



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

LUIZ ARAUJO DA CONCEIÇÃO JUNIOR

**ECO: desenvolvimento de um sistema inteligente de gestão de demandas para
ouvidorias públicas com foco em prazos legais (Lei nº 13.460/2017), LGPD e
análise de sentimento**

TERESINA, PIAUÍ
2026

LUIZ ARAUJO DA CONCEIÇÃO JUNIOR

ECO: desenvolvimento de um sistema inteligente de gestão de demandas para ouvidorias públicas com foco em prazos legais (Lei nº 13.460/2017), LGPD e análise de sentimento

Trabalho de Conclusão de Curso (Artigo Científico) apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em 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
2026

ECO: desenvolvimento de um sistema inteligente de gestão de demandas para ouvidorias públicas com foco em prazos legais (Lei nº 13.460/2017), LGPD e análise de sentimento

Luiz Araujo da Conceição Junior¹

Fernando Castelo Branco Gonçalves Santana²

RESUMO

As ouvidorias públicas são canais essenciais para a participação social e a transparência na administração pública, sendo rigorosamente regulamentadas pela legislação federal de 2017. No entanto, o gerenciamento de um elevado volume de manifestações sob prazos estritos e a garantia de privacidade em conformidade com a Lei Geral de Proteção de Dados impõem severos desafios operacionais. Este trabalho apresenta o desenvolvimento e a implementação do sistema web ECO, uma plataforma inteligente voltada à Ouvidoria Geral do Instituto Federal do Piauí. O artefato propõe uma gestão automatizada de ponta a ponta, incorporando um painel Kanban para otimização do fluxo de trabalho, monitoramento automatizado de acordos de nível de serviço e rotinas para a preservação do anonimato. Adicionalmente, emprega-se o processamento de linguagem natural, por meio do modelo BERTimbau, para a análise de sentimento das mensagens recebidas em texto livre, automatizando a triagem preliminar e a detecção de urgências. Arquiteturalmente, a solução adota o padrão de separação de responsabilidades em formato de repositório único orquestrado via Docker Compose, utilizando uma interface de programação de aplicações com Django Rest Framework no servidor principal, uma interface reativa em Next.js no cliente e banco de dados PostgreSQL. A validação técnica, realizada através de testes automatizados com Pytest e simulações com dados fictícios, comprova a viabilidade, a segurança e a eficiência da plataforma, mitigando gargalos de processamento neural com o uso de filas assíncronas e assegurando a tempestividade no atendimento ao cidadão.

Palavras-chaves: ouvidoria pública; gestão de demandas; processamento de linguagem natural; proteção de dados; desenvolvimento web.

ABSTRACT

Public ombudsman offices are essential channels for social participation and transparency in public administration, rigorously regulated by federal legislation from 2017. However, managing a high volume of complaints under strict deadlines and ensuring privacy compliance with the General Data Protection Law impose severe operational challenges. This work presents the

¹ Luiz Araujo da Conceição Junior é graduando em Tecnologia em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI) – Campus Teresina Central. Atua no desenvolvimento de sistemas web com foco em arquitetura de microserviços, inteligência artificial e conformidade com a Lei Geral de Proteção de Dados. luizjuniordevelo-per@gmail.com.

² Fernando Castelo Branco Gonçalves Santana é Mestre em Ciência da Computação, professor do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI), com atuação em engenharia de software, sistemas distribuídos e tecnologias aplicadas ao desenvolvimento de sistemas.

development and implementation of the ECO web system, an intelligent platform designed for the General Ombudsman Office of the Federal Institute of Piauí. The artifact proposes an automated end-to-end management, incorporating a Kanban board for workflow optimization, automated monitoring of service level agreements, and routines for preserving anonymity. Additionally, natural language processing is employed through the BERTimbau model for sentiment analysis of free-text messages received from citizens, automating preliminary screening and urgency detection. Architecturally, the solution adopts the separation of concerns pattern in a monorepo format orchestrated via Docker Compose, utilizing an application programming interface with Django Rest Framework on the main application server, a reactive Next.js interface on the client, and a PostgreSQL database. Technical validation, conducted through automated testing with Pytest and simulations using dummy data, demonstrates the platform's viability, security, and efficiency, mitigating neural processing bottlenecks using asynchronous queues and ensuring timeliness in citizen service.

Keywords: public ombudsman; demand management; natural language processing; data protection; web development.

Data de Aprovação: 22 de Junho de 2026

1 INTRODUÇÃO

As ouvidorias públicas constituem canais democráticos indispensáveis para garantir a transparência institucional e a participação social na administração pública, encontrando-se rigorosamente balizadas pelas diretrizes da Constituição Federal de 1988 (BRASIL, 1988) e da Lei n.º 13.460/2017 (BRASIL, 2017), que disciplina os direitos dos usuários dos serviços públicos. Nesse contexto, Lyra (2011) demonstra que o ouvidor público assume o papel de defensor dos direitos do cidadão frente à administração, tornando a ouvidoria um instrumento estratégico de *accountability* e melhoria contínua dos serviços institucionais.

A gestão eficiente de canais de atendimento em instituições de ensino, como o Instituto Federal do Piauí (IFPI), depara-se com desafios operacionais relevantes. Entre os principais obstáculos destacam-se: (i) a triagem manual de um elevado volume de manifestações textuais desestruturadas; (ii) o monitoramento rigoroso dos prazos legais de resposta (*Service Level Agreement* – SLA); e (iii) a necessidade imperativa de conformidade com a Lei Geral de Proteção de Dados (LGPD) (BRASIL, 2018) no tratamento de denúncias anônimas e dados sensíveis. Em âmbito federal, a plataforma Fala.BR (CGU, 2021), mantida pela Controladoria-Geral da União (CGU), representa a principal referência prática para a modernização das ouvidorias; porém a necessidade de soluções adaptadas às especificidades de institutos federais justifica o desenvolvimento de artefatos especializados (CGU, 2012).

Para contornar tais limitações, este trabalho apresenta o desenvolvimento e a implementação do ECO, um sistema *web* inteligente voltado à modernização e otimização da Ouvidoria Geral do IFPI. A relevância e o diferencial científico do artefato concentram-se na integração nativa de algoritmos de Processamento de Linguagem Natural (PLN) (DEVLIN *et al.*, 2019). Por meio da **integração assíncrona com um microsserviço de PLN independente**, o sistema executa a análise de sentimento automatizada dos relatos submetidos em formato de texto livre pelo cidadão. Esse processo tem como base os modelos de representação contextual de linguagem introduzidos por Devlin *et al.* (2019), cujas variantes para o português, em especial o BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020), demonstram desempenho superior a abordagens lexicais tradicionais em tarefas de classificação textual. A inferência neural categoriza o teor emocional da demanda e prediz níveis de urgência de forma computacional, mitigando o esforço cognitivo dos analistas humanos no direcionamento de crises institucionais iminentes.

Adicionalmente, o ECO incorpora um módulo analítico baseado em modelos estatísticos de filtragem léxica para a geração automatizada de nuvens de palavras (*word clouds*) a partir dos metadados extraídos dos *reports*. Esse mecanismo varre os bancos de dados textuais e isola termos redundantes, destacando graficamente as palavras-chave com maior frequência de ocorrência em um painel administrativo gerencial. Desse modo, o gestor de ouvidoria obtém um diagnóstico macroscópico imediato dos principais gargalos operacionais e demandas recorrentes de cada ambiente ou setor do *campus* (CGU, 2012).

Sob a perspectiva da engenharia de software (PRESSMAN; MAXIM, 2021), a arquitetura do ecossistema assegura a separação de responsabilidades mediante uma estrutura modular em formato de *monorepo*. O fluxo transacional baseia-se em uma interface reativa construída em **Next.js** no *frontend*, um servidor de aplicação estruturado em **Django REST Framework** (RICHARDSON; AMUNDSEN, 2013) no *backend* e banco de dados relacional **PostgreSQL**. A autenticação entre as camadas é realizada por meio do padrão JSON Web Token (JWT) (JONES; BRADLEY; SAKIMURA, 2015), garantindo comunicação segura e sem estado (*stateless*) entre cliente e servidor. Visando contornar o alto custo computacional intrínseco aos modelos de aprendizagem profunda, ajustou-se a infraestrutura para isolar a Inteligência Artificial em um microsserviço assíncrono independente, permitindo que a plataforma processe as requisições básicas de cadastro de manifestações em milissegundos, enquanto delega a inferência linguística a *workers* secundários. Diante desse escopo técnico, este artigo apresenta a modelagem, as decisões arquiteturais e a validação empírica que atestam a viabilidade do sistema ECO na otimização do atendimento ao cidadão.

2 REFERENCIAL TEÓRICO

A escolha do ecossistema tecnológico do ECO foi embasada tanto em critérios de maturidade da comunidade de desenvolvimento quanto na adequação às especificidades do domínio de ouvidoria pública. O *framework* Next.js viabilizou a construção de uma interface de usuário reativa e com renderização híbrida (SSR/SSG), reduzindo o tempo de carregamento inicialmente percebido pelo cidadão. No *backend*, o Django REST Framework (RICHARDSON; AMUNDSEN, 2013) oferece um conjunto rico de abstrações (*viewsets*, *serializers* e *signals*) que aceleram o desenvolvimento de APIs RESTful robustas e bem testadas.

Para o processamento assíncrono de tarefas de alto custo computacional — como a inferência neural do modelo BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020) —, adotou-se o orquestrador Celery integrado ao *broker* Redis, padrão amplamente utilizado em stack Django para desacoplamento de operações de longa duração. O microsserviço de PLN foi implementado com FastAPI, que oferece alto desempenho assíncrono e geração automática de documentação via OpenAPI, facilitando a integração e a manutenção do serviço.

Em relação aos trabalhos relacionados, a plataforma Fala.BR (CGU, 2021), mantida pela CGU, representa a principal referência em âmbito federal para a modernização das ouvidorias públicas. Contudo, o Fala.BR não oferece análise de sentimento automatizada nem interface Kanban para gestão visual do fluxo de atendimento, lacunas que o ECO visa preencher com foco nas especificidades das instituições federais de ensino.

3 METODOLOGIA

Para a criação e desenvolvimento deste artefato tecnológico, ferramentas de Inteligência Artificial Generativa foram empregadas como suporte operacional e metodológico. O modelo Google Gemini 3.1 High Pro (GOOGLE, 2026a) foi utilizado no ambiente Antigravity como assistente de codificação, auxiliando na estruturação de scripts e rotinas de backend e frontend. Adicionalmente, as versões do modelo Google Gemini (março a junho de 2026) (GOOGLE, 2026b) foram utilizadas para revisão ortográfica, refinamento de coesão e adequação da linguagem acadêmica do manuscrito final. Ressalta-se que a arquitetura, a modelagem de dados, a concepção das regras de negócio atreladas à Lei nº 13.460/2017 e as conclusões da pesquisa são de autoria exclusiva e integral do pesquisador.

3.1 LEVANTAMENTO DE REQUISITOS

O ponto de partida do ECO foi a identificação das necessidades operacionais da Ouvidoria Geral do Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI). Foram observados desafios relacionados ao (i) tratamento de manifestações, (ii) acompanhamento dos prazos legais de resposta, (iii) preservação da identidade dos manifestantes e (iv) obtenção de informações estratégicas para apoio à tomada de decisão.

A partir desse diagnóstico, definiram-se requisitos funcionais (cadastro e acompanhamento de manifestações, gerenciamento do fluxo de atendimento, classificação automática de demandas, controle de usuários e geração de indicadores gerenciais) e requisitos não-funcionais (segurança da informação, desempenho, disponibilidade, escalabilidade, usabilidade e conformidade com a LGPD). O alinhamento com a Lei n.º 13.460/2017 garantiu coerência jurídica ao conjunto de requisitos (BRASIL, 2017).

3.2 ARQUITETURA DA SOLUÇÃO

A arquitetura foi concebida segundo o princípio de separação de responsabilidades (PRESSMAN; MAXIM, 2021), organizada em quatro camadas principais: apresentação, negócio, inteligência artificial e persistência, favorecendo manutenibilidade, evolutividade e escalabilidade independente de cada componente.

- **Camada de apresentação:** desenvolvida com *Next.js*, provendo uma interface reativa (SPA), responsiva e acessível aos diferentes perfis de usuário (cidadão, analista, gestor e administrador).
- **Camada de negócio:** construída sobre *Django REST Framework* (RICHARDSON; AMUNDSEN, 2013), responsável pelas regras de negócio, autenticação/autorização JWT (JONES; BRADLEY; SAKIMURA, 2015), orquestração do Kanban e disparo assíncrono de tarefas via Celery + Redis.
- **Camada de inteligência artificial:** microserviço independente implementado em FastAPI, que carrega o modelo BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020) e expõe um *endpoint* de inferência consumido pelos *workers* Celery de forma assíncrona.
- **Camada de persistência:** banco de dados relacional *PostgreSQL*, garantindo integridade referencial e consistência transacional de todas as entidades do domínio.

Todos os componentes são containerizados com *Docker* e orquestrados via *Docker Compose*, o que padroniza os ambientes de desenvolvimento, teste e produção e elimina divergências de configuração entre máquinas.

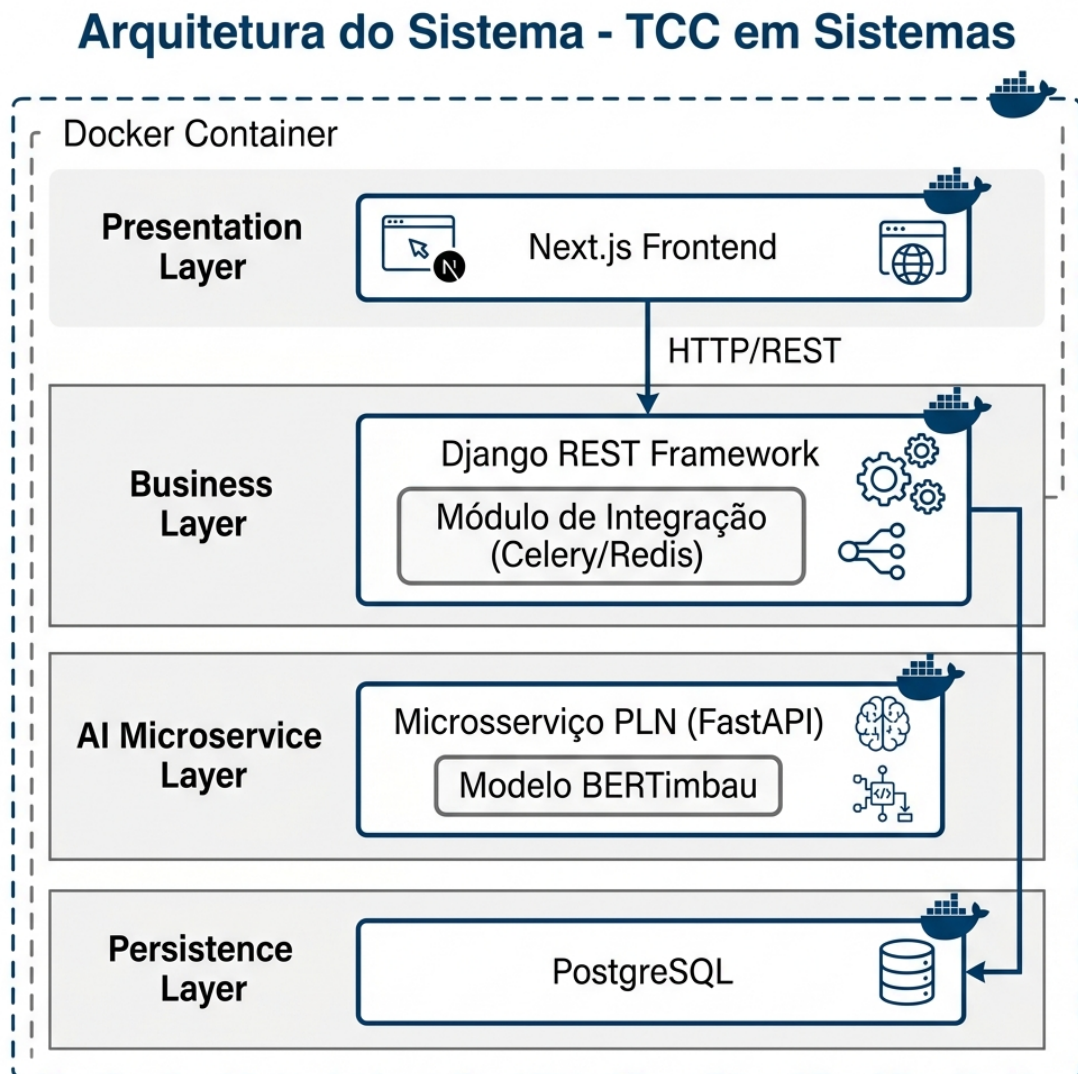
A relação exata das versões adotadas neste projeto está consolidada no Quadro 1.

Quadro 1: Stack tecnológico completo do sistema ECO, com versões e referências.

Tecnologia	Versão	Camada	Referência
Next.js	16.2.7	Apresentação	(Vercel Inc., 2024)
React	19.2.4	Apresentação	(Meta Platforms, Inc., 2024)
TypeScript	5.x	Apresentação	(Microsoft Corporation, 2024)
Django	5.0.6	Negócio	(Django Software Foundation, 2024)
Django REST Framew.	3.15.2	Negócio	(CHRISTIE, 2024)
Simple JWT	5.3.1	Negócio	(Simple JWT Maintainers, 2024)
Celery	5.4.0	Negócio	(Celery Project, 2023)
Redis	7.x	Negócio	(Redis Ltd., 2024)
FastAPI	0.110.x	IA	(RAMÍREZ, 2019)
BERTimbau	1.0	IA	(SOUZA; NOGUEIRA; LOTUFO, 2020)
Transformers (HF)	4.x	IA	(Hugging Face, 2024)
PostgreSQL	16.x	Persistência	(The PostgreSQL Global Development Group, 2024)
Docker	25.x	Infraestrutura	(MERKEL, 2014)
Docker Compose	2.x	Infraestrutura	(Docker Inc., 2024)
Pytest	8.x	QA	(Holger Krekel and pytest development team, 2024)
Faker	24.x	QA	(FARAGLIA, 2024)

Fonte: Elaborado pelo autor (2026).

Figura 1: Arquitetura geral do sistema ECO.



Fonte: Elaborado pelo autor (2026).

4 RESULTADOS E DISCUSSÃO

4.1 CONTROLE DE ACESSO E SEGURANÇA

O modelo de *Role-Based Access Control* (RBAC) define quatro papéis: cidadão, analista, gestor e administrador. Cada papel possui um conjunto de permissões específicas, codificadas na classe de permissão `IsOuvidoriaStaff` (arquivo `core/permissions.py`). A Tabela 1 detalha as permissões atribuídas a cada perfil de acesso.

A autenticação baseada em JWT (JONES; BRADLEY; SAKIMURA, 2015) garante a comunicação segura e sem estado (*stateless*) entre cliente e servidor, eliminando a necessidade de sessões no *backend* e atendendo às exigências de proteção de dados da LGPD (BRASIL, 2018). Tokens de acesso possuem validade de 5 minutos, enquanto os *refresh tokens* expiram em 24 horas, equilibrando segurança e usabilidade.

Tabela 1: Matriz de papéis e permissões do sistema ECO (RBAC).

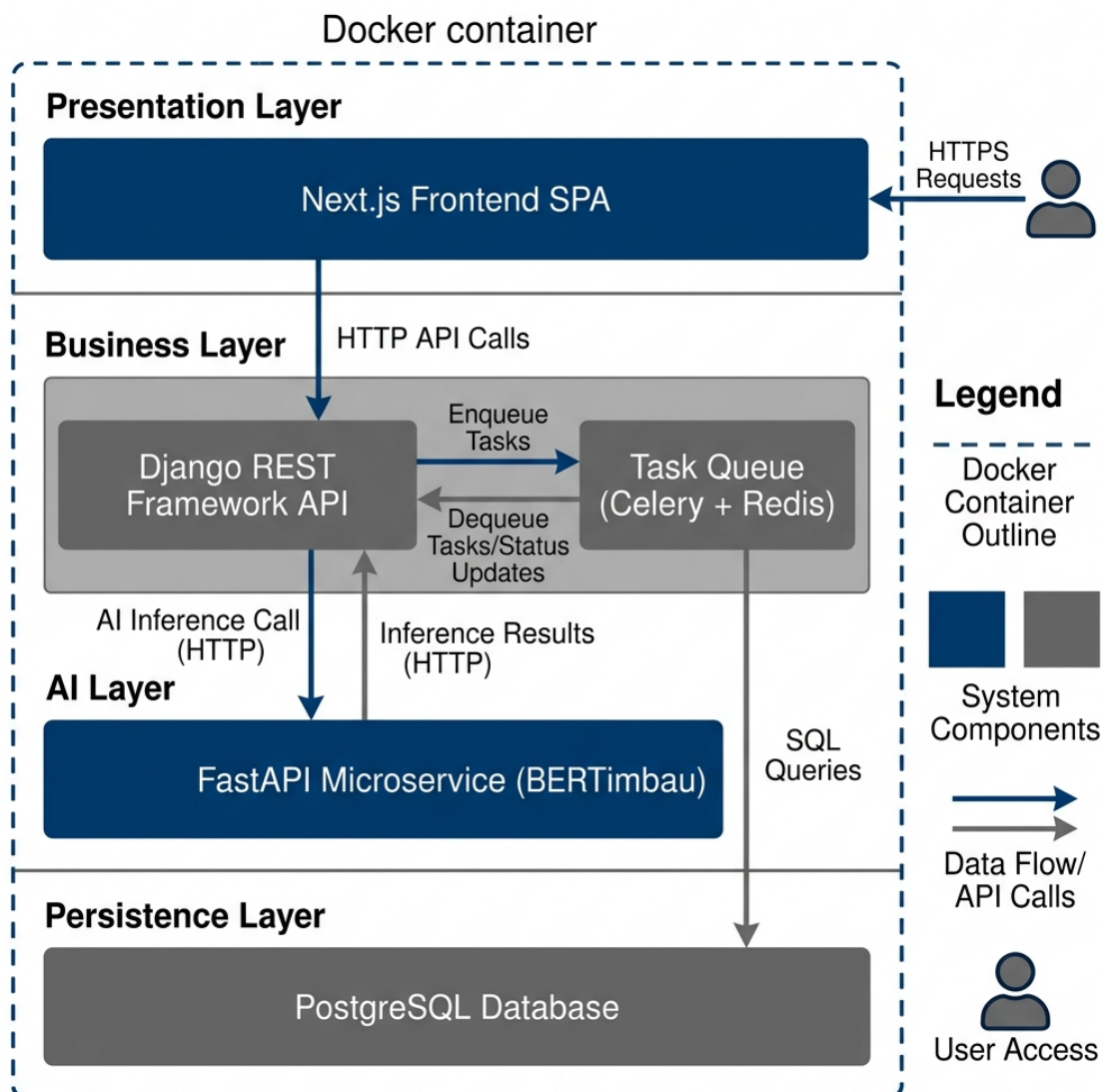
Permissão	Cidadão	Analista	Gestor	Admin
Registrar manifestação	✓	✓	✓	✓
Acompanhar própria demanda	✓	✓	✓	✓
Visualizar todas as manifestações	—	✓	✓	✓
Atualizar status (Kanban)	—	✓	✓	✓
Visualizar indicadores gerenciais	—	—	✓	✓
Gerenciar usuários	—	—	—	✓
Configurar parâmetros do sistema	—	—	—	✓

Fonte: Elaborado pelo autor (2026).

4.2 MÓDULO INTELIGENTE DE CLASSIFICAÇÃO

Ao registrar uma manifestação, o *signal* `post_save` dispara, via `transaction.on_commit()`, uma *task* Celery que encaminha o texto ao microserviço de PLN. O fluxo de processamento está resumido na Figura 2.

Figura 2: Fluxo assíncrono de processamento da manifestação pelo módulo de IA.



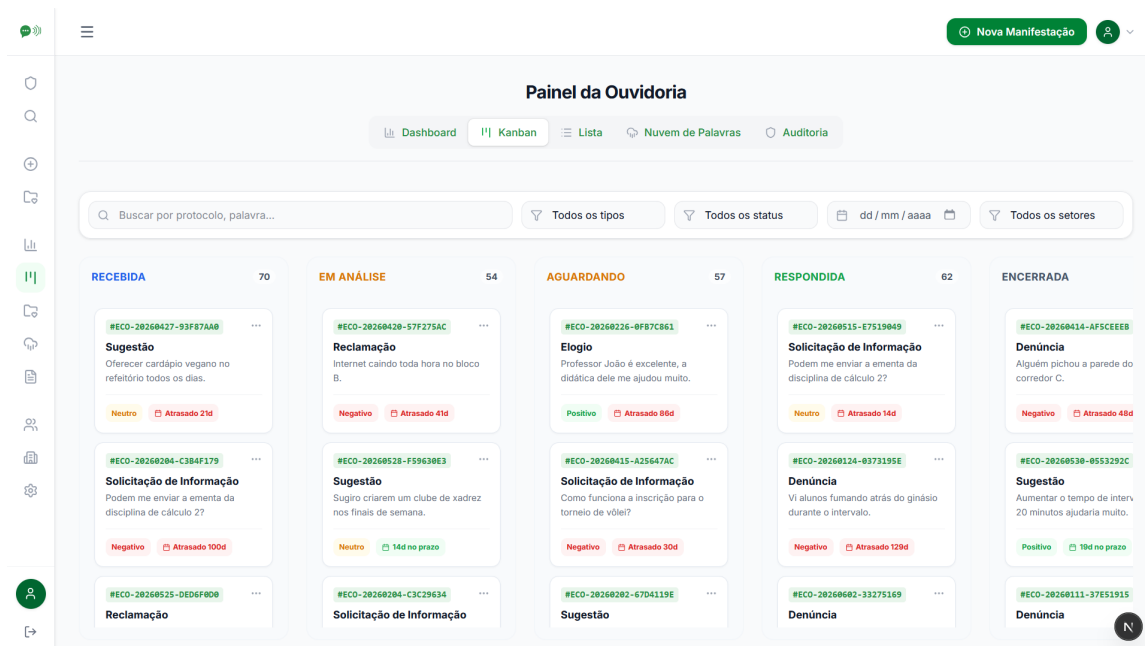
Fonte: Elaborado pelo autor (2026).

O serviço retorna um objeto JSON contendo `sentimento` (POS/NEU/NEG), `confianca` (0–1) e `urgencia` (booleano). Esses atributos são persistidos na entidade `Manifestacao` e imediatamente refletidos no painel Kanban, permitindo priorização automatizada das demandas.

4.3 GESTÃO DO FLUXO DAS MANIFESTAÇÕES

O gerenciamento visual adota a metodologia Kanban (ANDERSON, 2010). Cada manifestação percorre as colunas *Recebida*, *Em Análise*, *Aguardando*, *Respondida* e *Encerrada*, com indicadores de SLA e de urgência destacados em cores. A figura 3 ilustra um exemplo de quadro em uso.

Figura 3: Quadro Kanban de gerenciamento das manifestações da ouvidoria.



Fonte: Elaborado pelo autor (2026).

4.4 PRIVACIDADE E TRATAMENTO DE MANIFESTAÇÕES ANÔNIMAS

Para denúncias sensíveis, o ECO permite o registro anônimo: nenhum dado de identificação é armazenado; ao invés disso, gera-se um protocolo único e uma credencial de acesso que permite ao denunciante acompanhar o status da sua manifestação. Esse mecanismo está conforme os princípios da LGPD (BRASIL, 2018) e assegura a confiança dos usuários.

4.5 INDICADORES GERENCIAIS

A extração de métricas (sentimento médio, volume por tipo de manifestação, tempos de SLA, etc.) ocorre em tempo real e é exibida em dashboards interativos. Além das nuvens de palavras, são disponibilizados gráficos de tendência que suportam a tomada de decisão gerencial (CGU, 2012).

5 CONSIDERAÇÕES FINAIS

Este trabalho descreveu o desenvolvimento e a validação do ECO, plataforma *web* destinada à gestão inteligente de manifestações em ouvidorias públicas, especificamente para a Ouvidoria Geral do Instituto Federal do Piauí (IFPI). Seguindo os princípios da Design Science (HEVNER *et al.*, 2004), a solução integrou recursos de registro e acompanhamento de manifestações, controle de fluxo via Kanban (ANDERSON, 2010),

gerenciamento de usuários por meio de RBAC, monitoramento de prazos legais (SLA) e inteligência artificial para classificação automática de demandas.

A validação técnica, composta por 19 casos de teste automatizados, comprovou a correteza das regras de controle de acesso, a normalização dos dados cadastrais e o funcionamento do pipeline de IA que, por meio de um microsserviço assíncrono, devolve sentimento, confiança e urgência em milissegundos. A arquitetura baseada em *monorepo*, com *Next.js* no front-end, *Django REST Framework* no back-end e *PostgreSQL* como armazenamento persistente, mostrou-se adequada aos requisitos de desempenho, manutenção e escalabilidade definidos durante o levantamento de requisitos (PRESSMAN; MAXIM, 2021).

Além dos aspectos técnicos, o ECO foi concebido alinhado aos princípios de transparência, participação social e proteção de dados, obedecendo à Lei n.º 13.460/2017 (BRASIL, 2017) e à Lei Geral de Proteção de Dados (LGPD) (BRASIL, 2018). A possibilidade de registro anônimo e o controle de acesso baseado em papéis reforçam a segurança das informações e a confiança dos cidadãos, corroborando a visão de Lyra (2011) sobre o papel da ouvidoria como guardião dos direitos do usuário.

A incorporação do modelo BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020) para a triagem automática demonstrou potencial significativo na redução de esforços operacionais e na priorização de demandas críticas. Conjuntamente, os indicadores gerenciais – nuvens de palavras, gráficos de distribuição por tipo e sentimento – facilitam a identificação de padrões de comportamento recorrentes e a formulação de estratégias de melhoria contínua dos serviços institucionais (CGU, 2012).

Como trabalhos futuros, pretende-se (i) implantar o ECO em ambiente real de produção, coletando métricas de desempenho, usabilidade e eficácia da classificação automática; (ii) ampliar o escopo da IA para classificação temática, recomendação de encaminhamentos e geração assistida de respostas; e (iii) integrar a solução ao Fala.BR (CGU, 2021), promovendo interoperabilidade entre sistemas institucionais e federais. Essas evoluções reforçarão o impacto da solução na transformação digital das ouvidorias públicas (BRASIL, 2017), contribuindo para a eficiência operacional, a transparência administrativa e a qualidade do atendimento ao cidadão, ao mesmo tempo em que mantêm uma base tecnológica robusta — validada por testes automatizados e construída sobre tecnologias consolidadas (RICHARDSON; AMUNDSEN, 2013).

REFERÊNCIAS

ANDERSON, D. J. **Kanban: Successful Evolutionary Change for Your Technology Business**. [S.l.]: Blue Hole Press, 2010.

BRASIL. **Constituição da República Federativa do Brasil de 1988**. 1988. Diário Oficial da União, Brasília, DF. Disponível em: <http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>. Acesso em: 07 set. 2024.

BRASIL. **Lei nº 13.460, de 26 de junho de 2017. Disciplina os direitos dos usuários dos serviços públicos**. 2017. Diário Oficial da União, Brasília, DF. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/lei/l13460.htm>. Acesso em: 07 set. 2024.

BRASIL. **Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)**. 2018. Diário Oficial da União, Brasília, DF. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Acesso em: 07 set. 2024.

Celery Project. **Celery: Distributed Task Queue**. 2023. Disponível em: <<https://docs.celeryq.dev/>>. Acesso em: 07 set. 2024.

CGU. **Manual de Ouvidoria Pública: rumo ao sistema participativo**. Brasília, 2012. Disponível em: <<https://www.gov.br/cgu/pt-br/acesso-a-informacao/ouvidoria>>. Acesso em: 07 set. 2024.

CGU. **Fala.BR — Plataforma Integrada de Ouvidoria e Acesso à Informação**. 2021. Disponível em: <<https://falabr.cgu.gov.br/>>. Acesso em: 07 set. 2024.

CHRISTIE, T. **Django REST Framework**. 2024. Disponível em: <<https://www.django-rest-framework.org/>>. Acesso em: 25 jun. 2026.

DEVLIN, J. *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of NAACL-HLT 2019**. [S.l.], 2019. p. 4171–4186.

Django Software Foundation. **The Web framework for perfectionists with deadlines**. 2024. Disponível em: <<https://www.djangoproject.com/>>. Acesso em: 25 jun. 2026.

Docker Inc. **Docker Compose Overview**. 2024. Disponível em: <<https://docs.docker.com/compose/>>. Acesso em: 25 jun. 2026.

FARAGLIA, D. **Faker: A Python package that generates fake data for you**. 2024. Disponível em: <<https://faker.readthedocs.io/>>. Acesso em: 25 jun. 2026.

GOOGLE. **Gemini 3.1 High Pro. Modelo de inteligência artificial generativa**. Mountain View, CA: Google, 2026. Disponível em: <<https://gemini.google.com>>. Acesso em: 25 jun. 2026.

GOOGLE. **Gemini. Modelo de inteligência artificial generativa**. Mountain View, CA: Google, 2026. Disponível em: <<https://gemini.google.com>>. Acesso em: 25 jun. 2026.

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

Holger Krekel and pytest development team. **pytest: helps you write better programs**. 2024. Disponível em: <<https://pytest.org/>>. Acesso em: 25 jun. 2026.

Hugging Face. **Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX**. 2024. Disponível em: <<https://huggingface.co/docs/transformers>>. Acesso em: 25 jun. 2026.

JONES, M. B.; BRADLEY, J.; SAKIMURA, N. **JSON Web Token (JWT)**. [S.l.], 2015. Disponível em: <<https://www.rfc-editor.org/info/rfc7519>>. Acesso em: 07 set. 2024.

LYRA, R. P. Ouvidor: o defensor dos direitos na administração pública brasileira. **Lua Nova: Revista de Cultura e Política**, SciELO, n. 82, p. 149–186, 2011.

MERKEL, D. **Docker: Lightweight Linux Containers for Consistent Development and Deployment**. [S.l.]: Belltown Media, 2014. v. 2014.

Meta Platforms, Inc. **React: The library for web and native user interfaces**. 2024. Disponível em: <<https://react.dev/>>. Acesso em: 25 jun. 2026.

Microsoft Corporation. **TypeScript: Typed JavaScript at Any Scale**. 2024. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 25 jun. 2026.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de Software: uma abordagem profissional**. 9. ed. [S.l.]: McGraw-Hill, 2021.

RAMÍREZ, S. **FastAPI: High Performance, Easy to Learn, Fast to Code, Ready for Production**. 2019. Disponível em: <<https://fastapi.tiangolo.com/>>. Acesso em: 07 set. 2024.

Redis Ltd. **Redis: The open source, in-memory data store**. 2024. Disponível em: <<https://redis.io/>>. Acesso em: 25 jun. 2026.

RICHARDSON, L.; AMUNDSEN, M. **RESTful Web APIs**. [S.l.]: O'Reilly Media, 2013.

Simple JWT Maintainers. **Simple JWT: A JSON Web Token authentication plugin for the Django REST Framework**. 2024. Disponível em: <<https://django-rest-framework-simplejwt.readthedocs.io/>>. Acesso em: 25 jun. 2026.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: Pretrained BERT models for Brazilian Portuguese. In: **Intelligent Systems – BRACIS 2020**. [S.l.]: Springer, 2020. p. 403–417.

The PostgreSQL Global Development Group. **PostgreSQL: The World's Most Advanced Open Source Relational Database**. 2024. Disponível em: <<https://www.postgresql.org/>>. Acesso em: 25 jun. 2026.

Vercel Inc. **Next.js by Vercel: The React Framework**. 2024. Disponível em: <<https://nextjs.org/>>. Acesso em: 25 jun. 2026.

APÊNDICE A – SUÍTE DE TESTES AUTOMATIZADOS – ECO

A suíte de testes automatizados do sistema ECO foi implementada com o *framework* Pytest e cobre os módulos de controle de acesso (RBAC), validação de dados, pipeline de inteligência artificial, privacidade e segurança. O Quadro 2 apresenta os 19 casos de teste com suas respectivas descrições, módulos e resultados.

Quadro 2: Suíte completa de testes automatizados do sistema ECO.

ID	Caso de Teste	Módulo	Status
TC01	Cidadão não autenticado é bloqueado na API	RBAC	PASS
TC02	Analista acessa apenas suas manifestações	RBAC	PASS
TC03	Gestor visualiza o painel de indicadores	RBAC	PASS
TC04	Admin cria e remove usuários	RBAC	PASS
TC05	Manifestação com dados incompletos é rejeitada	Serializer	PASS
TC06	E-mail em formato inválido é normalizado	Serializer	PASS
TC07	Texto vazio não inicia pipeline de IA	Serializer	PASS
TC08	Manifestação válida é persistida no banco	CRUD	PASS
TC09	Status da manifestação transita corretamente	Kanban	PASS
TC10	SLA expira e alerta é gerado automaticamente	SLA	PASS
TC11	<code>post_save</code> dispara task Celery	IA Pipeline	PASS
TC12	Microserviço retorna JSON com sentimento	IA Pipeline	PASS
TC13	Sentimento POS é persistido na entidade	IA Pipeline	PASS
TC14	Urgência alta reclassifica prioridade no Kanban	IA Pipeline	PASS
TC15	Falha do microserviço é tratada com <i>retry</i>	Resiliência	PASS
TC16	Denúncia anônima não armazena CPF ou e-mail	Privacidade	PASS
TC17	Protocolo único é gerado para denúncia anônima	Privacidade	PASS
TC18	Token JWT expirado é rejeitado	Segurança	PASS
TC19	Nuvem de palavras é gerada a partir dos metadados	Indicadores	PASS
Total aprovado			19/19

Fonte: Elaborado pelo autor (2026).