



# Tô De Olho: Democratizando a Transparência do Senado Federal através de Dados Abertos

## Trabalho de Conclusão de Curso

Pedro Batista de Almeida Filho

Pablo Vieira Florentino  
Orientador

Instituto Federal da Bahia - IFBA  
Curso de Análise e Desenvolvimento de Sistemas  
Campus Salvador

Salvador, Bahia, Brasil  
Janeiro de 2026

# Resumo

O presente trabalho propõe o desenvolvimento do *Tô De Olho*, uma plataforma *web* de transparência parlamentar focada no Senado Federal. O Senado, apesar de sua relevância legislativa, carece de ferramentas de fiscalização cidadã integradas que facilitem o controle social, contrastando com o ecossistema consolidado para a Câmara dos Deputados. A solução adota uma arquitetura de monolito modular em linguagem Go, com interface implementada em Next.js, e integra dados de três APIs governamentais (Senado Legislativo, Senado Administrativo e Portal da Transparência) para consolidar informações sobre atividade legislativa, execução da Cota para o Exercício da Atividade Parlamentar (CEAPS) e alocação de emendas parlamentares. Como diferencial metodológico, apresenta-se um Índice de Efetividade Legislativa adaptado ao contexto brasileiro, que pondera produtividade, presença em plenário, economia de recursos e participação em comissões. Os resultados demonstram a viabilidade técnica da unificação de fontes de dados heterogêneas e o potencial da plataforma em reduzir a assimetria informacional, transformando transparência passiva em *accountability* ativa e qualificando o debate público para o cenário eleitoral de 2026. A plataforma adota premissas de acessibilidade (WCAG) e design *mobile-first* para democratizar o acesso conforme o perfil da população brasileira.

**Palavras-chaves:** Transparência Pública. Senado Federal. Dados Abertos. Accountability. Engenharia de Software.

# Abstract

This work proposes the development of *Tô De Olho*, a web platform for parliamentary transparency focused on the Federal Senate. The Senate, despite its legislative importance, lacks integrated citizen oversight tools to facilitate social control, in contrast to the consolidated ecosystem for the Chamber of Deputies. The solution adopts a modular monolith architecture in Go, with an interface implemented in Next.js, and integrates data from three government APIs (Legislative Senate, Administrative Senate, and Transparency Portal) to consolidate information on legislative activity, execution of the Quota for the Exercise of Parliamentary Activity (CEAPS), and parliamentary amendments allocation. As a methodological differentiator, a Legislative Effectiveness Index adapted to the Brazilian context is presented, weighting productivity, floor attendance, resource economy, and participation in committees. The results demonstrate the technical feasibility of unifying heterogeneous data sources and the platform's potential to reduce informational asymmetry, transforming passive transparency into active accountability and qualifying public debate for the 2026 electoral scenario. The platform adopts accessibility premises (WCAG) and mobile-first design to democratize access according to the Brazilian population profile.

**Keywords:** Public Transparency. Federal Senate. Open Data. Accountability. Software Engineering.

## Lista de Figuras

1	Arquitetura de implantação do sistema Tô De Olho . . . . .	28
2	Diagrama entidade-relacionamento do sistema Tô De Olho . . . . .	30
3	Página Inicial com Rankings e Destaques . . . . .	35
4	Recursos de Busca e Navegação por Mapa . . . . .	36
5	Ficha Parlamentar: Visão Geral e Gráficos . . . . .	37

# Lista de Tabelas

1	Cr�terios do algoritmo de ranking . . . . .	22
---	---	----

# Sumário

<b>1</b>	<b>Visão geral</b>	<b>1</b>
1.1	Objetivos	1
1.1.1	Objetivo Geral	1
1.1.2	Objetivos Específicos	1
1.2	Declaração do Problema	2
1.3	Proposta de Solução de Software	2
1.4	Trabalhos Relacionados	3
1.4.1	Portais Oficiais (Nível Informação)	3
1.4.2	Ferramentas de Fiscalização da Câmara dos Deputados	4
1.4.3	Experiências Internacionais	6
1.4.4	Lacuna Identificada e Diferencial do Tô De Olho	6
<b>2</b>	<b>Fundamentação Teórica</b>	<b>8</b>
2.1	Eixo Normativo-Institucional	8
2.1.1	Transparência Pública e Dados Abertos	8
2.1.2	Emendas PIX e Desafios de Transparência Orçamentária	10
2.2	Eixo Político-Participativo	10
2.2.1	Democracia Digital e Participação Cidadã	10
2.2.2	Teoria Principal-Agente e Accountability	11
2.2.3	Civic Tech e Sociedade Civil	13
2.3	Eixo Técnico-Metodológico	14
2.3.1	Métricas de Efetividade Legislativa	14
2.3.2	Visualização de Dados e Retórica Visual	15
2.3.3	Arquitetura de Software: Monolito Modular	16
2.3.4	Engenharia de Dados: APIs e Processos ETL	17
<b>3</b>	<b>Metodologia</b>	<b>18</b>
3.1	Abordagem de Desenvolvimento	19
3.2	Fontes de Dados	20
3.2.1	API Legislativa do Senado	20
3.2.2	API Administrativa do Senado	20
3.2.3	Portal da Transparência (CGU)	21
3.2.4	Limitações e Cobertura Temporal	21
3.3	Metodologia de Avaliação de Desempenho	21
3.3.1	Produtividade Legislativa	22
3.3.2	Presença em Votações	22
3.3.3	Economia na Cota Parlamentar	22
3.3.4	Participação em Comissões	22
3.3.5	Fórmula Final e Tratamento de Casos Especiais	23
<b>4</b>	<b>Requisitos</b>	<b>23</b>
4.1	Requisitos Funcionais	23
4.2	Requisitos Não-Funcionais	25

<b>5</b>	<b>Design</b>	<b>26</b>
5.1	Arquitetura do Sistema . . . . .	26
5.1.1	Organização em Módulos . . . . .	27
5.1.2	Visão de Implantação . . . . .	28
5.2	Stack Tecnológico . . . . .	28
5.2.1	Backend em Go . . . . .	28
5.2.2	Camada de Persistência . . . . .	29
5.2.3	Frontend em Next.js . . . . .	29
5.3	Estratégia de Ingestão de Dados . . . . .	29
5.4	Modelo de Dados . . . . .	30
<b>6</b>	<b>Testes de Software</b>	<b>31</b>
6.1	Testes Unitários . . . . .	31
6.2	Testes de Integração . . . . .	31
6.3	Testes de Contrato . . . . .	32
<b>7</b>	<b>Implantação e Infraestrutura</b>	<b>32</b>
7.1	Containerização . . . . .	32
7.2	Pipeline de Integração e Entrega Contínua (CI/CD) . . . . .	33
7.3	Infraestrutura Serverless no Google Cloud . . . . .	33
7.4	Estratégia Mobile-First e Acessibilidade . . . . .	34
<b>8</b>	<b>Manual do Usuário Simplificado</b>	<b>34</b>
8.1	Acesso e Visão Geral . . . . .	34
8.2	Análise do Perfil Parlamentar . . . . .	36
8.3	Ferramentas de Compartilhamento . . . . .	37
<b>9</b>	<b>Considerações Finais</b>	<b>38</b>



# 1 Visão geral

O *Tô De Olho* é uma plataforma *web* de transparência parlamentar focada no Senado Federal. Sua proposta é aproximar cidadãos dos dados legislativos oficiais, convertendo informação dispersa em conhecimento fiscalizável e de fácil compreensão. O projeto vai além dos dados abertos básicos, integrando fontes complexas como a Cota para o Exercício da Atividade Parlamentar dos Senadores (CEAPS) e as “emendas PIX”. Ao combinar uma arquitetura de **monolito modular** em Go, ingestão via APIs oficiais (Senado e Portal da Transparência) e um *front-end* em Next.js, a plataforma busca reduzir a assimetria de informação sobre os 81 senadores da República [1].

A literatura evidencia que TICs ampliam possibilidades de participação, mas só geram valor quando articuladas a contextos de uso. Avelino et al. mapeiam iniciativas, reforçando que tecnologias exigem visualizações para o controle social [2]. Com um corpo legislativo menor e mais “caro” *per capita* que a Câmara, o Senado carece de ferramentas que cruzem votações com a execução orçamentária. À luz desses estudos, o *Tô De Olho* procura transformar transparência passiva em *accountability* ativa [3, 4].

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Desenvolver uma plataforma *web* de transparência política que centralize, organize e simplifique o acesso aos dados públicos do Senado Federal, fomentando a fiscalização cidadã e o debate qualificado sobre a atuação dos 81 senadores, com ênfase no monitoramento de gastos e emendas parlamentares.

### 1.1.2 Objetivos Específicos

- Implementar um **backend em Go** com arquitetura de monolito modular para ingestão de dados das APIs oficiais do Senado e do Portal da Transparência;
- Desenvolver rotinas ETL para consumir as APIs Legislativa, Administrativa e do Portal da Transparência, priorizando fontes estruturadas;
- Criar algoritmos de *Ranking* para avaliar senadores com base em presença em votações, produtividade legislativa, economia na cota parlamentar, participação em comissões e transparência de dados;
- Construir uma interface *front-end* responsiva utilizando Next.js, permitindo a visualização intuitiva de perfis, despesas e *scorecards* de fiscalização;
- Garantir a acessibilidade e a usabilidade em dispositivos móveis seguindo diretrizes WCAG e abordagem *mobile-first*, considerando o perfil de acesso à internet da população brasileira.

## 1.2 Declaração do Problema

O Senado Federal disponibiliza dados públicos por meio de APIs próprias, enquanto a Controladoria-Geral da União (CGU) mantém o Portal da Transparência com dados de emendas parlamentares. Contudo, essas fontes encontram-se fragmentadas em órgãos distintos: a API Legislativa do Senado concentra informações sobre matérias e votações; a API Administrativa do Senado reúne dados da CEAPS e remunerações de gabinete; e o Portal da Transparência da CGU hospeda os registros de emendas e transferências federais. Para construir uma visão completa de um único senador, o cidadão precisaria consultar três sistemas de dois órgãos diferentes, com interfaces, formatos e periodicidades de atualização distintos.

Essa fragmentação adquire contornos mais graves quando analisamos as “Transferências Especiais” — popularmente conhecidas como “emendas PIX”. Criada em 2019, essa modalidade dispensa convênio e transfere recursos federais diretamente a estados e municípios. Alencar [5] demonstra que, do total de R\$ 20,5 bilhões transferidos por essa via, apenas R\$ 933 milhões tiveram prestação de contas adequada — menos de 5%. Em 2020, primeiro ano de vigência, as transferências especiais representavam 6,4% das emendas individuais; em 2023, esse percentual saltou para 32,4%. A distribuição é ainda mais desigual: no mesmo estado, alguns municípios receberam mais de R\$ 4.500 *per capita*, enquanto outros receberam menos de R\$ 1 — sem qualquer justificativa pública dos parlamentares.

Além da barreira técnica imposta pela fragmentação dos dados, há uma barreira social igualmente relevante. Segundo o Indicador de Alfabetismo Funcional [6], 29% da população brasileira entre 15 e 64 anos é funcionalmente analfabeta, o que limita severamente a capacidade de interpretar planilhas, gráficos e relatórios disponibilizados nos portais oficiais. Nesse contexto, a simples disponibilização de dados brutos não garante transparência efetiva: é necessária uma ferramenta que consolide as informações dispersas e as apresente de forma visual e acessível, permitindo ao cidadão comum avaliar qualitativamente seus representantes [2].

## 1.3 Proposta de Solução de Software

Diante da fragmentação de dados descrita e da barreira de letramento que impede o cidadão comum de interpretar planilhas e relatórios oficiais, propõe-se o *Tô De Olho*: uma plataforma *web* de código aberto concebida para centralizar a fiscalização do Senado Federal. A solução integra três APIs oficiais distintas — Legislativa do Senado, Administrativa do Senado e Portal da Transparência da CGU — consolidando informações dispersas em uma interface única e acessível.

O sistema organiza os dados em três dimensões complementares do mandato parlamentar:

- **Atividade Legislativa:** votações nominais, participação em comissões, proposições de autoria e relatorias;
- **Gestão de Recursos:** despesas detalhadas da Cota Parlamentar (CEAPS), incluindo identificação de fornecedores e categorias de gasto;

- **Articulação Orçamentária:** emendas parlamentares com destaque para Transferências Especiais (“emendas PIX”), permitindo rastrear o destino dos recursos.

O diferencial da plataforma reside em quatro pilares:

1. **Ranking Metodologicamente Fundamentado:** inspirado no *State Legislative Effectiveness Score* (SLES) de Volden e Wiseman [7], o algoritmo de avaliação pondera produtividade legislativa (35%), presença em votações (25%), economia na cota parlamentar (20%) e participação em comissões (20%). Os critérios e pesos são públicos, permitindo ao cidadão compreender — e questionar — a metodologia;
2. **Visualização Orientada à Ação:** seguindo os princípios de retórica visual de Hullman [8], cada dado absoluto é contextualizado com médias comparativas, reduzindo a possibilidade de interpretações manipuladas e estimulando conclusões informadas;
3. **Acessibilidade como Requisito:** a interface segue as diretrizes WCAG 2.1 nível AA, garantindo navegação por leitores de tela, contraste adequado e operação via teclado — essencial para atingir os 29% de brasileiros funcionalmente analfabetos identificados pelo INAF [6];
4. **Consolidação Multi-Fonte:** ao integrar dados de três órgãos distintos em uma única consulta, a plataforma elimina a necessidade de o cidadão navegar por sistemas heterogêneos com formatos e interfaces incompatíveis.

Em síntese, o *Tô De Olho* atua como um “auditor digital”, automatizando cruzamentos de dados que, manualmente, seriam inviáveis para o eleitor comum. O objetivo não é substituir a análise crítica do cidadão, mas fornecer-lhe ferramentas para exercê-la de forma qualificada.

## 1.4 Trabalhos Relacionados

Diversas iniciativas no Brasil e no mundo buscam promover a transparência política por meio da tecnologia. À luz da Escada de Participação de Arnstein [9], podemos classificar essas ferramentas conforme o grau de poder que conferem ao cidadão.

### 1.4.1 Portais Oficiais (Nível Informação)

**Portal da Transparência (CGU):** Lançado em novembro de 2004 pela Controladoria-Geral da União, o Portal da Transparência consolidou-se como a principal ferramenta oficial do governo federal para acesso a dados de gastos públicos, servidores e transferências [10]. Em 2018, o portal passou por reformulação completa para tornar a navegação mais intuitiva, e em 2024, ao completar 20 anos, recebeu novas atualizações que reafirmaram seu papel central no controle social. Os dados disponíveis abrangem execução orçamentária detalhada por órgão, remuneração individualizada de servidores, pagamentos de programas sociais (Bolsa Família, Auxílio Gás, Pé-de-Meia), licitações, contratos e — particularmente relevante para este trabalho — emendas parlamentares, incluindo registros relacionados à ADPF 854 sobre transparência orçamentária.

O portal registra entre 1,3 e 1,5 milhão de usuários únicos mensais, com aproximadamente

14 a 19 milhões de visualizações de página, demonstrando alto engajamento da sociedade civil e órgãos de controle. Para desenvolvedores, oferece uma API REST com limites de 90 requisições por minuto em horário comercial e 300 requisições por minuto durante a madrugada. Contudo, algumas limitações persistem: a periodicidade de atualização varia significativamente entre conjuntos de dados — enquanto despesas e emendas são atualizadas diariamente, dados de imóveis funcionais podem apresentar defasagem superior a seis meses. Além disso, a granularidade contábil (com termos técnicos do Siafi — Sistema Integrado de Administração Financeira do Governo Federal) representa barreira para cidadãos sem conhecimento em contabilidade pública.

**Portal de Dados Abertos do Senado Federal:** Lançado em 2012 em conformidade com a Lei de Acesso à Informação [11], o portal foi institucionalizado pelo Ato da Comissão Diretora n. 14 de 2013, que estabeleceu a Política de Dados Abertos do Senado [12]. O ecossistema divide-se em duas APIs principais: a **API Legislativa**, que oferece dados sobre matérias, votações nominais, senadores e atividades de comissões; e a **API Administrativa**, focada em transparência de gastos (CEAPS), gestão de pessoas, orçamento e contratos. Os formatos suportados incluem JSON, XML e CSV, com documentação técnica via Swagger UI. A API possui limite de 10 requisições por segundo para garantir estabilidade.

**Portal de Dados Abertos da Câmara dos Deputados:** O fornecimento de dados legislativos pela Câmara iniciou-se em 2006 através do sistema SIT Câmara (Web Services) [13]. Com a Lei de Acesso à Informação em 2011, o portal foi rebatizado como “Dados Abertos”, eliminando a obrigatoriedade de cadastro prévio. Em 2017, lançou-se a API RESTful v2, substituindo os antigos Web Services por arquitetura mais moderna. O portal oferece *endpoints* para deputados (perfis biográficos, discursos, frentes parlamentares), proposições (texto integral, tramitação), votações (incluindo voto individual de cada parlamentar) e cotas parlamentares (CEAP). Os formatos incluem JSON e XML via API, além de CSV, XLSX e ODS para *downloads* em massa. O portal é referência em transparência legislativa, alimentando projetos como a Operação Serenata de Amor, Radar Governamental e VotoBom.

Embora esses portais representem avanços significativos na transparência passiva, situam-se no degrau mais básico da Escada de Arnstein — informação bruta sem mediação interpretativa. O cidadão comum, sem conhecimento técnico sobre APIs ou contabilidade pública, enfrenta barreiras substanciais para transformar dados dispersos em fiscalização efetiva.

## 1.4.2 Ferramentas de Fiscalização da Câmara dos Deputados

O ecossistema de transparência para a Câmara dos Deputados é mais desenvolvido que para o Senado, contando com diversas iniciativas consolidadas:

**Operação Serenata de Amor:** Projeto pioneiro de código aberto, lançado em 2016 via financiamento coletivo, que utiliza inteligência artificial para detectar irregularidades em gastos parlamentares [14]. Desenvolvido pela Open Knowledge Brasil, é composto por dois sistemas complementares:

- **Rosie:** Algoritmo de *machine learning* desenvolvido em Python que audita a Cota para Exercício da Atividade Parlamentar (CEAP). Opera através de cinco classificadores

principais: (1) *meal price outlier* — identifica refeições com valores acima da média para o local; (2) *irregular companies* — detecta gastos em empresas com situação cadastral irregular na Receita Federal; (3) *traveled speeds* — cruza gastos para identificar deslocamentos fisicamente impossíveis; (4) *monthly subquota limit* — verifica excesso nos limites mensais por categoria; e (5) *election expenses* — identifica uso indevido da cota para financiar campanhas.

- **Jarbas:** Interface *web* desenvolvida em Django que permite aos cidadãos navegar pelos casos suspeitos identificados pela Rosie, visualizar notas fiscais digitalizadas e formalizar denúncias.

Até 2018, o projeto identificou **8.276 reembolsos suspeitos** envolvendo 735 deputados, totalizando aproximadamente **R\$ 3,6 milhões** em potenciais irregularidades. Um mutirão inicial resultou em 629 denúncias formais ao Congresso. Atualmente, a equipe principal migrou o foco para o projeto “Querido Diário”, que aplica técnicas similares a diários oficiais municipais.

**De Olho no Congresso:** Plataforma *web* moderna focada em gastos de Deputados Federais [15]. Com mais de **55 mil visitantes** e **95 mil consultas** realizadas, a ferramenta oferece interface acessível que inclui:

- **Rankings Múltiplos:** Top 50 deputados com maiores gastos, ranking de partidos por consumo da cota e ranking de empresas fornecedoras — incluindo filtro específico para “empresas com sanções” administrativas;
- **Painel de Alertas:** Sistema de detecção automática de despesas atípicas, incluindo: valores significativamente acima da média geral, pagamentos idênticos repetidos ao mesmo fornecedor, notas fiscais emitidas em finais de semana, e intervalos menores que 3 dias entre pagamentos;
- **Histórico Completo:** Gastos anuais e mensais com filtros por fornecedor, categoria e período, além de detalhamento de benefícios (auxílio-moradia, imóvel funcional) e equipe de gabinete.

A plataforma ressalta que os alertas são “indicativos que merecem investigação”, servindo como guia para auditoria cidadã. Limita-se, porém, à Câmara dos Deputados e não oferece métricas de desempenho legislativo.

**De Olho em Você:** Plataforma *web* com foco em “transparência que dá para entender”, abrangendo aproximadamente 549 parlamentares da Câmara dos Deputados [16]. A plataforma integra dados da API da Câmara e do Portal da Transparência, destacando-se pela cobertura das **Emendas PIX** (Transferências Especiais). Entre suas funcionalidades principais:

- **Mapas de Distribuição:** Cada perfil de deputado apresenta visualização geoespacial do destino de suas emendas parlamentares, permitindo identificar concentração de recursos por município;
- **Ranking de Cidades:** Classificação por faixa populacional das cidades que mais receberam Transferências Especiais (ex: municípios de até 20 mil habitantes);

- **Comparador de Parlamentares:** Ferramenta que permite selecionar de 2 a 5 deputados para comparação lado a lado de gastos de cota, equipe de gabinete, emendas enviadas e fornecedores em comum;
- **Painel de Fornecedores:** Ranking das empresas que mais recebem recursos, identificando padrões de concentração de gastos.

Entretanto, os rankings do “De Olho em Você” baseiam-se em métricas agregadas diretas (quem mais gastou, quem mais enviou emendas), **sem metodologia explícita de efetividade legislativa**. Além disso, a plataforma não contempla o Senado Federal.

### 1.4.3 Experiências Internacionais

**TheyWorkForYou (Reino Unido):** Lançada em 2004 e operada pela organização sem fins lucrativos mySociety [17], a plataforma monitora cinco parlamentos britânicos: Câmara dos Comuns, Câmara dos Lordes, Parlamento Escocês, Senedd (País de Gales) e Assembleia da Irlanda do Norte [18]. Em 2023/24, registrou mais de **4,8 milhões de visitas**. Entre suas funcionalidades:

- **Hansard Pesquisável:** Arquivo completo de todos os discursos e debates parlamentares;
- **Alertas por E-mail:** Notificações automáticas quando um parlamentar específico discursa ou quando uma palavra-chave é mencionada;
- **Busca por Código Postal:** Identificação imediata do representante local.

A plataforma consolidou-se como referência mundial em *civic tech* parlamentar, inspirando iniciativas em diversos países.

**OpenSecrets (Estados Unidos):** Principal organização de pesquisa sobre dinheiro na política americana, resultante da fusão em 2021 entre o *Center for Responsive Politics* (fundado em 1983 por dois ex-senadores) e o *National Institute on Money in State Politics* [19]. Vencedora de múltiplos *Webby Awards*, a plataforma rastreia:

- **Financiamento de Campanhas:** Contribuições individuais, PACs e Super PACs;
- **Lobbying:** Gastos de empresas e grupos de interesse para influenciar legislação;
- **Revolving Door:** Monitoramento de ex-congressistas que se tornaram lobistas;
- **Dark Money:** Análise de fundos de origem não divulgada que influenciam eleições.

É fonte primária para veículos como *The New York Times* e *The Washington Post*, oferecendo APIs e exportações de dados para pesquisadores acadêmicos.

### 1.4.4 Lacuna Identificada e Diferencial do Tô De Olho

A análise sistemática dos trabalhos relacionados evidencia um cenário paradoxal: enquanto a Câmara dos Deputados — com 513 parlamentares — dispõe de ao menos três plataformas consolidadas de fiscalização cidadã, o Senado Federal permanece como uma

“caixa preta” digital. Essa lacuna não é trivial. Os 81 senadores exercem mandatos de oito anos, atuam como câmara revisora de toda legislação federal e detêm competências exclusivas de alto impacto: confirmação de ministros do STF, julgamento de presidentes da República e aprovação de dívidas externas. A ausência de ferramentas de monitoramento específicas representa, portanto, uma falha sistêmica no ecossistema de *accountability* brasileiro. Adicionalmente, com a aproximação das eleições gerais de 2026, torna-se imperativo oferecer ao eleitorado ferramentas que permitam uma análise retrospectiva e fundamentada do mandato dos senadores que buscarão a reeleição ou outros cargos majoritários.

Mais do que apenas replicar soluções existentes para o âmbito senatorial, o *Tô De Olho* propõe-se a **synthetizar o melhor de cada iniciativa** analisada, superando limitações identificadas:

- Do “**De Olho em Você**”, incorporamos a **visualização geoespacial de Emendas PIX** — permitindo que o cidadão identifique, em mapas interativos, quais municípios receberam recursos de cada senador — e o **comparador de parlamentares**, que possibilita análise lado a lado de até cinco senadores em múltiplas dimensões;
- Do “**De Olho no Congresso**”, adotamos o **painel de alertas automáticos** para despesas atípicas — notas fiscais em finais de semana, valores acima da média, pagamentos repetidos em intervalos curtos — e o **ranking de fornecedores**, incluindo cruzamento com empresas sob sanção administrativa;
- Do “**Serenata de Amor**”, adotamos a filosofia de **código aberto** e documentação transparente.

O diferencial central do *Tô De Olho*, entretanto, reside em uma contribuição original: a implementação de um **Índice de Efetividade Legislativa** adaptado ao contexto brasileiro. Enquanto as ferramentas existentes limitam-se a ordenar parlamentares por *volume de gastos* — métrica que penaliza a parcimônia — ou *quantidade de proposições* — que ignora a qualidade e o impacto legislativo —, propomos um modelo multidimensional inspirado no *State Legislative Effectiveness Score* (SLES) de Volden e Wiseman [7].

Nosso índice pondera quatro dimensões objetivas, com pesos públicos e metodologia reprodutível:

1. **Produtividade Legislativa (35%)**: Avalia a capacidade de transformar proposições em leis, com multiplicadores por tipo (PEC: 3x, PLP: 2x) e estágio de tramitação alcançado;
2. **Presença em Votações (25%)**: Mensura o comparecimento efetivo às sessões deliberativas, descontando ausências justificadas por licença médica ou missão oficial;
3. **Economia na Cota Parlamentar (20%)**: Compara o uso individual da CEAPS com a mediana do Senado, premiando a eficiência no uso de recursos públicos;
4. **Participação em Comissões (20%)**: Pondera o engajamento em comissões permanentes e especiais, com bônus para cargos de liderança (presidente, relator).

A exposição pública de critérios e pesos não é mera formalidade: representa um compro-

misso ético com a **transparência metodológica**. Diferente de rankings opacos, o cidadão poderá compreender — e questionar — os fundamentos da classificação, evitando que a plataforma seja percebida como veículo de viés político.

Em síntese, o *Tô De Olho* posiciona-se como a **primeira plataforma integrada** de fiscalização cidadã voltada ao Senado Federal, combinando três vertentes:

1. **Consolidação Multi-Fonte:** Integra dados de três APIs oficiais (Legislativa, Administrativa e Portal da Transparência) em interface única;
2. **Inteligência de Dados:** Oferece alertas automáticos, rankings e visualizações que convertem dados brutos em informação acionável;
3. **Rigor Metodológico:** Fundamenta-se em literatura acadêmica sobre efetividade legislativa, com metodologia aberta a auditoria pública.

## 2 Fundamentação Teórica

O desenvolvimento de uma plataforma de transparência parlamentar situa-se na interseção de múltiplos campos do conhecimento: direito à informação, ciência política, teoria democrática, design de interação e engenharia de software. Compreender essas bases teóricas é essencial não apenas para fundamentar as decisões de projeto, mas também para posicionar a ferramenta no debate mais amplo sobre controle social e qualidade da democracia.

Esta seção organiza-se em três eixos complementares. O primeiro eixo — **normativo-institucional** — examina os marcos legais e conceituais que sustentam a transparência governamental no Brasil, desde a Lei de Acesso à Informação até os compromissos internacionais de governo aberto. O segundo eixo — **político-participativo** — aborda as teorias de democracia digital e accountability, investigando como tecnologias podem (ou não) fortalecer a participação cidadã e a responsabilização de agentes públicos. O terceiro eixo — **técnico-metodológico** — apresenta os fundamentos de mensuração de efetividade legislativa, visualização de dados e arquitetura de sistemas que orientam a implementação do *Tô De Olho*.

### 2.1 Eixo Normativo-Institucional

Este eixo examina os fundamentos jurídicos e conceituais que regulamentam o acesso à informação pública no Brasil, bem como os desafios de transparência em modalidades orçamentárias específicas.

#### 2.1.1 Transparência Pública e Dados Abertos

O sistema integra três fontes de dados oficiais, fundamentadas no arcabouço legal brasileiro de transparência pública. A Lei de Acesso à Informação (Lei n. 12.527/2011) estabelece como diretriz a “disponibilização de informações em formatos abertos, estruturados e legíveis por máquina” [11], princípio que as APIs governamentais operacionalizam — como exemplificado pelo Portal de Dados Abertos do Senado [12]. Conforme demonstrado pela Operação Serenata de Amor, tecnologias desenvolvidas sobre esses dados abertos podem gerar valor público ao



facilitar o controle social do gasto parlamentar [14]. A LAI impõe aos órgãos públicos o dever de divulgação proativa de informações de interesse coletivo, incluindo dados sobre despesas, contratos e remunerações de servidores.

A literatura distingue duas modalidades complementares de transparência. A **transparência ativa** ocorre quando o Estado disponibiliza informações de forma proativa em portais e bases de dados, independentemente de solicitação — exemplificada pelos Portais de Dados Abertos do Senado Federal e da Câmara dos Deputados. A **transparência passiva**, por sua vez, responde às solicitações dos cidadãos via canais específicos como o Sistema Eletrônico do Serviço de Informação ao Cidadão (e-SIC). Enquanto a primeira amplia o acesso massivo a dados estruturados, a segunda garante o direito individual à informação específica [2]. Pesquisas de Michener *et al.* demonstram que, cinco anos após a promulgação da LAI, a transparência permanece significativamente mais fraca nos níveis estadual e municipal, com índices de compliance inferiores a 50% em muitos entes federativos [20].

O conceito de *Open Government Data* (Dados Governamentais Abertos) vai além da simples disponibilização: preconiza que as informações públicas devem ser liberadas em formatos abertos, processáveis por máquina e livres de licenças restritivas. Tim O'Reilly, em ensaio seminal que cunhou o termo “Government as a Platform”, argumenta que governos devem atuar como **plataformas** que habilitam a inovação cidadã, em vez de provedores diretos de serviços isolados [21]. Nessa visão, APIs de dados abertos funcionam como infraestrutura sobre a qual a sociedade civil constrói ferramentas de fiscalização — exatamente o que o *Tô De Olho* realiza. Tim Berners-Lee, inventor da *World Wide Web*, complementa essa perspectiva ao propor em 2010 uma escala de cinco estrelas para avaliar a qualidade dos dados abertos [22]:

1. **Uma estrela:** Dados disponíveis na web em qualquer formato, sob licença aberta (ex: PDF escaneado);
2. **Duas estrelas:** Dados estruturados e legíveis por máquina (ex: planilha Excel);
3. **Três estrelas:** Dados em formato aberto não-proprietário (ex: CSV ao invés de Excel);
4. **Quatro estrelas:** Dados identificados por URIs, seguindo padrões W3C como RDF;
5. **Cinco estrelas:** Dados linkados formando rede interoperável (*Linked Open Data*).

Os portais brasileiros de dados abertos situam-se predominantemente entre duas e três estrelas: oferecem arquivos CSV e JSON processáveis por máquina, mas raramente implementam identificadores únicos (URIs) ou linkagem semântica entre bases de dados de diferentes órgãos. A fragmentação identificada na Declaração do Problema — três APIs de dois órgãos distintos — exemplifica essa limitação: embora os dados sejam tecnicamente “abertos”, a ausência de interoperabilidade impõe barreiras significativas à consolidação e análise integrada [2].

O Brasil integra a *Open Government Partnership* (OGP) desde 2011, tendo desenvolvido seis planos de ação nacionais com participação da sociedade civil. Essas iniciativas resultaram em 130 reformas voltadas à melhoria da governança e ao fortalecimento da Lei de Acesso

à Informação [23]. Contudo, como demonstram as pesquisas sobre dados abertos, a mera disponibilização de informações não garante transparência efetiva: é necessário que os dados alcancem o público, que este tenha capacidade de processá-los, e que existam mecanismos institucionais para responsabilização dos agentes públicos.

### 2.1.2 Emendas PIX e Desafios de Transparência Orçamentária

As Transferências Especiais, popularmente conhecidas como “emendas PIX”, constituem modalidade de repasse de recursos federais a estados e municípios criada pela Emenda Constitucional nº 105/2019. Diferentemente de convênios tradicionais, que exigem plano de trabalho prévio e prestação de contas detalhada, as emendas PIX transferem recursos diretamente às contas dos entes federativos com discricionariedade ampla sobre sua aplicação.

A pesquisa de Alencar [5] revela deficiências graves de transparência fiscal nesta modalidade. Do total de R\$ 20,5 bilhões transferidos via emendas PIX até 2023, apenas R\$ 933 milhões — menos de 5% — tiveram prestação de contas adequada. A evolução do instrumento é expressiva: em 2020, primeiro ano de vigência, as transferências especiais representavam 6,4% das emendas individuais; em 2023, esse percentual saltou para 32,4%.

A distribuição territorial revela disparidades extremas: no mesmo estado, alguns municípios receberam mais de R\$ 4.500 *per capita* via emendas PIX, enquanto outros receberam menos de R\$ 1 — sem qualquer justificativa pública dos parlamentares sobre os critérios de alocação. Esta opacidade compromete as três vertentes de *accountability* discutidas posteriormente nesta seção: a **vertical** (eleitores não conseguem avaliar escolhas de seus representantes), a **horizontal** (tribunais de contas enfrentam dificuldades de fiscalização) e a **social** (jornalistas e pesquisadores encontram dados fragmentados e incompletos).

O *Tô De Olho* aborda esta lacuna ao integrar dados de emendas do Portal da Transparência com informações legislativas do Senado, permitindo que o cidadão visualize, para cada senador: o total de recursos destinados via transferências especiais, os municípios beneficiados e a evolução temporal dos repasses. Ao consolidar informações dispersas em interface única, a plataforma contribui para reduzir a opacidade que caracteriza esta modalidade orçamentária.

## 2.2 Eixo Político-Participativo

Este eixo investiga as bases teóricas que fundamentam a relação entre tecnologia, participação cidadã e responsabilização de agentes públicos. Examina-se como ferramentas digitais podem fortalecer — ou apenas simular — o controle social.

### 2.2.1 Democracia Digital e Participação Cidadã

O conceito de democracia digital refere-se ao emprego de tecnologias de informação e comunicação (TICs) para produzir “mais democracia e melhores democracias” [1]. Wilson Gomes identifica três fases históricas neste campo: a **teledemocracia** (anos 1970-90), marcada por experimentos com televisão interativa e enquetes eletrônicas; a **fase da internet** (1995-2005), caracterizada pelo debate sobre potenciais e limites da rede para a participação política; e a **autonomização contemporânea**, onde subtemas como governo aberto, *smart cities*

e parlamento digital desenvolvem-se de forma independente, com metodologias e agendas próprias [4].

A participação cidadã mediada por tecnologia pode assumir diferentes níveis de profundidade e poder real. Sherry Arnstein, em seu trabalho seminal de 1969, propõe a “Escada da Participação Cidadã” (*Ladder of Citizen Participation*), uma tipologia de oito degraus que classifica o grau de poder efetivamente conferido aos cidadãos [9]:

- **Não-participação** (degraus 1-2): *Manipulação e Terapia* — formas em que o objetivo é “educar” ou “curar” os participantes, não ouvi-los. Comitês consultivos sem poder deliberativo exemplificam essa categoria.
- **Participação simbólica** (degraus 3-5): *Informação, Consulta e Pacificação* — cidadãos podem ouvir e ser ouvidos, mas sem garantia de que suas vozes influenciem decisões. Audiências públicas e pesquisas de opinião situam-se neste nível.
- **Poder cidadão** (degraus 6-8): *Parceria, Delegação de Poder e Controle Cidadão* — redistribuição efetiva de poder decisório. Orçamentos participativos vinculantes e conselhos com poder de veto exemplificam esses degraus superiores.

Ferramentas de transparência como o *Tô De Olho* situam-se primariamente no degrau da **informação**: proveem ao cidadão dados estruturados sobre a atuação parlamentar, condição necessária — mas não suficiente — para o exercício pleno da fiscalização. Como adverte Arnstein, “participação sem redistribuição de poder é um processo vazio e frustrante para os desprovidos de poder” [9]. Reconhecendo essa limitação estrutural, o projeto busca ir além da mera disponibilização de dados: ao oferecer *rankings*, comparativos e visualizações contextualizadas, a plataforma **empodera** o cidadão para uma fiscalização mais qualificada, fornecendo-lhe ferramentas para exercer pressão informada sobre seus representantes.

No contexto brasileiro, a brecha digital representa obstáculo adicional à democracia digital. Segundo o INAF, 29% da população entre 15 e 64 anos é funcionalmente analfabeta [6], limitando severamente a capacidade de interpretar planilhas, gráficos e relatórios técnicos. Complementarmente, autores como Saikali [24] e Lemos [25] alertam que iniciativas de “Cidades Inteligentes” e ferramentas digitais correm o risco de reproduzir desigualdades sociais se não forem desenhadas com foco explícito em inclusão. Farias argumenta que as possibilidades democráticas da internet dependem crucialmente de desenhos institucionais que facilitem a apropriação tecnológica por camadas populares [26]. Avelino et al. mapeiam iniciativas de governo aberto no âmbito federal, identificando avanços significativos na disponibilização de dados, porém reforçam que tecnologias precisam ser mediadas por visualizações claras e linguagem acessível para efetivação do controle social [2].

## 2.2.2 Teoria Principal-Agente e Accountability

A relação entre cidadãos e representantes eleitos pode ser analisada através da lente da **teoria econômica de agência**, originalmente desenvolvida para compreender relações contratuais em organizações. Neste modelo, os cidadãos (eleitores) atuam como **principais** que delegam autoridade a **agentes** (parlamentares e burocratas) para tomar decisões em seu

nome. O problema fundamental surge da **assimetria informacional**: os agentes possuem mais informação sobre suas próprias ações, esforços e competências do que os principais que os monitoram [27].

Esta assimetria manifesta-se de duas formas principais. O **risco moral** (*moral hazard*) ocorre quando os principais não conseguem observar plenamente as ações dos agentes após a delegação: um parlamentar pode “enrolar” em suas funções, ausentar-se de votações ou priorizar interesses particulares sem que o eleitorado perceba. A **seleção adversa** (*adverse selection*) surge antes da delegação: candidatos podem exagerar suas qualificações ou ocultar incompetências durante campanhas eleitorais. Em ambos os casos, a divergência entre os interesses do principal e do agente gera “custos de agência” — ineficiências, corrupção ou políticas distorcidas.

O conceito de **accountability** — frequentemente traduzido como “prestação de contas” ou “responsabilização” — emerge como mecanismo para mitigar esses problemas. A literatura distingue três vertentes complementares [5]:

- **Accountability vertical**: Exercida pelos eleitores através do voto. Parlamentares que não atendem às expectativas podem ser punidos nas urnas. Contudo, eleições são instrumentos “grosseiros”: ocorrem periodicamente (a cada 4-8 anos para senadores), envolvem múltiplas dimensões de avaliação simultâneas, e dependem de que o eleitor tenha informação suficiente para julgar o desempenho.
- **Accountability horizontal**: Exercida por instituições de controle como tribunais de contas, controladorias, Ministério Público e o próprio Poder Judiciário. Esses órgãos possuem competência técnica e acesso privilegiado a informações, mas enfrentam limitações de capacidade operacional diante do volume de atos a fiscalizar.
- **Accountability social**: Exercida por organizações da sociedade civil, imprensa investigativa, acadêmicos e cidadãos organizados. Esta modalidade complementa as anteriores ao ampliar a capacidade de monitoramento e pressionar por transparência.

A transparência é condição necessária — mas não suficiente — para a accountability. Pesquisas demonstram que para a divulgação de informações traduzir-se em responsabilização efetiva, três condições devem ser satisfeitas: (1) a informação deve efetivamente alcançar o público relevante; (2) o público deve ter capacidade de processá-la e reagir a ela; e (3) devem existir mecanismos institucionais que permitam consequências para os agentes. A simples disponibilização de dados brutos — o que alguns autores chamam de “governo nu” (*naked government*) — pode até amplificar percepções negativas sem gerar accountability efetiva.

O *Tô De Olho* posiciona-se como ferramenta de **accountability social**, reduzindo a assimetria informacional entre eleitores e senadores. Ao consolidar dados fragmentados, contextualizar valores com médias comparativas e oferecer rankings metodologicamente fundamentados, a plataforma amplia a capacidade do cidadão de monitorar seus representantes — condição prévia para o exercício tanto da accountability vertical (voto informado) quanto da pressão por accountability horizontal (denúncias fundamentadas a órgãos de controle). Esta abordagem alinha-se aos princípios de governo aberto defendidos pela Open Government

Partnership, que destacam a importância de intermediários tecnológicos para traduzir dados brutos em inteligência cívica acionável [28, 29].

### 2.2.3 Civic Tech e Sociedade Civil

O termo **civic tech** (tecnologia cívica) refere-se ao uso de tecnologias digitais para fortalecer a participação cidadã, a transparência governamental e a colaboração entre sociedade e Estado. Diferencia-se de *GovTech* (tecnologia para eficiência governamental interna) por seu foco na interface com o cidadão e no empoderamento da sociedade civil. Lathrop e Ruma, em coletânea seminal sobre governo aberto, reúnem contribuições de pioneiros como Tim O'Reilly, Beth Noveck e Aaron Swartz, estabelecendo três pilares para a *civic tech*: **transparência** (dados abertos), **participação** (engajamento cidadão) e **colaboração** (parcerias entre governo e sociedade) [30].

No Brasil, o ecossistema de civic tech consolidou-se a partir de 2010, impulsionado pela Lei de Acesso à Informação (2011) e pelo crescimento de organizações especializadas. Entre as iniciativas mais relevantes:

- **Transparência Brasil**: Fundada em 2000, é uma das principais organizações dedicadas à promoção da integridade e supervisão cívica no setor público. Desenvolve pesquisas, relatórios e ferramentas para monitoramento de políticas públicas [31].
- **Open Knowledge Brasil**: Responsável pela Operação Serenata de Amor e pelo projeto “Querido Diário”, que aplica técnicas de inteligência artificial para auditar diários oficiais municipais.
- **Fiquem Sabendo**: Organização sem fins lucrativos que utiliza civic tech de código aberto para expor gastos governamentais não divulgados, tendo revelado mais de 500 bilhões de reais em despesas não reportadas ao longo de 27 anos.
- **Associação Brasileira de Jornalismo Investigativo (Abraji)**: Emprega jornalismo de dados e ferramentas tecnológicas para investigar corrupção e má gestão de recursos públicos.

O cenário internacional oferece referências consolidadas que inspiram o movimento brasileiro. No Reino Unido, a plataforma *TheyWorkForYou* [18] é pioneira em simplificar a linguagem parlamentar para o cidadão comum. No Brasil, além das iniciativas citadas, destacam-se a *Base dos Dados* [32], que universaliza o acesso a dados públicos tratados, e o *Colab* [33], focado na gestão participativa municipal. O Portal Brasileiro de Dados Abertos [34], embora institucional, fornece a infraestrutura de dados brutos que alimenta muitas dessas aplicações.

Apesar dos avanços, o ecossistema enfrenta desafios significativos. A **sustentabilidade financeira** é precária: muitas iniciativas dependem de financiamento coletivo, bolsas internacionais ou trabalho voluntário, limitando sua capacidade de operação contínua. A **brecha digital** restringe o alcance a populações com menor acesso à internet ou habilidades tecnológicas limitadas. E a **fragmentação de esforços** — múltiplas ferramentas com escopos parcialmente sobrepostos — dispersa recursos e dificulta o engajamento do cidadão comum.

O *Tô De Olho* posiciona-se neste ecossistema com um diferencial claro: o foco exclusivo no **Senado Federal**, câmara legislativa até então carente de ferramentas específicas de fiscalização cidadã. Enquanto a Câmara dos Deputados conta com ao menos três plataformas consolidadas (Serenata de Amor, De Olho no Congresso, De Olho em Você), os 81 senadores — que exercem mandatos de oito anos e detêm competências exclusivas de alto impacto — permanecem em relativa “sombra” digital.

## 2.3 Eixo Técnico-Metodológico

Este eixo expõe fundamentos técnicos e metodológicos que orientam a implementação do *Tô De Olho*. Aborda avaliação parlamentar, visualização de dados e arquitetura de software.

### 2.3.1 Métricas de Efetividade Legislativa

A avaliação quantitativa do desempenho parlamentar constitui tema relevante na ciência política contemporânea. Craig Volden e Alan E. Wiseman, co-diretores do *Center for Effective Lawmaking*, desenvolveram o *Legislative Effectiveness Score* (LES), uma métrica que mensura a capacidade de parlamentares em conduzir suas proposições através do processo legislativo [35, 7].

A metodologia do LES fundamenta-se em cinco estágios do processo legislativo, cada qual representando um grau crescente de sucesso na agenda do parlamentar:

1. **Introdução:** O projeto é formalmente apresentado;
2. **Ação em comissão:** O projeto recebe parecer ou é debatido em comissão temática;
3. **Votação em plenário (câmara de origem):** O projeto é levado à votação na casa onde foi apresentado;
4. **Aprovação na câmara de origem:** O projeto é aprovado e segue para a outra casa;
5. **Conversão em lei:** O projeto completa a tramitação e é sancionado.

Crucialmente, nem todos os projetos possuem igual peso na metodologia. Volden e Wiseman categorizam as proposições em três níveis de significância: **projetos comemorativos** (como denominação de logradouros), que recebem peso mínimo; **projetos substantivos**, que alteram políticas públicas de forma moderada; e **projetos substantivos e significativos**, que promovem mudanças estruturais relevantes. Esta ponderação evita que parlamentares “inflatem” suas estatísticas com proposições triviais [36].

Os autores identificaram fatores consistentemente correlacionados à maior efetividade legislativa: senioridade no mandato, ocupação de posições em comissões estratégicas (especialmente presidências e relatorias), pertencimento ao partido majoritário e experiência prévia em legislaturas estaduais. Interessantemente, pesquisas subsequentes demonstraram que, controlando demais variáveis, parlamentares mulheres tendem a ser mais efetivas que seus colegas homens [7].

Embora desenvolvida para o contexto norte-americano, a metodologia oferece um *framework* adaptável para avaliar parlamentares brasileiros. No *Tô De Olho*, o “Score” do senador inspira-se nesta abordagem, combinando quatro dimensões com pesos públicos:

- **Produtividade Legislativa (35%):** Avalia proposições de autoria e relatorias, com multiplicadores por tipo (PEC: 3x, PLP: 2x) e estágio de tramitação alcançado;
- **Presença em Votações (25%):** Mensura comparecimento efetivo às sessões deliberativas;
- **Economia na Cota Parlamentar (20%):** Compara uso individual da CEAPS com a mediana do Senado;
- **Participação em Comissões (20%):** Pondera engajamento em comissões, com bônus para cargos de liderança.

A transparência metodológica — expor publicamente os critérios e pesos utilizados — é fundamental para que o *ranking* seja percebido como ferramenta de informação, não de manipulação política. Diferente de classificações opacas, o cidadão poderá compreender — e questionar — os fundamentos da avaliação.

### 2.3.2 Visualização de Dados e Retórica Visual

A apresentação de dados ao cidadão não é neutra: escolhas de *design* influenciam profundamente a interpretação das informações. Jessica Hullman e Nicholas Diakopoulos investigaram os “efeitos de enquadramento” (*framing effects*) em visualizações narrativas, demonstrando que técnicas retóricas como seleção, omissão, ênfase e sequenciamento podem direcionar a leitura do público de forma consciente ou inconsciente [8].

Os autores identificam quatro categorias de técnicas retóricas em visualizações de dados:

1. **Proveniência:** Identificação da origem e credibilidade dos dados. Visualizações que ocultam fontes ou datas de atualização comprometem a verificabilidade. No *Tô De Olho*, cada gráfico exibe a fonte oficial (API do Senado ou Portal da Transparência) e a data da última sincronização.
2. **Mapeamento visual:** Como elementos gráficos representam variáveis numéricas. Escalas inconsistentes, truncamento de eixos ou escolhas de cores podem distorcer percepções. A plataforma adota escalas consistentes em gráficos comparativos e paletas de cores acessíveis.
3. **Anotações linguísticas:** Textos, títulos e legendas que guiam a interpretação. Valores absolutos sem contexto podem induzir conclusões equivocadas (ex: “Senador X gastou R\$ 100 mil” parece muito sem saber que a média é R\$ 150 mil). O *Tô De Olho* contextualiza valores com médias e percentis.
4. **Interatividade:** Controles que permitem ao usuário explorar dados por conta própria reduzem a dependência de narrativas pré-construídas. A plataforma oferece filtros por partido, estado e período, permitindo análises personalizadas.

A **literacia em visualização de dados** (*Data Visualization Literacy* — DVL) refere-se à capacidade de interpretar corretamente representações visuais de informações. Pesquisas demonstram que mesmo populações com alta escolaridade frequentemente cometem erros de interpretação em gráficos aparentemente simples. No contexto brasileiro, onde 29% da população é funcionalmente analfabeta [6], o desafio é ainda maior: visualizações complexas podem excluir justamente os cidadãos mais vulneráveis à falta de transparência.

Para o *Tô De Olho*, esses princípios orientam decisões de *design*: priorizar visualizações simples e intuitivas; oferecer múltiplas formas de apresentação (gráficos, tabelas, textos explicativos); e testar a compreensibilidade com usuários de diferentes perfis. O objetivo é maximizar a transparência metodológica, evitando que a plataforma seja percebida como veículo de viés político.

### 2.3.3 Arquitetura de Software: Monolito Modular

A escolha arquitetural de um sistema de software envolve *trade-offs* fundamentais entre complexidade operacional, velocidade de desenvolvimento e capacidade de evolução. A literatura distingue três abordagens principais: o **monolito tradicional**, onde todos os componentes residem em uma única base de código sem separação clara; os **microsserviços**, que fragmentam a aplicação em serviços independentes comunicando-se via rede; e o **monolito modular**, que combina a simplicidade operacional do primeiro com a organização interna do segundo [37, 38].

O monolito modular organiza o código em módulos discretos, fracamente acoplados e orientados a domínio, porém dentro de uma única unidade de *deploy*. Diferente do monolito tradicional, que tende a se tornar uma “bola de lama” (*big ball of mud*) com o crescimento, o monolito modular impõe fronteiras lógicas explícitas entre componentes, facilitando a manutenção e permitindo que equipes trabalhem de forma relativamente independente em diferentes partes do sistema [38].

A escolha por monolito modular no *Tô De Olho* fundamenta-se em critérios técnicos e contextuais:

- **Simplicidade de Deploy:** Um único contêiner Docker simplifica a infraestrutura e reduz custos operacionais. Plataformas *serverless* como Google Cloud Run beneficiam-se particularmente desta característica, oferecendo escala automática (inclusive a zero) sem a complexidade de orquestração de múltiplos serviços;
- **Ausência de Latência de Rede:** A comunicação entre módulos ocorre via chamadas de função em memória, eliminando a latência e os riscos de falha associados a chamadas de rede. Em microsserviços, cada interação entre componentes adiciona overhead de serialização, transmissão e desserialização;
- **Consistência Transacional:** Operações que envolvem múltiplos módulos podem compartilhar uma única transação de banco de dados, garantindo atomicidade. Em arquiteturas distribuídas, esse padrão requereria implementação de *sagas* ou *two-phase commit*, aumentando significativamente a complexidade;



- **Adequação ao Contexto:** Para equipes pequenas e projetos acadêmicos, a complexidade operacional de microsserviços — monitoramento distribuído, orquestração de contêineres, *service discovery*, gerenciamento de configurações — frequentemente supera os benefícios de escalabilidade granular.

A literatura recente sobre monolito modular destaca a importância de princípios do **Domain-Driven Design** (DDD) na definição de fronteiras entre módulos. Eric Evans, criador do DDD, propõe o conceito de *Bounded Contexts* — regiões do sistema onde um modelo de domínio específico se aplica, com interfaces bem definidas para comunicação com outros contextos [39]. No *Tô De Olho*, cada módulo interno (*senador*, *ceaps*, *votacao*, *emenda*, *ranking*) representa um *bounded context* com responsabilidades claras.

Frameworks modernos têm formalizado o padrão de monolito modular. O Google Service Weaver permite escrever aplicações como monolitos modulares que podem ser implantados como processos únicos ou microsserviços distribuídos, sem alteração de código [40]. O Spring Modulith oferece suporte similar no ecossistema Java, com verificação automatizada de fronteiras entre módulos [41]. Estudos recentes indicam que esta abordagem tem ganhado tração na indústria como alternativa pragmática à complexidade prematura de microsserviços [38]. Embora o *Tô De Olho* não utilize esses frameworks específicos (dado o backend em Go), a arquitetura interna segue os mesmos princípios: módulos com interfaces explícitas, dependências unidirecionais e comunicação via contratos bem definidos.

Esta escolha não descarta a possibilidade de evolução futura. Caso o sistema alcance escala que justifique escalabilidade granular de componentes específicos, a estrutura modular facilitará a extração de microsserviços independentes — uma estratégia de “decomposição incremental” recomendada pela literatura como alternativa ao risco de migração completa [37].

### 2.3.4 Engenharia de Dados: APIs e Processos ETL

O padrão ETL (*Extract, Transform, Load*) constitui a espinha dorsal de sistemas de integração de dados desde os primórdios da computação analítica. Ralph Kimball, pioneiro em *data warehousing*, define ETL como o processo responsável por “extrair dados de sistemas fonte, aplicar transformações de qualidade e conformidade, e entregar os dados em formato dimensional” [42]. Vassiliadis, em revisão abrangente da literatura, estima que processos ETL consomem entre 60% e 80% do esforço de desenvolvimento de projetos de integração de dados [43].

A estratégia de ingestão do *Tô De Olho* segue as três fases clássicas do ETL:

1. **Extração (Extract):** Coleta de dados brutos das fontes oficiais via APIs RESTful. O sistema consome três APIs distintas: a API Legislativa do Senado (matérias, votações nominais, comissões), a API Administrativa do Senado (CEAPS, remunerações de gabinete) e a API do Portal da Transparência (emendas parlamentares). Cada fonte possui características próprias de paginação, limites de requisição e formatos de resposta, exigindo adaptadores específicos;
2. **Transformação (Transform):** Normalização, limpeza e enriquecimento dos dados ex-

traídos. Esta fase inclui: padronização de nomes de senadores (tratamento de variações como “José da Silva” vs “JOSE DA SILVA”), conversão de formatos de data, cálculo de métricas derivadas (como percentuais de presença) e detecção de anomalias (valores negativos, datas futuras, registros duplicados);

3. **Carga (Load):** Persistência dos dados transformados no banco PostgreSQL, escolhido por sua robustez e suporte nativo a tipos JSONB, facilitando o armazenamento de estruturas semi-estruturadas das APIs [44]. Utilizam-se estratégias de upsert para evitar duplicação e manutenção de histórico de alterações.

O processo combina duas estratégias complementares de carga:

- **Backfill (Carga Histórica):** Execução única para ingestão do histórico completo desde um ano configurável (padrão: 2019). Esta flexibilidade, implementada via variável de ambiente, permite ajustar o escopo temporal conforme necessidades específicas — por exemplo, restringir a dados de 2023 em ambiente de desenvolvimento para acelerar testes, ou expandir para 2015 para análises de longo prazo;
- **Sincronização Incremental:** Tarefas agendadas (*CronJobs*) executadas diariamente às 03:00 BRT, capturando apenas atualizações desde a última sincronização. Esta abordagem otimiza o consumo de recursos e respeita os limites de requisição das APIs oficiais.

A escolha por APIs oficiais — em detrimento de técnicas de *web scraping* — fundamenta-se em critérios de confiabilidade e manutenibilidade. APIs possuem contratos documentados, formatos estruturados (JSON/XML) e versionamento explícito, enquanto o *scraping* de páginas HTML é intrinsecamente frágil a mudanças de layout. Kimball adverte que “a qualidade dos dados de entrada determina o teto de qualidade do sistema final” [45] — premissa que reforça a preferência por fontes oficiais com garantias de integridade.

Adicionalmente, as APIs oficiais brasileiras oferecem vantagens práticas: o Portal da Transparência disponibiliza dados de emendas com granularidade diária; a API Legislativa do Senado permite consultas por período específico; e a API Administrativa provê arquivos CSV consolidados para carga em massa. Essas características distintas exigem que o módulo ETL do *Tô De Olho* implemente adaptadores especializados para cada fonte, unificando os dados em um modelo dimensional coerente antes da carga.

### 3 Metodologia

O desenvolvimento do *Tô De Olho* enquadra-se no paradigma da *Design Science Research* (DSR), abordagem metodológica adequada para pesquisas que visam a construção e avaliação de artefatos tecnológicos destinados a resolver problemas organizacionais identificados [46]. O problema abordado — a fragmentação de dados públicos sobre a atuação de senadores federais em múltiplas APIs governamentais não integradas — demanda a construção de um artefato de software capaz de consolidar, processar e apresentar essas informações de forma acessível ao cidadão.

Esta seção detalha a abordagem de desenvolvimento iterativo adotada, as fontes de dados governamentais integradas e a metodologia de avaliação de desempenho parlamentar inspirada no *Legislative Effectiveness Score*.

## 3.1 Abordagem de Desenvolvimento

A natureza do projeto — integração de múltiplas APIs governamentais com estruturas de dados heterogêneas e documentação variável — demandou uma abordagem de desenvolvimento capaz de acomodar descobertas incrementais e ajustes frequentes de escopo. Metodologias tradicionais de desenvolvimento em cascata, que pressupõem requisitos estáveis e bem definidos desde o início, mostraram-se inadequadas para este contexto de exploração de APIs públicas com comportamentos nem sempre previsíveis.

Optou-se, portanto, por uma abordagem **iterativa e incremental**, na qual o trabalho foi organizado em ciclos de desenvolvimento focados em entregas funcionais. Cada ciclo produzia um incremento utilizável do sistema, permitindo validação contínua das funcionalidades implementadas e ajustes baseados nos aprendizados obtidos durante a integração com cada API.

O desenvolvimento contou com apoio de ferramentas de inteligência artificial generativa como assistentes de codificação, seguindo a tendência contemporânea de *AI-assisted software development* [47]. Tais ferramentas, baseadas em modelos de linguagem de grande escala (LLMs), foram empregadas para aceleração de tarefas operacionais como geração de código *boilerplate*, refatoração e *debugging*. O uso de assistentes de IA em desenvolvimento de software representa uma evolução natural das ferramentas de produtividade, análoga à adoção de IDEs com *autocomplete* e analisadores estáticos de código.

É fundamental distinguir entre **assistência operacional** e **autoria intelectual**. As decisões estruturantes do projeto — arquitetura do sistema, design do algoritmo de ranking, seleção de fontes de dados, modelagem do domínio e interpretação de resultados — foram integralmente concebidas, avaliadas e validadas pelo desenvolvedor. A ferramenta de IA atuou como acelerador de implementação, não como substituto do julgamento técnico.

A divisão do trabalho ocorreu em cinco fases principais:

1. **Fundação:** Estruturação do projeto em Golang, implementação do cliente para a API Legislativa do Senado, criação das *migrations* do banco de dados e configuração inicial do *frontend* em Next.js utilizando Bun como *runtime*;
2. **Ingestão de Dados:** Implementação do cliente para a API Administrativa, configuração do *scheduler* para tarefas agendadas, coleta de votações nominais e carga de dados históricos;
3. **Ranking e API:** Desenvolvimento do serviço de cálculo de rankings, criação dos *endpoints* REST para consumo pelo *frontend*, configuração do cache Redis e implementação de testes automatizados;
4. **Frontend:** Desenvolvimento do *dashboard* principal, interface de ranking interativo e

páginas de perfil dos senadores;

5. **Emendas e Polimento:** Integração com o Portal da Transparência para dados de emendas parlamentares, visualizações de dados e preparação para *deploy*.

## 3.2 Fontes de Dados

O sistema integra três fontes de dados oficiais, fundamentadas no arcabouço legal brasileiro de transparência pública. A Lei de Acesso à Informação (Lei n. 12.527/2011) estabelece como diretriz a “disponibilização de informações em formatos abertos, estruturados e legíveis por máquina” [11], princípio que as APIs governamentais operacionalizam. Conforme demonstrado pela Operação Serenata de Amor, tecnologias desenvolvidas sobre esses dados abertos podem gerar valor público ao facilitar o controle social do gasto parlamentar [14].

A seleção dos *endpoints* seguiu três critérios: (i) **relevância** — priorizando dados de gastos, votações e atuação legislativa; (ii) **confiabilidade** — selecionando fontes com documentação oficial; e (iii) **viabilidade técnica** — considerando formatos estruturados (JSON/XML) e limites de requisição adequados.

### 3.2.1 API Legislativa do Senado

Documentada em `<legis.senado.leg.br/dadosabertos>`, esta API RESTful fornece dados do processo legislativo. A URL base para requisições é `<https://legis.senado.leg.br/dadosabertos>`. Os dados são retornados em formato JSON, com suporte a paginação e limite de 10 requisições por segundo. Os principais recursos utilizados incluem: lista de senadores em exercício, detalhes biográficos, histórico de mandatos e licenças, votações nominais em plenário e comissões, orientação partidária, proposições de autoria, relatorias, participação em comissões e discursos.

### 3.2.2 API Administrativa do Senado

Documentada em `<adm.senado.gov.br/adm-dadosabertos/swagger-ui>`, esta interface disponibiliza dados financeiros e administrativos. Os recursos utilizados incluem: despesas da Cota Parlamentar (CEAPS) com detalhamento por fornecedor e tipo, opção por auxílio-moradia, escritórios de apoio, lista de servidores de gabinete, remunerações mensais e mapeamento de lotações.

### 3.2.3 Portal da Transparência (CGU)

A API do Portal da Transparência (`<api.portaldatransparencia.gov.br>`) fornece dados de emendas parlamentares e transferências federais. O acesso requer autenticação via chave de API no *header* `chave-api-dados`. O filtro `tipoEmenda=Transferência Especial` permite identificar as “emendas PIX”, modalidade de repasse que dispensa convênio.

### 3.2.4 Limitações e Cobertura Temporal

Cada fonte de dados apresenta limitações que impactam o escopo das análises:

- **API Legislativa:** Votações nominais disponíveis apenas a partir de 2019 (legislatura 56). Limite de 10 requisições por segundo;
- **API Administrativa:** Despesas CEAPS disponíveis desde 2008 em CSV; API REST cobre apenas o ano corrente. Não há *endpoint* para vincular servidores diretamente ao gabinete;
- **Portal da Transparência:** Emendas disponíveis a partir de 2015. Busca por autor utiliza nome textual, exigindo normalização de grafia. Limite de 300 requisições por minuto.

### 3.3 Metodologia de Avaliação de Desempenho

A avaliação objetiva do desempenho parlamentar é elemento central para a participação cidadã. Conforme a escada de participação de Arnstein [9], o acesso a informações claras é pré-requisito para que cidadãos avancem de níveis consultivos para controle social efetivo. A construção de rankings requer transparência metodológica [8]. Ressalta-se que o índice proposto não pretende mensurar “qualidade moral” ou “mérito político” subjetivo dos parlamentares, mas sim oferecer um instrumento comparativo fundamentado em critérios observáveis e quantificáveis de produtividade e gestão fiscal.

O cálculo do *score* de cada senador é inspirado no *Legislative Effectiveness Score* (LES), metodologia desenvolvida por Volden e Wiseman para avaliar a efetividade legislativa de parlamentares americanos [7]. O LES define efetividade legislativa como “a capacidade comprovada de avançar itens da agenda de um parlamentar através do processo legislativo até sua transformação em lei” [35].

Para a adaptação brasileira, manteve-se a filosofia de valorizar o avanço de proposições através do processo legislativo, incorporando critérios adicionais relevantes ao contexto de fiscalização cidadã. O índice proposto compõe-se de quatro dimensões objetivamente mensuráveis:

Tabela 1: Critérios do algoritmo de ranking

Critério	Peso	Justificativa
Produtividade Legislativa	35%	Core do LES: capacidade de aprovar proposições
Presença em Votações	25%	Compromisso efetivo com o mandato
Economia na Cota (CEAPS)	20%	Responsabilidade fiscal
Participação em Comissões	20%	Trabalho técnico especializado

#### 3.3.1 Produtividade Legislativa

Este critério avalia o avanço de proposições de autoria do senador através do processo legislativo, atribuindo pontuação crescente por estágio alcançado (Apresentado: 1 ponto, Em Comissão: 2, Aprovado em Comissão: 4, Aprovado em Plenário: 8, Transformado em Lei: 16). Os multiplicadores refletem a maior complexidade de Propostas de Emenda Constitucional (PEC:  $\times 3.0$ ) e Projetos de Lei Complementar (PLP:  $\times 2.0$ ). Senadores que atuam como relatores recebem pontuação adicional.

### 3.3.2 Presença em Votações

Calculada como a razão entre votações participadas e votações disponíveis:

$$\text{Presença} = \frac{\text{Votações Participadas}}{\text{Votações Disponíveis}} \times 100 \quad (1)$$

As votações disponíveis excluem períodos de licença oficial (médica ou para cargo executivo), garantindo que afastamentos justificados não penalizem o senador. Obstrução **não** conta como presença.

### 3.3.3 Economia na Cota Parlamentar

Calculada como a proporção não utilizada do teto da CEAPS:

$$\text{Economia} = \left( 1 - \frac{\text{Gasto Senador}}{\text{Teto CEAPS}} \right) \times 100 \quad (2)$$

O teto da CEAPS varia por UF (R\$ 26.000 para DF até R\$ 44.000+ para estados da região Norte). Senadores com gasto acima de 120% do teto recebem pontuação zero neste critério.

### 3.3.4 Participação em Comissões

Pontuação atribuída conforme o nível de engajamento: membro titular (+2 pontos), suplente (+1), vice-presidente (+3), presidente (+5). Comissões estratégicas (CAE, CCJ) recebem multiplicador de  $\times 1.5$ .

### 3.3.5 Fórmula Final e Tratamento de Casos Especiais

Cada métrica é normalizada de 0 a 100 antes da ponderação:

$$\text{Score} = (P_L \times 0.35) + (P_V \times 0.25) + (E_C \times 0.20) + (P_C \times 0.20) \quad (3)$$

Onde:

- $P_L$ : Produtividade Legislativa;
- $P_V$ : Presença em Votações;
- $E_C$ : Economia na Cota;
- $P_C$ : Participação em Comissões.

Casos especiais são tratados da seguinte forma: senadores novos (suplentes) requerem período mínimo de 30 dias para entrar no ranking; licenças curtas mantêm o senador no ranking com ajuste nos denominadores; licenças longas ( $\geq 30$  dias) congelam o score; dados indisponíveis causam reponderação proporcional dos demais critérios.

## 4 Requisitos

A elicitação de requisitos seguiu as diretrizes da norma ISO/IEC/IEEE 29148 [49], que estabelece boas práticas para especificação de requisitos em projetos de software. Os requisitos foram organizados em duas categorias: funcionais (RF), que descrevem as funcionalidades do sistema, e não-funcionais (RNF), que definem atributos de qualidade.

Para a categorização dos requisitos não-funcionais, adotou-se o modelo de qualidade da ISO/IEC 25010 [50], que define oito características de qualidade de software: funcionalidade, eficiência de desempenho, compatibilidade, usabilidade, confiabilidade, segurança, manutenibilidade e portabilidade. Essa estrutura permite uma cobertura sistemática dos atributos de qualidade esperados para a plataforma.

### 4.1 Requisitos Funcionais

Conforme destacado na análise de trabalhos relacionados (Seção 1.6), o *Tô De Olho* sintetiza funcionalidades de plataformas consolidadas como *De Olho em Você* e *De Olho no Congresso*, adaptando-as ao contexto do Senado Federal. Os requisitos funcionais foram organizados em sete módulos:

#### **Módulo de Senadores:**

- **[RF01]** O sistema deve apresentar a lista atualizada dos 81 senadores com foto, partido e estado.
- **[RF02]** O sistema deve permitir a busca de senadores por nome, sigla partidária ou UF.
- **[RF03]** O sistema deve exibir o perfil completo do senador com abas organizadas (Visão Geral, Gastos, Gabinete, Votações, Emendas).

#### **Módulo de Transparência Financeira (CEAPS):**

- **[RF04]** O sistema deve importar os lançamentos da Cota Parlamentar (CEAPS) através das APIs de Dados Abertos do Senado.
- **[RF05]** O sistema deve permitir visualizar o gasto acumulado por tipo de despesa (passagens, correios, consultorias, combustível).
- **[RF06]** O sistema deve exibir os fornecedores que mais receberam recursos de um determinado senador.
- **[RF07]** O sistema deve gerar alertas automáticos para despesas atípicas: valores acima da média, pagamentos em finais de semana, intervalos menores que 3 dias entre pagamentos ao mesmo fornecedor.

#### **Módulo de Emendas e Orçamento:**

- **[RF08]** O sistema deve integrar com o Portal da Transparência para buscar emendas de autoria do senador.

- **[RF09]** O sistema deve destacar valores destinados via “Transferências Especiais” (emendas PIX).
- **[RF10]** O sistema deve exibir mapas interativos de distribuição de emendas por município, permitindo identificar concentração geográfica de recursos.

#### **Módulo de Atividade Legislativa:**

- **[RF11]** O sistema deve listar as votações nominais recentes e o voto de cada senador (Sim/Não/Abstenção).
- **[RF12]** O sistema deve exibir a participação do senador em comissões permanentes e especiais, incluindo cargo ocupado (titular, suplente, presidente, relator).
- **[RF13]** O sistema deve listar proposições de autoria do senador, com indicação do tipo (PEC, PLP, PL) e estágio de tramitação.
- **[RF14]** O sistema deve exibir discursos proferidos pelo senador em plenário, com data e tema.
- **[RF15]** O sistema deve exibir a agenda pública de reuniões de comissões, permitindo visualizar pautas futuras.
- **[RF16]** O sistema deve apresentar links para os perfis oficiais do senador em redes sociais (Twitter/X, Instagram, Facebook).

#### **Módulo de Gabinete:**

- **[RF17]** O sistema deve exibir a lista de servidores do gabinete do senador, incluindo cargo e vínculo.
- **[RF18]** O sistema deve apresentar a folha de pagamento do gabinete, com remuneração mensal por servidor.

#### **Módulo de Comparação e Análise:**

- **[RF19]** O sistema deve permitir comparar de 2 a 5 senadores lado a lado em múltiplas dimensões: despesas, emendas, votações e fornecedores em comum.
- **[RF20]** O sistema deve exibir ranking de fornecedores com cruzamento de sanções administrativas, identificando empresas com situação cadastral irregular.
- **[RF21]** O sistema deve mostrar indicadores de confiança: data da última sincronização, completude dos dados e fonte de cada informação.

#### **Módulo de Ranking e Score:**

- **[RF22]** O sistema deve calcular e exibir o *Score* de efetividade legislativa de cada senador, baseado no algoritmo de ranking do projeto.
- **[RF23]** O sistema deve apresentar gráfico radar com as quatro dimensões do *Score*:



Produtividade Legislativa, Presença em Votações, Economia na Cota e Participação em Comissões.

- **[RF24]** O sistema deve permitir ordenar e filtrar senadores por cada critério individual do ranking.

## 4.2 Requisitos Não-Funcionais

### Desempenho:

- **[RNF01]** O sistema deve responder a requisições de consulta em até 2 segundos sob condições normais de uso.
- **[RNF02]** A arquitetura deve suportar escalabilidade horizontal para lidar com picos de acesso em períodos eleitorais.

### Usabilidade e Acessibilidade:

- **[RNF03]** O sistema deve ser acessível via navegadores *web* em dispositivos *desktop* e *mobile*.
- **[RNF04]** A interface deve seguir o padrão *mobile-first* para garantir boa experiência em dispositivos móveis.
- **[RNF05]** O sistema deve seguir as diretrizes de acessibilidade WCAG 2.1 nível AA.

### Confiabilidade:

- **[RNF06]** Os dados devem ser sincronizados diariamente com as APIs oficiais do Senado Federal e Portal da Transparência.
- **[RNF07]** O sistema deve manter disponibilidade mínima de 99% durante o período eleitoral.

### Segurança e Privacidade:

- **[RNF08]** As comunicações devem ser criptografadas utilizando HTTPS/TLS.
- **[RNF09]** O sistema deve estar em conformidade com a LGPD (Lei Geral de Proteção de Dados).

### Manutenibilidade:

- **[RNF10]** A arquitetura modular deve permitir atualizações isoladas de cada módulo sem impacto em outros componentes.
- **[RNF11]** O código deve seguir padrões de desenvolvimento (*linting*, formatação) e estar documentado.
- **[RNF12]** O sistema deve possuir *pipelines* de CI/CD para integração e deploy contínuos.

## 5 Design

O design de sistemas de informação voltados à fiscalização cidadã deve conciliar requisitos aparentemente conflitantes: robustez suficiente para processar grandes volumes de dados governamentais, simplicidade operacional compatível com equipes reduzidas, e acessibilidade que permita ao cidadão comum compreender informações complexas. Conforme o paradigma de Design Science Research adotado neste trabalho [46], as decisões de design devem ser orientadas tanto pela utilidade prática do artefato quanto pelo rigor teórico que fundamenta cada escolha.

Esta seção organiza as decisões técnicas em quatro dimensões interdependentes: a arquitetura de software que estrutura o sistema, o stack tecnológico que a implementa, a estratégia de ingestão que alimenta a base de dados, e o modelo relacional que organiza as informações. Cada decisão é apresentada com suas alternativas consideradas e a justificativa para a escolha final, seguindo a tradição de documentação arquitetural proposta por Dragoni et al. [37].

### 5.1 Arquitetura do Sistema

A escolha arquitetural representa uma das decisões mais consequentes em projetos de software, com implicações que se estendem por todo o ciclo de vida do sistema. Em aplicações que integram múltiplas APIs externas — como é o caso do *Tô De Olho*, que consome dados de três órgãos governamentais distintos — a arquitetura deve equilibrar a flexibilidade para evolução futura com a simplicidade operacional exigida pelo contexto acadêmico.

Após análise de alternativas, optou-se pelo padrão de **monolito modular**. Esta abordagem, sistematizada em revisão de literatura recente [38], combina a organização interna característica de microsserviços — com fronteiras claras entre domínios de negócio — com a simplicidade de implantação dos monolitos tradicionais. A decisão fundamentou-se em quatro critérios:

- **Simplicidade Operacional:** Um único contêiner Docker elimina a complexidade de orquestração distribuída. Em plataformas serverless como Google Cloud Run, essa característica traduz-se em escala automática (inclusive a zero) e custos proporcionais ao uso efetivo;
- **Latência Previsível:** A comunicação entre módulos ocorre via chamadas de função em memória, eliminando a variabilidade de latência inerente a redes — fator crítico para operações que agregam dados de múltiplos domínios;
- **Consistência Transacional:** Operações que envolvem múltiplos módulos compartilham uma única transação de banco de dados, garantindo atomicidade sem os patterns complexos (sagas, two-phase commit) exigidos por sistemas distribuídos;
- **Estratégia MonolithFirst:** Martin Fowler argumenta que “quase todos os casos de sucesso de microsserviços começaram com um monolito que cresceu grande demais e foi decomposto” [51]. A estrutura modular atual permite extração futura de serviços independentes apenas onde houver necessidade comprovada.

### 5.1.1 Organização em Módulos

A estrutura interna segue os princípios de *Bounded Contexts* do Domain-Driven Design [39], onde cada módulo representa um contexto de domínio com responsabilidades claramente delimitadas. O diretório `internal/` organiza-se em cinco módulos principais, cada um encapsulando um aspecto da fiscalização parlamentar:

- `internal/senador`: Gerencia dados cadastrais e mandatos históricos, atuando como “fonte de verdade” para identificação de parlamentares;
- `internal/ceaps`: Processa despesas da Cota Parlamentar, implementando detecção de anomalias e agregações por categoria;
- `internal/votacao`: Coleta votações nominais e calcula métricas como presença e fidelidade partidária;
- `internal/emenda`: Integra dados do Portal da Transparência, com tratamento especial para “emendas PIX”;
- `internal/ranking`: Orquestra o cálculo de scores conforme metodologia definida na Seção 3.3.

Módulos transversais complementam a estrutura: `internal/api` expõe endpoints REST, `internal/scheduler` gerencia tarefas agendadas, e `internal/cache` abstrai operações no Redis. Os módulos comunicam-se exclusivamente através de interfaces Go, respeitando o princípio de inversão de dependência.

### 5.1.2 Visão de Implantação

A Figura 1 apresenta a arquitetura de implantação do sistema, ilustrando o fluxo de dados desde as fontes governamentais até o usuário final. O backend executa no Google Cloud Run, consumindo dados armazenados em PostgreSQL (Cloud SQL) e Redis (Memorystore). O frontend, desenvolvido em Next.js, é hospedado na Vercel para otimização de distribuição global.

## 5.2 Stack Tecnológico

A seleção de tecnologias foi orientada por três critérios: desempenho em cenários de alta concorrência (característicos de pipelines de ingestão), manutenibilidade a longo prazo (considerando a natureza open-source do projeto), e adequação ao domínio (priorizando ferramentas com suporte robusto a operações analíticas).

### 5.2.1 Backend em Go

A linguagem Go foi selecionada após avaliação comparativa com Node.js e Python. O fator decisivo foi o modelo de **goroutines**: threads leves gerenciadas pelo runtime, consumindo aproximadamente 2KB de memória inicial — em contraste com cerca de 1MB por thread de sistema operacional [52]. Essa eficiência viabiliza o processamento paralelo de centenas de

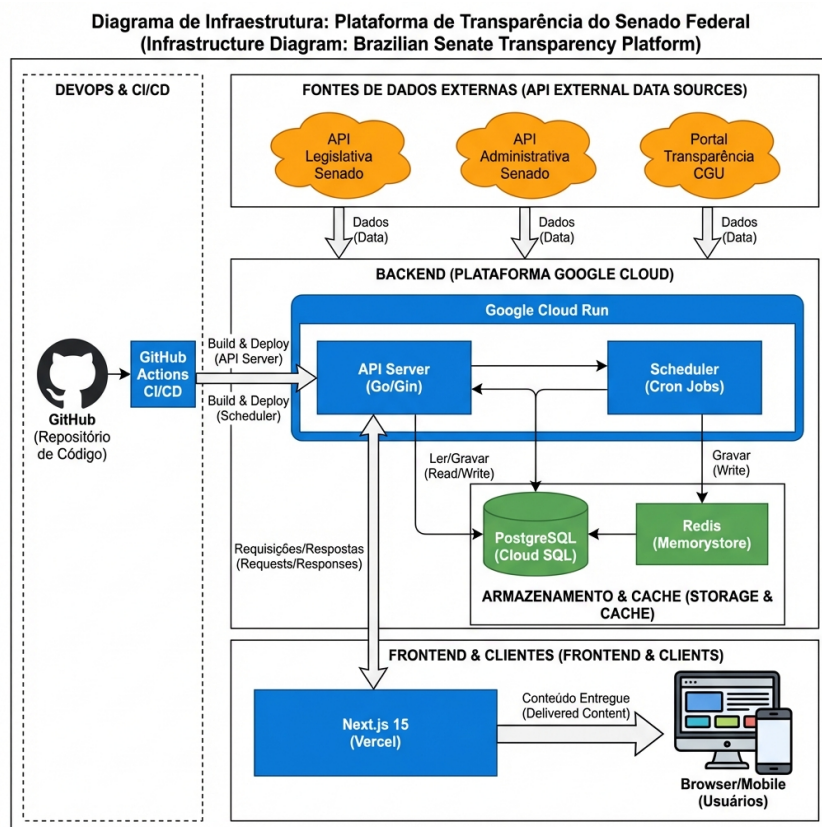


Figura 1: Arquitetura de implantação do sistema Tô De Olho  
Fonte: Autoria Própria

requisições HTTP durante a ingestão.

O framework **Gin** foi escolhido por seu roteamento baseado em radix tree, com desempenho até 40 vezes superior a alternativas [53]. O ORM **GORM** oferece migrações automáticas e operações de upsert essenciais para garantia de idempotência. Para ambientes de produção, o sistema implementa connection pooling com limites configuráveis de conexões ociosas e máximas.

### 5.2.2 Camada de Persistência

**PostgreSQL** foi selecionado por sua robustez em consultas analíticas e suporte a CTEs (*Common Table Expressions*), essenciais para agregações complexas. Originado em 1986 na UC Berkeley [54], o banco garante conformidade ACID crítica para operações de ingestão concorrente.

**Redis** atua como cache para rankings pré-computados, alcançando latências de 100–500 microssegundos por operação [55]. Esta camada evita recálculos custosos a cada requisição, materializando resultados após cada ciclo de sincronização.

### 5.2.3 Frontend em Next.js

A escolha de **Next.js 15** justifica-se pela necessidade crítica de indexação por mecanismos de busca [56]. Diferentemente de SPAs que renderizam conteúdo apenas via JavaScript, Next.js oferece Server-Side Rendering (SSR) e Static Site Generation (SSG), entregando HTML pré-renderizado aos crawlers — requisito fundamental para plataformas de fiscalização que dependem de descoberta orgânica.

A biblioteca **Recharts** foi selecionada para visualização de dados por sua integração declarativa com React. **Tailwind CSS 4** permite desenvolvimento ágil com classes utilitárias e geração otimizada de CSS em produção [57].

A escolha do **Bun** como *runtime* JavaScript fundamenta-se em estudos comparativos recentes que demonstram seu desempenho superior em operações síncronas e processamento JSON [48]. Diferentemente do Node.js, que utiliza o motor V8, o Bun é construído sobre o JavaScriptCore (WebKit), resultando em tempos de inicialização significativamente menores — característica vantajosa para o ciclo de desenvolvimento iterativo. Além do *runtime*, o Bun incorpora gerenciador de pacotes nativo, eliminando dependências externas como npm ou Yarn e simplificando o *pipeline* de *build*.

## 5.3 Estratégia de Ingestão de Dados

A consolidação de dados dispersos em três APIs governamentais representa um dos maiores desafios técnicos do projeto. O processo segue o paradigma *Extract, Transform, Load* (ETL), que conforme Vassiliadis [43] tipicamente consome entre 60% e 80% do esforço total em projetos de data warehousing.

A estratégia adotada combina duas abordagens complementares: **backfill** para carga histórica inicial e **sincronização contínua** para manutenção incremental. Conforme Kimball e Ross [??], operações de carga histórica devem privilegiar throughput sobre latência, utilizando arquivos CSV consolidados e bulk inserts. As fontes históricas incluem CEAPS (2008–2025, 500 mil registros), votações (2019–presente) e emendas (2015–presente).

A sincronização contínua opera via polling periódico, calibrado pela volatilidade de cada domínio: votações diárias às 03:00 BRT (baixo tráfego nas APIs), remunerações semanais aos domingos, e emendas mensalmente após fechamento contábil.

Para garantir consistência, o sistema opera sob modelo *at-least-once* com **idempotência** via chaves naturais compostas [58]. Falhas transientes são tratadas com retry exponencial ( $t_{wait} = base \times 2^{tentativa}$ ) conforme recomendação AWS [59], timeout de 30 segundos e falha graciosa que permite continuidade da sincronização mesmo com fontes temporariamente indisponíveis.

## 5.4 Modelo de Dados

O modelo relacional constitui o alicerce sobre o qual todas as funcionalidades são construídas. O projeto priorizou dois objetivos: garantir integridade referencial durante ingestão

concorrente, e otimizar consultas analíticas frequentes como agregações de gastos por senador, período e categoria.

A Figura 2 apresenta o diagrama entidade-relacionamento.

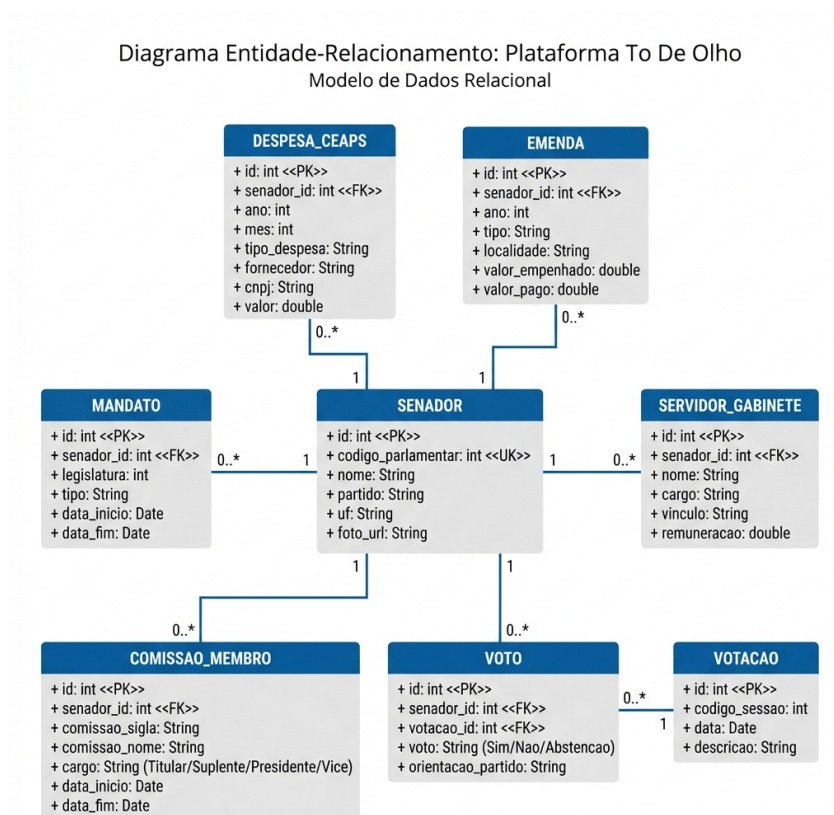


Figura 2: Diagrama entidade-relacionamento do sistema Tô De Olho  
Fonte: Autoria Própria

As entidades organizam-se em torno de **Senador**, a entidade central identificada por código parlamentar único. Relacionamentos 1:N conectam o senador a: **Mandato** (legislatura, tipo, período), **Despesa CEAPS** (com chave composta para idempotência), **Votação** (tabela associativa com sessão), **Servidor de Gabinete** (nome, cargo, remuneração), **Emenda** (incluindo Transferências Especiais) e **Comissão Membro** (participação com cargo e período).

Para otimizar consultas frequentes, foram criados os seguintes índices compostos:

- `idx_despesa_senador_ano` — totalização de gastos por período;
- `idx_votacao_sessao_senador` — cálculo de presença em votações;
- `idx_emenda_senador_tipo` — filtros por modalidade de emenda;
- `idx_comissao_senador` — agregação de participação em comissões.

## 6 Testes de Software

A garantia da qualidade em sistemas de transparência pública assume papel crítico, dado que falhas na apresentação de dados podem comprometer a confiança do cidadão nas instituições monitoradas. A estratégia de validação do *Tô De Olho* fundamentou-se no modelo da Pirâmide de Testes [60], priorizando uma base sólida de testes automatizados de baixo nível para garantir *feedback* rápido durante o ciclo de desenvolvimento.

A implementação seguiu as diretrizes de *Continuous Testing*, integrando as rotinas de verificação ao *pipeline* de entrega contínua. Para isso, utilizou-se o ferramental nativo da linguagem Go, complementado por bibliotecas específicas para cenários de integração.

### 6.1 Testes Unitários

Os testes unitários constituem a primeira linha de defesa contra regressões, focando na validação isolada de componentes de lógica de negócio e utilitários. A escolha da linguagem Go favoreceu esta prática através de seu suporte nativo no pacote `testing`, que elimina a necessidade de *frameworks* externos complexos.

Adotou-se o padrão de *Table-Driven Tests* (Testes Orientados a Tabelas), idiomático em Go. Esta técnica permite testar múltiplos cenários (casos de sucesso, erros de borda, entradas inválidas) reutilizando a mesma lógica de asserção, o que aumenta a legibilidade e facilita a manutenção.

- **Cálculo de Scores:** Validação dos algoritmos de pontuação dos senadores, garantindo que os pesos da metodologia SLES (Seção 3) sejam aplicados corretamente sobre os dados brutos.
- **Parsers de Dados:** Verificação das rotinas de tratamento de dados provenientes das APIs do Senado, assegurando a correta conversão de formatos (XML, JSON) e tipos numéricos monetários.

A cobertura de código (*code coverage*) foi monitorada com a ferramenta nativa `go test -cover`, estabelecendo-se uma meta mínima de 80% para pacotes críticos do domínio.

### 6.2 Testes de Integração

Diferentemente dos testes unitários, que utilizam *mocks* para isolar dependências, os testes de integração validam a interação entre os módulos do sistema e componentes de infraestrutura real, como o banco de dados PostgreSQL e o cache Redis.

Para viabilizar esses testes de forma isolada e reproduzível, utilizou-se a biblioteca `testcontainers-go`. Esta ferramenta permite instanciar contêineres Docker descartáveis durante a execução da suíte de testes.

1. **Persistência de Dados:** Testes que verificam se as operações de escrita e leitura no PostgreSQL respeitam as restrições de integridade referencial e unicidade, essenciais

para a idempotência da ingestão.

2. **Mecanismos de Cache:** Validação das estratégias de invalidação e expiração de chaves no Redis, garantindo que o usuário final não visualize dados obsoletos.

## 6.3 Testes de Contrato

Considerando que o *Tô De Olho* depende integralmente da estabilidade de APIs de terceiros (Senado e Portal da Transparência), implementaram-se testes de contrato para detectar mudanças não anunciadas nos formatos de resposta (*breaking changes*).

Estes testes realizam requisições controladas às APIs externas e validam se a estrutura do JSON/XML retornado corresponde aos esquemas esperados pelos *structs* da aplicação. Falhas nesta camada disparam alertas para a equipe de desenvolvimento, sinalizando a necessidade de adaptação nos adaptadores de integração antes que o erro impacte o ambiente de produção.

## 7 Implantação e Infraestrutura

A arquitetura de implantação do *Tô De Olho* foi projetada para garantir alta disponibilidade, escalabilidade elástica e baixo custo operacional, alinhando-se aos princípios da metodologia *Twelve-Factor App* [61]. Balalaie et al. argumentam que a adoção de arquiteturas orientadas a serviços e práticas de DevOps é crucial para permitir o desenvolvimento ágil e a entrega contínua de valor em sistemas complexos [62]. Seguindo essa premissa, a infraestrutura baseia-se integralmente em serviços gerenciados na nuvem (*Cloud Native*), eliminando a necessidade de provisionamento e manutenção de servidores físicos ou virtuais tradicionais.

### 7.1 Containerização

A portabilidade da aplicação é assegurada pelo uso de contêineres Docker. O processo de construção das imagens utiliza a técnica de *Multi-Stage Builds*, recomendada pelas melhores práticas de segurança [63]. Esta abordagem divide o processo em dois estágios:

1. **Estágio de Build:** Utiliza uma imagem base completa (`golang:1.21-alpine`) contendo compiladores e ferramentas de construção necessárias para gerar o binário estático da aplicação.
2. **Estágio de Runtime:** Copia apenas o binário compilado para uma imagem minimalista (`gcr.io/distroless/static`), isenta de gerenciadores de pacotes ou *shell*.

O resultado são imagens finais extremamente leves (aproximadamente 25MB) e com superfície de ataque reduzida, visto que não contêm ferramentas que poderiam ser exploradas por atacantes.



## 7.2 Pipeline de Integração e Entrega Contínua (CI/CD)

A automação do ciclo de vida do software é gerenciada via *GitHub Actions*, configurada para executar pipelines distintos baseados em eventos do repositório. A adoção de CI/CD permite reduzir o tempo de *feedback* e mitigar erros humanos em processos manuais de *deploy* [64]. O fluxo principal (*pipeline* de produção) é composto pelas seguintes etapas:

- **Verificação (Linting & Vet):** Análise estática do código para garantir conformidade com os padrões de estilo da linguagem Go e detecção prévia de construções suspeitas.
- **Testes Automatizados:** Execução de todas as suítes de testes unitários e de integração. A falha em qualquer teste bloqueia imediatamente o processo de entrega.
- **Build e Push:** Construção da imagem Docker e publicação no *Google Artifact Registry*, versionada com o *hash* do *commit* (*SHA*).
- **Deploy:** Atualização do serviço no Google Cloud Run, utilizando a estratégia de *rolling update* para garantir zero *downtime* durante a transição de versões.

## 7.3 Infraestrutura Serverless no Google Cloud

O ambiente de execução principal é o **Google Cloud Run**, uma plataforma de computação *serverless* que abstrai a complexidade de gerenciamento de infraestrutura Kubernetes [65]. Esta escolha arquitetural oferece vantagens estratégicas para o projeto:

- **Escalabilidade Automática:** O serviço ajusta automaticamente o número de instâncias com base no tráfego de requisições, podendo escalar a zero (shutdown total) em momentos de inatividade, otimizando custos.
- **Resiliência e Graceful Shutdown:** A aplicação implementa rotinas de desligamento gracioso para lidar com os sinais de interrupção (*SIGTERM*) da plataforma. Isso garante que requisições em andamento sejam finalizadas e conexões com o banco de dados sejam encerradas corretamente antes da destruição do contêiner [66].

Para mitigar o problema de *Cold Start* (latência inicial na subida de novas instâncias) [67], característico de arquiteturas *serverless*, utilizam-se otimizações como a inicialização preguiçosa (*lazy initialization*) de conexões pesadas e configurações de *min-instances* em períodos críticos. Adicionalmente, estratégias de resiliência como *Exponential Backoff* com *Jitter* são empregadas na comunicação entre serviços para evitar o efeito de manada (*thundering herd*) em caso de falhas transientes, conforme recomendado pela arquitetura de referência da AWS [59]. Para evoluções futuras, planeja-se a implementação do padrão *Circuit Breaker* [68] e o aprofundamento da observabilidade distribuída baseada nos conceitos de *Dapper* [69] e nos três pilares de monitoramento modernos [70].

## 7.4 Estratégia Mobile-First e Acessibilidade

A decisão de priorizar a experiência em dispositivos móveis (*Mobile-First*) fundamenta-se na realidade do acesso à internet no Brasil. Dados da PNAD Contínua TIC 2024 revelam que o

celular é o dispositivo de acesso exclusivo para a grande maioria dos internautas brasileiros, atingindo 99% de penetração nas classes D e E [71]. Portanto, garantir uma interface leve, responsiva e performática em conexões móveis não é apenas uma escolha técnica de *frontend*, mas um requisito de inclusão digital indispensável para uma ferramenta de controle social.

## 8 Manual do Usuário Simplificado

O *Tô De Olho* foi concebido sob a premissa de máxima acessibilidade, removendo barreiras de entrada para o cidadão comum. Diferentemente de sistemas governamentais tradicionais, a plataforma não exige cadastro, *login* ou conhecimentos técnicos prévios. Esta seção descreve os fluxos principais de interação, desenhados para permitir uma fiscalização intuitiva e eficiente.

### 8.1 Acesso e Visão Geral

Ao acessar o endereço <https://todeolho.org>, o usuário é recepcionado pela **Página Inicial**, que atua como um painel de síntese ("*dashboard*"). O destaque visual é dado aos rankings dinâmicos, apresentando os "Top 3" senadores em critérios positivos, como maior economia na Cota Parlamentar e maior assiduidade em votações (Figura 3).

Figura 3: Página Inicial com Rankings e Destaques



Fonte: O Autor (2026).

1. **Barra de Busca Global:** Localizada no topo de todas as páginas, permite localizar senadores por nome, apelido parlamentar ou estado (UF). A busca possui recurso de autocompletar, facilitando a navegação.
2. **Mapa Interativo:** Visualização geográfica que permite filtrar senadores clicando diretamente nos estados do mapa do Brasil (Figura 4).

Figura 4: Recursos de Busca e Navegação por Mapa



Fonte: O Autor (2026).

## 8.2 Análise do Perfil Parlamentar

Ao selecionar um senador, o cidadão tem acesso à **Ficha Parlamentar**, estruturada em abas temáticas para evitar sobrecarga cognitiva:

- **Visão Geral:** Apresenta o "Score de Efetividade"(gráfico radar) e dados biográficos essenciais.
- **Análise de Gastos:** Gráficos de evolução temporal das despesas da CEAPS. É possível "mergulhar"nos dados (*drill-down*) clicando nas barras do gráfico para visualizar a lista detalhada de notas fiscais daquele mês.
- **Emendas e Recursos:** Exibe o destino das emendas parlamentares. O destaque

é o mapa de "Emendas PIX", onde cada ponto representa um município beneficiado, permitindo ao usuário verificar se recursos foram enviados para sua cidade sem finalidade definida.

- **Votações:** Histórico de posicionamentos do senador (Sim/Não/Abstenção/Ausência) em votações nominais no plenário.

A Figura 5 ilustra a consolidação dessas informações na interface do usuário.

Figura 5: Ficha Parlamentar: Visão Geral e Gráficos



Fonte: O Autor (2026).

### 8.3 Ferramentas de Compartilhamento

Reconhecendo o papel das redes sociais na disseminação política, todas as visualizações (gráficos e rankings) possuem botões de compartilhamento direto para WhatsApp, Twitter/X e

Facebook. Ao clicar, o sistema gera automaticamente uma imagem sintetizada ("card") com as estatísticas do senador, facilitando a viralização de informações factuais e a fiscalização coletiva.

## 9 Considerações Finais

Este trabalho apresentou o *Tô De Olho*, uma plataforma *web* que centraliza e democratiza o acesso aos dados do Senado Federal. A arquitetura de **monolito modular** em Go, combinada com a ingestão de dados via APIs oficiais, mostrou-se adequada para consolidar informações dispersas em três fontes distintas. O *front-end* em Next.js oferece ao cidadão uma interface acessível para fiscalizar despesas da CEAPS, acompanhar votações nominais e avaliar o desempenho dos 81 senadores por meio de um ranking com metodologia transparente.

Uma limitação relevante deste trabalho é a ausência do módulo de fórum para debate cívico, inicialmente planejado mas não implementado devido a restrições de prazo. Conforme aponta Costa [72], a gestão pública digital efetiva requer não apenas transparência, mas também canais de participação direta. O *Tô De Olho* avança significativamente na dimensão da *accountability*, porém ainda não contempla espaços deliberativos.

A principal barreira encontrada durante o desenvolvimento, contudo, transcendeu os desafios técnicos de ingestão de dados. A tradução da complexidade do Regimento Interno do Senado em regras de negócio determinísticas revelou-se um exercício de contínuo compromisso. Simplificar a atuação parlamentar — intrinsecamente política e negociada — em métricas quantitativas impôs reducionismos necessários ao modelo de dados, mas que devem ser interpretados com cautela. A experiência reforçou que ferramentas de *civic tech* são meios, e não fins; elas iluminam dados, mas a interpretação do contexto permanece uma atribuição insubstituível do cidadão.

Para trabalhos futuros, sugere-se: (1) a implementação do Fórum de Cidadania para debate qualificado; (2) a expansão do escopo para incluir a Câmara dos Deputados, tornando a plataforma bicameral; e (3) a aplicação de técnicas de aprendizado de máquina para detecção de padrões anômalos em despesas parlamentares, seguindo o exemplo da Operação Serenata de Amor. Essas evoluções fortaleceriam o controle social e aproximariam a ferramenta dos degraus superiores da Escada de Arnstein. Por fim, ressalta-se o potencial impacto da ferramenta no pleito de 2026. Ao fornecer um histórico estruturado da atuação parlamentar, o *Tô De Olho* pode qualificar o debate público e auxiliar eleitores em suas decisões de voto.

## Agradecimentos

Agradeço primeiramente a Deus, por ter permitido que meu caminho fosse o melhor possível, trazendo boas vibrações e oportunidades que jamais teria sem o Seu amor único e incondicional.

À minha mãe, Veruschka, minha grande heroína. Ela sempre fez de tudo para que seu filho único se desenvolvesse e nunca soltou a minha mão, nem mesmo nos momentos mais difíceis.

Sua força ao “segurar as pontas” e seu incentivo constante para que eu estudasse e crescesse foram fundamentais.

À minha esposa Priscila, pelo amor, confiança e parceria inabaláveis. Você sempre me motivou a seguir em frente com o TCC, apoiando-me mesmo nos momentos de menor produtividade.

Ao Governo Federal e ao IFBA, por proporcionarem e manterem uma educação de extrema qualidade, gratuita e acessível, base da minha formação desde o ensino técnico em Mecânica até o curso superior em ADS. A toda a comunidade acadêmica de ADS do IFBA, agradeço pelo ensino, vivências e conhecimentos compartilhados.

Ao meu orientador, Pablo Vieira Florentino, pelos incentivos, motivações e pelas conversas enriquecedoras. Agradeço a paciência durante as alterações de tema e, principalmente, por sempre ter acreditado no meu potencial.

Ao professor Manoel Neto, pela disponibilidade constante em apoiar minha formação, pela torcida e pela grande amizade construída desde a disciplina de Programação Web.

Aos amigos Ricardo, Camila, Victor e Yan, parceiros da faculdade, que sempre me impulsionaram a concluir este projeto. Os “puxões de orelha” foram essenciais e deram certo.

Por fim, agradeço a mim mesmo pela resiliência. Apesar das mudanças de tema e dos períodos de desmotivação, persisti para fechar este ciclo com um projeto de relevância política ímpar, pronto para impactar o contexto das eleições de 2026.

## Referências

- 1 GOMES, W. Democracia digital: Que democracia? **Compolítica**, 2010. Disponível em: [http://www.compolitica.org/home/wp-content/uploads/2011/01/gt\\_ip-wilson.pdf](http://www.compolitica.org/home/wp-content/uploads/2011/01/gt_ip-wilson.pdf).
- 2 AVELINO, D. P. de; POMPEU, J. C. B.; FONSECA, I. F. da. Democracia digital: mapeamento de experiências em dados abertos, governo digital e ouvidorias públicas. **Instituto de Pesquisa Econômica Aplicada (Ipea)**, 2021.
- 3 PATEMAN, C. **Participation and Democratic Theory**. [S.l.: s.n.], 1970.
- 4 GOMES, W. **A democracia no mundo digital**. [S.l.]: Sesc, 2019.
- 5 ALENCAR, H. N. **O problema da falta de transparência das “emendas PIX” no orçamento constitucional brasileiro**. Tese (Tese de Doutorado em Direito Constitucional) — Instituto Brasileiro de Ensino, Desenvolvimento e Pesquisa (IDP), Brasília, 2024. Orientação: Profa. Dra. Mariana Barbosa Cirne.
- 6 Ação Educativa; Instituto Paulo Montenegro. **Indicador de Alfabetismo Funcional (INAF) 2024**. São Paulo, 2024. 29% da população brasileira entre 15 e 64 anos é funcionalmente analfabeta.
- 7 VOLDEN, C.; WISEMAN, A. E. Legislative effectiveness in the american states. **American Political Science Review**, Cambridge University Press, v. 112, n. 4, p. 1–17, 2018. Metodologia

do State Legislative Effectiveness Score (SLES).

8 HULLMAN, J.; DIAKOPOULOS, N. Visualization rhetoric: Framing effects in narrative visualization. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 17, n. 12, p. 2231–2240, 2011.

9 ARNSTEIN, S. R. A ladder of citizen participation. **Journal of the American Institute of Planners**, v. 35, n. 4, p. 216–224, 1969.

10 Controladoria-Geral da União. **Portal da Transparência do Governo Federal**. 2024. <<https://portaldatransparencia.gov.br>>. Lançado em novembro de 2004, reformulado em 2018, registra 1,3 a 1,5 milhão de usuários mensais.

11 Brasil. **Lei n. 12.527, de 18 de novembro de 2011 (Lei de Acesso à Informação)**. 2011. <[https://www.planalto.gov.br/ccivil\\_03/\\_ato2011-2014/2011/lei/l12527.htm](https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm)>. Regula o acesso a informações públicas previsto na Constituição Federal.

12 Senado Federal. **Portal de Dados Abertos do Senado Federal**. 2024. <<https://dadosabertos.senado.leg.br>>. Institucionalizado pelo Ato da Comissão Diretora n. 14 de 2013.

13 Câmara dos Deputados. **Portal de Dados Abertos da Câmara dos Deputados**. 2024. <<https://dadosabertos.camara.leg.br>>. APIs disponíveis desde 2006 (SIT Câmara), modernizado em 2017 com API RESTful v2.

14 ALBUQUERQUE, O. J. d.; ALMEIDA, B. d.; COSTA, L. V. Valor público por meio de tecnologias desenvolvidas com dados governamentais abertos: o caso da operação serenata de amor. **Revista de Administração Pública**, SciELO Brasil, v. 52, n. 4, p. 610–629, 2018.

15 De Olho no Congresso. **De Olho no Congresso: Fiscalize os Gastos dos Deputados Federais**. 2024. <<https://deolhonocongresso.com.br>>. Plataforma de transparência focada em gastos parlamentares da Câmara.

16 De Olho em Você. **De Olho em Você: Transparência que Dá para Entender**. 2024. <<https://deolhoemvoce.com.br>>. Plataforma de transparência focada em Deputados Federais, com destaque para Emendas PIX e mapas de distribuição.

17 mySociety. **mySociety: Digital Tools for Democracy**. 2024. <<https://www.mysociety.org>>. Organização britânica pioneira em civic tech, responsável pelo TheyWorkForYou.

18 \_\_\_\_\_. **TheyWorkForYou: Parliamentary Monitoring**. 2024. <<https://www.theyworkforyou.com>>. Plataforma de monitoramento parlamentar do Reino Unido, ativa desde 2004.

19 OpenSecrets. **OpenSecrets: Following the Money in Politics**. 2024. <<https://www.opensecrets.org>>. Organização não-partidária que rastreia financiamento de campanhas nos EUA.

20 MICHENER, G.; CONTRERAS, E.; NISKIER, I. From opacity to transparency? evaluating access to information in brazil five years later. **Revista de Administração Pública**, SciELO Brasil, v. 52, n. 4, p. 610–629, 2018. Avaliação da implementação da Lei de Acesso à Informação (LAI) no Brasil.

21 O'REILLY, T. Government as a platform. In: LATHROP, D.; RUMA, L. (Ed.). **Open Government: Collaboration, Transparency, and Participation in Practice**. Sebastopol, CA: O'Reilly Media, 2010. p. 11–40. ISBN 978-0596804350. Ensaio seminal que cunhou o termo “Government as a Platform” (GaaP).



- 22 BERNERS-LEE, T. **Linked Data - Design Issues**. 2010. <<https://www.w3.org/DesignIssues/LinkedData.html>>. Princípios fundamentais de Linked Data e esquema de 5 estrelas para dados abertos.
- 23 Open Government Partnership. **Brazil: Open Government Partnership Member Page**. 2024. <<https://www.opengovpartnership.org/members/brazil/>>. Brasil membro desde 2011, com 6 planos de ação e 130 reformas implementadas.
- 24 SAIKALI, L. B. Cidades inteligentes para todos: o desafio de reduzir a desigualdade social diante da exclusão digital. **International Journal of Digital Law**, 2021.
- 25 LEMOS, A. Cidades inteligentes. **GV-EXECUTIVO**, v. 12, n. 2, p. Art. 2, abril 2013.
- 26 FARIAS, V. V. M. As possibilidades da democracia digital no brasil. **Mídias e direitos da sociedade em rede**, v. 2, p. 495–509, 2013.
- 27 MOE, T. M. The new economics of organization. **American Journal of Political Science**, v. 28, n. 4, p. 739–777, 1984. Artigo seminal sobre teoria de agência aplicada a organizações públicas.
- 28 Open Government Partnership. **Parliamentary Openness, Transparency and Accountability**. [S.l.], 2022. Disponível em: <<https://www.opengovpartnership.org>>.
- 29 VERHULST, S. G.; YOUNG, A. Open data impact: When demand and supply meet. **GovLab**, NYU Governance Lab, 2016. Estudo sobre impacto de dados abertos governamentais.
- 30 LATHROP, D.; RUMA, L. (Ed.). **Open Government: Collaboration, Transparency, and Participation in Practice**. Sebastopol, CA: O'Reilly Media, 2010. 402 p. Coleção de ensaios sobre governo aberto, incluindo contribuições de Tim O'Reilly, Beth Noveck e Aaron Swartz. ISBN 978-0596804350.
- 31 Transparência Brasil. **Transparência Brasil**. 2024. <<https://www.transparencia.org.br>>. Organização fundada em 2000, dedicada à promoção da integridade no setor público.
- 32 Base dos Dados. **De Olho na Câmara: Dados Abertos da Câmara dos Deputados**. 2024. <<https://basedosdados.org>>. Plataforma que facilita acesso a dados legislativos da Câmara dos Deputados.
- 33 Colab. **Colab: Tecnologia Cívica para Cidades Inteligentes**. 2024. <<https://colab.com.br>>. Plataforma brasileira de civic tech para participação cidadã municipal, fundada em 2013.
- 34 Governo Federal do Brasil. **Portal Brasileiro de Dados Abertos**. 2024. <<https://dados.gov.br>>. Ferramenta central para acesso a dados públicos do governo federal.
- 35 VOLDEN, C.; WISEMAN, A. E. **Legislative Effectiveness in the United States Congress: The Lawmakers**. [S.l.]: Cambridge University Press, 2014. Obra fundamental sobre metodologia do Legislative Effectiveness Score.
- 36 Center for Effective Lawmaking. **The Center for Effective Lawmaking**. 2024. <<https://thelawmakers.org>>. Centro de pesquisa co-dirigido por Volden e Wiseman, mantendo dados de efetividade legislativa.
- 37 DRAGONI, N. et al. Microservices: Yesterday, today, and tomorrow. **Present and Ulterior Software Engineering**, Springer, p. 195–216, 2017. Discussão de trade-offs entre

microsserviços e monólitos.

38 LAIGNER, R.; KALINOWSKI, M. et al. A systematic literature review on modular monolith architecture. **Applied Sciences**, MDPI, v. 14, n. 8, p. 3234, 2024. Revisão sistemática sobre arquitetura de monólito modular.

39 EVANS, E. **Domain-Driven Design: Tackling Complexity in the Heart of Software**. [S.l.]: Addison-Wesley, 2003. Obra fundamental sobre design orientado a domínio e bounded contexts. ISBN 978-0321125217.

40 Google. **Service Weaver: A Programming Framework for Writing Distributed Applications**. 2024. <<https://serviceweaver.dev>>. Framework que suporta desenvolvimento modular com deploy flexível.

41 VMware. **Spring Modulith: Building Well-Structured Spring Boot Applications**. 2024. <<https://spring.io/projects/spring-modulith>>. Framework para construção de aplicações modulares em Java.

42 KIMBALL, R.; CASERTA, J. **The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data**. [S.l.]: Wiley, 2004. Obra de referência para processos ETL em data warehousing. ISBN 978-0764567575.

43 VASSILIADIS, P. A survey of extract–transform–load technology. **International Journal of Data Warehousing and Mining**, IGI Global, v. 5, n. 3, p. 1–27, 2009. Revisão abrangente de tecnologias e técnicas ETL.

44 PostgreSQL Global Development Group. **PostgreSQL: The World's Most Advanced Open Source Relational Database**. 2024. <<https://www.postgresql.org>>. Banco de dados relacional com conformidade ACID e suporte a consultas analíticas.

45 KIMBALL, R.; ROSS, M. **The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling**. 3rd. ed. [S.l.]: Wiley, 2013. Guia definitivo de modelagem dimensional e arquitetura bottom-up. ISBN 978-1118530801.

46 HEVNER, A. R. et al. Design science in information systems research. **MIS Quarterly**, Management Information Systems Research Center, v. 28, n. 1, p. 75–105, 2004. Artigo seminal sobre Design Science Research em sistemas de informação.

47 PENG, S. et al. The impact of ai on developer productivity: Evidence from github copilot. **arXiv preprint arXiv:2302.06590**, 2023. Estudo controlado randomizado com 95 desenvolvedores mostrando 55.8% aumento na velocidade de conclusão de tarefas.

48 SMIRNOV, A. A. et al. A comparative analysis of the performance of javascript code execution environments: Node.js, deno and bun. **Russian Journal of Computer Science and Information Science (RCSI)**, April 2024. Estudo comparativo demonstrando desempenho superior do Bun em operações síncronas e processamento JSON.

49 ISO/IEC/IEEE. **ISO/IEC/IEEE 29148:2018 - Systems and software engineering – Life cycle processes – Requirements engineering**. 2018. <<https://www.iso.org/standard/72089.html>>. Padrão internacional que substitui IEEE 830-1998 para especificação de requisitos de software.

50 ISO/IEC. **ISO/IEC 25010:2011 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models**. 2011. <<https://www.iso.org/standard/35733.html>>. Define oito características de

qualidade de produto de software: funcionalidade, eficiência, compatibilidade, usabilidade, confiabilidade, segurança, manutenibilidade e portabilidade.

51 FOWLER, M. Monolithfirst. **martinfowler.com**, 2015. Artigo seminal defendendo a estratégia de iniciar projetos como monolitos antes de microserviços. Disponível em: <https://martinfowler.com/bliki/MonolithFirst.html>.

52 NANZ, S.; FURIA, C. A. A comparative study of programming languages in rosetta code. In: IEEE. **2015 IEEE/ACM 37th IEEE International Conference on Software Engineering**. [S.l.], 2015. p. 778–788. Estudo comparativo de linguagens incluindo Go e Python.

53 ALFIAN, M. et al. Analyzing the performance of golang web frameworks utilizing gorm in the oil and gas industry. In: IEEE. **2024 IEEE 9th International Conference on Information Technology and Digital Applications (ICITDA)**. [S.l.], 2024. Avaliação de performance de frameworks Go incluindo Gin.

54 STONEBRAKER, M.; ROWE, L. A. The design of postgres. **ACM SIGMOD Record**, ACM, v. 15, n. 2, p. 340–355, 1986. Paper fundacional do PostgreSQL, apresentado na UC Berkeley.

55 Redis Ltd. **Redis: The Real-time Data Platform**. 2024. <https://redis.io>. Banco de dados em memória com latências em microssegundos.

56 SALIM, M. et al. Evaluating the efficacy of next.js: A comparative analysis with react.js on performance, seo, and global network equity. **arXiv preprint arXiv:2403.13350**, 2024. Comparação de performance e SEO entre Next.js e React.

57 Tailwind Labs. **Tailwind CSS: A Utility-First CSS Framework**. 2024. <https://tailwindcss.com>. Framework CSS com abordagem utility-first e purge automatico.

58 HUMMER, W. et al. A model-driven approach for document-orientated restful apis. In: IEEE. **2013 IEEE 6th International Conference on Service-Oriented Computing and Applications**. [S.l.], 2013. p. 247–254. Discute idempotência como propriedade essencial para consistência em APIs REST.

59 Amazon Web Services. **Exponential Backoff And Jitter**. 2015. <https://aws.amazon.com/blogs/architecture/exponential-backoff-and-jitter/>. Best practice da AWS para retry com backoff exponencial e jitter.

60 MYERS, G. J.; SANDLER, C.; BADGETT, T. **The Art of Software Testing**. 3rd. ed. Hoboken, NJ: John Wiley & Sons, 2011. Obra clássica sobre fundamentos de teste de software, definindo a Pirâmide de Testes e distinção entre verificação e validação. ISBN 978-1118031964.

61 WIGGINS, A. **The Twelve-Factor App**. 2017. <https://12factor.net>. Metodologia para construção de aplicações cloud-native, incluindo logs como streams e processos stateless.

62 BALALAIE, A.; HEYDARNOORI, A.; JAMSHIDI, P. Microservices architecture enables devops: Migration to a cloud-native architecture. In: **IEEE Software**. [S.l.]: IEEE, 2016. v. 33, n. 3, p. 42–52. Análise de padrões de microserviços incluindo Circuit Breaker e API Gateway.

63 Google Cloud. **Best Practices for Building Containers**. 2024. <https://cloud.google.com/architecture/best-practices-for-building-containers>. Guia oficial do Google sobre otimização de imagens Docker.

64 KINSMAN, T. et al. How do software developers use github actions to automate their

workflows? In: **2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)**. [S.l.]: IEEE, 2021. p. 420–431. Análise empírica de workflows GitHub Actions em mais de 10.000 repositórios.

65 Google Cloud. **Cloud Run: Serverless Container Platform**. 2024. <<https://cloud.google.com/run>>. Plataforma serverless para containers com escala automática e cobrança por uso.

66 NYGARD, M. T. **Release It!: Design and Deploy Production-Ready Software**. 2nd. ed. [S.l.]: Pragmatic Bookshelf, 2018. Obra de referência sobre padrões de estabilidade e resiliência (Circuit Breaker, Bulkheads). ISBN 978-1680502398.

67 SILVA, P.; FIREMAN, D.; PEREIRA, T. E. A survey on serverless computing: Challenges and research opportunities. **Future Generation Computer Systems**, Elsevier, v. 115, p. 213–236, 2020. Revisão sistemática de desafios serverless, incluindo estratégias de mitigação de cold start.

68 MONTESI, F.; WEBER, J. Circuit breakers, discovery, and api gateways in microservices. **arXiv preprint arXiv:1609.05830**, 2016. Análise detalhada do padrão Circuit Breaker em arquiteturas de microserviços.

69 SIGELMAN, B. H. et al. Dapper, a large-scale distributed systems tracing infrastructure. **Google Technical Report**, 2010. Fundamentos de distributed tracing, base para OpenTelemetry e Jaeger.

70 SRIDHARAN, C. Distributed systems observability: A guide to building robust systems. O'Reilly Media, 2018. Formalização dos três pilares: logs, métricas e traces.

71 IBGE. **PNAD Contínua TIC 2024**. [S.l.], 2024. Disponível em: <<https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=2102144>>.

72 COSTA, E. A. D. **Gestão Pública Digital: o poder das TIC na democracia brasileira**. Dissertação (Dissertação de Mestrado em Gestão Pública e Sociedade) — Universidade Federal de Alfenas (UNIFAL-MG), Varginha, MG, 2019. Disponível no Repositório Institucional UNIFAL-MG.

## Apêndice A — Glossário, Siglas e Abreviações

**ACID** *Atomicity, Consistency, Isolation, Durability* — propriedades que garantem integridade em transações de banco de dados.

**ADPF** Arguição de Descumprimento de Preceito Fundamental — ação judicial perante o STF para questionar violações à Constituição.

**API** *Application Programming Interface* — interface que permite comunicação entre sistemas.

**CAE** Comissão de Assuntos Econômicos — comissão permanente do Senado Federal.

**CCJ** Comissão de Constituição, Justiça e Cidadania — comissão permanente do Senado Federal.

**CEAP** Cota para o Exercício da Atividade Parlamentar — verba destinada aos Deputados Federais (equivalente à CEAPS para senadores).

**CEAPS** Cota para o Exercício da Atividade Parlamentar dos Senadores — verba destinada ao custeio de despesas relacionadas ao exercício do mandato parlamentar.

**CGU** Controladoria-Geral da União — órgão responsável pelo controle interno do Poder Executivo Federal e pela transparência pública.

**CI/CD** *Continuous Integration / Continuous Deployment* — práticas de integração e implantação contínuas de software.

**CSS** *Cascading Style Sheets* — linguagem de folhas de estilo utilizada para definir a apresentação de documentos *web*.

**CSR** *Client-Side Rendering* — renderização de páginas no navegador do usuário.

**CSV** *Comma-Separated Values* — formato de arquivo de texto para dados tabulares.

**DSR** *Design Science Research* — abordagem metodológica de pesquisa focada na criação e avaliação de artefatos.

**DVL** *Data Visualization Literacy* — literacia em visualização de dados; capacidade de interpretar representações visuais de informações.

**ETL** *Extract, Transform, Load* — processo de extração, transformação e carga de dados.

**FCP** *First Contentful Paint* — métrica de performance web que mede o tempo até o primeiro conteúdo visível.

**GCP** *Google Cloud Platform* — plataforma de serviços em nuvem do Google.

**HTML** *HyperText Markup Language* — linguagem de marcação padrão para documentos desenhados para serem exibidos em um navegador.

**HTTPS** *HyperText Transfer Protocol Secure* — extensão segura do protocolo HTTP para

comunicação em redes de computadores.

**IDE** *Integrated Development Environment* — ambiente de desenvolvimento integrado; software que auxilia programadores no desenvolvimento de software.

**INAF** Indicador de Alfabetismo Funcional — pesquisa que avalia níveis de alfabetização no Brasil.

**JIT** *Just-In-Time* — compilação sob demanda; no contexto de CSS, geração de estilos apenas quando utilizados.

**JSON** *JavaScript Object Notation* — formato leve de intercâmbio de dados.

**LAI** Lei de Acesso à Informação (Lei nº 12.527/2011) — legislação que garante o direito de acesso a informações públicas.

**LCP** *Largest Contentful Paint* — métrica de performance web que mede o tempo até o maior elemento visível.

**LES** *Legislative Effectiveness Score* — metodologia para avaliação de efetividade legislativa desenvolvida por Volden e Wiseman.

**LGPD** Lei Geral de Proteção de Dados (Lei nº 13.709/2018) — legislação brasileira sobre privacidade e proteção de dados pessoais.

**LLM** *Large Language Model* — modelo de linguagem grande; tipo de inteligência artificial treinado em vastas quantidades de texto.

**MVP** *Minimum Viable Product* — produto mínimo viável, versão inicial com funcionalidades essenciais.

**OGP** *Open Government Partnership* — Parceria para Governo Aberto; iniciativa multilateral para promoção da transparência.

**ORM** *Object-Relational Mapping* — técnica de mapeamento objeto-relacional para persistência de dados.

**PAC** *Political Action Committee* — comitê de ação política nos EUA para arrecadação de fundos eleitorais.

**PEC** Proposta de Emenda à Constituição — visa alterar a Constituição; exige quórum de 3/5 em duas votações.

**PIX** Sistema de pagamentos instantâneos do Banco Central; no contexto parlamentar, refere-se às “Emendas PIX” (Transferências Especiais).

**PL** Projeto de Lei — proposição destinada a dispor sobre matéria de competência normativa do Poder Legislativo.

**PLP** Projeto de Lei Complementar — proposição que regulamenta matérias específicas previstas na Constituição.

**RDF** *Resource Description Framework* — modelo de dados para representação de informações na web semântica.

**REST** *Representational State Transfer* — estilo arquitetural para APIs *web*.

**SEO** *Search Engine Optimization* — otimização para motores de busca.

**SHA** *Secure Hash Algorithm* — família de funções de hash criptográficas.

**SIAFI** Sistema Integrado de Administração Financeira do Governo Federal — sistema de contabilidade pública do governo brasileiro.

**SLES** *State Legislative Effectiveness Score* — versão estadual do LES, adaptada para legislaturas subnacionais.

**SQL** *Structured Query Language* — linguagem de consulta estruturada; padrão para gerenciamento de bancos de dados relacionais.

**SSG** *Static Site Generation* — geração estática de páginas web em tempo de build.

**SSR** *Server-Side Rendering* — renderização de páginas no servidor.

**STF** Supremo Tribunal Federal — órgão máximo do Poder Judiciário brasileiro.

**TIC** Tecnologias da Informação e Comunicação.

**TLS** *Transport Layer Security* — protocolo de segurança para comunicações criptografadas na internet.

**TTL** *Time To Live* — tempo de vida de dados em cache.

**UF** Unidade Federativa — estado brasileiro.

**URI** *Uniform Resource Identifier* — identificador único de recursos na web.

**URL** *Uniform Resource Locator* — endereço de um recurso disponível em uma rede (como a Web).

**W3C** *World Wide Web Consortium* — organização internacional que desenvolve padrões para a web.

**WCAG** *Web Content Accessibility Guidelines* — diretrizes de acessibilidade para conteúdo web.

**XML** *eXtensible Markup Language* — linguagem de marcação para estruturação de dados.