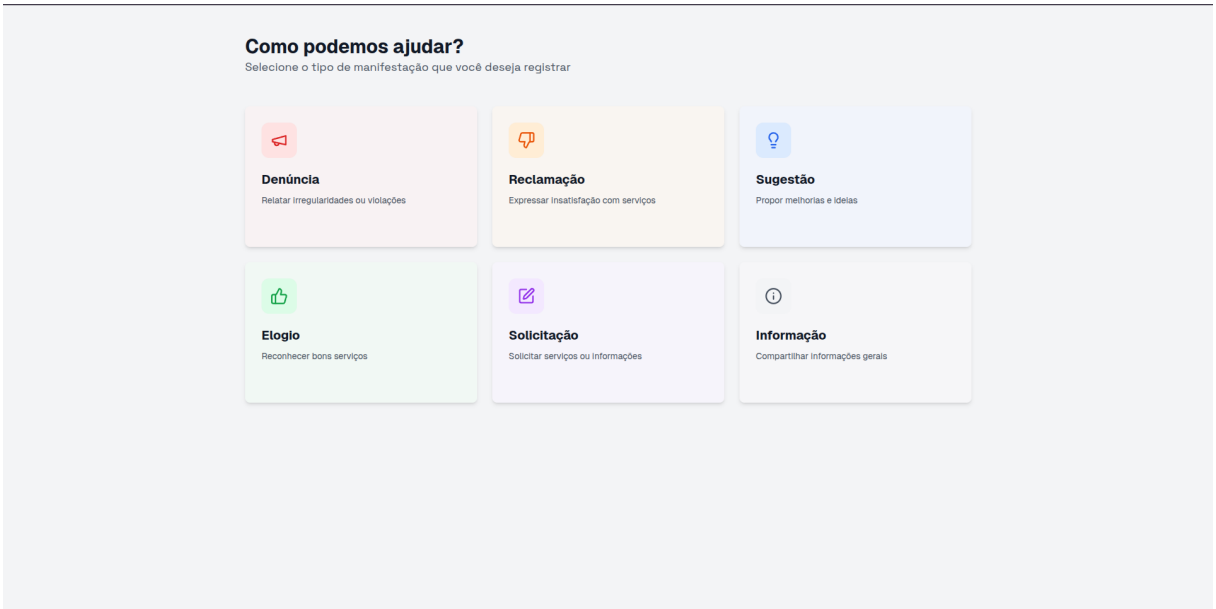
4.3 FLUXO OPERACIONAL DO OUVIDOR.IA

Esta seção apresenta o funcionamento do sistema do ponto de vista operacional, descrevendo como as manifestações são processadas desde o registro inicial até a resolução final, integrando recursos de inteligência artificial para classificação e roteamento automático com a gestão humana do atendimento.

4.3.1 Registro da Manifestação

O fluxo do Ouvidor.IA se inicia quando o usuário acessa a plataforma web para registrar uma manifestação. Para a realização dessa tarefa, não é necessário estar autenticado no sistema para utilização do formulário de cadastro, o que possibilita que o relato seja anônima, caso seja de interesse do manifestante.

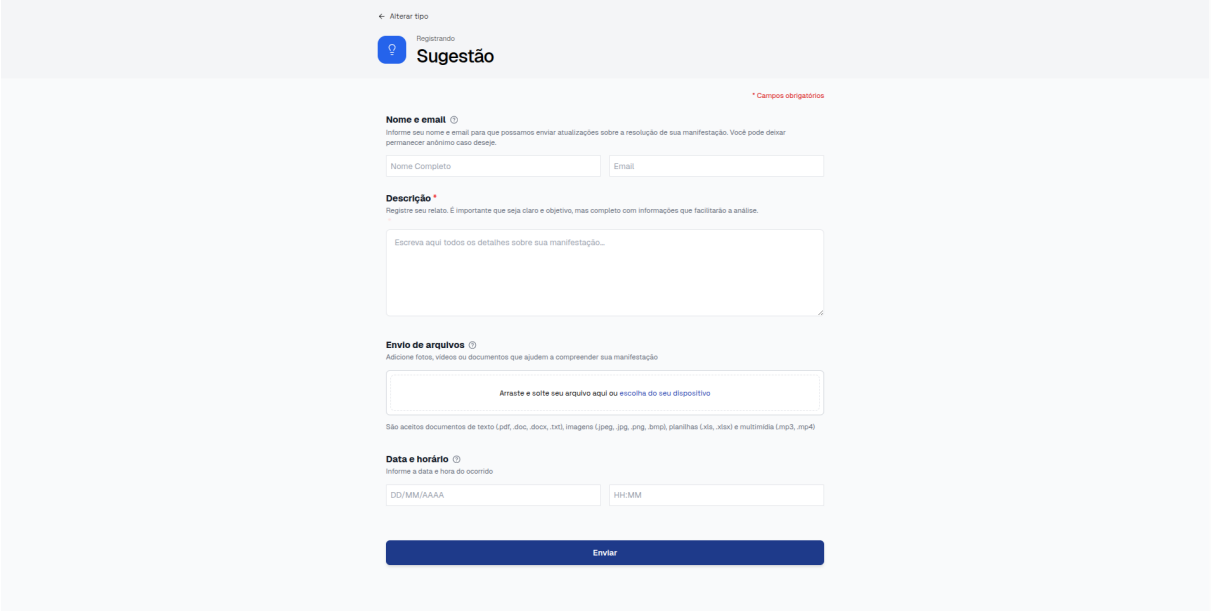
Figura 6 – Tela de escolha do tipo de manifestação



Fonte: Elaborado pelo autor (2025)

O primeiro passo é escolher o tipo de manifestação, que incluem seis possibilidades: denúncia, reclamação, sugestão, informação, solicitação ou elogio. Após essa seleção, o sistema exibe o formulário principal com campos de identificação (nome e email do manifestante) e campos de descrição do ocorrido. O nome e email são usados estritamente para facilitar a comunicação e enviar o parecer final quando o caso for resolvido. Os campos sobre a ocorrência incluem a descrição do fato, data e hora, e anexos de mídia como fotos ou documentos que possam ajudar a complementar o detalhamento do caso. Entre esses campos, apenas o campo descrição é obrigatório.

Figura 7 – Tela de formulário da manifestação



Fonte: Elaborado pelo autor (2025)

4.3.2 Triagem e Classificação da Manifestação

Após o cadastro da manifestação, o sistema aciona o processamento de forma assíncrona. Isso permite que o usuário receba confirmação instantânea do registro enquanto a análise da manifestação ocorre em segundo plano.

Com isso, o *pipeline* de processamento segue uma sequência de etapas:

- **Geração de Embedding:** O modelo BGE-M3 transforma o texto da descrição em um vetor numérico armazenado no índice FAISS.
- **Classificação de Gravidade:** O sistema aplica o `CLASSIFICATION_PROMPT` ao modelo Qwen3-30B-A3B. Este *prompt* fornece ao modelo uma lista de critérios que definem situações graves no ambiente escolar: violência física ou psicológica, assédio, corrupção, discriminação, riscos à saúde dos alunos, ou condutas que comprometam seriamente o ambiente escolar. O modelo retorna um JSON contendo a classificação binária (“GRAVE” ou “NÃO GRAVE”) e um percentual de confiança, auxiliando os operadores a identificarem casos que necessitam priorização.
- **Geração de Título:** O `TITLE_PROMPT` instrui o modelo a criar um título curto (máximo 50 caracteres) que captura o tema principal da manifestação.

4.3.3 Agrupamento por Categoria

O agrupamento por categoria busca identificar se a nova manifestação é similar a casos anteriores já categorizados. Este processo utiliza um algoritmo de decisão baseado em variáveis configuráveis que podem ser ajustadas conforme as necessidades da instituição:

- **Limiar de Escolha (`CHOICE_THRESHOLD`):** Define o nível mínimo de similaridade necessário para agrupar manifestações.
- **Peso Temporal (`TIME_WEIGHT`):** Determina a influência do fator temporal na decisão de agrupamento.
- **Peso Semântico (`SENSE_WEIGHT`):** Determina a influência da similaridade semântica na decisão.
- **Janela Temporal Máxima (`MAX_GROUPING_DAYS`):** Define por quantos dias uma categoria permanece ativa para receber novos agrupamentos. Isso evita agrupar casos novos com categorias antigas já resolvidas ou inativas.
- **Limiar de Confiança para Despacho (`AUTO_DISPATCH_CONFIDENCE_THRESHOLD`):** Define o nível mínimo de confiança necessário para que o sistema realize despacho automático.

O algoritmo considera duas variáveis principais no processo de decisão:

- **Similaridade Semântica:** Calculada através da similaridade de cosseno entre o vetor do novo relato e os vetores das categorias existentes no índice FAISS.
- **Proximidade Temporal:** Avalia se a categoria recebeu manifestações recentemente, dentro da janela temporal configurada.

O sistema calcula uma pontuação combinada ponderando a similaridade semântica e a proximidade temporal conforme os pesos configurados. Se essa pontuação superar o limiar de escolha e a categoria estiver dentro da janela temporal ativa, o novo relato é adicionado àquela categoria existente. Caso contrário, o sistema identifica que se trata de um tema novo e cria uma categoria provisória.

4.3.4 Roteamento Automático

O resultado do agrupamento determina como o relato será encaminhado:

- **Despacho para Categoria Existente:** Se o relato foi adicionado a uma categoria que já possui setor responsável definido, ele é automaticamente despachado para aquele setor.

- **Criação de Nova Categoria:** Se o relato inaugurou uma nova categoria, o sistema executa análises adicionais para caracterizá-la. O processo utiliza os seguintes *prompts*:
    - **Geração de Metadados:**
        - `TOPIC_PROMPT`: Cria um nome conciso (máximo 100 caracteres) que capture o tema central da categoria.
        - `DESCRIPTION_PROMPT`: Gera uma descrição sintética (máximo 150 caracteres) que contextualize o tipo de manifestações do agrupamento.
    - **Classificação de Setor:** O `SECTOR_CLASSIFICATION_PROMPT` fornece ao modelo um catálogo dos setores disponíveis com suas atribuições. Exemplos:
        - Infraestrutura e Manutenção: Instalações físicas, equipamentos e conservação do ambiente escolar.
        - Coordenação Pedagógica: Ensino-aprendizagem, metodologias e relações professor-aluno.
        - Recursos Humanos: Relações trabalhistas e comportamento de funcionários.

O modelo analisa o conteúdo da nova categoria e retorna um JSON estruturado contendo o identificador do setor mais apropriado, o índice de confiança na classificação, e uma breve justificativa do raciocínio aplicado.

O despacho automático ocorre apenas se o índice de confiança superar o limiar configurado. Quando a confiança é insuficiente, o relato permanece no status “Classificado” sem setor atribuído, sendo encaminhado para triagem manual.

Todas as etapas do processamento são registradas no histórico do relato, incluindo o horário, identificadores dos modelos utilizados, índices de confiança, setores envolvidos e justificativas, o que possibilita auditorias posteriores.

4.3.5 Gestão das Manifestações

Após essa fase de processamento com o auxílio de IA, a interface do usuário autenticado mostra os relatórios consolidados por categoria, ordenados por gravidade e data. O usuário visualiza apenas as manifestações do seu setor, com exceção do administrador do sistema e dos usuários do setor de maior hierarquia da instituição (Direção), que podem acessar e interagir com todas as manifestações independentemente do setor ao qual estão atribuídas.

O ciclo de atendimento possui as seguintes operações:

- **Atribuição:** O operador atribui o relato para si mesmo, mudando o status para "Em Atendimento".

- **Acompanhamento:** Durante a investigação, o operador adiciona comentários documentando as ações tomadas, criando um histórico narrativo do processo de resolução.
- **Resolução:** Ao concluir a investigação, o operador marca o relato como "Resolvido". Essa ação faz com que um email com o parecer final seja enviado para o manifestante, fechando o ciclo de comunicação.
- **Arquivamento:** Relatórios resolvidos podem ser arquivados, saindo da visualização das manifestações ativas no momento.
- **Redespacho:** Podem mover manualmente um relato para outro setor se identificarem que o roteamento automático foi inadequado.
- **Recategorização:** Podem alterar a categoria de agrupamento, movendo manifestações entre as categorias quando o operador julgar necessário.
- **Cancelamento:** Quando um relato é identificado como spam, duplicado ou inadequado e não está apto a ser analisado, o operador pode cancelá-lo. Neste caso, ocorre a remoção do vetor correspondente do índice FAISS. Essa limpeza evita que manifestações inválidas "contaminem"futuras análises de similaridade e decisões de agrupamento.

4.3.6 Geração de Relatórios

O sistema oferece geração de relatórios consolidados em formato PDF, funcionalidade restrita aos usuários com setores privilegiados (Direção) e por usuário administrador. Esses documentos apresentam as seguintes métricas operacionais:

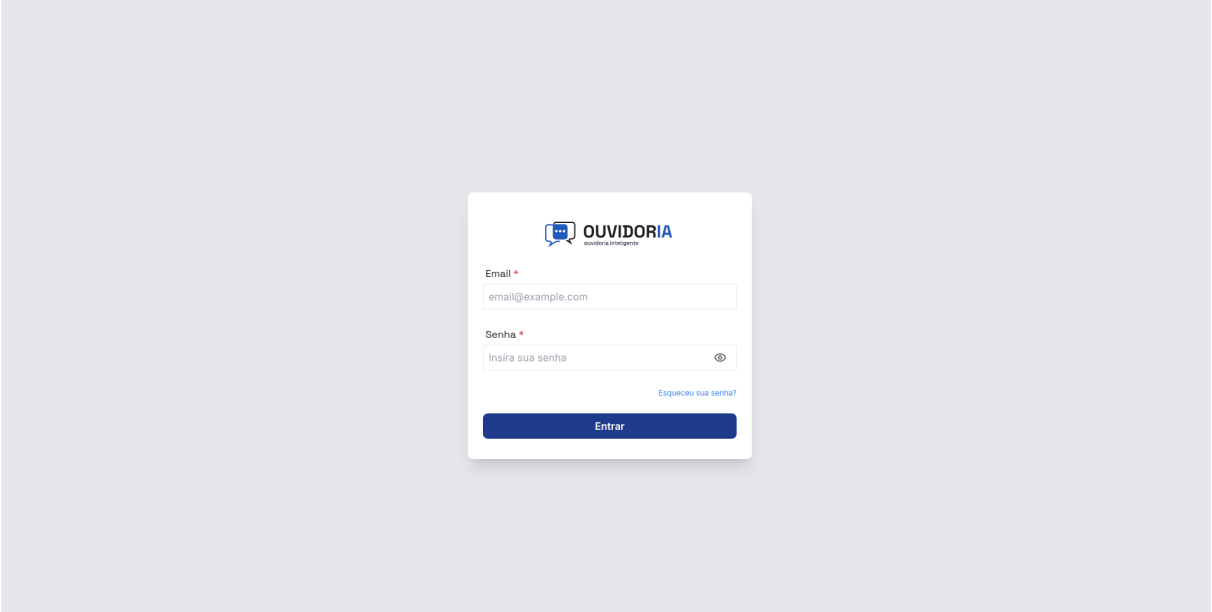
- Indicadores gerais: total de manifestações, média diária, taxa de resolução e projeção mensal.
- Distribuição por tipo de manifestação e por status do atendimento, com quantidades e percentuais.
- Consolidado por setor contendo volume total de manifestações, distribuição detalhada por status e taxa de resolução individualizada.
- Detalhamento por setor com listagem completa das manifestações incluindo tipo, status, data e categoria.

Essas análises permitem aos gestores identificar padrões recorrentes, avaliar o desempenho dos setores e tomar decisões estratégicas baseadas em dados concretos sobre o funcionamento da ouvidoria.

4.4 APRESENTAÇÃO DO SISTEMA

O Ouvidor.IA foi desenvolvido com foco na usabilidade e na eficiência do fluxo de trabalho. Essa seção apresenta as principais interfaces que compõem o sistema.

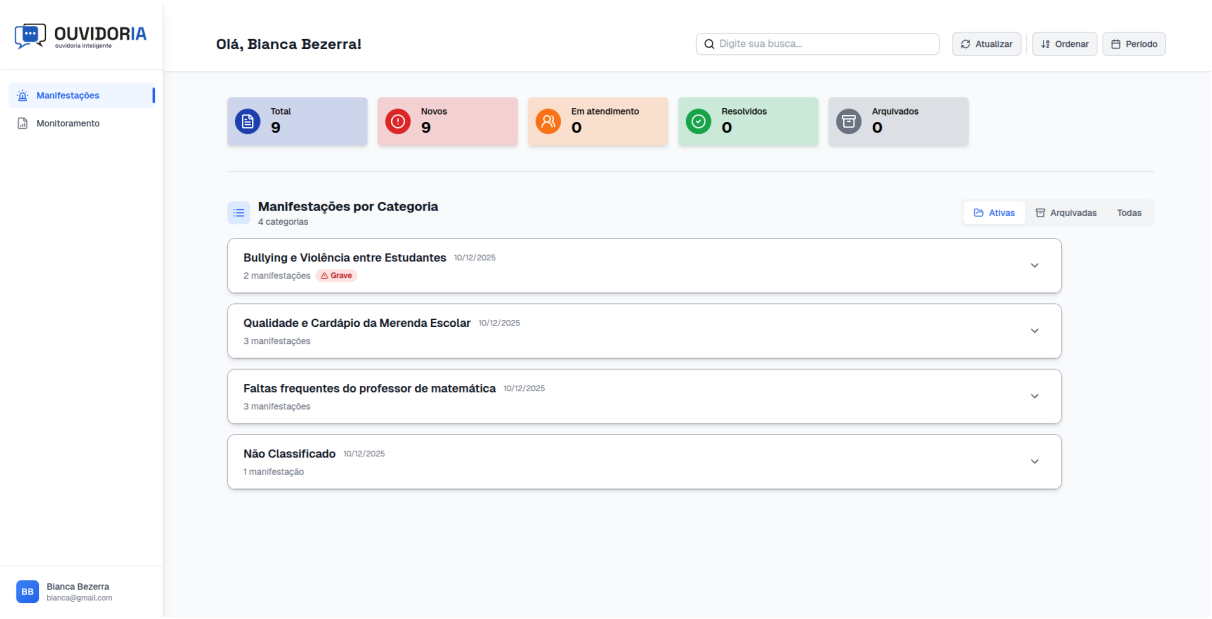
Figura 8 – Tela de Login



Fonte: Elaborado pelo autor (2025)

A Figura 8 apresenta a tela de login, ponto de acesso inicial para operadores do sistema. A interface solicita email e senha para autenticação, garantindo acesso restrito às funcionalidades de gestão da ouvidoria.

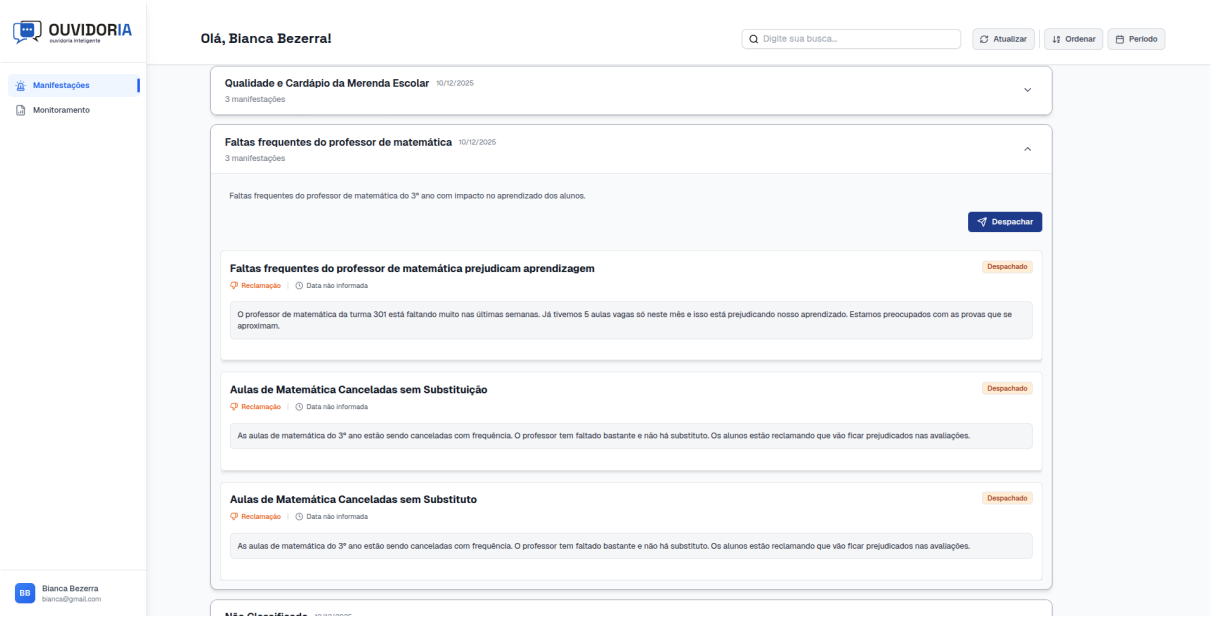
Figura 9 – Tela da Página Principal



Fonte: Elaborado pelo autor (2025)

A Figura 9 mostra a página principal do sistema, onde o operador tem visão geral do fluxo de trabalho. Na parte superior, *cards* apresentam indicadores quantitativos, total de manifestações, casos novos, em atendimento, resolvidos e arquivados. Abaixo, a seção “Manifestações por Categoria” lista os agrupamentos categorizados por inteligência artificial. A interface permite expandir cada categoria para visualizar as manifestações individuais.

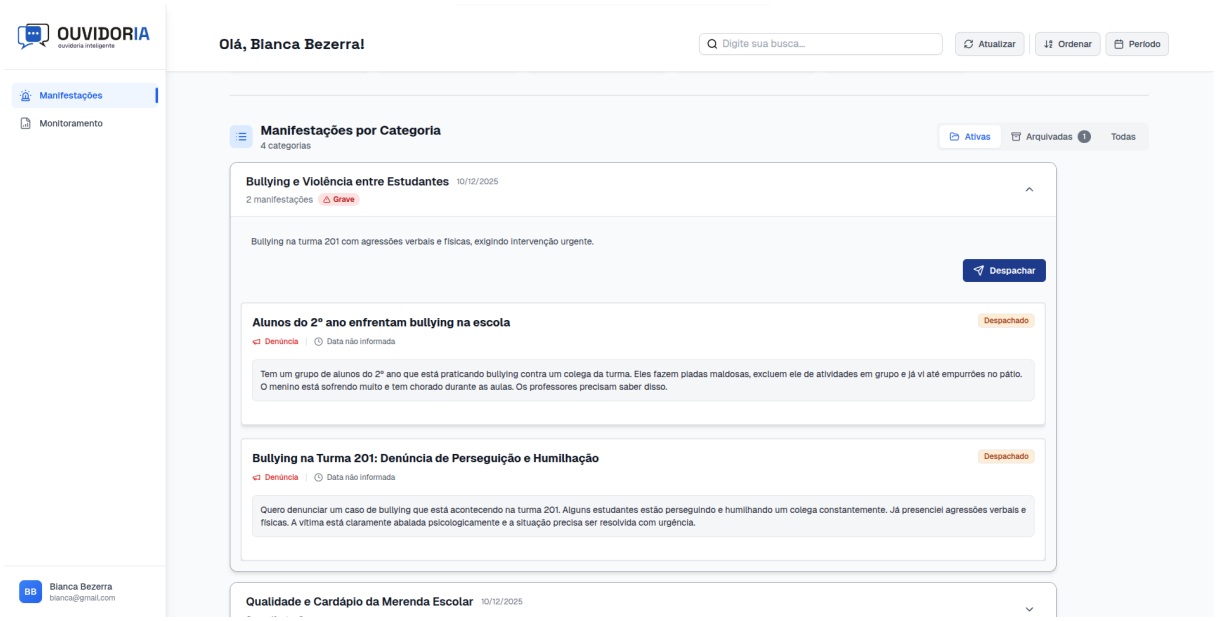
Figura 10 – Visualização do agrupamento de manifestações



Fonte: Elaborado pelo autor (2025)

A Figura 10 detalha a visualização expandida de um agrupamento específico. Cada manifestação individual é apresentada como um *card* contendo o título gerado, tipo de manifestação, *status* atual e data de recebimento. O botão “Despachar” no topo permite enviar múltiplas manifestações simultaneamente para um setor responsável.

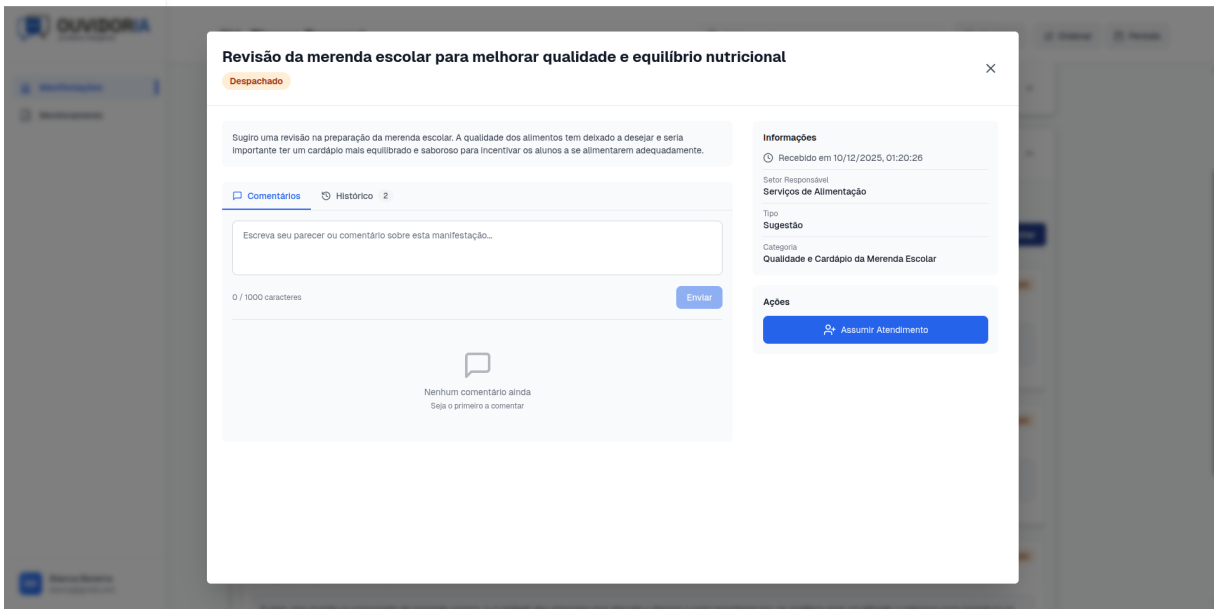
Figura 11 – Priorização de manifestações



Fonte: Elaborado pelo autor (2025)

A Figura 11 ilustra a funcionalidade de priorização automática. Categorias identificadas como graves recebem um badge vermelho de alerta para facilitar a identificação visual pelos operadores. Neste exemplo, a categoria “Bullying e Violência entre Estudantes” está marcada como grave e posicionada no topo da listagem, para garantir que casos críticos recebam prioridade.

Figura 12 – Detalhamento da manifestação

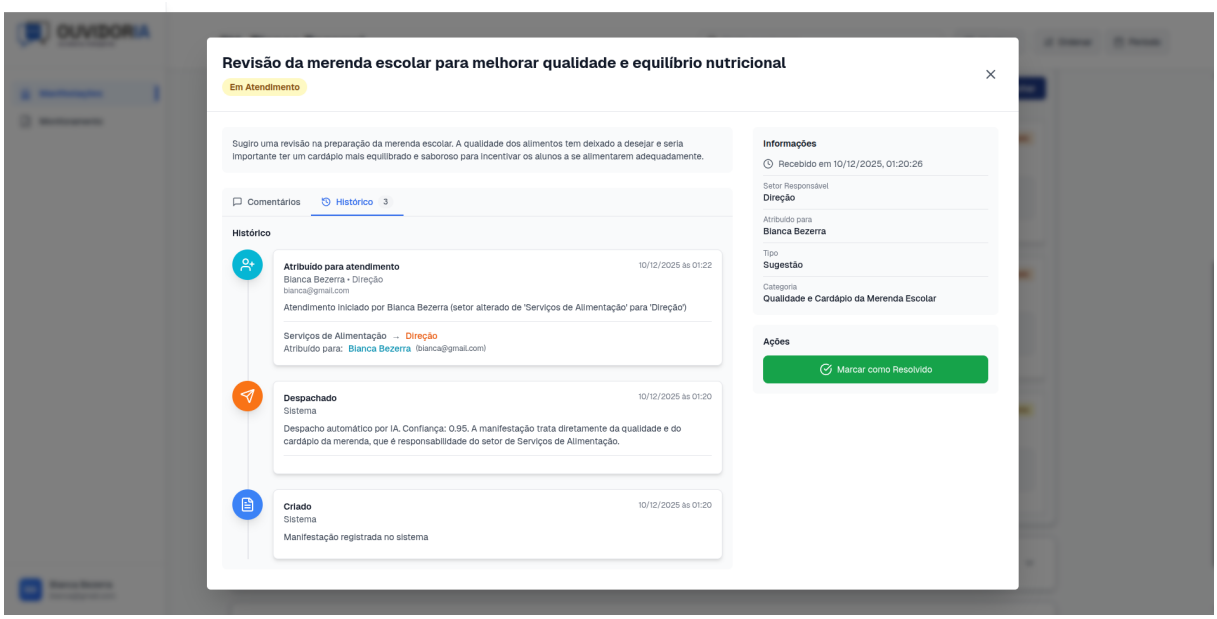


Fonte: Elaborado pelo autor (2025)

A Figura 12 apresenta a visualização completa de uma manifestação individual.

A interface exibe o título gerado pela IA, status atual destacado visualmente (neste exemplo, “Em Atendimento” em amarelo), descrição completa submetida pelo manifestante, e painel lateral com as informações data de recebimento, setor responsável, usuário atribuído, tipo de manifestação e categoria de agrupamento. O botão “Marcar como Resolvido” permite finalizar o atendimento.

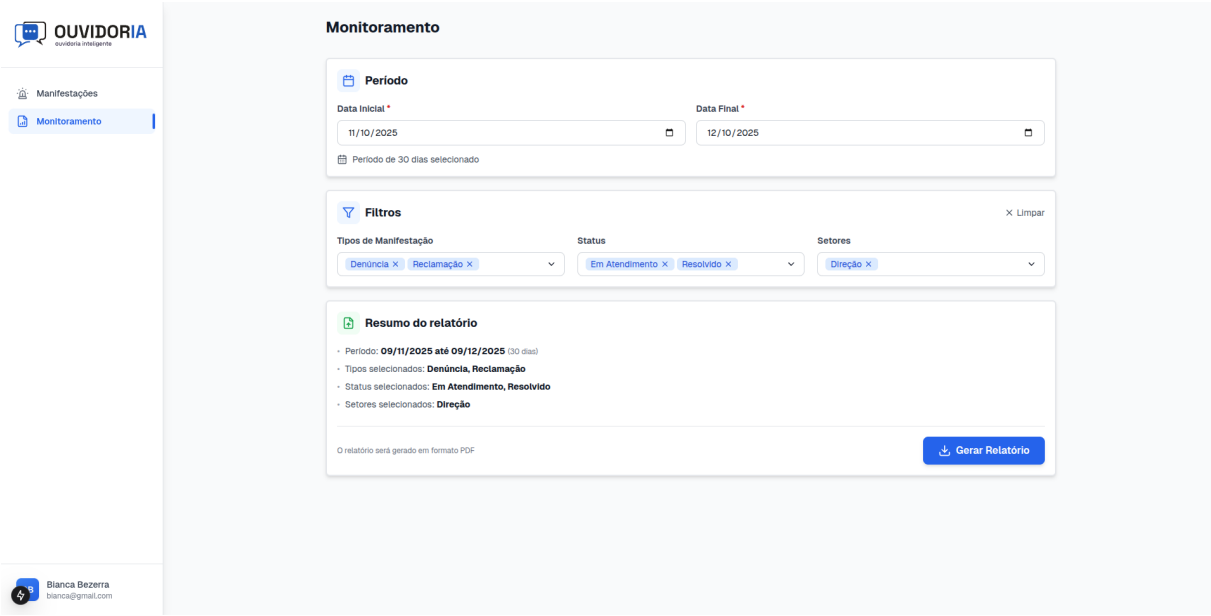
Figura 13 – Histórico da manifestação



Fonte: Elaborado pelo autor (2025)

A Figura 13 detalha o histórico de auditoria de uma manifestação. O sistema registra cronologicamente todas as ações executadas: criação do relato, classificação automática pela IA, despacho para setor (incluindo movimentações entre setores), e atribuição para atendimento. Cada entrada mostra o horário, usuário responsável pela ação, setor de origem e destino quando aplicável.

Figura 14 – Tela de geração de relatório



Fonte: Elaborado pelo autor (2025)

Por fim, a Figura 14 apresenta a tela de geração de relatórios, funcionalidade restrita aos usuários administradores e do setor Direção. A interface permite selecionar período de análise através de campos de data inicial e final. Filtros adicionais possibilitam segmentar por tipo de manifestação, status e setores específicos. O botão “Gerar Relatório” produz um documento PDF consolidado contendo métricas operacionais sobre o funcionamento da ouvidoria no período selecionado.

5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento do Ouvidor.IA, uma plataforma de ouvidoria escolar que utiliza inteligência artificial para automatizar a triagem, classificação e roteamento de manifestações. O projeto surgiu da constatação de que o crescente volume de demandas nas ouvidorias exige mecanismos eficientes para organizar informações, identificar padrões e priorizar temas sensíveis, equilibrando agilidade e qualidade no atendimento.

O sistema alcançou os objetivos propostos ao implementar módulos de recuperação semântica para agrupar manifestações similares, encaminhamento automático para setores responsáveis, geração de títulos e descrições sintetizadas, além de interfaces adequadas para manifestantes e operadores. A solução combina técnicas de processamento de linguagem natural e busca vetorial para automatizar tarefas que normalmente exigem análise manual extensiva.

A abordagem adotada equilibra automação e supervisão humana. O sistema utiliza limiares de confiança configuráveis para evitar decisões automatizadas inadequadas em casos ambíguos, mantendo o operador humano como tomador final de decisões. O registro detalhado de todas as etapas do processamento garante a transparência e rastreabilidade necessárias em sistemas de ouvidoria institucional.

O Ouvidor.IA demonstra que técnicas modernas de processamento de linguagem natural podem ser aplicadas com sucesso em plataformas de ouvidoria e contextos educacionais. A solução reduz o esforço em tarefas repetitivas e permite que operadores concentrem-se em análises que realmente exigem julgamento humano, contribuindo para modernizar a gestão de ouvidorias escolares.

5.1 TRABALHOS FUTUROS

Como continuidade deste trabalho, algumas melhorias podem ser implementadas. Primeiramente, o desenvolvimento de um modelo de aprendizado de máquina especializado para roteamento, substituindo o papel da LLM nessa tarefa, treinado com dados históricos de despachos realizados pelos operadores, poderia reduzir significativamente o custo computacional e melhorar a precisão do sistema. Associado a isso, a implementação de um mecanismo de retreinamento periódico (conhecido como "Human-in-the-loop") permitiria que correções manuais dos operadores alimentassem ciclos de ajuste do modelo, promovendo melhoria contínua da precisão ao longo do tempo.

Além disso, um ajuste fino dos modelos de linguagem utilizando dados específicos de manifestações escolares brasileiras poderia aprimorar substancialmente a

qualidade das classificações de gravidade e das gerações de texto. Por fim, o desenvolvimento de dashboards com visualizações em tempo real forneceria aos gestores um acompanhamento mais dinâmico das métricas operacionais da ouvidoria, facilitando a tomada de decisões baseadas em dados atualizados.

REFERÊNCIAS

Amazon Web Services. **O que é o Docker?** 2025. Disponível em: https://aws.amazon.com/pt/docker/. Acesso em: 9 dez. 2025.

ASGARI, T. *et al.* Identifying key success factors for startups with sentiment analysis using text data mining. **International Journal of Engineering Business Management**, v. 14, p. 18479790221131612, 2022. Disponível em: https://doi.org/10.1177/18479790221131612. Acesso em: 01 dez. 2025.

BAAI. **BGE-M3: A Versatile Embedding Model**. Beijing Academy of Artificial Intelligence, 2024. Disponível em: https://github.com/FlagOpen/FlagEmbedding. Acesso em: 9 dez. 2025.

BRASIL. Conselho Nacional de Justiça. **Ouvidoria do CNJ registra aumento de 34% nas demandas e reforça canais de atendimento**. 2025. Disponível em: https://www.cnj.jus.br/ouvidoria-do-cnj-registra-aumento-de-34-nas-demandas-e-reforca-canais-de-atendimento/. Acesso em: 9 dez. 2025.

CHASE, H. **LangChain**. GitHub, 2022. https://github.com/langchain-ai/langchain. Disponível em: https://github.com/langchain-ai/langchain. Acesso em: 9 dez. 2025.

CHEN, B. *et al.* Unleashing the potential of prompt engineering for large language models. **Patterns**, Elsevier BV, v. 6, n. 6, p. 101260, jun. 2025. ISSN 2666-3899. Disponível em: http://dx.doi.org/10.1016/j.patter.2025.101260. Acesso em: 20 nov. 2025.

CHEN, J. *et al.* **BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings**. 2024. Disponível em: https://arxiv.org/abs/2402.03216. Acesso em: 20 nov. 2025.

Controladoria-Geral da União (CGU). **Manual de Ouvidoria Pública: rumo ao sistema participativo**. [S.l.], 2019. Disponível em: https://repositorio.cgu.gov.br/bitstream/1/29959/14/manual_de_ouvidoria_publica.pdf. Acesso em: 9 dez. 2025.

DANDE, A. A.; PUND, M. A. A review study on applications of natural language processing. **International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)**, v. 10, n. 2, p. 122–126, 2023. Disponível em: https://ijsrset.com/paper/8827.pdf. Acesso em: 20 nov. 2025.

DISTRITO FEDERAL. Ouvidoria-Geral. **Ouvidoria-Geral do DF chega aos 25 anos com 1 milhão de cidadãos cadastrados**. 2024. Disponível em: https://ouvidoria.df.gov.br/ouvidoria-geral-do-df-chega-aos-25-anos-com-1-milhao-de-cidadaos-cadastrados/. Acesso em: 9 dez. 2025.

Django Ninja. **Django Ninja Documentation**. 2024. Disponível em: https://django-ninja.dev. Acesso em: 9 dez. 2025.

FALCãO, L.; LOPES, B.; SOUZA, R. R. Absorção das tarefas de processamento de linguagem natural (NLP) pela ciência da informação (CI): uma revisão da literatura para tangibilização do uso de NLP pela CI. **Em Questão**, Universidade Federal do Rio Grande do Sul, Porto Alegre, v. 28, n. 1, p. 13–34, 2022. Disponível em: https://www.redalyc.org/journal/4656/465669358002/. Acesso em: 9 dez. 2025.

FAN, W. *et al.* **A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models**. 2024. Disponível em: https://arxiv.org/abs/2405.06211. Acesso em: 27 nov. 2025.

GAHMAN, N.; ELANGOVAN, V. A comparison of document similarity algorithms. **International Journal of Artificial Intelligence and Applications (IJAIA)**, AIRCC Publishing Corporation, v. 14, n. 2, p. 41–50, March 2023. Disponível em: https://arxiv.org/pdf/2304.01330. Acesso em: 20 nov. 2025.

GAO, T.; YAO, X.; CHEN, D. SimCSE: Simple contrastive learning of sentence embeddings. In: **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021. p. 6894–6910. Disponível em: https://aclanthology.org/2021.emnlp-main.552/. Acesso em: 20 nov. 2025.

HOSSAIN, E. *et al.* Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review. **Natural Language Processing Journal**, Elsevier, 2024. Disponível em: https://www.sciencedirect.com/science/article/pii/S2949719124000074. Acesso em: 20 nov. 2025.

KRANTZ, T.; JONKER, A. What is cosine similarity? **IBM Think**, 2024. Disponível em: https://www.ibm.com/think/topics/cosine-similarity. Acesso em: 9 dez. 2025.

LangChain Inc. **LangChain Documentation**. 2024. Disponível em: https://python.langchain.com/docs/. Acesso em: 9 dez. 2025.

LIMA, J. A. de O. **Poly-Vector Retrieval: Reference and Content Embeddings for Legal Documents**. 2025. Disponível em: https://arxiv.org/abs/2504.10508. Acesso em: 20 nov. 2025.

MARVIN, G. *et al.* Prompt engineering in large language models. In: JACOB, I. J.; PIRAMUTHU, S.; FALKOWSKI-GILSKI, P. (Ed.). **Data Intelligence and Cognitive Informatics**. Singapore: Springer Nature Singapore, 2024. p. 387–402.

MATTIUZZI, C. F. **Triagem automatizada de currículos usando busca vetorial e modelos de linguagem de larga escala**. 2025. Disponível em: https://repositorio.ifes.edu.br/handle/123456789/6883. Acesso em: 1 dez. 2025.

MCTI. **Plano Brasileiro de Inteligência Artificial 2024-2028**. 2024. Disponível em: https://www.gov.br/mcti/pt-br/acompanhe-o-mcti/noticias/2024/07/plano-brasileiro-de-ia-tera-supercomputador-e-investimento-de-r-23-bilhoes-em-quatro-anos/ia_para_o_bem_de_todos.pdf/view. Acesso em: 9 dez. 2025.

Meta AI. **FAISS: A Library for Efficient Similarity Search**. 2024. Disponível em: https://faiss.ai. Acesso em: 9 dez. 2025.

MIKOLOV, T. *et al.* Efficient estimation of word representations in vector space. In: **Proceedings of Workshop at ICLR**. [S.l.: s.n.], 2013.

MINAEE, S. *et al.* **Large Language Models: A Survey**. 2025. Disponível em: https://arxiv.org/abs/2402.06196. Acesso em: 8 dez. 2025.

MinIO, Inc. **MinIO: High Performance Object Storage**. 2024. Disponível em: https://min.io/. Acesso em: 9 dez. 2025.

MIRANDA, S. S. M.; CRUZ, F. N. B. As ouvidorias estudantis como instrumento da gestão democrática. **Jornal de Políticas Educacionais**, v. 16, 2022.

MONIR, S. S. *et al.* **VectorSearch: Enhancing Document Retrieval with Semantic Embeddings and Optimized Search**. 2024. Disponível em: https://arxiv.org/abs/2409.17383. Acesso em: 18 nov. 2025.

MUENNIGHOFF, N. *et al.* Mteb: Massive text embedding benchmark. In: **Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics**. Dubrovnik, Croatia: Association for Computational Linguistics, 2023. p. 2014–2037. Disponível em: https://aclanthology.org/2023.eacl-main.148/. Acesso em: 20 nov. 2025.

Next.js Documentation. **Next.js by Vercel**. 2024. Disponível em: https://nextjs.org/docs. Acesso em: 9 dez. 2025.

OUYANG, L. *et al.* **Training language models to follow instructions with human feedback**. 2022. Disponível em: https://arxiv.org/abs/2203.02155. Acesso em: 20 nov. 2025.

PAIVA, E. S. de. **Classificação de Denúncias: Utilização de Processamento de Linguagem Natural para Aprimorar Atividades de Ouvidoria**. Tese (Tese de Doutorado) — Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, RJ, março 2023. Programa de Pós-graduação em Engenharia Civil.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. GloVe: Global vectors for word representation. In: **Proceedings of EMNLP**. [S.l.: s.n.], 2014. p. 1532–1543.

PICCINI, C.; REIS, D. F. P. d. As ouvidorias públicas como instrumento de transparência: aspectos jurídicos e federativos. **Seqüência: Estudos Jurídicos e Políticos**, v. 43, n. 92, p. 1–29, 2023. Disponível em: https://periodicos.ufsc.br/index.php/sequencia/article/view/86824. Acesso em: 20 nov. 2025.

PostgreSQL Global Development Group. **About**. 2024. Disponível em: https://www.postgresql.org/about/. Acesso em: 9 dez. 2025.

Qwen Team. **Qwen3-30B-A3B**. Hugging Face, 2025. Accessed: 2024-12-09. Disponível em: https://huggingface.co/Qwen/Qwen3-30B-A3B. Acesso em: 9 dez. 2025.

ROUMELIOTIS, K. I.; TSELIKAS, N. D.; NASIOPOULOS, D. K. Next-generation spam filtering: Comparative fine-tuning of LLMs, NLPs, and CNN models for email spam classification. **Electronics**, MDPI, v. 13, n. 11, p. 2034, 2024. Disponível em: https://www.mdpi.com/2079-9292/13/11/2034. Acesso em: 25 nov. 2025.

RUSANOV, M. The ai-powered ombudsman: A boon or a curse? **Alternative Dispute Resolution Journal**, August 2025. Published under Creative Commons Attribution-NonCommercial 4.0 International License.

SANTOS, P. A. d. *et al.* Análise das manifestações à ouvidoria-geral do SUS. **Ciência & Saúde Coletiva**, v. 27, p. 785–798, 2022.

SCLAR, M. *et al.* **Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting**. 2024. Disponível em: https://arxiv.org/abs/2310.11324. Acesso em: 20 nov. 2025.

SHUO, L.; AFFENDEY, L. S.; SIDI, F. Content-based image retrieval using transfer learning and vector database. **International Journal of Advanced Computer Science and Applications**, The Science and Information Organization, v. 15, n. 9, 2024. Disponível em: http://dx.doi.org/10.14569/IJACSA.2024.0150985. Acesso em: 20 nov. 2025.

SPILLEIR, D. d. P.; SILVA, D. H. d.; SOUZA, A. V. d. A inteligência artificial e a ouvidoria do século xxi. **Revista Científica da Associação Brasileira de Ouvidores/Ombudsman**, v. 4-5, n. 4, p. 11–23, 2021.

TABASSUM, A.; PATIL, R. R. A survey on text pre-processing & feature extraction techniques in natural language processing. **International Research Journal of Engineering and Technology (IRJET)**, v. 7, n. 06, p. 4864–4867, 2020.

TEAM, Q. **Qwen3 Technical Report**. 2025. Disponível em: https://arxiv.org/abs/2505.09388. Acesso em: 20 nov. 2025.

VASWANI, A. *et al.* Attention is all you need. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. v. 30, p. 5998–6008.

WANG, J. *et al.* Prompt engineering for healthcare: Methodologies and applications. **Meta-Radiology**, p. 100190, 2025. ISSN 2950-1628. Disponível em: https://www.sciencedirect.com/science/article/pii/S295016282500058X. Acesso em: 20 nov. 2025.

WU, T.; WANG, Y.; QUACH, N. **Advancements in Natural Language Processing: Exploring Transformer-Based Architectures for Text Understanding**. 2025. Disponível em: https://arxiv.org/abs/2503.20227. Acesso em: 20 nov. 2025.

YADAV, K.; JANU, A. Comprehensive analysis of natural language processing. **Global Journal of Engineering and Technology Advances**, v. 19, n. 1, p. 83–90, 2024.

ZOUHAR, V. *et al.* Tokenization and the noiseless channel. In: ROGERS, A.; BOYD-GRABER, J.; OKAZAKI, N. (Ed.). **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Toronto, Canada: Association for Computational Linguistics, 2023. p. 5184–5207. Disponível em: https://aclanthology.org/2023.acl-long.284/. Acesso em: 17 nov. 2025.