

# Sonar Municipal: Recomendação de Políticas Públicas a partir de Projetos de Lei Municipais e Indicadores Oficiais

Autor(a) 1\*  
Instituição  
email@exemplo.com

Autor(a) 2  
Instituição  
email@exemplo.com

## Abstract

Municipal governments often need to draft, compare, and justify legislative proposals under limited time and resources. This report describes a reproducible method to discover Brazilian SAPL instances, extract municipal Bills (Projetos de Lei), transform their short summaries (ementas) into action-oriented recommendations, and retrieve relevant actions via semantic search. We also describe how time-series indicators from official sources can be used to approximate real-world effects, and how similar proposals can be grouped into policy clusters to improve statistical robustness. Finally, we present a web platform that operationalizes the method for non-technical users. **Keywords:** municipal legislation; SAPL; semantic search; text-to-text models; open data; reproducibility

## Resumo

Gestores municipais frequentemente precisam redigir, comparar e justificar proposições legislativas com restrição de tempo e de recursos. Este relatório descreve um método reproduzível para (i) descobrir instâncias do SAPL no Brasil, (ii) extrair Projetos de Lei (PLs), (iii) transformar ementas em recomendações de ação, e (iv) recuperar ações relevantes por busca semântica. Também descrevemos como indicadores em séries temporais de fontes oficiais podem ser usados para estimar efeitos no mundo real, e como agrupar projetos similares em “políticas públicas” aumenta o poder estatístico e a explicabilidade. Por fim, apresentamos uma plataforma web que torna o método acessível a usuários sem perfil técnico. **Palavras-chave:** legislação municipal; SAPL; busca semântica; modelos texto-para-texto; dados abertos; reprodutibilidade

## 1 Introdução

A produção legislativa municipal no Brasil ocorre em um cenário de alta diversidade institucional. Mesmo quando diferentes municípios enfrentam problemas semelhantes (por exemplo, segurança pública ou evasão escolar), as soluções propostas podem variar em linguagem, escopo e grau de detalhamento. Isso dificulta a reutilização de conhecimento legislativo entre municípios e reduz a eficiência da tomada de decisão baseada em evidências.

Este trabalho descreve o projeto **Sonar Municipal** (plataforma web: <https://sonar-municipal.vercel.app/>), cujo objetivo é apoiar a elaboração e análise de Projetos de Lei por meio de:

- descoberta e coleta sistemática de PLs em instâncias do SAPL;
- tradução de ementas para uma linguagem de ação recomendada;
- busca semântica de ações a partir da pergunta do usuário;
- simulação aproximada de efeitos em indicadores oficiais ao longo do tempo;
- agrupamento de PLs similares em “políticas públicas” para análise conjunta.

**Contribuições.** (C1) Método auditável para descoberta de instâncias SAPL e extração exaustiva de PLs via API.

(C2) Tradução de ementas para ações recomendáveis. (C3) Busca semântica por embeddings. (C4) Análise com indicadores e agrupamento. (C5) Plataforma web para usuários não técnicos.

## 2 Fundamentação e conceitos utilizados

### 2.1 SAPL e padronização do processo legislativo

O **SAPL** é um sistema mantido pelo Programa Interlegis e amplamente adotado por casas legislativas brasileiras para informatizar e dar publicidade ao processo legislativo (SENADO FEDERAL; PROGRAMA INTERLEGIS, 2026). Por ser de código aberto (INTERLEGIS, 2026), muitas instâncias compartilham estruturas semelhantes de consulta e, em vários casos, expõem uma API.

### 2.2 APIs HTTP, REST, OpenAPI e Swagger

Uma **API** (Interface de Programação de Aplicações) é um conjunto de regras que permite que programas acessem

funcionalidades e dados de um serviço. Neste trabalho, consideramos APIs acessadas por HTTP.

O estilo **REST** (Representational State Transfer) organiza a API em recursos acessados por URLs, com operações padronizadas (por exemplo, GET para consulta) (FIELDING, 2000). Um **endpoint** é a URL associada a um recurso específico (por exemplo, uma lista de matérias).

**OpenAPI** descreve formalmente endpoints, parâmetros e respostas (OPENAPI INITIATIVE, 2024). O **Swagger UI** renderiza essa descrição em uma interface navegável para exploração e teste (SWAGGER, 2026).

## 2.3 Paginação

APIs costumam retornar coleções de forma **paginada**, dividindo resultados em páginas para reduzir carga e tamanho da resposta. Paginação é essencial para extração exaustiva de bases volumosas (DJANGO REST FRAMEWORK, 2026).

## 2.4 Embeddings, busca semântica e banco vetorial

**Embedding** é um vetor numérico que representa um texto de modo que textos semanticamente próximos fiquem próximos no espaço vetorial. **Busca semântica** recupera itens por similaridade entre embeddings, e não por coincidência literal.

Um **banco vetorial** armazena vetores e permite consultas eficientes por similaridade. Neste projeto, utilizamos o Qdrant (QDRANT, 2026b,a).

## 2.5 Proveniência e FAIR

**Proveniência** é informação sobre origem e transformações do dado, útil para auditoria e reuso. O modelo PROV-DM do W3C formaliza esse conceito (W3C, 2013). Também buscamos aderência prática aos princípios **FAIR** (Encontrável, Acessível, Interoperável e Reutilizável) (WILKINSON; DUMONTIER; AALBERSBERG et al., 2016).

# 3 Metodologia

## 3.1 Visão geral do pipeline

A Figura 1 resume o pipeline: (E1) municípios, (E2) descoberta SAPL, (E3) extração de PLs, (E4) ementa para ação, (E5) indexação vetorial e (E6) indicadores e agrupamento.

## 3.2 Descoberta de instâncias SAPL e extração de Projetos de Lei

### 3.2.1 Universo de busca e cobertura territorial

O quadro amostral (lista de municípios) foi obtido via **API de localidades do IBGE** (IBGE, 2026). Essa escolha reduz viés de seleção por listas manuais e permite replicação a partir da mesma fonte.

### 3.2.2 Geração de candidatos e validação por evidência pública

A descoberta foi implementada como **geração de candidatos + validação**. A geração usa **heurísticas** (regra prática baseada em padrões observáveis) para criar endereços prováveis de portais SAPL. A validação checa se o conteúdo e o comportamento de rotas públicas são compatíveis com o sistema.

---

**Algorithm 1** Descoberta e validação de instâncias SAPL (visão conceitual)

---

**Entrada:** Lista de municípios  $M$  (IBGE)

**Saída:** Lista de instâncias SAPL validadas  $S$

```
1:  $S \leftarrow \emptyset$ 
2: for cada município  $m \in M$  do
3:    $C \leftarrow$  gerar candidatos de URL para  $m$  (heurísticas)
4:   for cada candidato  $c \in C$  do
5:     if rota pública de consulta responde e conteúdo
       é compatível com SAPL then
6:       adicionar  $c$  em  $S$ 
7:     break ▷ evita duplicidade para o mesmo
       município
8:   end if
9: end for
10: end for
11: return  $S$ 
```

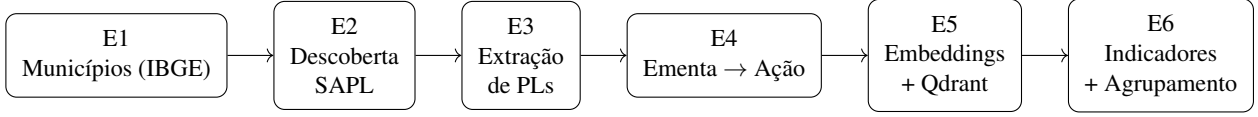
---

### 3.2.3 Estrutura conceitual da API do SAPL

Quando exposta, a API tende a seguir REST (FIELDING, 2000), com endpoints de coleções e itens. Em várias instâncias, a descrição OpenAPI e a interface Swagger UI podem estar disponíveis (OPENAPI INITIATIVE, 2024; SWAGGER, 2026).

### 3.2.4 Extração de Projetos de Lei

A extração foi desenhada para ser adaptável ao **vocabulário local**. O método consulta o catálogo de tipos de matéria e identifica quais rótulos correspondem a “Projeto de Lei”. Em seguida, percorre todas as páginas até o esgotamento (paginação) (DJANGO REST FRAMEWORK, 2026).



**Figura 1:** Pipeline conceitual do Sonar Municipal.

**Algorithm 2** Extração exaustiva de PLs via API (visão conceitual)

**Entrada:** Instância SAPL validada  $s$

**Saída:** Conjunto de PLs  $P$  extraídos de  $s$

```

1:  $P \leftarrow \emptyset$ 
2:  $T \leftarrow$  consultar catálogo de tipos de matéria em  $s$ 
3:  $T_{pl} \leftarrow$  selecionar tipos que correspondem a “Projeto de Lei”
4: for cada tipo  $t \in T_{pl}$  do
5:    $page \leftarrow 1$ 
6:   while há resultados na página  $page$  do
7:      $R \leftarrow$  requisitar lista paginada de matérias do tipo  $t$  na página  $page$ 
8:      $P \leftarrow P \cup R$ 
9:      $page \leftarrow page + 1$ 
10:  end while
11: end for
12: return  $P$ 
  
```

### 3.2.5 Resultado da primeira execução e customizações locais

Na primeira execução dos coletores, foram encontradas 1.259 instâncias SAPL; porém, a extração automatizada de PLs foi possível em 322 instâncias, totalizando 220.065 PLs coletados. A discrepância é explicada por customizações locais (mudanças de rotas, proxies, autenticação, regras anti-robô e parametrizações).

### 3.2.6 Rastreabilidade

Para auditoria, cada registro mantém proveniência mínima: instância de origem, identificadores e carimbo temporal. O PROV-DM formaliza esse conceito (W3C, 2013).

## 3.3 Construção do tradutor de ementas para ações

Ementas têm variação alta e linguagem abreviada. Para reduzir ruído na busca e melhorar a leitura do resultado, o projeto traduz ementas para uma **recomendação de ação**, mantendo a semântica.

Foi criado um dataset de pares *ementa*  $\rightarrow$  *ação* com 1.000 amostras sintéticas, geradas por um LLM da família GPT-5.1 (OPENAI, 2026, 2025). O objetivo é transferência de aprendizado, isto é, aproveitar um modelo já treinado para aprender a tarefa com menos exemplos reais.

O modelo tradutor foi treinado como tarefa **seq2seq** (sequência para sequência). O ajuste foi feito com QLoRA-4bit

(DETTMERS et al., 2023). A qualidade foi avaliada por BERTScore (ZHANG et al., 2019), obtendo 84% na avaliação interna do projeto.

Após o treinamento, todas as ementas coletadas foram convertidas em ações, formando um domínio fechado de ações recomendáveis.

## 3.4 Codificação das ações para busca semântica

Para indexar ações e consultas no mesmo espaço vetorial, foi utilizado o Multilingual E5 (WANG et al., 2024), adequado ao cenário de pergunta e resposta. As embeddings foram armazenadas no Qdrant (QDRANT, 2026b,a), permitindo recuperar as top- $K$  ações mais similares a uma pergunta.

Para viabilizar hospedagem gratuita do serviço de embeddings, uma opção prática é usar Text Embeddings Inference (TEI) (HUGGING FACE, 2026).

## 3.5 Indicadores para simulação de efeito no mundo real

Indicadores em séries temporais foram construídos por município, usando fontes oficiais. No indicador educacional, por exemplo, utilizam-se microdados do Censo Escolar (INEP) (INEP, 2026). A comparação do indicador na data de apresentação do PL com valores meses ou anos depois produz uma estimativa operacional de variação, com horizonte escolhido pelo usuário.

**Observação:** essa estimativa é associativa, não causal. Ela descreve o que mudou depois, mas não prova que o PL causou a mudança.

## 3.6 Agrupamento de projetos em políticas públicas

Para aumentar robustez, PLs similares são agrupados em clusters chamados “políticas públicas”. A análise conjunta permite calcular efeito médio e dispersão. Para ranquear políticas, foi definida a métrica:

$$Q = \frac{n_{\text{positivos}}}{n} \times \frac{n}{n+1}$$

onde  $n_{\text{positivos}}$  é o número de municípios com efeito considerado positivo (de acordo com o objetivo do indicador) e  $n$  é o total de municípios no grupo.

### 3.7 Desenvolvimento da plataforma web

A plataforma foi desenvolvida em React com Next.js (VERCEL, 2026) e hospedada na Vercel (VERCEL, 2025). A arquitetura integra: interface web, serviço de embeddings e banco vetorial (Qdrant).

Funcionalidades principais:

1. recomendação de políticas públicas com estimativa de efeito esperado;
2. detalhamento de PLs (ementa, data, link do SAPL e gráfico do indicador);
3. busca semântica no acervo para apoiar ideação e comparação de propostas.

## 4 Resultados

**Tabela 1:** Resumo da primeira execução de descoberta e extração.

Métrica	Valor
Instâncias SAPL encontradas	1.259
Instâncias com extração bem-sucedida	322
Projetos de Lei coletados (total)	220.065

O tradutor ementa → ação atingiu BERTScore de 84% na avaliação interna, indicando preservação semântica adequada para uso prático.

## 5 Limitações e ameaças à validade

- disponibilidade operacional de instâncias (instabilidade e mudanças de URL);
- heterogeneidade e customizações locais que quebram padrões de extração;
- variação semântica de rótulos e tipos de proposição entre instâncias;
- dificuldade de atribuir variações de indicadores a um único PL;
- uso de dataset sintético pode introduzir vieses do gerador.

## 6 Conclusão

O Sonar Municipal integra coleta sistemática de PLs via SAPL, tradução de ementas para ações, busca semântica por embeddings, análise com indicadores e agrupamento de políticas públicas. O resultado é uma ferramenta que aproxima evidência legislativa e dados oficiais do fluxo de trabalho de gestores públicos, com foco em reprodutibilidade e auditoria.

## Referências

DETTMERS, Tim et al. QLoRA: Efficient Finetuning of Quantized LLMs. **arXiv**, 2023. Disponível em: <<https://arxiv.org/abs/2305.14314>>. Acesso em: 7 jan. 2026.

DJANGO REST FRAMEWORK. **Pagination**. 2026. Disponível em: <<https://www.django-rest-framework.org/api-guide/pagination/>>. Acesso em: 7 jan. 2026.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado) – University of California, Irvine. Disponível em: <<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>. Acesso em: 7 jan. 2026.

HUGGING FACE. **Text Embeddings Inference (TEI)**. 2026. Disponível em: <<https://huggingface.github.io/text-embeddings-inference/>>. Acesso em: 7 jan. 2026.

IBGE. **API de Localidades (documentação)**. 2026. Disponível em: <<https://servicodados.ibge.gov.br/api/docs/localidades>>. Acesso em: 7 jan. 2026.

INEP. **Censo Escolar: Microdados (dados abertos)**. 2026. Disponível em: <<https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados/censo-escolar>>. Acesso em: 7 jan. 2026.

INTERLEGIS. **SAPL: Sistema de Apoio ao Processo Legislativo (repositório oficial)**. 2026. Disponível em: <<https://github.com/interlegis/sapl>>. Acesso em: 7 jan. 2026.

OPENAI. **GPT-5.1 Instant and GPT-5.1 Thinking: System Card Addendum**. 2025. Disponível em: <<https://openai.com/index/gpt-5-system-card-addendum-gpt-5-1/>>. Acesso em: 7 jan. 2026.

\_\_\_\_\_. **GPT-5.1 Model (OpenAI API)**. 2026. Disponível em: <<https://platform.openai.com/docs/models/gpt-5.1>>. Acesso em: 7 jan. 2026.

OPENAPI INITIATIVE. **OpenAPI Specification (v3.2.0)**. 2024. Disponível em: <<https://spec.openapis.org/oas/v3.2.0.html>>. Acesso em: 7 jan. 2026.

QDRANT. **Collections (conceitos)**. 2026. Disponível em: <<https://qdrant.tech/documentation/concepts/collections/>>. Acesso em: 7 jan. 2026.

\_\_\_\_\_. **Qdrant Documentation**. 2026. Disponível em: <<https://qdrant.tech/documentation/>>. Acesso em: 7 jan. 2026.

SENADO FEDERAL; PROGRAMA INTERLEGIS.  
**Sistema de Apoio ao Processo Legislativo (SAPL).** 2026.  
Disponível em: <<https://www12.senado.leg.br/interlegis/produtos/sapl>>. Acesso em: 7 jan. 2026.

SWAGGER. **Swagger UI: REST API Documentation Tool.** 2026. Disponível em:  
<<https://swagger.io/tools/swagger-ui/>>.  
Acesso em: 7 jan. 2026.

VERCEL. **Next.js Documentation.** 2026. Disponível em:  
<<https://nextjs.org/docs>>. Acesso em: 7 jan. 2026.

\_\_\_\_\_. **Next.js on Vercel (documentação).** 2025.  
Disponível em: <<https://vercel.com/docs/frameworks/full-stack/nextjs>>. Acesso em: 7 jan. 2026.

W3C. **PROV-DM: The PROV Data Model.** 2013.  
Disponível em:  
<<https://www.w3.org/TR/prov-dm/>>. Acesso em: 7 jan. 2026.

WANG, Liang et al. Multilingual E5 Text Embeddings: A Technical Report. **arXiv**, 2024. Disponível em:  
<<https://arxiv.org/abs/2402.05672>>.  
Acesso em: 7 jan. 2026.

WILKINSON, Mark D.; DUMONTIER, Michel;  
AALBERSBERG, IJsbrand Jan et al. The FAIR Guiding Principles for scientific data management and stewardship. **Scientific Data**, v. 3, p. 160018, 2016. DOI: 10.1038/sdata.2016.18.

ZHANG, Tianyi et al. BERTScore: Evaluating Text Generation with BERT. **arXiv**, 2019. Disponível em:  
<<https://arxiv.org/abs/1904.09675>>.  
Acesso em: 7 jan. 2026.