



# 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  
Fevereiro de 2026

# Sumário

<b>1 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 Tecnologias adotadas . . . . .	3
1.5 Trabalhos Relacionados . . . . .	5
1.5.1 Portais Oficiais (Nível Informação) . . . . .	5
1.5.2 Ferramentas de Fiscalização da Câmara dos Deputados . . . . .	5
1.5.3 Experiências Internacionais . . . . .	6
1.5.4 Lacuna Identificada e Diferencial do To De Olho . . . . .	6
<b>2 Fundamentação Teórica</b>	<b>6</b>
2.1 Transparência Pública e Dados Abertos . . . . .	6
2.2 Democracia Digital e Participação Cidadã . . . . .	7
2.3 Métricas de Efetividade Legislativa . . . . .	7
2.4 Visualização de Dados e Retórica Visual . . . . .	8
2.5 Arquitetura de Software: Monolito Modular . . . . .	8
2.6 Engenharia de Dados: APIs e Processos ETL . . . . .	9
2.7 Emendas PIX e Desafios de Transparência . . . . .	9
<b>3 Metodologia</b>	<b>9</b>
3.1 Abordagem de Desenvolvimento . . . . .	9
3.2 Fontes de Dados . . . . .	10
3.2.1 API Legislativa do Senado . . . . .	10
3.2.2 API Administrativa do Senado . . . . .	10
3.2.3 Portal da Transparência (CGU) . . . . .	10
3.3 Estratégia de Ingestão . . . . .	11
3.4 Arquitetura do Sistema . . . . .	11
3.5 Stack Tecnológico . . . . .	11
3.6 Infraestrutura e Implantação . . . . .	12
3.7 Algoritmo de Ranking . . . . .	12
<b>4 Requisitos</b>	<b>13</b>
4.1 Requisitos Funcionais . . . . .	13
4.2 Requisitos Não-Funcionais . . . . .	13
<b>5 Design</b>	<b>14</b>
5.1 Projeto UML . . . . .	14
5.1.1 Diagrama de Classes . . . . .	14
5.1.2 Diagrama de Implantação . . . . .	15
5.2 Visão arquitetural . . . . .	16
5.3 Modelo de Banco de Dados . . . . .	17

<b>6 Testes de Software</b>	<b>18</b>
6.1 Projeto de Testes . . . . .	18
<b>7 Implantação</b>	<b>18</b>
7.1 Projeto de Implantação . . . . .	18
<b>8 Manual do Usuário Simplificado</b>	<b>18</b>
<b>9 Considerações Finais</b>	<b>19</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 de democracia digital evidencia que TICs ampliam possibilidades de participação, mas só geram valor quando articuladas a contextos de uso e inclusão. Avelino et al. mapam iniciativas de dados abertos, reforçando que tecnologias precisam ser mediadas por visualizações claras para efetivação do controle social [2]. Com um corpo legislativo menor e mais “caro” per capita que a Câmara, o Senado carece de ferramentas focadas que cruzem votações nominais com a execução orçamentária de emendas. À luz desses estudos, o *Tô De Olho* procura transformar a transparência passiva em *accountability* ativa, oferecendo rankings e métricas objetivas de desempenho parlamentar [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.

## 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), com 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 *Legislative Effectiveness Score* de Volden e Wiseman [7], o algoritmo de avaliação pondera presença em votações (25%), produtividade legislativa (25%), economia na cota parlamentar (20%), participação em comissões (15%) e transparência de dados (15%). 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 Tecnologias adotadas

A arquitetura do *Tô De Olho* foi projetada seguindo o padrão de **monolito modular** [9], escolhido por sua simplicidade de desenvolvimento e deploy, mantendo a organização interna em módulos bem definidos para garantir manutenibilidade. A seguir, justificamos cada escolha tecnológica.

### **Backend — Golang:**

A linguagem Go foi selecionada para o *backend* após avaliação de alternativas como Node.js e Python. O fator decisivo foi o modelo de **concorrência nativo**: as *goroutines* permitem processar milhares de conexões simultâneas com baixo consumo de memória, característica essencial para a ingestão paralela de dados de três APIs distintas [10]. Adicionalmente, a tipagem estática e a compilação para binário único simplificam o *deploy* em contêineres.

O framework **Gin** foi escolhido por seu roteamento HTTP de alto desempenho [11], enquanto **GORUM** oferece mapeamento objeto-relacional com suporte a migracoes, reduzindo a complexidade de gerenciamento do esquema de banco.

### **Banco de Dados — PostgreSQL e Redis:**

**PostgreSQL** foi selecionado como banco relacional por sua robustez em consultas analíticas e suporte a índices compostos [12], essenciais para agregações por senador, período e tipo de despesa. A conformidade ACID garante integridade nas operações de ingestão.

**Redis** atua como camada de cache para rankings pré-computados e totalizadores de gastos [13]. Com tempos de resposta em microssegundos, elimina a necessidade de recalcular métricas a cada requisição, atendendo ao requisito de latência inferior a 2 segundos (RNF01).

### **Frontend — Next.js 15:**

A escolha de **Next.js** sobre React “puro” justifica-se pela necessidade de **SEO** e performance inicial [14]. Aplicações React tradicionais utilizam renderização no cliente (CSR), onde motores de busca recebem uma página HTML vazia que depende de JavaScript para exibir conteúdo — prejudicando a indexação. Next.js resolve esse problema com *Server-Side Rendering* (SSR) e *Static Site Generation* (SSG), entregando HTML pré-renderizado aos *crawlers*.

A biblioteca **Recharts** foi selecionada para visualização de dados por sua integração nativa com React e suporte a gráficos responsivos [15]. **Tailwind CSS** permite estilização eficiente com classes utilitárias, reduzindo o tamanho do CSS final através de *tree-shaking* [16].

### **Infraestrutura e DevOps:**

O sistema é containerizado com **Docker** e implantado no **Google Cloud Run**, plataforma serverless que oferece escala automática (inclusive a zero, eliminando custos quando inativo). A escolha por Cloud Run ao invés de Kubernetes foi motivada pela simplicidade operacional: um único Dockerfile para toda a aplicação, sem necessidade de gerenciar clusters ou manifestos complexos.

O pipeline de **CI/CD** com GitHub Actions automatiza *build*, testes e *deploy*, garantindo que cada *commit* na branch principal passe por validação antes de chegar à produção.

### **Fontes de Dados:**

O sistema integra três fontes oficiais com características distintas:

- **API Legislativa do Senado:** dados de votações, matérias e comissões em formato JSON, com paginação para grandes volumes;
- **API Administrativa do Senado:** despesas CEAPS disponíveis via endpoint REST, com opção de download em CSV para carga histórica;
- **API do Portal da Transparéncia (CGU):** dados de emendas parlamentares, requerendo autenticação via chave de API.

## 1.5 Trabalhos Relacionados

Diversas iniciativas no Brasil e no mundo buscam promover a transparencia politica por meio da tecnologia. A luz da Escada de Participacao de Arnstein [17], podemos classificar essas ferramentas conforme o grau de poder que conferem ao cidadao.

### 1.5.1 Portais Oficiais (Nivel Informacao)

**Portal da Transparecia (CGU):** Principal ferramenta oficial do governo federal para acesso a dados de gastos publicos, servidores e transferencias. Embora abrangente, sua interface e voltada para consultas tecnicas, exigindo conhecimento previo sobre a estrutura orçamentaria. O cidadao tem acesso aos dados brutos, mas sem ferramentas de analise comparativa ou visualizacoes contextualizadas.

**Portais de Dados Abertos (Camara e Senado):** Disponibilizam APIs e arquivos para download seguindo princípios de *Open Government Data*. Essenciais para desenvolvedores e pesquisadores, porem inacessiveis ao cidadao comum por exigirem conhecimento tecnico. Situam-se no degrau mais basico da informacao — dados brutos sem interpretacao.

### 1.5.2 Ferramentas de Fiscalizacao da Camara dos Deputados

O ecossistema de transparencia para a Camara dos Deputados e mais desenvolvido que para o Senado, contando com diversas iniciativas:

**Operacao Serenata de Amor:** Projeto de codigo aberto que utiliza inteligencia artificial para detectar irregularidades em gastos parlamentares [18]. Composto pela *Rosie* (algoritmo de *machine learning* que analisa notas fiscais) e pelo *Jarbas* (interface web para consulta de suspeitas). Entre 2016 e 2018, analisou mais de 3 milhoes de reembolsos e gerou milhares de denuncias formais ao Congresso.

**De Olho no Congresso:** Plataforma *web* moderna focada em gastos de Deputados Federais [19]. Oferece interface acessivel com: (1) rankings comparativos de gastos por deputado e partido; (2) graficos de evolucao mensal de despesas; (3) **painel de alertas** que sinaliza automaticamente gastos suspeitos — valores acima da media, despesas repetidas, pagamentos em fins de semana; e (4) visualizacao de emendas parlamentares por area. Com mais de 94 mil consultas realizadas, demonstra alta aceitacao entre usuarios. Limita-se, porem, a Camara dos Deputados e nao oferece metricas de desempenho legislativo.

**Base dos Dados — De Olho na Camara:** Plataforma que disponibiliza dados legislativos em *datalake* publico [20]. Publica analises sobre producao legislativa e despesas parlamentares, voltada para analistas e jornalistas de dados. Diferente do *De Olho no Congresso*, seu foco e fornecer dados estruturados para reuso, nao visualizacoes para o cidadao comum.

**De Olho em Voce:** Plataforma de codigo aberto que oferece rankings de gastos e visualizacao de emendas PIX em mapas geoespaciais. Inspirou a concepcao inicial do *To De Olho*, porem nao explicita a metodologia de seus rankings.

### 1.5.3 Experiencias Internacionais

**TheyWorkForYou (Reino Unido):** Desenvolvida pela organizacao mySociety [21], monitora os parlamentos britanicos desde 2004 [22]. Permite aos cidadaos identificar representantes locais, acompanhar votacoes e receber alertas sobre temas especificos. Referencia em *civic tech* parlamentar.

**OpenSecrets (Estados Unidos):** Organizacao nao-partidaria que rastreia financiamento de campanhas e gastos de lobbying desde 1983 [23]. Foco na influencia do dinheiro na politica, consolidou-se como referencia para jornalistas investigativos.

### 1.5.4 Lacuna Identificada e Diferencial do To De Olho

A analise dos trabalhos relacionados revela uma **lacuna significativa**: enquanto a Camera dos Deputados conta com multiplas ferramentas de fiscalizacao (De Olho no Congresso, Serenata de Amor, De Olho em Voce), o **Senado Federal** carece de solucao equivalente. Essa assimetria e agravada pelo fato de cada senador representar maior custo per capita e mandato mais longo (8 anos vs. 4 anos).

O *To De Olho* posiciona-se para preencher essa lacuna, diferenciando-se por:

1. **Foco no Senado Federal:** Casa legislativa menos coberta por ferramentas de fiscalizacao;
2. **Consolidacao Multi-Fonte:** Integra tres APIs distintas (Legislativa, Administrativa e Portal da Transparencia) em interface unica;
3. **Metodologia Transparente:** Inspirado no *Legislative Effectiveness Score* de Volden e Wiseman [7], expoe claramente criterios e pesos do ranking;
4. **Rastreamento de Emendas PIX:** Monitora transferencias especiais, modalidade criada em 2020 com baixa prestacao de contas [5].

## 2 Fundamentação Teórica

Esta seção apresenta os conceitos fundamentais que embasam o desenvolvimento do *Tô De Olho*, abrangendo transparência pública, democracia digital e arquitetura de software.

### 2.1 Transparência Pública e Dados Abertos

A transparéncia governamental constitui pilar fundamental do Estado Democrático de Direito. No Brasil, a Lei de Acesso à Informação (LAI – Lei nº 12.527/2011) estabelece que o acesso é a regra e o sigilo, a exceção, garantindo aos cidadãos o direito de solicitar e receber informações públicas sem necessidade de justificativa [2].

A literatura distingue duas modalidades de transparéncia: a **transparéncia ativa**, na qual o Estado disponibiliza informações de forma proativa em portais e bases de dados; e a **transparéncia passiva**, que responde às solicitações dos cidadãos via canais específicos. O Por-

tal de Dados Abertos do Senado Federal exemplifica a primeira modalidade, disponibilizando APIs e arquivos para consulta pública.

O conceito de *Open Government Data* (Dados Governamentais Abertos) preconiza que as informações públicas devem ser disponibilizadas em formatos abertos, processáveis por máquina e livres de licenças restritivas. Tim Berners-Lee propôs uma escala de cinco estrelas para avaliar a qualidade dos dados abertos, sendo o nível máximo aquele em que os dados são linkados (*Linked Open Data*), permitindo cruzamentos entre diferentes fontes [2].

## 2.2 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]. Gomes identifica três fases históricas neste campo: a teledemocracia (anos 1970-90), marcada por experimentos com televisão interativa; a fase da internet (1995-2005), caracterizada pelo debate sobre potenciais e limites da rede; e a autonomização contemporânea, onde subtemas como governo aberto, *smart cities* e parlamento digital desenvolvem-se de forma independente [4].

A participação cidadã mediada por tecnologia pode assumir diferentes níveis de profundidade. Sherry Arnstein, em seu trabalho seminal de 1969, propõe a “Escada da Participação Cidadã”, uma tipologia de oito degraus que classifica o grau de poder real conferido aos cidadãos [17]. Os degraus inferiores — manipulação e terapia — representam formas de **não-participação**, onde o objetivo é “educar” ou “curar” os participantes em vez de ouvi-los. Os degraus intermediários — informação, consulta e pacificação — constituem níveis de **participação simbólica**, nos quais cidadãos podem ouvir e ser ouvidos, mas sem garantia de que suas vozes influenciem decisões. Apenas nos degraus superiores — parceria, delegação de poder e controle cidadão — observa-se redistribuição efetiva de poder decisório.

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” [17]. Reconhecendo essa limitação, o projeto busca ir além da mera disponibilização de dados, oferecendo *rankings*, comparativos e visualizações que **empoderam** o cidadão para uma fiscalização mais qualificada.

No contexto brasileiro, Avelino et al. mapeiam iniciativas de governo aberto no âmbito federal, identificando avanços significativos na disponibilização de dados, porém alertando para desafios como a brecha digital e o analfabetismo funcional, que limitam o acesso efetivo da população às informações disponibilizadas [2].

## 2.3 Métricas de Efetividade Legislativa

A avaliação quantitativa do desempenho parlamentar constitui tema relevante na ciência política contemporânea. Volden e Wiseman desenvolveram o *Legislative Effectiveness Score*

(LES), uma métrica que mensura a capacidade de parlamentares em conduzir suas proposições através do processo legislativo [7]. O modelo considera múltiplas dimensões: quantidade de projetos apresentados, taxa de aprovação em comissões, progressão para votação em plenário e conversão em lei.

Os autores identificaram fatores correlacionados à maior efetividade: senioridade no mandato, posição em comissões estratégicas, pertencimento ao partido majoritário e experiência prévia em cargos legislativos estaduais. 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 indicadores objetivos como presença em votações, produtividade legislativa (proposições de autoria e relatorias), economia na utilização da cota parlamentar e participação em comissões. A transparência metodológica — expor claramente os critérios e pesos utilizados — é fundamental para que o *ranking* seja percebido como ferramenta de informação, não de manipulação.

## 2.4 Visualização de Dados e Retórica Visual

A apresentação de dados ao cidadão não é neutra: escolhas de *design* influenciam a interpretação das informações. Hullman e 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 [8].

Os autores identificam quatro categorias de técnicas retóricas em visualizações: (1) **proveniência** — identificar a origem e credibilidade dos dados; (2) **mapeamento visual** — como elementos gráficos representam variáveis; (3) **anotações linguísticas** — textos que guiam a interpretação; e (4) **interatividade** — controles que permitem ao usuário explorar os dados por conta própria.

Para o *Tô De Olho*, esses princípios orientam decisões de *design*: exibir sempre a fonte oficial e data de atualização (**proveniência**); utilizar escalas consistentes em gráficos comparativos (**mapeamento**); contextualizar valores absolutos com médias e percentis (**anotação**); e permitir filtros por partido, estado e período (**interatividade**). O objetivo é maximizar a transparência metodológica, evitando que a plataforma seja percebida como veículo de viés político.

## 2.5 Arquitetura de Software: Monolito Modular

A arquitetura de **monolito modular** representa uma abordagem intermediária entre o monolito tradicional e os microsserviços, estruturando uma aplicação como módulos bem definidos dentro de um único artefato de deploy. Esta escolha é particularmente adequada para equipes pequenas e projetos acadêmicos, oferecendo benefícios de organização e manutenibilidade sem a complexidade operacional de sistemas distribuídos.

Os principais benefícios desta arquitetura incluem: simplicidade de deploy, com um único container em ambiente serverless (Cloud Run); baixa latência entre módulos, já que a comunicação ocorre via chamadas de função em memória; e facilidade de evolução, permitindo eventual

migração para microsserviços se a escala justificar.

Para o *Tô De Olho*, esta abordagem organiza a aplicação em módulos internos (ingestão, domínio, API) que podem ser desenvolvidos e testados de forma independente, mantendo a simplicidade operacional necessária para um projeto acadêmico com prazo definido.

## 2.6 Engenharia de Dados: APIs e Processos ETL

A estratégia de ingestão de dados do *Tô De Olho* fundamenta-se no padrão ETL (*Extract, Transform, Load*): extração dos dados brutos das fontes oficiais, transformação para normalização e enriquecimento, e carga no banco de dados da aplicação.

A abordagem adotada consome exclusivamente APIs RESTful oficiais: a API Legislativa do Senado (matérias, votações, comissões), a API Administrativa (CEAPS, remunerações de servidores) e a API do Portal da Transparência (emendas parlamentares). O processo combina *backfill* para carga histórica e sincronização diária via tarefas agendadas.

## 2.7 Emendas PIX e Desafios de Transparência

As Transferências Especiais, popularmente conhecidas como “emendas PIX”, constituem modalidade de repasse de recursos federais a estados e municípios sem necessidade de convênio, criadas pela Emenda Constitucional nº 105/2019. Alencar [5] demonstra sérias deficiências de transparência fiscal nesta modalidade: do total de R\$ 20,5 bilhões transferidos, apenas R\$ 933 milhões (menos de 5%) tiveram prestação de contas adequada.

O autor identifica três vertentes de *accountability* relevantes para a fiscalização parlamentar: a **accountability vertical**, exercida pelo eleitorado nas urnas; a **accountability horizontal**, desempenhada por instituições como tribunais de contas; e a **accountability social**, protagonizada por organizações da sociedade civil e imprensa. A fragmentação das fontes de dados dificulta todas essas formas de controle, evidenciando a necessidade de ferramentas que consolidem informações dispersas.

# 3 Metodologia

Esta seção descreve a metodologia adotada para o desenvolvimento do *Tô De Olho*, detalhando a abordagem de desenvolvimento, as fontes de dados utilizadas, a arquitetura do sistema e a infraestrutura de implantação.

## 3.1 Abordagem de Desenvolvimento

O desenvolvimento do projeto seguiu uma abordagem iterativa e incremental, inspirada em práticas ágeis. O trabalho foi organizado em ciclos de desenvolvimento focados em entregas funcionais, permitindo validação contínua das funcionalidades implementadas.

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;
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 governamentais oficiais, cada uma com características e formatos distintos:

### 3.2.1 API Legislativa do Senado

Disponível em `legis.senado.leg.br/dadosabertos`, esta API RESTful fornece dados do processo legislativo: lista de senadores em exercício, histórico de mandatos, votações nominais em plenário e comissões, proposições de autoria parlamentar, discursos proferidos e composição das comissões. Os dados são retornados em formato JSON, com paginação para grandes volumes.

### 3.2.2 API Administrativa do Senado

Acessível em `adm.senado.gov.br/adm-dadosabertos`, esta interface disponibiliza dados administrativos: lançamentos da Cota para o Exercício da Atividade Parlamentar dos Senadores (CEAPS), informações sobre auxílio-moradia, escritórios de apoio, lista de servidores por lotação e remunerações mensais. Alguns dados sensíveis são disponibilizados apenas em arquivos CSV para *download*.

### 3.2.3 Portal da Transparência (CGU)

A API do Portal da Transparência (`api.portaldatransparencia.gov.br`) fornece dados de emendas parlamentares e transferências da União. Para acesso, é necessária autenticação via chave de API. O filtro `tipoEmenda=Transferência Especial` permite identificar as chamadas “emendas PIX”, modalidade de repasse que dispensa convênio e apresenta menor rastreabilidade.

### 3.3 Estratégia de Ingestão

A ingestão de dados segue uma estratégia híbrida que combina carga inicial massiva com atualização contínua:

**Backfill (Carga Histórica):** Para a população inicial do banco de dados com dados históricos, podem ser utilizados arquivos em massa CSV quando disponíveis. A API Administrativa do Senado fornece endpoint REST para consulta de despesas CEAPS por ano, sendo a fonte principal para sincronização contínua.

**Sincronização Contínua:** Tarefas agendadas (*cron jobs*) executam diariamente a coleta de atualizações via APIs. O serviço de ingestão verifica a data da última atualização de cada entidade e solicita apenas os registros novos ou modificados, otimizando o consumo de recursos e respeitando os limites de requisição das APIs.

### 3.4 Arquitetura do Sistema

O *Tô De Olho* adota uma arquitetura de **monolito modular**, na qual os componentes são organizados internamente em módulos bem definidos:

- *internal/senador*: Gerencia dados cadastrais dos parlamentares e cálculo de rankings;
- *internal/ceaps*: Processa e totaliza despesas da Cota Parlamentar;
- *internal/emenda*: Integra dados de emendas e transferências do Portal da Transparência;
- *internal/votacao*: Coleta e armazena votações nominais;
- *internal/ranking*: Orquestra o cálculo de scores e agregações.

Os módulos internos comunicam-se através de chamadas de função diretas, simplificando o fluxo de dados e facilitando a depuração. Para operações de ingestão, utilizam-se *goroutines* para processamento concorrente.

### 3.5 Stack Tecnológico

A escolha das tecnologias foi orientada por critérios de desempenho, manutenibilidade e adequação ao domínio do problema:

**Backend:** Golang foi selecionada pela sua eficiência em processamento concorrente, característica essencial para a ingestão paralela de múltiplas fontes de dados. O framework Gin provê roteamento HTTP de alto desempenho, enquanto GORM oferece mapeamento objeto-relacional com suporte a migrações.

**Frontend:** Next.js 15 com *App Router* possibilita renderização híbrida (servidor e cliente), otimizando o tempo de carregamento inicial e o SEO. A biblioteca Recharts é utilizada para visualização de dados, permitindo gráficos interativos de evolução de gastos e comparativos entre senadores.

**Banco de Dados:** PostgreSQL armazena os dados de forma estruturada, com índices otimizados para as consultas mais frequentes (agregações por senador, período e tipo de despesa). Redis atua como cache para rankings pré-computados e resultados de consultas complexas.

## 3.6 Infraestrutura e Implantação

Todos os componentes são containerizados com Docker, utilizando *multi-stage builds* para otimização das imagens. A implantação ocorre via Google Cloud Run, que oferece escala automática (inclusive a zero) e deploy simplificado a partir de um único Dockerfile.

O *pipeline* de CI/CD, implementado com GitHub Actions, automatiza as etapas de *build*, testes e *deploy*:

1. **Build:** Compilação dos binários Go e verificação de erros de sintaxe;
2. **Test:** Execução de testes unitários e de integração, utilizando *testcontainers* para instâncias efêmeras de PostgreSQL e Redis;
3. **Publish:** Construção da imagem Docker e envio para o Google Container Registry;
4. **Deploy:** Atualização automática do serviço no Google Cloud Run.

## 3.7 Algoritmo de Ranking

O cálculo do *score* de cada senador baseia-se em cinco critérios objetivos, ponderados conforme sua relevância para a fiscalização cidadã:

- **Presença em Votações (25%):** Percentual de participação nas votações nominais em plenário;
- **Produtividade Legislativa (25%):** Quantidade de proposições de autoria e relatorias assumidas;
- **Economia na Cota (20%):** Relação entre o valor utilizado e o teto disponível da CEAPS;
- **Participação em Comissões (15%):** Envolvimento efetivo nas comissões permanentes e temporárias;
- **Transparência (15%):** Disponibilidade de informações públicas sobre o mandato.

Cada métrica é normalizada em uma escala de 0 a 100 antes do cálculo final, permitindo comparabilidade entre critérios de natureza distinta. A fórmula resultante é:

$$Score = (\text{Presença} \times 0.25) + (\text{Produtividade} \times 0.25) + (\text{Economia} \times 0.20) + (\text{Comissões} \times 0.15) + (\text{Transparência} \times 0.15)$$

Os rankings são recalculados diariamente após a conclusão das tarefas de ingestão e

armazenados em cache Redis, garantindo resposta imediata às consultas do *frontend*.

## 4 Requisitos

### 4.1 Requisitos Funcionais

#### 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 "Score" do senador baseado no algoritmo de ranking do projeto.

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

- [RF04] O sistema deve importar os lançamentos da Cota Parlamentar (CEAPS) através de arquivos CSV/Dados Abertos.
- [RF05] O sistema deve permitir visualizar o gasto acumulado por tipo de despesa (passagens, correios, consultorias).
- [RF06] O sistema deve exibir os fornecedores que mais receberam recursos de um determinado senador.

#### Módulo de Emendas e Orçamento:

- [RF07] O sistema deve integrar com o Portal da Transparência para buscar emendas de autoria do senador.
- [RF08] O sistema deve destacar valores destinados via "Transferências Especiais"(emendas PIX).

#### Módulo de Atividade Legislativa:

- [RF09] O sistema deve listar as votações nominais recentes e o voto de cada senador (Sim/Não/Abstenção).

### 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 Transparéncia.
- **[RNF07]** O sistema deve manter disponibilidade mínima de 99% durante o período eleitoral.

### **Segurança:**

- **[RNF08]** As comunicações devem ser criptografadas utilizando HTTPS/TLS.
- **[RNF10]** O sistema deve estar em conformidade com a LGPD (Lei Geral de Proteção de Dados).

### **Manutenibilidade:**

- **[RNF09]** O sistema deve estar em conformidade com a LGPD (Lei Geral de Proteção de Dados).
- **[RNF10]** A arquitetura modular deve permitir atualizações isoladas de cada módulo.
- **[RNF11]** O código deve seguir padrões de desenvolvimento e estar documentado.
- **[RNF12]** O sistema deve possuir *pipelines* de CI/CD para integração e deploy contínuos.

## **5 Design**

### **5.1 Projeto UML**

O projeto utiliza a Unified Modeling Language (UML) para documentar visualmente a estrutura e o comportamento do sistema. Abaixo são apresentados os diagramas essenciais para o entendimento da arquitetura proposta.

#### **5.1.1 Diagrama de Classes**

O Diagrama de Classes modela a estrutura estática do domínio. As principais entidades identificadas são:

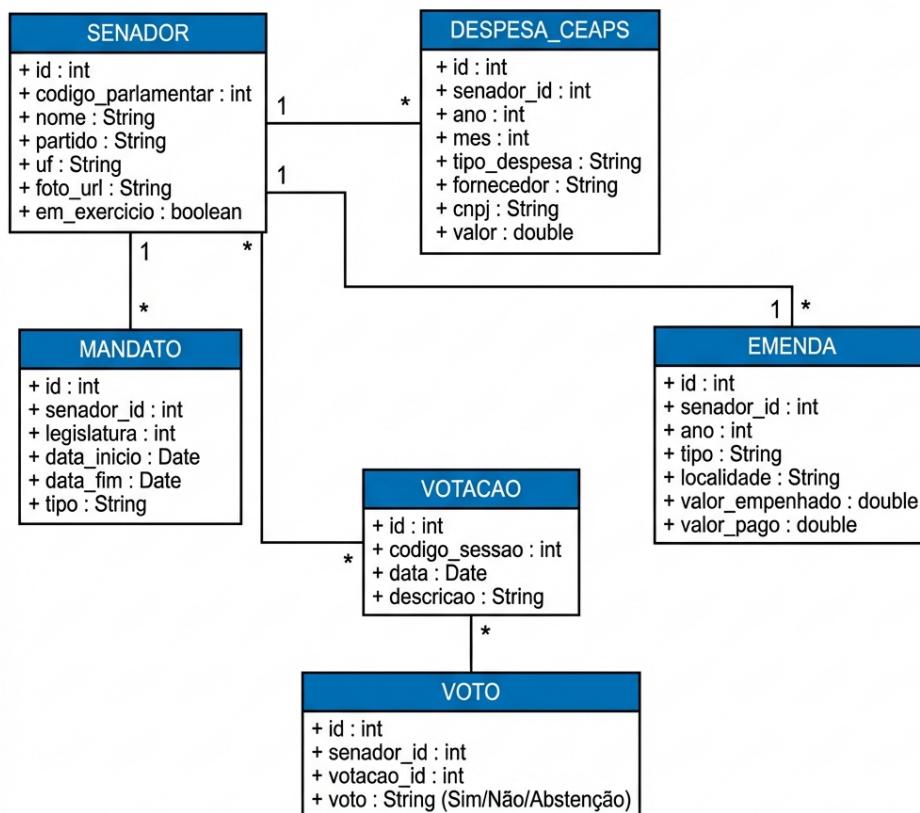
- **Senador:** Representa o parlamentar, contendo atributos como Identificador, Nome, Par-

tido e Estado.

- **Despesa:** Representa um lançamento na CEAPS, associada a um Senador e um Fornecedor.
- **Votacao:** Representa uma sessão deliberativa no Plenário, composta por múltiplos Votos.
- **Emenda:** Representa verba orçamentária destinada pelo senador, contendo Valor, Ano e Beneficiário.

Figura 1: Diagrama de Classes Simplificado (Conceitual)

### Diagrama de Classes UML: Plataforma de Transparência "Tô De Olho"



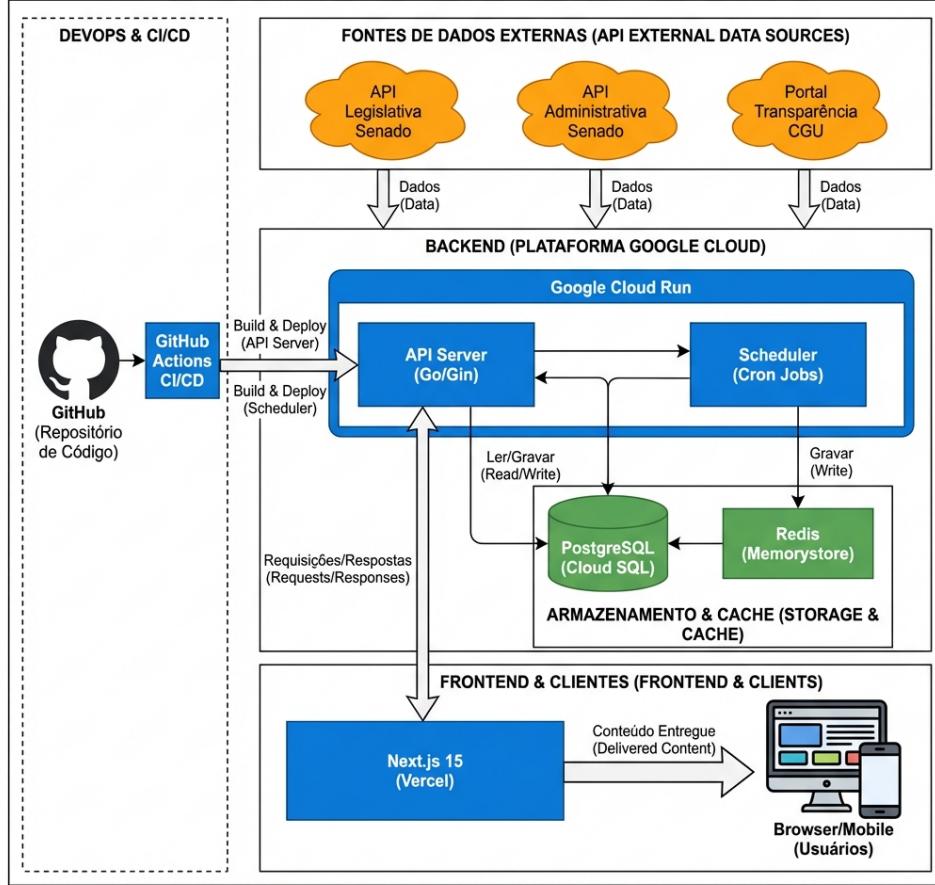
Fonte: Autoria Própria

#### 5.1.2 Diagrama de Implantação

O sistema é implantado em nuvem (GCP), utilizando orquestração de containers.

Figura 2: Diagrama de Implantação e Infraestrutura

Diagrama de Infraestrutura: Plataforma de Transparéncia do Senado Federal  
(Infrastructure Diagram: Brazilian Senate Transparency Platform)



Fonte: Autoria Própria

## 5.2 Visão arquitetural

O *Tô De Olho* adota uma arquitetura de **monolito modular**, onde os componentes são organizados internamente em módulos bem definidos, mas compartilham um único processo de deploy. Essa escolha arquitetural foi motivada pelos seguintes fatores:

- **Simplicidade de Deploy:** um único container simplifica a infraestrutura e reduz custos operacionais;
- **Manutenibilidade:** a organização em módulos internos permite atualizações isoladas sem a complexidade de orquestração distribuída;
- **Evolução Futura:** a estrutura modular permite migração gradual para microsserviços, se necessário.

### Módulos Internos:

- `internal/senador`: Gerencia dados cadastrais dos parlamentares e mandatos;
- `internal/ceaps`: Processa e totaliza despesas da Cota Parlamentar;

- `internal/votacao`: Armazena e consulta votações nominais em plenário;
- `internal/emenda`: Integração com Portal da Transparência para emendas parlamentares;
- `internal/ranking`: Cálculo e agregação de scores de desempenho.

#### **Estratégia de Ingestão de Dados:**

A plataforma utiliza uma estratégia híbrida de ingestão:

1. **Backfill**: carga inicial histórica através das APIs do Senado, processando dados desde 2019;
2. **Sincronização Contínua**: *CronJobs* diários às 03:00 BRT que consomem as APIs oficiais para manter os dados atualizados.

#### **Infraestrutura:**

O sistema é containerizado com Docker e implantado no Google Cloud Run, que oferece:

- **Escala a Zero**: sem custos quando não há tráfego;
- **Deploy Simplificado**: um único Dockerfile para toda a aplicação;
- **Escalabilidade Automática**: instâncias são criadas sob demanda durante picos de acesso.

### **5.3 Modelo de Banco de Dados**

O modelo de dados relacional foi projetado para garantir integridade e eficiência nas consultas analíticas. As tabelas principais são:

- **TB\_SENADORES**: Tabela mestre. PK: `codigo_senado`. Colunas: `nome`, `partido`, `uf`, `url_foto`.
- **TB\_MANDATOS**: Histórico de legislaturas. FK para `TB_SENADORES`.
- **TB\_DESPESAS\_CEAPS**: Armazena cada nota fiscal reembolsada. FK para `TB_SENADORES`. Colunas: `valor`, `data`, `tipo_despesa`, `cnpj_fornecedor`, `url_documento`. Indexada por senador e ano para performance em buscas.
- **TB\_VOTACOES**: Cabeçalho das votações. PK: `codigo_sessao`.
- **TB\_VOTOS**: Tabela de junção (*many-to-many*) entre Senadores e Votações, registrando o voto individual (Sim/Não).
- **TB\_EMENDAS**: Registra valores destinados. Colunas: `valor_empenhado`, `valor_pago`, `beneficiario`, `modalidade` (ex: Pix).

# 6 Testes de Software

## 6.1 Projeto de Testes

A estratégia de qualidade do *Tô De Olho* combina testes em diferentes níveis, aproveitando o ferramental nativo da linguagem Go:

- **Testes Unitários:** Validam regras de negócio isoladas, como o cálculo do "Score" do senador e parsers de CSV. Implementados com o pacote padrão `testing` do Go, utilizando a técnica de *Table-Driven Tests* para cobrir múltiplos cenários de borda.
- **Testes de Integração:** Validam a comunicação entre os componentes e o banco de dados. Utiliza-se a biblioteca `testcontainers-go` para subir instâncias efêmeras do PostgreSQL e Redis durante a execução da pipeline, garantindo que as queries e a persistência funcionem como esperado num ambiente controlado.
- **Testes de Contrato (API):** Asseguram que as respostas dos serviços externos (Senado/Transparéncia) continuam respeitando os formatos esperados, alertando sobre "Quebras de API" em dependências externas.

# 7 Implantação

## 7.1 Projeto de Implantação

A implantação do sistema segue as práticas de *GitOps* e Infraestrutura como Código.

**Ambiente de Execução:** O sistema é empacotado em uma imagem Docker otimizada (*multi-stage build*) e implantado no Google Cloud Run. Esta plataforma serverless oferece escala automática baseada na demanda, incluindo escala a zero quando não há tráfego, otimizando custos operacionais.

**Pipeline de CI/CD:** Utiliza-se GitHub Actions para automação. A cada *push* na branch principal:

1. *Build*: Compilação dos binários Go e checagem de erros;
2. *Test*: Execução dos testes unitários e de integração;
3. *Publish*: Criação da imagem Docker e envio para o Google Container Registry;
4. *Deploy*: Atualização automática do serviço no Cloud Run.

# 8 Manual do Usuário Simplificado

A plataforma é pública e não requer cadastro para consulta, maximizando a transparéncia.

1. **Acesso:** Navegue para <https://todeolho.org.br>.

2. **Ranking:** Na página inicial, visualize os "Top 3" senadores em Economia e Presença.
3. **Busca:** Utilize a barra superior para digitar o nome de um senador.
4. **Detalhes:** Ao clicar em um senador, navegue pelas abas "Gastos"(para ver notas fiscais detalhadas) e "Emendas"(para ver destino das verbas).
5. **Fiscalização:** Use o botão "Compartilhar" para enviar ficha do senador nas redes sociais.

## 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 [24], 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.

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 Sere-nata de Amor. Essas evoluções fortaleceriam o controle social e aproximariam a ferramenta dos degraus superiores da Escada de Arnstein.

## Agradecimentos

Agradeço ao meu orientador, Prof. Pablo Vieira Florentino, pela orientação e apoio durante o desenvolvimento deste trabalho. Aos colegas de curso que contribuíram com discussões e sugestões. À minha família pelo apoio incondicional. Por fim, às comunidades de código aberto que mantiveram as ferramentas e documentações utilizadas neste projeto.

## 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.I.]: Cambridge University Press, 1970.
- 4 GOMES, W. **A democracia no mundo digital**. [S.I.]: 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 DRAGONI, N. et al. Microservices: Yesterday, today, and tomorrow. **Present and Ulterior Software Engineering**, Springer, p. 195–216, 2017. Discussão de trade-offs entre microserviços e monólitos.
- 10 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.I.], 2015. p. 778–788. Estudo comparativo de linguagens incluindo Go e Python.
- 11 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.I.], 2024. Avaliação de performance de frameworks Go incluindo Gin.
- 12 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.
- 13 Redis Ltd. **Redis: The Real-time Data Platform**. 2024. (<https://redis.io>). Banco de dados em memória com latências em microsssegundos.
- 14 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.
- 15 Recharts Community. **Recharts: A Composable Charting Library Built on React Components**. 2024. (<https://recharts.org>). Biblioteca de visualização de dados para React.
- 16 Tailwind Labs. **Tailwind CSS: A Utility-First CSS Framework**. 2024. (<https://tailwindcss.com>). Framework CSS com abordagem utility-first e purge automatico.

- 17 ARNSTEIN, S. R. A ladder of citizen participation. **Journal of the American Institute of Planners**, v. 35, n. 4, p. 216–224, 1969.
- 18 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.
- 19 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.
- 20 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.
- 21 mySociety. **mySociety: Digital Tools for Democracy**. 2024. <<https://www.mysociety.org>>. Organização britânica pioneira em civic tech, responsável pelo TheyWorkForYou.
- 22 \_\_\_\_\_. **TheyWorkForYou: Parliamentary Monitoring**. 2024. <<https://www.theyworkforyou.com>>. Plataforma de monitoramento parlamentar do Reino Unido, ativa desde 2004.
- 23 OpenSecrets. **OpenSecrets: Following the Money in Politics**. 2024. <<https://www.opensecrets.org>>. Organização não-partidária que rastreia financiamento de campanhas nos EUA.
- 24 COSTA, E. A. D. Gestão pública digital: O poder das tic na democracia brasileira. **Revista de Administração Pública**, v. 25, p. 123–145, 2019.

# Apêndices

## Glossário, Siglas e Abreviações

- API** *Application Programming Interface* — Interface de programação que permite a comunicação entre sistemas de software.
- CEAPS** Cota para o Exercício da Atividade Parlamentar dos Senadores — verba destinada ao custeio de despesas relacionadas ao exercício do mandato parlamentar.
- CI/CD** *Continuous Integration / Continuous Deployment* — práticas de integração e implantação contínuas de software.
- ETL** *Extract, Transform, Load* — processo de extração, transformação e carga de dados.
- GCP** *Google Cloud Platform* — plataforma de serviços em nuvem do Google.
- LES** *Legislative Effectiveness Score* — metodologia para avaliação de efetividade legislativa desenvolvida por Volden e Wiseman.
- LGPD** Lei Geral de Proteção de Dados — legislação brasileira sobre privacidade e proteção de dados pessoais.
- MVP** *Minimum Viable Product* — produto mínimo viável, versão inicial com funcionalidades essenciais.
- ORM** *Object-Relational Mapping* — técnica de mapeamento objeto-relacional para persistência de dados.
- PIX** Sistema de pagamentos instantâneos do Banco Central; no contexto parlamentar, refere-se às “Emendas PIX” (Transferências Especiais) — modalidade de repasse que dispensa convênio.
- REST** *Representational State Transfer* — estilo arquitetural para APIs web.
- SSR** *Server-Side Rendering* — renderização de páginas no servidor.
- TIC** Tecnologias da Informação e Comunicação.
- TTL** *Time To Live* — tempo de vida de dados em cache.
- UF** Unidade Federativa — estado brasileiro.
- WCAG** *Web Content Accessibility Guidelines* — diretrizes de acessibilidade para conteúdo web.