AUDITORIA ELEITORAL

A Geometria do Voto

Reflexões, perguntas e hipóteses sobre auditoria eleitoral

Maurício Ferreira Sarmento

Direitos autorais

Copyright © 2026 Maurício Ferreira Sarmento. Todos os direitos reservados.

Primeira edição, 2026. Esta obra apresenta reflexões do autor, hipóteses de engenharia e simulações didáticas. Os cenários e números inventados não constituem prova de fraude nem medem a probabilidade de manipulação em uma eleição real. As opiniões são do autor e não representam necessariamente as instituições citadas. Os procedimentos eleitorais e as regras podem mudar. As referências ao final indicam as fontes consultadas para esta edição.

Como ler este livro

Nos capítulos, explico os conceitos e as perguntas em linguagem direta. Nos Anexos A a D, deixo os programas completos para que o leitor possa repetir as contas e alterar as premissas. Os valores das simulações são hipotéticos; quando falo de procedimentos reais, aponto as fontes consultadas.

Ao longo da leitura, diferencio fato documentado, opinião, hipótese e resultado matemático. Candidato A, Candidato B e Candidatos C-F são nomes genéricos usados nos exemplos, não referências a pessoas específicas.

Sumário

Prefácio — Da experiência à pergunta 5

Parte I — O que significa verificar 6

Capítulo 1 — Uma eleição é uma cadeia de registros 6

Capítulo 2 — O que uma prova precisa mostrar 7

Parte II — Arquitetura, barreiras e hipóteses de falha 9

Capítulo 3 — Onde começa a confiança no software 9

Capítulo 4 — Isolamento físico, mídias e registros 10

Parte III — Simulações que o leitor pode repetir 12

Capítulo 5 — A matemática por trás de uma tela em degraus 12

Capítulo 6 — Que composição de lote faria a tela mudar? 14

Capítulo 7 — Uma porta, várias fechaduras 19

Parte IV — Soluções que precisam ser comparadas 27

Capítulo 8 — Criptografia, registros distribuídos e prova de resultado 27

Conclusão — O que as simulações me permitem afirmar 30

Anexo A — Simulação dos percentuais acumulados 32

Anexo B — Árvore lógica de barreiras 35

Anexo C — Que composição de lote faz a tela mudar? 40

Anexo D — Duas hipóteses, a mesma escadinha 48

Referências e fontes consultadas 55

Prefácio — Da experiência à pergunta

Minha trajetória pessoal e profissional me levou a olhar para as instituições brasileiras com perguntas que continuam comigo. Uma delas é simples de formular e difícil de responder: como uma pessoa comum pode verificar um sistema do qual depende a confiança de milhões?

Passei anos montando redes, consertando equipamentos e desenvolvendo sistemas. Esse trabalho ensina uma lição simples: quando algo parece estranho, a primeira tarefa é descobrir *o que foi medido*. Um gráfico é uma representação; um registro é produzido por um processo; e um processo tem regras, limites e pessoas responsáveis por cada etapa. Uma impressão visual pode motivar uma investigação. Para sustentar uma conclusão, porém, preciso de dados que outra pessoa consiga examinar.

Este livro nasce desse compromisso. Quero entender quais evidências o sistema eleitoral produz, o que elas permitem conferir e onde permanecem perguntas legítimas de engenharia. Vou expor hipóteses de falha sem apresentá-las como acontecimentos comprovados. Também vou examinar as barreiras que já existem, porque uma análise que ignora as defesas de um sistema não consegue medir seus riscos.

Quando houver código, deixarei os parâmetros visíveis e indicarei as fontes dos números. O leitor poderá repetir as contas e avaliar as conclusões. A confiança pública exige perguntas claras e respostas verificáveis.

Parte I — O que significa verificar

Capítulo 1 — Uma eleição é uma cadeia de registros

Para compreender um sistema eleitoral, não basta olhar a tela que mostra o resultado. Entre a escolha feita pelo eleitor e o número divulgado há etapas distintas: a coleta do voto, o registro no equipamento, a apuração de cada seção, a transmissão dos resultados locais, a totalização e a publicação. Cada etapa produz ou utiliza registros diferentes. Uma conferência válida precisa dizer exatamente qual ligação dessa cadeia está sendo testada.

No Brasil, ao final da votação em uma seção, a urna produz um Boletim de Urna (BU) com os totais daquela seção. O documento identifica a seção e informa, entre outros dados, a votação de candidatos, votos brancos e nulos. O BU impresso pode ser comparado com a versão divulgada na internet. A regulamentação de 2026 também descreve o Registro Digital do Voto (RDV), os arquivos de resultado e procedimentos de fiscalização. [1]

Comparar o BU impresso com o BU publicado responde a uma pergunta importante: o resultado local divulgado coincide com o documento produzido na seção? Somar boletins de forma independente ajuda a conferir a totalização. Essas verificações não são, por si só, uma observação independente de cada escolha feita dentro da cabina. São verificações de etapas determinadas.

Reconhecer esse alcance torna a auditoria mais precisa, não menos útil.

O teste de integridade faz outra pergunta. Nele, votos previamente conhecidos são inseridos em urnas selecionadas para comparar a contagem produzida com a contagem esperada. O procedimento testa o comportamento das unidades auditadas nas condições do teste. Para avaliar seu alcance, é preciso conhecer regras de seleção, quantidade de unidades, condições de execução e diferenças possíveis entre o teste e a votação comum. [2]

Há ainda a inspeção de código, as assinaturas digitais, a custódia de mídias e a análise de registros de operação. Uma assinatura pode mostrar que um arquivo corresponde a uma identidade criptográfica esperada; ela não torna correto, por definição, o comportamento do programa assinado. Ao mesmo tempo, uma hipótese de ataque à cadeia de desenvolvimento não demonstra que um ataque ocorreu. O trabalho técnico consiste em formular a hipótese, identificar as barreiras e procurar evidências capazes de confirmá-la ou refutá-la.

Capítulo 2 — O que uma prova precisa mostrar

Em engenharia, risco e ocorrência são coisas diferentes. Se um programa admite entrada de dados por mídia removível, existe uma superfície que merece análise. Para afirmar que houve alteração indevida, seria necessário mostrar o caminho da alteração, sua capacidade de atravessar as barreiras existentes e algum vestígio verificável de seu efeito. A gravidade de uma hipótese não substitui essa demonstração.

Também é necessário definir o adversário do modelo. Uma pessoa com acesso físico breve a uma urna enfrenta condições diferentes das de alguém que administra o ambiente de compilação. Uma falha de programação sem agente malicioso é diferente de uma operação coordenada entre várias organizações. Ao misturar essas situações, uma simulação pode produzir um número impressionante que não descreve nenhum risco identificável.

Usarei quatro perguntas para examinar cada cenário deste livro:

1. Qual etapa e qual resultado o cenário tenta alterar?
2. Que acesso e que recursos o agente hipotético possui?
3. Que barreiras existentes ele precisaria superar?
4. Que registro externo poderia revelar o desvio?

Essas perguntas também ajudam a comparar soluções. Um registro físico conferido pelo eleitor pode oferecer uma fonte de verificação independente do software, desde que sua produção, custódia e auditoria funcionem. Há propostas criptográficas de verificabilidade de ponta a ponta que procuram atingir uma finalidade semelhante por outros mecanismos. O princípio da independência do software trata da possibilidade de detectar uma mudança indevida no resultado mesmo quando há erro ou alteração não percebida no programa; ele não determina uma tecnologia única. [3]

O objetivo, portanto, não é chamar um mecanismo de perfeito. É explicar o que ele verifica, de que pressupostos depende e como uma falha poderia ser descoberta. Esse vocabulário será a base para ler os capítulos sobre código, chaves, mídias, boletins, simulações e propostas de auditoria.

Parte II — Arquitetura, barreiras e hipóteses de falha

Capítulo 3 — Onde começa a confiança no software

Quando um programa é instalado, o usuário costuma perguntar se o arquivo é “original”. Uma assinatura digital ajuda a responder: ela permite verificar se um arquivo corresponde ao que foi assinado por determinada chave e se foi modificado depois disso. É uma proteção importante contra substituições no caminho. Sua resposta, contudo, depende da proteção da chave e da confiança no processo que produziu o arquivo assinado.

Essa distinção é a raiz do problema da cadeia de desenvolvimento. O código escrito por programadores passa por bibliotecas, compiladores, ambientes de construção, testes, assinatura, distribuição e instalação. Um erro pode surgir em mais de uma dessas etapas. Na hipótese clássica apresentada por Ken Thompson, até um compilador comprometido pode produzir um programa diferente do que um leitor espera ao examinar apenas o código-fonte. O exemplo é uma demonstração conceitual de risco na cadeia de confiança; não é uma descrição de algo ocorrido em uma eleição brasileira. [5]

Por isso, inspecionar código, verificar assinaturas e acompanhar a preparação das urnas são medidas complementares. Uma busca por vulnerabilidades em um ambiente controlado pode revelar falhas e levar a correções. Seu resultado precisa ser lido com o

escopo do teste em mãos: quais versões foram examinadas, quais planos de ataque foram executados e quais condições não foram testadas? O TSE publica procedimentos de fiscalização e relatórios do Teste Público de Segurança; eles devem fazer parte da análise, inclusive quando a pergunta inicial do pesquisador for crítica. [2] [6]

Para discutir uma hipótese de comprometimento da cadeia de desenvolvimento, não basta dizer que “um código malicioso poderia ser assinado”. É necessário especificar como ele chegaria ao ambiente, quais controles teria de atravessar, como alteraria o resultado e qual evidência independente poderia revelar seu efeito. Sem esse percurso, temos uma possibilidade genérica de segurança da informação. Com ele, temos uma hipótese examinável.

Capítulo 4 — Isolamento físico, mídias e registros

Retirar um equipamento da internet reduz a oportunidade de ataques remotos. Esse isolamento costuma ser chamado de *air gap*. Ele não faz os dados deixarem de circular: sistemas precisam ser preparados, operados e, em algum momento, produzir resultados. Cada passagem de arquivos ou mídias entre ambientes pede regras de origem, integridade e custódia. A regulamentação eleitoral brasileira descreve etapas de geração de mídias, preparação, votação e arquivos produzidos no encerramento. [1]

Um cenário hipotético de ataque por mídia removível só é plausível se indicar quem teria acesso à mídia, quando o acesso ocorreria e por que os controles de autenticação e preparação não impediriam a alteração. A simples existência de uma porta ou de um dispositivo removível não demonstra que esses controles falharam.

Os registros de operação, ou *logs*, são úteis para reconstruir eventos. Mas também têm uma limitação conhecida: se forem gerados e guardados exclusivamente pelo componente sob suspeita, dependem da integridade desse componente. Comparações com registros externos ajudam a reduzir essa dependência. O BU impresso na seção, por exemplo, permite conferir o número divulgado para aquela seção; já não resolve, sozinho, uma hipótese que diga respeito ao instante em que o voto foi registrado. Cada fonte de evidência responde por uma etapa diferente.

Essa leitura por camadas impede dois saltos comuns no debate público. Uma barreira existente não torna impossível qualquer falha futura. Uma falha teoricamente possível não significa que a barreira tenha sido atravessada. Entre as duas frases existe trabalho técnico: descrever o ataque, testar as condições e procurar vestígios.

Parte III — Simulações que o leitor pode repetir

Capítulo 5 — A matemática por trás de uma tela em degraus

Em uma transmissão de resultados, percentuais aparecem com poucas casas decimais. Isso pode produzir uma imagem curiosa: a linha fica parada por algum tempo e depois avança um décimo de ponto percentual. A primeira pergunta não é se a imagem parece regular. É quanto os números completos mudaram entre duas atualizações.

Imagine uma disputa hipotética com dois candidatos, A e B, e apenas votos válidos para um deles. Começamos com 100 milhões de votos: 52 milhões para A e 48 milhões para B. Acrescentamos dez lotes de 500 mil votos. Em um cenário, B recebe 70% de cada lote. Em outro, recebe 70% nos três primeiros, 50% no quarto, 30% nos dois seguintes e 68% nos quatro últimos. Esses percentuais foram escolhidos para ilustrar comportamentos possíveis; não vieram de uma eleição.

Se B tinha V votos em um total N, e um novo lote de tamanho L contém uma fração q de votos para B, a nova fração acumulada é:

```
p_novo = (V + q × L) / (N + L)
```

Como p = V / N, a mudança também pode ser escrita assim:

```
p_novo − p = (q − p) × L / (N + L)
```

A fórmula mostra duas coisas. Primeiro, a direção da curva depende da comparação entre a participação de B *no novo lote* e sua participação *acumulada*. Se q for maior que p, a curva sobe; se for menor, desce. Segundo, à medida que o total N cresce, um lote de mesmo tamanho tem efeito menor. Trata-se de uma proporção acumulada; o programa não calcula média móvel nem simula um banco de dados saturado.

Com lotes repetidos de 70%, o percentual exibido para B sobe de 48,1% para 49,0% em dez degraus sucessivos. No segundo cenário, a tela mostra 48,1%, 48,2%, 48,3%, 48,3%, 48,2%, 48,2%, 48,3%, 48,3%, 48,4% e 48,5%. No quarto lote, B recebe 50% dos novos votos: seu percentual real aumenta um pouco, mas o número com uma casa decimal permanece em 48,3%. No quinto lote, com 30% para B, a tendência acumulada se inverte e a tela volta a 48,2%.

O Anexo A contém o código integral e a instrução para reproduzir a tabela. A interface didática do exemplo **arredonda** para uma casa decimal. O modelo tem exatamente dois candidatos, motivo pelo qual o ganho percentual de um corresponde à perda percentual do outro. Essa reciprocidade precisa ser reavaliada se o denominador incluir outras candidaturas ou categorias de voto.

O experimento demonstra a relação entre composição dos lotes, tamanho do total e arredondamento. Ele não mostra a ordem efetiva de chegada de boletins em uma eleição, nem prova o uso de um algoritmo para suavizar uma curva. Para avançar dessa demonstração para uma investigação empírica, precisamos definir um pleito, obter os registros necessários, reconstruir as atualizações por horário quando isso for possível e comparar os números completos com os exibidos. Os BUs publicados e a documentação de resultados são pontos de partida; a

disponibilidade e a granularidade dos horários precisam ser verificadas antes de prometer essa reconstrução. [1] [4]

Capítulo 6 — Que composição de lote faria a tela mudar?

O exemplo anterior tinha apenas dois candidatos. Isso facilita a primeira conta, mas não representa uma eleição com várias candidaturas. Quando há outros candidatos, a queda de um nome não precisa ser igual ao crescimento de outro: parte dos votos novos pode ir para qualquer uma das demais opções. Para entender uma atualização, precisamos considerar o lote inteiro.

Vamos usar uma conta inventada, com números redondos. Antes de uma atualização, há 100 milhões de votos válidos: 47 milhões para o Candidato A, 45 milhões para o Candidato B e 8 milhões para o grupo dos Candidatos C, D, E e F. Agora entram 5 milhões de votos. Se esse novo lote tiver 44,9% para o Candidato A, 47,1% para o Candidato B e 8% para os demais, a tela passa de 47,0% para 46,9% para A e de 45,0% para 45,1% para B. Os demais ficam em 8,0%.

Isso não é um palpite sobre o que chegou na apuração. É um exemplo construído para mostrar uma composição que produz exatamente aqueles passos. Com 5 milhões de votos, os números correspondem a 2.245.000 votos para A, 2.355.000 para B e 400.000 para os demais. As três parcelas somam os 5 milhões do lote.

Mas a composição não pode ser deduzida como um número único apenas olhando a tela. Como 46,9% representa uma faixa de valores exatos e 45,1% representa outra, várias proporções do lote

podem produzir as mesmas casas decimais. Neste exemplo específico, a faixa calculada fica em torno de 43,85% a 45,95% para A e de 46,05% a 48,15% para B; a parte restante cabe aos demais candidatos, e as três participações precisam somar 100%. Essas faixas mudam quando mudam os votos já acumulados e o tamanho do lote.

Para fazer a conta ao contrário, usamos esta regra: participação necessária no lote = [percentual acumulado desejado depois × (total anterior + tamanho do lote) – percentual acumulado anterior × total anterior] / tamanho do lote. Como a tela arredonda, o percentual desejado depois é uma faixa, então o resultado também costuma ser uma faixa.

Agora mudemos a composição para 32,5% para A, 62,5% para B e 5% para os demais. Nesse mesmo exemplo, o acumulado vai para aproximadamente 46,3% e 45,8% - uma mudança bem maior que um décimo para cada um. Se o lote vier com 15% para A, 80% para B e 5% para os demais, o acumulado muda para cerca de 45,5% e 46,7%; com 20%, 75% e 5%, fica perto de 45,7% e 46,4%. Com 40%, 55% e 5%, os valores ficam perto de 46,7% e 45,5%. Já com 50%, 45% e 5%, A volta a subir e B praticamente estaciona. O tamanho do lote e a distância entre sua composição e o acumulado explicam essas diferenças.

Há um detalhe que a tela esconde: arredondar para uma casa decimal transforma vários valores exatos no mesmo número mostrado. Por exemplo, conforme a regra de arredondamento, a tela pode mostrar 46,9% para valores exatos aproximadamente entre 46,85% e 46,95%. Portanto, uma tela que mostra 46,9% não nos diz o percentual exato, e uma passagem de 47,0% para 46,9%

não quer dizer que o valor real caiu exatamente 0,1 ponto naquele lote.

Isso impede uma conclusão apressada: não existe um percentual único do lote que possamos deduzir apenas olhando os números já arredondados. Para resolver a conta de uma atualização real, precisamos dos votos acumulados antes, do total de votos que entrou, dos votos novos por candidato e da regra de arredondamento usada pela tela. O Anexo C explica o cálculo e traz um programa para o leitor experimentar outras composições.

Também é importante separar fontes. O TSE informa que o Painel da evolução da totalização permite rever a evolução dos resultados ao longo do tempo. Já o arquivo final de boletins por seção é uma fotografia de cada seção depois da apuração, não um histórico cronológico das chegadas. Eu não reproduzi, a partir do conjunto de dados que consultei, uma série completa de cada atualização e não usei a cronologia do painel como dado empírico desta simulação. Isso delimita o que fiz nesta edição; não prova que a cronologia inexista ou tenha sido escondida. Para avaliar uma sequência observada, é preciso indicar qual tela, quais horários e quais dados foram comparados. [11] [12]

O objetivo do teste não é declarar que um padrão foi normal ou suspeito sem medir. É reconstruir cada atualização com os dados disponíveis e perguntar: que regiões e seções entraram naquele intervalo, quantos votos trouxeram para cada candidatura e como isso alterou o acumulado? Sem essas informações, qualquer chance numérica seria inventada. Esta simulação demonstra possibilidades matemáticas, não afirma fraude nem calcula probabilidade de fraude.

Duas hipóteses para a mesma escadinha

Neste livro, vamos colocar lado a lado duas explicações possíveis para uma sequência em que um candidato aparece caindo um décimo enquanto outro sobe um décimo. A primeira hipótese não inclui fraude nem algoritmo de amortização: os resultados chegam em lotes cuja composição, somada ao acumulado e ao arredondamento da tela, produz aquela sequência. A segunda supõe, apenas para o exercício, um algoritmo que escolhe ou ajusta a composição divulgada para tentar produzir os mesmos passos.

Para que eu compare as ideias, uso a mesma base inventada nas duas hipóteses: 100 milhões de votos acumulados, com 47 milhões para o Candidato A, 45 milhões para o Candidato B e 8 milhões para os Candidatos C a F. Cada atualização acrescenta 5 milhões de votos. A sequência exibida é a mesma nos dois casos: 47,0% / 45,0%; depois 46,9% / 45,1%; 46,8% / 45,2%; e 46,7% / 45,3%.

Na hipótese sem algoritmo, posso construir lotes com estas composições: 44,9% / 47,1% / 8,0%; depois 44,7% / 47,3% / 8,0%; e 44,5% / 47,5% / 8,0%. A conta acumulada produz a sequência da tabela. Isso demonstra que a sequência é matematicamente compatível com aquelas chegadas. Não demonstra que foi assim que os votos chegaram, nem calcula a probabilidade de uma composição natural produzir o padrão.

Na hipótese de amortização, um mecanismo hipotético poderia calcular a participação necessária de cada grupo para que a próxima tela mostrasse o degrau desejado. Para o primeiro lote de 5 milhões deste exemplo, a faixa aproximada é de 43,85% a 45,95% para A e de 46,05% a 48,15% para B; os demais recebem o que sobra, e as três participações precisam somar 100%. A faixa depende do ponto inicial, do tamanho do lote e do

arredondamento. Ela não é um limite universal para todas as atualizações: depois de cada lote, o acumulado muda e a conta precisa ser refeita.

Uma pequena mudança pode ou não aparecer na tela. Se a composição variar, mas continuar dentro da faixa que arredonda para o mesmo degrau, o número mostrado permanece igual. Se cruzar o limite da faixa, a atualização pode ficar parada, mudar dois décimos ou alterar o passo de um dos candidatos. Neste exemplo, 44,8% / 47,0% / 8,2% ainda produz 46,9% / 45,1%; já 43,8% / 47,1% / 9,1% produz 46,8% / 45,1%. Isso é uma análise de sensibilidade de um lote hipotético, não uma afirmação sobre lotes reais.

O mesmo gráfico, portanto, pode ser compatível com duas narrativas diferentes. Na primeira, a sequência aparece pela soma de lotes recebidos e pelo arredondamento; na segunda, um controlador hipotético seleciona as composições para atingir os degraus. Uma simulação consegue mostrar que cada narrativa é possível sob certas premissas. Sozinha, ela não diz qual narrativa descreve uma eleição real, nem quão provável é cada uma.

O leitor deve prestar atenção à diferença entre “possível” e “demonstrado”. Para avaliar qual hipótese merece mais confiança, seria preciso examinar evidências independentes e também explicitar o modelo usado para dizer o que seria esperado sem amortização. Neste exercício, não tenho uma série cronológica utilizável das atualizações para estimar essa frequência. Embora o TSE descreva um painel de evolução, eu não consegui obter, na consulta que fiz para esta edição, uma série completa que me permitisse reproduzir cada passo. Isso é um relato limitado da minha busca, não uma afirmação de que o painel inexista ou de que

haja ocultação deliberada. Por isso, os gráficos deste anexo são assumidamente hipotéticos e não são apresentados como reconstrução do que aconteceu.

O Anexo D traz o código para gerar as duas explicações e testar variações na composição dos lotes. O programa não decide pelo leitor: ele deixa visíveis as premissas e mostra quais mudanças alteram o degrau exibido.

Capítulo 7 — Uma porta, várias fechaduras

Imagine uma sala que guarda um documento importante. Uma pessoa protege a porta; outra confere a identidade de quem entra; uma terceira compara o documento que sai com uma cópia guardada fora da sala. Nenhuma dessas pessoas é uma fechadura perfeita. O valor está em não depender de uma única pessoa, de uma única chave ou de uma única conferência.

Uma eleição também tem etapas diferentes. Há o programa usado na urna, a preparação do equipamento, a votação, a contagem na própria seção, o transporte dos resultados e a soma final. Em cada etapa podemos fazer uma pergunta simples: **o que poderia dar errado, quem perceberia e com qual prova?** Chamarei de *barreira* qualquer controle capaz de impedir uma alteração ou de deixar um sinal que permita descobri-la.

Antes de imaginar a falha de todas as barreiras, precisamos conhecer o que elas tentam proteger. Vou separar o processo em cinco grupos. Esta divisão é minha, para facilitar a leitura; não é a lista oficial de “cinco barreiras” da Justiça Eleitoral.

1. Programa, assinaturas e chaves. Uma assinatura digital funciona, em parte, como um lacre que pode ser conferido. Ela ajuda a detectar uma troca de arquivo depois da assinatura. O *hash* é uma espécie de impressão digital calculada a partir do arquivo: se o arquivo muda, o valor esperado normalmente deixa de coincidir. Na preparação das urnas há procedimentos de assinatura, lacração e conferência dos programas instalados, inclusive por representantes de entidades fiscalizadoras. [1] [9]

As chaves criptográficas precisam ser protegidas, assim como as pessoas e os equipamentos autorizados a usá-las. Mas é importante não confundir coisas diferentes. Ter acesso a uma senha ou a uma chave não é, automaticamente, controlar o programa, a urna, o teste e o resultado. Nas normas e explicações oficiais que consultei para este capítulo, não localizei base para afirmar que exista uma **"chave mestra" única** capaz de abrir e controlar todas as urnas. Vou usar essa expressão apenas como nome curto para uma hipótese extrema de acesso indevido; ela não descreve um recurso comprovado do sistema.

2. Preparação, mídias e conferência antes da votação. *Mídia* é, aqui, um dispositivo que transporta ou guarda arquivos. As normas distinguem mídia de carga, mídia de votação e mídia de resultado. Não são nomes intercambiáveis. Durante a preparação, são feitos procedimentos de demonstração de votação e de verificação de integridade e autenticidade de sistemas instalados; problemas com mídias têm procedimentos de tratamento e registro. Na abertura da seção, a urna emite a *Zerésima*, relatório que mostra os totais antes de começar a votação. Isso é diferente do teste de integridade feito com urnas escolhidas para auditoria. [1]

Se uma conferência encontrar assinatura ou resumo digital incompatível, isso é um alerta que precisa ser tratado conforme a etapa e o procedimento aplicável. Não posso resumir o processo dizendo que “uma chave errada invalida automaticamente a eleição inteira”. O efeito de uma divergência tem de ser descrito com precisão: qual arquivo, qual urna, qual momento e qual providência prevista.

3. Teste de integridade. Pense numa calculadora: se digitamos contas cujo resultado já conhecemos, podemos comparar o que ela mostra com o resultado esperado. No teste de integridade, votos previamente conhecidos são digitados em urnas selecionadas para auditoria, e o resultado é comparado com essas escolhas. Em 2026, a norma prevê o teste no mesmo dia e horário da votação oficial, inclusive em modalidades com e sem biometria. Assim, chamá-lo apenas de “teste antes da urna” confundiria o leitor. O teste ajuda a verificar as unidades examinadas nas condições em que o exame ocorreu; seu alcance depende da seleção e do desenho da auditoria. [2]

4. Resultado na seção e verificações posteriores. Ao encerrar a votação, a urna gera o Boletim de Urna, ou **BU**, com os totais daquela seção. A norma também prevê o **RDV**, registro digital dos votos sem identificar a pessoa que votou, e a assinatura digital desses arquivos. Cópias do BU podem ser afixadas na seção e entregues a interessados previstos nas regras; uma mídia de resultado segue para transmissão. Depois, podem ser comparados o BU impresso e o recebido pela totalização, ou o RDV e o BU. [1] [10]

Aqui mora uma diferença decisiva. Se alguém mudasse **apenas a soma depois da transmissão**, uma cópia independente do BU

daquela seção poderia mostrar a divergência. Se a hipótese for de alteração **antes da impressão do BU**, a comparação entre BU impresso e BU publicado pode dar igual, pois os dois já partiram do mesmo número. Nesse caso, seria preciso procurar outra evidência: a comparação com o RDV, com registros de teste ou com uma fonte independente da etapa suspeita. Se a hipótese disser que BU e RDV foram alterados de modo coerente, a simples comparação entre os dois tampouco resolve tudo. Não basta apontar uma prova; precisamos perguntar **de onde ela nasceu e se depende do mesmo componente suspeito**.

5. Fiscalização externa e resposta das instituições. Fiscais, partidos, Ministério Público, imprensa, pesquisadores e cidadãos podem observar ou conferir aspectos diferentes do processo, dentro das regras de cada procedimento. Uma foto de BU e uma comparação feita por terceiros criam uma referência fora da transmissão central. Se surgir uma divergência, ainda é necessário apurar, preservar provas e decidir o que fazer. Essa resposta institucional é outra camada. Ela não altera retroativamente um BU já fotografado, mas pode determinar se a divergência será levada a sério. [1] [10]

A hipótese das “duas mídias”

Agora podemos discutir sua ideia sem apresentá-la como fato. Imagine, **somente como exercício**, que um grupo consiga fazer o equipamento se comportar corretamente quando é auditado e de outra maneira durante uma votação real. Talvez se imagine uma mídia usada num contexto de teste e outra em contexto de produção. Para o livro, vou chamar isso de **hipótese de comportamento diferente entre teste e uso real**. Essa expressão é mais precisa que afirmar que hoje existam duas mídias oficiais com essas funções.

O cenário supõe mais do que a existência de dois dispositivos. Teríamos de explicar como o equipamento distinguiria os contextos, por que as assinaturas e conferências não mostrariam a diferença, como o teste de integridade deixaria de percebê-la e em qual momento o número seria alterado. Se a mudança aparecesse **só no papel impresso**, o RDV e os arquivos digitais poderiam divergir do BU. Se aparecesse no BU e na mídia de resultado, mas não no RDV, outra comparação poderia revelar o desvio. Se todos esses registros fossem adulterados de forma coerente, estaríamos supondo uma capacidade muito maior. Cada frase acrescenta uma condição à hipótese; nenhuma delas foi demonstrada por esta simulação.

O mesmo vale para a ideia de corrupção prolongada. Podemos imaginar, como cenário extremo, que pessoas com funções internas sejam subornadas ao longo de anos e que órgãos de fiscalização ou de resposta deixem de agir. Isso é uma premissa do exercício, **não uma acusação de que tais fatos ocorreram**. E não significa que toda pessoa, toda instituição ou todo registro externo passe a

obedecer a um mesmo grupo. Precisamos dizer exatamente quais verificações deixam de funcionar no cenário.

Vamos subir um degrau de cada vez

Para não esconder etapas dentro da palavra “fraude”, faremos seis paradas. Em cada parada, assumimos que **mais uma** camada foi comprometida. A pergunta é: *quais verificações ainda restam?* A tabela não informa porcentagens reais.

Etapa do exercício	O que passamos a supor	O que ainda poderia barrar ou revelar o desvio
0	Nenhuma camada foi comprometida.	Todas as verificações continuam disponíveis.
1	Pessoas ou credenciais ligadas a chaves e autorizações foram comprometidas.	Preparação das mídias, conferência do software instalado, teste, BU/RDV e fiscalização externa. Acesso a chaves, sozinho, não prova acesso a todas as urnas.
2	Além disso, um programa ou uma mídia indevida consegue atravessar a preparação.	O teste de integridade, os registros locais e a comparação externa ainda podem revelar comportamento errado.
3	Além disso, o comportamento indevido não aparece no teste.	BU, RDV, logs e cópias externas continuam sendo perguntas obrigatórias.
4	Além disso, os registros locais relevantes são alterados ou sua conferência falha.	Cópias ou evidências realmente independentes e a fiscalização externa ainda importam.
5	Além disso, a fiscalização e a resposta institucional consideradas no modelo falham.	Nenhuma das **cinco camadas que escolhemos** interrompe esse caminho hipotético; ainda pode haver controles ou vestígios que o modelo omitiu.

Na etapa 1, dizer “roubaram todas as chaves” parece encerrar a história. Não encerra. É preciso mostrar o que essas chaves permitem fazer, se têm escopo limitado e quais verificações independem delas. Na etapa 2, o caminho avança, mas ainda enfrenta um teste. Na etapa 3, superar o teste não apaga um BU impresso nem um RDV. Na etapa 4, mesmo registros internos coerentes podem ser confrontados com material que saiu do domínio controlado. Na etapa 5, o modelo chega ao limite que você propôs: todas as proteções que decidimos incluir deixaram de bloquear aquele caminho.

Isso **abre uma possibilidade no cenário**, mas ainda não nos dá uma chance medida de fraude. Para chegar a uma fraude efetiva, alguém teria de tentar, executar uma alteração que funcionasse, mudar votos ou resultado de modo relevante e não ser detido por controles que nossa tabela não incluiu. Se definirmos todas essas coisas como verdadeiras por hipótese, o computador devolverá 100%. Esse 100% é a resposta à frase que escrevemos, não uma descoberta sobre uma eleição.

No Anexo B, o leitor pode repetir a árvore lógica e os pequenos sorteios feitos pelo programa. Os números de 5%, 2% ou 80% usados lá são **peças de brinquedo matemático**, escolhidas para mostrar como uma conta funciona. Não são a chance de corromper pessoas, contornar testes ou alterar uma eleição. Mesmo multiplicar esses números exige uma suposição adicional: que as etapas sejam independentes. Num cenário de corrupção coordenada, essa suposição pode ser especialmente fraca.

Portanto, ao final de cada degrau, a resposta honesta à pergunta “qual é a chance real?” é: **ainda não sabemos**. Para calcular algo defensável, precisaríamos de evidências sobre acesso,

funcionamento dos controles, dependência entre falhas, extensão do ataque e tentativas observadas. Enquanto não tivermos isso, o exercício serve para organizar perguntas e identificar provas necessárias. É assim que uma hipótese forte se torna uma investigação séria.

Parte IV — Soluções que precisam ser comparadas

Capítulo 8 — Criptografia, registros distribuídos e prova de resultado

Uma tecnologia interessante não resolve automaticamente todas as etapas de uma eleição. Antes de propor uma ferramenta, precisamos perguntar qual problema ela pretende resolver: preservar o sigilo, mostrar que cada voto foi incluído, conferir a soma, resistir a uma troca de software ou permitir a correção de um resultado contestado? As respostas podem exigir mecanismos diferentes.

A criptografia homomórfica permite executar certas operações sobre dados cifrados. Em esquemas de votação, ela pode ajudar a somar votos sem revelar individualmente cada escolha. Isso não basta, isoladamente, para assegurar que a escolha foi registrada como o eleitor pretendia, que cada voto válido entrou uma única vez ou que a abertura do total cifrado foi correta. Protocolos verificáveis precisam de provas e procedimentos adicionais. Há propostas estudadas pela literatura de verificabilidade de ponta a ponta; seu interesse está no conjunto do protocolo, não apenas na propriedade matemática da cifra. [7]

Um registro distribuído pode tornar alterações posteriores mais visíveis quando cópias independentes e regras de validação são bem definidas. Se os validadores estiverem sob o mesmo controle, ou se dados incorretos forem inseridos desde a origem, a palavra

blockchain não oferece independência por si só. Nesse caso, é mais útil descrever quem publica, quem valida, quem pode contestar e qual registro permite detectar a divergência.

O papel conferido pelo eleitor propõe outra família de evidência: um registro físico que pode ser comparado com a apuração eletrônica. Sua força depende da oportunidade real de conferência, da preservação do sigilo, da custódia dos registros e de uma auditoria com amostragem e critérios definidos. Impressoras podem falhar, eleitores podem deixar de conferir e registros podem ser mal guardados. A proposta precisa enfrentar essas condições, além do custo de implantação. O NIST descreve auditorias por amostragem como forma de testar resultados com registros em papel, desde que os registros e a seleção da amostra sustentem o método. [8]

Por que ainda pergunto sobre um registro em papel

Eu não afirmo que seja impossível auditar o sistema eleitoral brasileiro. As normas e páginas oficiais descrevem verificações de assinaturas e resumos digitais, reimpressão e comparação de boletins de urna, análise do Registro Digital do Voto e inspeção de código-fonte por entidades habilitadas. Essas possibilidades precisam ser consideradas quando avalio o sistema. [10]

Minha pergunta é mais específica: que evidência independente do próprio software permite a uma eleitora conferir, depois de votar, que a escolha que fez foi registrada como pretendia? Um registro físico verificável poderia oferecer uma resposta diferente, mas seu desenho também teria de proteger o sigilo, evitar coerção, prever custódia e definir auditorias. Por isso, defendo que essa alternativa continue sendo debatida; não trato sua ausência como prova de fraude.

Também é importante descrever corretamente por que o modelo atual não imprime um comprovante individual do voto. Em 2020, o Supremo Tribunal Federal julgou inconstitucional a regra legal que exigia a impressão do registro individual, apontando riscos ao sigilo e à liberdade do voto. Portanto, não descrevo a situação como uma simples “recusa do TSE”: houve uma decisão judicial sobre a lei então vigente. Posso questionar se outros desenhos conciliariam auditabilidade e sigilo, mas preciso apresentar a decisão e os argumentos institucionais com fidelidade. [13]

Minha desconfiança é uma opinião sobre o grau de confiança que considero suficiente e sobre a independência das provas disponíveis. Ela não é, por si só, evidência de adulteração. Do mesmo modo, a existência de procedimentos oficiais não obriga todo cidadão a considerá-los suficientes: cabe explicar o que cada um verifica, quais pressupostos exige e quais perguntas permanecem abertas.

Essas três linhas de solução merecem uma comparação honesta. Para cada uma, procuro identificar que evidência independente ela produz, quais pessoas ou equipamentos ainda precisam ser confiáveis e o que acontece quando algo falha. Esse critério ajuda a avaliar propostas de voto impresso auditável no contexto brasileiro sem tratar nenhuma tecnologia como solução mágica.

Conclusão — O que as simulações me permitem afirmar

Ao longo deste livro, procurei separar três coisas que muitas vezes aparecem misturadas: o que os procedimentos documentados dizem, o que uma simulação hipotética consegue mostrar e o que seria necessário provar sobre uma eleição real.

Nos exemplos de percentuais, construí a mesma sequência de atualizações de duas maneiras: com lotes hipotéticos que chegam sem um mecanismo de controle e com lotes escolhidos por um amortizador também hipotético. Isso mostra que um gráfico sozinho não identifica a causa que o produziu. A faixa de composição capaz de manter um número arredondado depende do total acumulado, do tamanho do lote e do ponto de partida; por isso, ela precisa ser recalculada a cada etapa. Uma mudança pequena pode continuar invisível dentro da faixa de arredondamento ou alterar o degrau quando cruza seu limite.

Esse resultado é matemático, mas não é uma estimativa da frequência com que lotes reais cairiam nessas faixas. Não defini uma distribuição realista para a ordem, o tamanho e a composição das atualizações, nem reconstruí aqui uma série cronológica completa de um pleito real. Portanto, não uso o exercício para afirmar que houve algoritmo ou fraude. Também não uso a falta de uma série que eu tenha conseguido reproduzir como prova de que ela não exista ou de que tenha sido ocultada de propósito.

Na árvore de barreiras, o resultado de 100% aparece somente quando todas as condições modeladas são fixadas como superadas. O programa não calcula a chance de alguém tentar uma manipulação, de a alteração funcionar ou de o resultado ser efetivamente mudado. Esse número descreve o cenário lógico que escrevi, não uma probabilidade eleitoral.

As regras e os procedimentos oficiais consultados descrevem formas de fiscalização e comparação. Reconhecê-las é parte de uma análise honesta; explicar seus limites e perguntar que evidência independente cada uma oferece também é. Minha desconfiança é uma opinião sobre o grau de confiança que considero suficiente. Ela não substitui evidência, assim como uma simulação possível não comprova um acontecimento.

Termino, então, sem pedir ao leitor que aceite uma conclusão pronta. Peço que examine as premissas, repita as contas, consulte as fontes e distinga possibilidade de demonstração. Para mim, a pergunta que permanece é como ampliar a transparência e a verificabilidade independente sem comprometer o sigilo e a liberdade do voto. O debate pode continuar aberto justamente porque dúvidas legítimas merecem respostas verificáveis — e porque acusações sobre fatos precisam de provas proporcionais ao seu peso.

Anexo A — Simulação dos percentuais acumulados

Este anexo acompanha o capítulo “A matemática por trás de uma tela em degraus”. O mesmo código está disponível como arquivo executável em simulacao_fluxo_apuracao.py. Para repetir o experimento, salve o arquivo e execute no terminal:

```
python simulacao_fluxo_apuracao.py
```

O programa usa somente a biblioteca padrão do Python 3. Os dados são hipotéticos: 100 milhões de votos válidos iniciais, distribuídos entre dois candidatos, e dez lotes novos de 500 mil votos cada. O cenário homogêneo atribui 70% de todos os lotes ao candidato B. O segundo altera essa participação no meio da sequência.

O resultado impresso contém o percentual de B com quatro casas decimais e os percentuais de A e B arredondados a uma casa decimal. Isso permite ver pequenas mudanças que desaparecem na apresentação simplificada.

Código integral

```
# -*- coding: utf-8 -*-
"""Anexo A: efeito do arredondamento em percentuais acumulados.

Execute com Python 3: python simulacao_fluxo_apuracao.py

Os números são hipotéticos. Este programa não lê boletins de urna, não usa
dados de uma eleição e não simula a infraestrutura da Justiça Eleitoral.
"""

VOTOS_INICIAIS_A = 52_000_000
VOTOS_INICIAIS_B = 48_000_000
TAMANHO_LOTE = 500_000
```

```
CENARIOS = {
    "Fluxo homogêneo": [70] * 10,
    "Fluxo com mudança de perfil": [70, 70, 70, 50, 30, 30, 68, 68, 68,
68],
}


def simular(percentuais_b):
    """Retorna as dez atualizações de uma disputa hipotética entre A e B.

    Cada item de percentuais_b é a participação inteira de B no novo lote.
    O percentual exibido usa arredondamento a uma casa decimal (``.1f``).
    """
    votos_a = VOTOS_INICIAIS_A
    votos_b = VOTOS_INICIAIS_B
    resultado = []

    for numero, percentual_b_lote in enumerate(percentuais_b, start=1):
        if not 0 <= percentual_b_lote <= 100:
            raise ValueError("Cada participação deve estar entre 0 e
100%.")

        novos_votos_b = TAMANHO_LOTE * percentual_b_lote // 100
        novos_votos_a = TAMANHO_LOTE - novos_votos_b
        votos_a += novos_votos_a
        votos_b += novos_votos_b
        total = votos_a + votos_b

        percentual_b_exato = 100 * votos_b / total
        percentual_a_exato = 100 * votos_a / total
        resultado.append(
            (
                numero,
                percentual_b_lote,
                percentual_b_exato,
                f"{percentual_b_exato:.1f}",
                f"{percentual_a_exato:.1f}",
            )
        )

    return resultado


def main():
    print("SIMULAÇÃO DIDÁTICA: PERCENTUAIS ACUMULADOS")
    print("Início: A = 52.000.000; B = 48.000.000 votos válidos")
    print("Cada lote: 500.000 votos; interface: uma casa decimal")

    for nome, percentuais in CENARIOS.items():
        print(f"\n{nome}")
```

```
        print("Lote | B no lote | B exato | B exibido | A exibido")
        print("-----+-----------+----------+-----------+----------")
        for numero, perfil, exato, exibido_b, exibido_a in
simular(percentuais):
            print(
                f"{numero:>4} | {perfil:>8}% | {exato:>7.4f}% |"
                f" {exibido_b:>8}% | {exibido_a:>8}%"
            )

    print("\nA sequência escolhida mostra possibilidades matemáticas;")
    print("não permite inferir como chegaram votos de uma eleição real.")


if __name__ == "__main__":
    main()
```

Como interpretar a saída

No primeiro cenário, B sobe de 48,1% para 49,0% na apresentação com uma casa decimal. No segundo, a sequência é 48,1%; 48,2%; 48,3%; 48,3%; 48,2%; 48,2%; 48,3%; 48,3%; 48,4%; 48,5%. O lote em que B recebe 50% não inverte a curva real, pois 50% ainda é maior que sua participação acumulada naquele momento. Seu efeito apenas fica oculto pelo arredondamento.

O código produz uma demonstração determinística: executar novamente com as mesmas entradas produz a mesma tabela. Ele não calcula a probabilidade de fraude, não executa 10.000 simulações e não utiliza dados de votação real. Para estudar uma eleição específica, seria necessário substituir as entradas hipotéticas por dados documentados e verificar como a interface usada naquele caso calcula e exibe percentuais.

Anexo B — Árvore lógica de barreiras

Este anexo acompanha o capítulo “Uma porta, várias fechaduras”. O arquivo executável correspondente é simulacao_arvore_barreiras.py. Para repetir o exercício:

```
python simulacao_arvore_barreiras.py
```

O programa usa somente a biblioteca padrão do Python 3. Em cada um dos três primeiros cenários, sorteia dez mil vezes se etapas hipotéticas foram superadas. Para multiplicar as probabilidades, o modelo **supõe independência entre as etapas**. Os valores foram escolhidos para ensinar a mecânica do cálculo; não vieram de medições de uma eleição. O resultado do sorteio pode diferir do valor esperado, especialmente quando os eventos são raros.

Depois, ele mostra os **degraus** do capítulo: primeiro nenhuma camada comprometida, depois uma, duas e assim por diante. Para que apareçam números, usamos uma moeda imaginária: cada camada ainda ativa teria 50% de chance de ser superada em uma tentativa fictícia. Essa escolha é deliberadamente simples. Ela **não mede** a chance de violar chaves, enganar uma auditoria, subornar alguém ou fraudar uma eleição. Quando todas as cinco camadas são dadas como comprometidas, a conta chega a 100% de passagem pelas **camadas restantes do exercício**, pois não resta nenhuma. Não chega a 100% de fraude real.

No cenário limite, todas as condições são fixadas como verdadeiras. A porta AND permanece operando: basta uma entrada falsa para que a saída também seja falsa. O programa não sorteia esse cenário

nem imprime uma taxa de fraude. Ele também não modela se houve uma tentativa de manipulação ou uma alteração efetiva dos votos.

Código integral

```
# -*- coding: utf-8 -*-
"""Anexo B: árvore lógica e Monte Carlo com parâmetros didáticos.

Execute com Python 3: python simulacao_arvore_barreiras.py

Os parâmetros abaixo foram escolhidos apenas para ensinar a lógica do
modelo.
Não foram medidos em uma eleição e não estimam risco de fraude real.
A saída conta percursos pelas etapas escolhidas, não manipulações de
votos.
"""

from math import prod
from random import Random

RODADAS = 10_000
SEMENTE = 42

# Cada número é a chance HIPOTÉTICA de o atacante superar uma etapa.
# O modelo pressupõe independência entre as etapas; sistemas reais podem
# apresentar dependências, controles adicionais e outros caminhos de
ataque.
CENARIOS = (
    (
        "I — cadeia de desenvolvimento",
        (
            ("acesso à raiz", 0.05),
            ("evasão de teste", 0.02),
            ("ausência de alerta nos registros", 0.80),
        ),
    ),
    (
        "II — acesso interno assumido",
        (
            ("acesso à raiz", 1.00),
            ("registros internos dominados", 1.00),
            ("evasão de teste", 0.02),
            ("ausência de contestação externa", 0.05),
        ),
    ),
    (
        "III — evasão de teste assumida",
        (
```

```
            ("acesso à raiz", 1.00),
            ("registros internos dominados", 1.00),
            ("evasão de teste", 1.00),
            ("superação de controle de hardware", 0.20),
            ("ausência de contestação externa", 0.30),
        ),
    ),
)

# Outro exercício: uma camada é dada como comprometida a cada degrau.
# 50% é uma moeda imaginária, NÃO uma estimativa da segurança eleitoral.
CAMADAS = (
    "chaves e autorizações",
    "preparação e conferência de mídias",
    "teste de integridade",
    "BU, RDV e conferências locais",
    "fiscalização e resposta externa",
)
CHANCE_DIDATICA_POR_CAMADA_RESTANTE = 0.50


def simular(etapas, semente):
    """Conta quantas rodadas passam por todas as etapas hipotéticas."""
    gerador = Random(semente)
    caminhos_completos = 0

    for _ in range(RODADAS):
        resultados = [gerador.random() < chance for _, chance in etapas]
        if all(resultados):
            caminhos_completos += 1

    taxa_teorica = prod(chance for _, chance in etapas)
    return caminhos_completos, taxa_teorica


def mostrar_degraus():
    """Mostra a mudança no modelo ao assumir mais camadas
comprometidas."""
    print("DEGRAUS — UMA NOVA CAMADA COMPROMETIDA A CADA LINHA")
    print("Exemplo fictício: chance de passar por CADA camada ainda ativa
= 50%.")
    print("A coluna de porcentagens NÃO é chance de fraude real.")
    for quantidade in range(len(CAMADAS) + 1):
        ativas = CAMADAS[quantidade:]
        chance_no_exercicio = CHANCE_DIDATICA_POR_CAMADA_RESTANTE **
len(ativas)
        print(
            f" {quantidade} comprometida(s), {len(ativas)} ativa(s): "
            f"passagem pelas camadas RESTANTES no brinquedo =
{chance_no_exercicio:.3%}"
        )
```

```
        if ativas:
            print(" Ainda ativa(s): " + "; ".join(ativas))
        else:
            print(" Nenhuma camada MODELADA resta; tentativa e alteração
não foram modeladas.")
    print()


def main():
    print("ÁRVORE DE BARREIRAS — EXERCÍCIO DIDÁTICO")
    print(f"{RODADAS:,} rodadas por cenário; semente = {SEMENTE}")
    print("Os parâmetros são hipóteses arbitrárias, não medições
eleitorais.\n")

    for numero, (nome, etapas) in enumerate(CENARIOS):
        caminhos_completos, taxa_teorica = simular(etapas, SEMENTE +
numero)
        print(nome)
        for etapa, chance in etapas:
            print(f" {etapa}: {chance:.0%}")
        print(f" Caminhos completos no sorteio:
{caminhos_completos}/{RODADAS}")
        print(f" Valor esperado: {RODADAS * taxa_teorica:.1f}/{RODADAS}")
        print(f" Taxa assumida pelo modelo: {taxa_teorica:.2%}\n")

    mostrar_degraus()

    # Cenário limite: cada condição abaixo é uma premissa, não uma
medição.
    # A tentativa de manipular votos e a alteração efetiva não são
modeladas.
    condicoes_superadas = {
        "acesso à raiz": True,
        "evasão do teste": True,
        "ausência de alerta interno": True,
        "ausência de contestação externa efetiva": True,
    }
    caminho_sem_bloqueio = all(condicoes_superadas.values())
    condicoes_com_barreira_ativa = dict(condicoes_superadas)
    condicoes_com_barreira_ativa["evasão do teste"] = False

    print("LIMITE LÓGICO — CONDIÇÕES FIXAS, SEM SORTEIO")
    print(f" Todas as condições modeladas superadas: {'sim' if
caminho_sem_bloqueio else 'não'}")
    print(
        " Uma barreira ativa interrompe o caminho: "
        f"{'sim' if not all(condicoes_com_barreira_ativa.values()) else
'não'}"
    )
    print(" Tentativa de manipulação: não modelada")
    print(" Alteração efetiva do resultado: não modelada")
```

```
    print(" Probabilidade de fraude real: não calculada")


if __name__ == "__main__":
    main()
```

O que a tabela permite concluir

Sob as hipóteses escolhidas e de independência, os valores esperados dos três primeiros cenários são 8, 10 e 600 caminhos completos em 10.000 tentativas. Uma execução com a semente indicada pode apresentar quantidades diferentes por causa da variação aleatória. O cenário limite apenas avalia uma expressão lógica com entradas fixas; não produz uma quarta taxa.

Alterar uma porcentagem no código explora a sensibilidade do modelo àquela premissa. Não transforma a porcentagem escolhida em medida de segurança de uma instituição. Além disso, uma comparação entre o BU impresso e o BU publicado testa a correspondência entre esses registros; não detecta, por si só, todo possível erro anterior à produção do BU.

Nos degraus, os números do brinquedo são 3,125%, 6,25%, 12,5%, 25%, 50% e 100%. Eles sobem porque retiramos uma camada por vez **e porque decidimos**, sem medições, usar a mesma chance de 50% em cada camada restante. Se qualquer parâmetro mudar, a tabela muda. Se as camadas dependerem umas das outras, nem a multiplicação simples será adequada.

Anexo C — Que composição de lote faz a tela mudar?

Este anexo responde a uma pergunta diferente da simulação de dois candidatos do Anexo A: que participação de votos para cada grupo em um lote novo faria os percentuais acumulados subirem, descerem ou permanecerem iguais na tela?

O exemplo usa uma base inventada de 100 milhões de votos válidos: 47 milhões para o Candidato A, 45 milhões para o Candidato B e 8 milhões para os Candidatos C, D, E e F. Depois, acrescenta um lote hipotético de 5 milhões de votos. Esses números foram escolhidos para a conta ficar fácil de conferir; não são os dados de uma apuração real.

O que significa “porcentagem do lote”

Se um lote contém 5 milhões de votos válidos, dizer “44,9% para o Candidato A” significa que 2.245.000 votos daquele lote foram para A. As parcelas de A, B e dos demais candidatos precisam somar 100%. Não podemos escolher livremente, por exemplo, 35% para A, 65% para B e ainda acrescentar votos para outros candidatos: essa soma já seria 100%.

Em cada atualização, a porcentagem acumulada é calculada dividindo os votos acumulados do candidato pelo total acumulado de votos válidos. Em palavras:

```
novo percentual = (votos anteriores + votos do lote) /
(total anterior + tamanho do lote)
```

Para descobrir o efeito do lote, compare a participação dele com a participação que cada candidato já tinha. Se o lote der a A uma fatia menor que seus 47% acumulados, seu percentual tende a cair. Se der a B uma fatia maior que seus 45% acumulados, o percentual de B tende a subir. O tamanho do lote também importa: cinco milhões de votos mexem mais na conta do que cinquenta mil.

Um lote que dá exatamente os dois passos

Com a base e o lote hipotéticos acima, uma composição de **44,9% para A, 47,1% para B e 8,0% para os demais** produz estes totais:

Grupo	Base antes	Votos no lote	Percentual acumulado antes	Percentual acumulado depois
Candidato A	47.000.000	2.245.000	47,0%	46,9%
Candidato B	45.000.000	2.355.000	45,0%	45,1%
Candidatos C-F	8.000.000	400.000	8,0%	8,0%

Nesse exemplo, os valores exatos depois do lote são 46,9%, 45,1% e 8,0%. Assim, o modelo mostra como uma atualização pode produzir o passo de menos 0,1 ponto para A e mais 0,1 ponto para B, sem que um percentual seja simplesmente o espelho do outro: há outros candidatos no denominador.

Também podemos calcular uma faixa, em vez de procurar um único número. Com arredondamento comum para uma casa decimal, a tela mostrar 46,9% para A significa que o valor exato ficou aproximadamente entre 46,85% e 46,95%. Para B mostrar 45,1%, o valor exato ficou aproximadamente entre 45,05% e

45,15%. Com a base de 100 milhões e lote de 5 milhões deste exemplo, isso permite várias composições de lote: cerca de 43,85% a 45,95% para A e 46,05% a 48,15% para B. O que sobra pode ir para os demais candidatos. Essas faixas são ligadas entre si: as três partes precisam somar 100%.

Em linguagem matemática, se `p` é a participação acumulada antes do lote, `p_novo` é a participação acumulada desejada depois, `N` é o total antes e `L` é o tamanho do lote, a participação necessária no próprio lote é:

```
participação no lote = [p_novo × (N + L) − p × N] / L
```

Para a tela arredondada, `p_novo` não é um único valor, mas uma faixa. Por isso, o resultado também é uma faixa. Se ela exigir mais que 100% ou menos que 0% de votos para um candidato, aquele lote, com aquele tamanho e aquela base, não conseguiria produzir o passo pretendido.

Os números necessários mudam conforme o total que já foi contado e o tamanho do próximo lote. Se o total acumulado for muito maior, um lote pequeno geralmente desloca menos a porcentagem real. Ainda assim, a tela pode trocar de 47,0% para 46,9% quando o valor exato cruza o limite entre duas faixas de arredondamento.

As faixas sugeridas e outras composições

Podemos testar, por exemplo, uma composição: 32,5% para A, 62,5% para B e 5% para os demais. Com o lote hipotético de 5 milhões, a tela sairia de 47,0% / 45,0% para aproximadamente **46,3% / 45,8%**. Essa composição não produz apenas um passo de 0,1: ela causa mudanças maiores, porque o lote é grande e está bem distante das participações acumuladas iniciais.

Outras composições também mudam o desenho. Se o lote trouxer 15% para A, 80% para B e 5% para os demais, este exemplo termina perto de 45,5% e 46,7%; com 20%, 75% e 5%, termina perto de 45,7% e 46,4%. Com 40%, 55% e 5%, os valores ficam perto de 46,7% e 45,5%. Já com 50%, 45% e 5%, A volta a crescer e B fica praticamente parado na tela. A tabela completa pode ser reproduzida com `python simulacao_composicao_lotes.py`.

Esses exemplos demonstram que pequenas ou grandes mudanças no perfil dos lotes podem produzir curvas diferentes. Não dizem qual foi a composição dos lotes recebidos em uma eleição real.

O arredondamento esconde uma faixa, não um número único

Uma tela que mostra uma casa decimal não revela o percentual exato. Pela regra usual de arredondamento, mostrar 46,9% pode corresponder a valores exatos desde aproximadamente 46,85% até antes de 46,95%. O limite exato depende da regra usada pela interface para os casos que ficam exatamente no meio.

Por isso, ver 46,9% numa atualização não prova que o valor real era exatamente 46,9000%. E ver a sequência 47,0%, 46,9%, 46,8% não significa que cada lote alterou o valor real em precisamente 0,1 ponto. A tela só informa que o valor atravessou uma faixa de arredondamento.

Para calcular a composição necessária num caso real, precisamos de quatro coisas em cada atualização: votos acumulados antes do lote, tamanho do lote novo, votos novos de cada candidato e regra de arredondamento da tela. Se só temos os percentuais arredondados, existem muitas composições possíveis; não dá para recuperar um único percentual “necessário”.

O que os dados permitem investigar

O TSE informa que seu Painel da evolução da totalização permite rever percentuais e totais ao longo do tempo. Na consulta que fiz para esta edição, eu não consegui obter, em formato que pudesse reproduzir, uma sequência completa de cada atualização com seus horários e totais intermediários. Esse relato descreve o limite da minha busca; não prova que a série inexista nem que alguém a tenha ocultado. Por isso, não apresento este exercício como reconstrução da cronologia real. O arquivo final de votação por seção permite conferir totais locais, mas não é, por si só, uma lista cronológica de chegada de cada boletim. São materiais diferentes, com granularidades diferentes. [1] [2]

Isso não confirma nem refuta, por si só, a impressão de que uma sequência vista em uma transmissão foi incomum. A falta de uma sequência reproduzível também não prova, sozinha, que ela foi escondida de propósito. O que podemos fazer sem essa cronologia é comparar explicações hipotéticas para o mesmo gráfico, deixando claro o que cada uma assume. O Anexo D faz essa comparação.

Também não basta dizer que uma variação “seria normal” ou “deveria acontecer”. Uma comparação estatística precisa considerar de quais regiões vieram os votos de cada lote, quantas seções entraram em cada atualização, o tamanho dos lotes, os percentuais dos demais candidatos e o momento da apuração. Se esses fatores não forem conhecidos, não é possível atribuir uma probabilidade confiável a uma sequência.

Este exercício, portanto, não afirma fraude nem calcula uma chance de fraude. Ele ensina a pergunta matemática: **dadas as contagens antes e depois de uma atualização, qual composição de lote seria compatível com os percentuais**

exibidos? Quando as atualizações reais não estão disponíveis numa série verificável, usamos números hipotéticos e não os apresentamos como se fossem observações.

Fontes e arquivos

1. TSE, Saiba como foi a totalização dos votos no 1º turno das Eleições 2026. A página descreve o painel de evolução e afirma que ele permite refazer a totalização ao longo do tempo.
2. TSE, Resultados — 2026. O conjunto inclui os arquivos finais por seção e informa que são atualizados ao final de cada turno.

O script associado, `simulacao_composicao_lotes.py`, usa somente a biblioteca padrão do Python 3. Seus cenários são didáticos e editáveis; para analisar uma eleição, substitua-os por contagens documentadas e preserve a fonte de cada dado.

Código integral

Salve o código abaixo como `simulacao_composicao_lotes.py` e execute `python simulacao_composicao_lotes.py`.

```
# -*- coding: utf-8 -*-
"""Simulação didática de lotes com três grupos de votos.

Execute com Python 3:
    python simulacao_composicao_lotes.py

Os dados são inteiramente hipotéticos. O programa não lê resultados
eleitorais
reais e não calcula a probabilidade de fraude.
"""

from decimal import Decimal, ROUND_HALF_UP


VOTOS_INICIAIS = {
    "Candidato A": 47_000_000,
```

```
    "Candidato B": 45_000_000,
    "Candidatos C-F": 8_000_000,
}
TAMANHO_LOTE = 5_000_000

# Participação de cada grupo no lote novo. Cada cenário precisa somar
100%.
CENARIOS = {
    "Composição que produz exatamente -0,1/+0,1 ponto": {
        "Candidato A": Decimal("44.9"), "Candidato B": Decimal("47.1"),
"Candidatos C-F": Decimal("8.0")
    },
    "Exemplo dentro das faixas sugeridas (32,5 / 62,5 / 5)": {
        "Candidato A": Decimal("32.5"), "Candidato B": Decimal("62.5"),
"Candidatos C-F": Decimal("5.0")
    },
    "Variação (15 / 80 / 5)": {
        "Candidato A": Decimal("15.0"), "Candidato B": Decimal("80.0"),
"Candidatos C-F": Decimal("5.0")
    },
    "Variação (20 / 75 / 5)": {
        "Candidato A": Decimal("20.0"), "Candidato B": Decimal("75.0"),
"Candidatos C-F": Decimal("5.0")
    },
    "Variação (40 / 55 / 5)": {
        "Candidato A": Decimal("40.0"), "Candidato B": Decimal("55.0"),
"Candidatos C-F": Decimal("5.0")
    },
    "Variação (50 / 45 / 5)": {
        "Candidato A": Decimal("50.0"), "Candidato B": Decimal("45.0"),
"Candidatos C-F": Decimal("5.0")
    },
}


def exibir_percentual(fracao):
    """Converte uma fração em porcentagem e arredonda para 1 casa
decimal."""
    percentual = Decimal(fracao) * Decimal(100)
    return percentual.quantize(Decimal("0.1"), rounding=ROUND_HALF_UP)


def validar(cenario):
    if any(valor < 0 or valor > 100 for valor in cenario.values()):
        raise ValueError("Cada participação deve estar entre 0% e 100%.")
    if sum(cenario.values()) != Decimal("100.0"):
        raise ValueError("As participações do lote precisam somar 100%.")


def simular(cenario):
    validar(cenario)
```

```
    total_inicial = sum(VOTOS_INICIAIS.values())
    total_final = total_inicial + TAMANHO_LOTE
    votos_lote = {
        nome: int(Decimal(TAMANHO_LOTE) * participacao / Decimal(100))
        for nome, participacao in cenario.items()
    }
    # Corrige eventual sobra de arredondamento no último grupo para
preservar o lote.
    votos_lote["Candidatos C-F"] += TAMANHO_LOTE -
sum(votos_lote.values())

    print(f"\n{sum(cenario.values())}% do lote; lote = {TAMANHO_LOTE:,}
votos")
    print("Grupo | No lote | Votos no lote | Acumulado exato | Exibido
antes -> depois")
    print("--------+---------+--------------+-----------------+---------
--------------")
    for nome, votos_antes in VOTOS_INICIAIS.items():
        votos_depois = votos_antes + votos_lote[nome]
        exibido_antes = exibir_percentual(Decimal(votos_antes) /
Decimal(total_inicial))
        exibido_depois = exibir_percentual(Decimal(votos_depois) /
Decimal(total_final))
        exato = Decimal(votos_depois * 100) / Decimal(total_final)
        print(
            f"{nome:<7} | {cenario[nome]:>6}% | {votos_lote[nome]:>13,} |
"
            f"{exato:>14.4f}% | {exibido_antes:>5}% ->
{exibido_depois:>5}%"
        )


def main():
    total = sum(VOTOS_INICIAIS.values())
    print("SIMULAÇÃO DIDÁTICA — COMPOSIÇÃO DE UM LOTE E PERCENTUAIS
ACUMULADOS")
    print(f"Base hipotética: {total:,} votos válidos")
    for nome, votos in VOTOS_INICIAIS.items():
        print(f" {nome}: {votos:,} ({Decimal(votos * 100) /
Decimal(total):.2f}%)")
    print("A soma de A, B e dos Candidatos C-F no lote deve ser 100%.")
    for nome, cenario in CENARIOS.items():
        print(f"\n{nome}")
        simular(cenario)
    print("\nSão cenários matemáticos inventados, não observações de uma
eleição.")


if __name__ == "__main__":
    main()
```

Anexo D — Duas hipóteses, a mesma escadinha

Este anexo coloca duas explicações hipotéticas lado a lado. A primeira não supõe fraude nem algoritmo: lotes de votos chegam com determinadas composições e, depois da soma e do arredondamento, a tela mostra uma escadinha. A segunda supõe um mecanismo de amortização que escolhe a composição de cada lote para atingir os mesmos degraus.

As duas explicações serão comparadas com os mesmos números. Isso não quer dizer que tenham a mesma evidência a favor. O exercício mostra apenas que uma mesma sequência exibida pode ser construída sob premissas diferentes. Nenhum dos cenários abaixo reconstrói uma eleição real ou calcula a probabilidade de fraude.

A base comum

Começo com 100 milhões de votos válidos: 47 milhões para o Candidato A, 45 milhões para o Candidato B e 8 milhões para os Candidatos C, D, E e F. Cada atualização acrescenta 5 milhões de votos válidos. A tela apresenta uma casa decimal.

Nos dois cenários, queremos visualizar esta mesma sequência:

Momento	Candidato A	Candidato B	Candidatos C-F
Antes dos lotes	47,0%	45,0%	8,0%
Depois do lote 1	46,9%	45,1%	8,0%

Momento	Candidato A	Candidato B	Candidatos C-F
Depois do lote 2	46,8%	45,2%	8,0%
Depois do lote 3	46,7%	45,3%	8,0%

Hipótese 1 — Sem algoritmo de amortização

Nesta hipótese, ninguém ajusta os números para atingir um gráfico. Os resultados simplesmente chegam em três lotes cuja composição é:

Lote	Candidato A	Candidato B	Candidatos C-F
1	44,9%	47,1%	8,0%
2	44,7%	47,3%	8,0%
3	44,5%	47,5%	8,0%

Somando cada lote ao total anterior, a tela apresenta os quatro pontos da sequência acima. Isto é uma demonstração de possibilidade matemática. Como escolhemos essas composições para construir o exemplo, não podemos usar o resultado para dizer que a sequência seria comum, provável ou foi o que ocorreu numa apuração real.

Uma explicação sem algoritmo também pode incluir lotes de tamanhos e composições diferentes. O efeito exibido depende tanto da porcentagem dentro do lote quanto do peso dele em relação ao total já acumulado. A fórmula é:

```
novo percentual = (votos anteriores + votos do lote) /
(total anterior + tamanho do lote)
```

Assim, lotes menores tendem a deslocar menos o percentual acumulado; lotes maiores podem deslocá-lo mais. O arredondamento pode esconder uma parte dessa mudança.

Hipótese 2 — Com um amortizador hipotético

Agora imaginamos um programa que, a cada etapa, calcula quais proporções de votos fariam a tela mostrar o próximo degrau escolhido. Para os mesmos três passos, o programa encontra as mesmas composições da tabela anterior: 44,9% / 47,1% / 8,0%; depois 44,7% / 47,3% / 8,0%; e, por fim, 44,5% / 47,5% / 8,0%.

Neste cenário, as proporções são escolhidas pelo mecanismo hipotético; no primeiro, elas são as composições assumidas para os lotes que chegaram sem esse mecanismo. O gráfico final é idêntico. A diferença entre as hipóteses está na causa presumida, não nos pontos desenhados.

Para o primeiro lote de 5 milhões, a tela mostrar 46,9% para A e 45,1% para B admite várias composições. Com arredondamento convencional para uma casa decimal, as faixas aproximadas são 43,85% a 45,95% para A e 46,05% a 48,15% para B. O restante cabe aos demais. As faixas são interdependentes: A, B e os demais precisam somar 100%.

Uma composição que permaneça dentro das faixas pode mudar sem alterar o número exibido. Por exemplo, 44,8% / 47,0% / 8,2% também mostra 46,9% / 45,1%. Já 43,8% / 47,1% / 9,1% mostra 46,8% / 45,1%: o passo de A muda. Se B recebe 48,2%, com 44,9% para A e 6,9% para os demais, a tela mostra 46,9% / 45,2%: o passo de B muda.

Portanto, a frase precisa ser precisa: **uma variação que ultrapasse o limite de arredondamento compatível com o**

degrau altera a leitura exibida; uma variação que permaneça dentro da faixa pode ficar invisível. A faixa não é fixa para sempre. Ela precisa ser recalculada a cada atualização, porque o total acumulado, o tamanho do lote e os percentuais anteriores mudam.

O que esse teste permite dizer

O exemplo permite ao leitor verificar três coisas:

1. Uma sequência de degraus pode ser produzida matematicamente sem incluir um algoritmo no modelo.
2. Um algoritmo hipotético também poderia ser descrito como mecanismo que seleciona composições para atingir os mesmos degraus.
3. Fora das faixas compatíveis com o arredondamento, o degrau mostrado muda; dentro delas, alterações podem não aparecer.

O exemplo não permite estimar a chance de os lotes naturais caírem nessas faixas, porque não definimos uma distribuição realista para tamanho, ordem e composição dos lotes. Também não demonstra que um algoritmo existiu. Para escolher entre explicações sobre uma eleição real, seriam necessárias evidências além do gráfico sintético. O leitor pode comparar as premissas e os limites, mas não deve confundir uma hipótese simulada com um fato provado.

Código para repetir a comparação

Salve o código como `simulacao_duas_hipoteses.py` e execute `python simulacao_duas_hipoteses.py`. Ele calcula a mesma sequência sob as duas narrativas e mostra exemplos de variação dentro e fora da faixa. Todos os valores são inventados.

```
"""Compara duas explicações hipotéticas para a mesma escadinha.

Execute com Python 3. Os números são didáticos e não representam uma
eleição real.
"""

from decimal import Decimal, ROUND_HALF_UP


INICIAL = {"Candidato A": 47_000_000, "Candidato B": 45_000_000,
"Candidatos C-F": 8_000_000}
TAMANHO_LOTE = 5_000_000
ALVOS = [
    (Decimal("46.9"), Decimal("45.1")),
    (Decimal("46.8"), Decimal("45.2")),
    (Decimal("46.7"), Decimal("45.3")),
]


def tela(votos, total):
    return {
        nome: (Decimal(voto) * 100 / Decimal(total)).quantize(
            Decimal("0.1"), rounding=ROUND_HALF_UP
        )
        for nome, voto in votos.items()
    }


def somar_lote(votos, lote):
    return {nome: votos[nome] + lote[nome] for nome in votos}


def imprimir_cenario(titulo, lotes):
    votos = INICIAL.copy()
    total = sum(votos.values())
    print(f"\n{titulo}")
    atual = tela(votos, total)
    print(f"Antes: Candidato A {atual['Candidato A']}% | Candidato B
{atual['Candidato B']}%")
    for numero, lote in enumerate(lotes, start=1):
        votos = somar_lote(votos, lote)
```

```
        total += TAMANHO_LOTE
        atual = tela(votos, total)
        print(f"Lote {numero}: Candidato A {atual['Candidato A']}% |
Candidato B {atual['Candidato B']}%")


def lote_de_composicao(participacao_a, participacao_b):
    participacao_a = Decimal(str(participacao_a))
    participacao_b = Decimal(str(participacao_b))
    outros = Decimal("100") - participacao_a - participacao_b
    if min(participacao_a, participacao_b, outros) < 0:
        raise ValueError("As participações precisam somar 100%, sem
valores negativos.")
    votos_a = int(Decimal(TAMANHO_LOTE) * participacao_a / 100)
    votos_b = int(Decimal(TAMANHO_LOTE) * participacao_b / 100)
    return {
        "Candidato A": votos_a,
        "Candidato B": votos_b,
        "Candidatos C-F": TAMANHO_LOTE - votos_a - votos_b,
    }


def lote_alvo(votos, total, alvo_a, alvo_b):
    """Calcula votos para levar os acumulados ao centro dos alvos
escolhidos."""
    tamanho = Decimal(TAMANHO_LOTE)
    total_novo = Decimal(total) + tamanho
    votos_a = total_novo * alvo_a / 100 - Decimal(votos["Candidato A"])
    votos_b = total_novo * alvo_b / 100 - Decimal(votos["Candidato B"])
    votos_outros = tamanho - votos_a - votos_b
    if min(votos_a, votos_b, votos_outros) < 0:
        raise ValueError("O alvo não é possível com o tamanho de lote
escolhido.")
    a_int = int(votos_a)
    b_int = int(votos_b)
    return {
        "Candidato A": a_int,
        "Candidato B": b_int,
        "Candidatos C-F": TAMANHO_LOTE - a_int - b_int,
    }


def main():
    # Hipótese 1: os lotes chegam com estas composições, sem controlador.
    lotes_sem_algoritmo = [
        lote_de_composicao("44.9", "47.1"),
        lote_de_composicao("44.7", "47.3"),
        lote_de_composicao("44.5", "47.5"),
    ]
```

```
    # Hipótese 2: um amortizador hipotético escolhe os lotes para atingir
os alvos.
    votos = INICIAL.copy()
    total = sum(votos.values())
    lotes_com_amortizador = []
    for alvo_a, alvo_b in ALVOS:
        lote = lote_alvo(votos, total, alvo_a, alvo_b)
        lotes_com_amortizador.append(lote)
        votos = somar_lote(votos, lote)
        total += TAMANHO_LOTE

    imprimir_cenario("Hipótese 1 — sem algoritmo", lotes_sem_algoritmo)
    imprimir_cenario("Hipótese 2 — amortizador hipotético",
lotes_com_amortizador)

    imprimir_cenario(
        "Variação dentro da faixa",
        [lote_de_composicao("44.8", "47.0")],
    )
    imprimir_cenario(
        "Variação fora da faixa de A",
        [lote_de_composicao("43.8", "47.1")],
    )
    imprimir_cenario(
        "Variação fora da faixa de B",
        [lote_de_composicao("44.9", "48.2")],
    )
    print("\nTodos os cenários são hipotéticos; não são dados de uma
eleição real.")


if __name__ == "__main__":
    main()
```

Referências e fontes consultadas

Fontes online consultadas em 12 de outubro de 2026.

[1] Tribunal Superior Eleitoral. Resolução nº 23.751/2026: votação, apuração e boletins.

[2] Tribunal Superior Eleitoral. Resolução nº 23.758/2026: procedimentos de fiscalização e teste de integridade.

[3] Ronald L. Rivest e John P. Wack. On the notion of software independence in voting systems; ver também a discussão do NIST sobre auditabilidade.

[4] Tribunal Superior Eleitoral. Divulgação e publicação de resultados e boletins em 2026.

[5] Ken Thompson. Reflections on Trusting Trust, Communications of the ACM, 1984.

[6] Tribunal Superior Eleitoral. Teste Público de Segurança dos Sistemas Eleitorais e resultados do teste de 2025.

[7] Ronald L. Rivest e Madars Virza. Software Independence Revisited, 2016; National Institute of Standards and Technology. Desirable Properties of Voting Systems.

[8] National Institute of Standards and Technology. Auditing Use Case.

[9] Tribunal Superior Eleitoral. Assinatura digital e lacração dos sistemas.

[10] Tribunal Superior Eleitoral. Possibilidades de auditoria do voto eletrônico.

[11] Tribunal Superior Eleitoral. Saiba como foi a totalização dos votos no 1º turno das Eleições 2026.

[12] Tribunal Superior Eleitoral. Resultados — 2026: conjunto de dados.

[13] Supremo Tribunal Federal. ADI 5889: STF declara inconstitucional a impressão do registro do voto, 16 de setembro de 2020.