Revista Ibero-Americana de Humanidades, Ciências e Educação — REASE

OPEN ACCESS

doi.org/10.51891/rease.v12i11.33390

FERRAMENTAS GENÉRICAS DE QUALIDADE DE DADOS DIANTE DE UM DEFEITO POINT-IN-TIME: AVALIAÇÃO DO DBT E DO GREAT EXPECTATIONS COM CONJUNTOS GÊMEOS E O VIÉS DE SOBREVIVÊNCIA NO MERCADO ACIONÁRIO BRASILEIRO

GENERIC DATA-QUALITY TOOLS FACING A POINT-IN-TIME DEFECT: AN EVALUATION OF DBT AND GREAT EXPECTATIONS USING TWIN DATASETS AND SURVIVORSHIP BIAS IN THE BRAZILIAN STOCK MARKET

João Vinícius Passos Castello Branco Carvalho[1]
Lia Mariana Nascimento Brito[2]

RESUMO: Um conjunto de dados pode estar livre de erros aparentes e ainda falsear o passado, por lhe faltarem registros que existiam na data que pretendia retratar. Esse defeito, chamado de defeito point-in-time, tem um exemplo conhecido no contexto do viés de sobrevivência. Este trabalho avalia se duas ferramentas de qualidade de dados, o dbt e o Great Expectations, detectam esse defeito em configuração padrão, sob a hipótese de que constatam defeitos sintáticos, mas não essa falha. Propõe-se um método de avaliação com conjuntos gêmeos, aplicado ao mercado acionário brasileiro de 2010 a 2025: o conjunto correto reúne os papéis negociados em cada mês, e o gêmeo contaminado omite os papéis das companhias que deixaram o mercado até o fim de 2025. Na configuração padrão, nenhum teste do dbt detecta o defeito; no Great Expectations, apenas duas combinações de verificação e parâmetros o detectam, sem indicar o que falta, de modo que a hipótese não se confirma integralmente. Quando recebem o universo correto como referência, as verificações nativas de ambas as ferramentas sinalizam o defeito. Conclui-se que a aprovação por meio de verificações genéricas não garante que um conjunto esteja isento de viés de sobrevivência.

Palavras-chave: Qualidade de dados. Correção point-in-time. Viés de sobrevivência.

ABSTRACT: A dataset may be free of apparent errors and still misrepresent the past, because it lacks records that existed on the date it claims to describe. This defect, called a point-in-time defect, has a known example in the context of survivorship bias. This study evaluates whether two data-quality tools, dbt and Great Expectations, detect this defect in their default configuration, under the hypothesis that they detect syntactic defects but not this one. A twin-dataset evaluation method is proposed and applied to the Brazilian stock market from 2010 to 2025: the correct dataset comprises the securities traded in each month, and the contaminated twin omits the securities of the companies that left the market by the end of 2025. In the default configuration, no dbt test detects the defect; in Great Expectations, only two combinations of check and parameters detect it, without indicating what is missing, so the hypothesis is not fully confirmed. When given the correct universe as a reference, the native checks of both tools detect the defect. It is concluded that passing generic checks does not guarantee that a dataset is free of survivorship bias.

Keywords: Data quality. Point-in-time correctness. Survivorship bias.

[1] Graduando em Engenharia de Software. Instituto de Ensino Superior iCEV.
[2] Orientadora. Coordenadora do curso de Engenharia de Software. Instituto de Ensino Superior iCEV. Bacharel em Engenharia de Software (iCEV/PI), Mestranda em Gestão (UFPI/PI).

1 INTRODUÇÃO

1.1 Contexto e problema

Pipelines[3] de dados são hoje infraestrutura de que dependem decisões de negócio e modelos de aprendizado de máquina (PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026, p. 1:1; REDYUK et al., 2021, p. 61; SCHELTER et al., 2018, p. 1781) e, em setores regulados, relatórios exigidos por norma (DAMA INTERNATIONAL, 2017, p. 30, 87-88). Entre a origem e o uso, o dado atravessa cadeias de transformação de muitos estágios, e cada estágio pode degradá-lo (DAMA INTERNATIONAL, 2017, p. 462).

Parte da resposta da engenharia de software a essa dependência foi levar ao dado a disciplina de teste que já se aplicava ao código. O trabalho seminal de Schelter et al. (2018) formulou a ideia como *unit tests for data*, por analogia declarada com as práticas de teste da engenharia de software (SCHELTER et al., 2018, p. 1782), e o sistema descrito no artigo é identificado como Deequ por Papastergios, Ehrlinger e Gounaris (2026, p. 1:5). O mesmo princípio — declarar expectativas sobre o dado e verificá-las contra ele — organiza uma camada de ferramentas de teste e validação, entre as quais se dispõem dbt[4], Great Expectations[5], Deequ e Soda Core (PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026, p. 1:5).

Essas ferramentas operacionalizam bem uma parte do vocabulário clássico de qualidade de dados, cujas taxonomias de referência organizam a qualidade em dimensões (BATINI; SCANNAPIECO, 2016; DAMA INTERNATIONAL, 2017; WANG; STRONG, 1996). Uma restrição de não nulidade cobre a completude de atributo obrigatório, uma restrição de unicidade cobre duplicidade, e uma verificação de domínio cobre validade; as quatro ferramentas trazem as três verificações prontas (PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026, p. 1:35-1:36). A verificação, uma vez declarada, passa a rodar de forma automatizada como parte do *pipeline* de ingestão (SCHELTER et al., 2018, p. 1792).

Há, porém, uma classe de defeitos que atravessa esse catálogo sem ser tocada. Um conjunto de dados pode ser sintaticamente impecável, isto é, sem valores nulos, sem duplicatas, com esquema íntegro e domínios respeitados, e ainda assim descrever o passado de maneira

[3] Sequências automatizadas de etapas que coletam, transformam e entregam dados para uso, como relatórios e modelos.
[4] Ferramenta de transformação de dados; os testes que nela se declaram sobre os dados produzidos estão descritos na subseção 2.2.
[5] Biblioteca de validação de dados em que cada verificação se chama expectation (subseção 2.2).

falsa, porque foi montado com informação que não existia na data que ele afirma retratar. Um exemplo basta. Um analista que hoje monte a lista das ações negociáveis em 2015 a partir da relação atual de empresas listadas na B3 S.A. – Brasil, Bolsa, Balcão (B3) obtém um conjunto sem nulos, sem duplicatas e com esquema íntegro. Nele, porém, faltam as empresas cujas ações deixaram a bolsa desde então, por fechamento de capital, incorporação ou cancelamento de registro, e nenhuma verificação sobre as linhas presentes tem como notar.

É o problema da correção *point-in-time*[6]: o conjunto é correto quando descreve cada data passada apenas com o que era verdadeiro e conhecido naquela data. Das duas condições, este trabalho reconstrói e verifica a primeira, o que era verdadeiro em cada data; a segunda, o que se sabia em cada data, a fonte usada não permite verificar (subseção 3.2). O caso do exemplo, em que o que vaza é a própria composição do universo, tem nome próprio, *survivorship bias*, o viés de sobrevivência (ELTON; GRUBER; BLAKE, 1996, p. 1098; LÓPEZ DE PRADO, 2018, cap. 11).

As verificações típicas dessas ferramentas examinam cada linha presente no conjunto, e são desse tipo os quatro testes que o dbt traz de fábrica (DBT LABS, 2026b): esta coluna não é nula, esta chave não se repete, este valor pertence ao conjunto aceito, esta chave existe na tabela referenciada. O defeito *point-in-time* é diferente: é um defeito de ausência, em que faltam linhas que deveriam estar no conjunto. Como nenhuma linha presente está errada, uma verificação que só examina as linhas presentes não tem como acusá-lo, por melhor que esteja configurada: ela é estruturalmente cega. Essa limitação decorre da forma das verificações, e não dos dados, e por isso é tomada aqui como ponto de partida.

1.2 Objetivos, hipótese e questões de pesquisa

Uma delimitação governa todo o desenho deste trabalho: o objeto de estudo são as ferramentas, não o viés de sobrevivência. Não se pretende demonstrar que o viés existe, nem estimar seu efeito sobre retornos, o que é conhecido há mais de três décadas (BROWN et al., 1992; BROWN; GOETZMANN; ROSS, 1995; CARHART, 1997; ELTON; GRUBER; BLAKE, 1996). O viés de sobrevivência entra aqui como sonda: um defeito escolhido por ser *point-in-time* por construção, por possuir uma referência oficial, pública e gratuita, que se toma como verdade

[6] Do inglês point in time, “ponto no tempo”.

para julgar cada detecção, e por ser injetável de maneira controlada e reversível em um conjunto de dados real.

Já existem trabalhos que plantam vazamento temporal de propósito e medem o seu efeito ou a sua detecção (subseção 2.4). A pergunta que este trabalho faz é distinta e deliberadamente mais modesta: os testes que o dbt traz de fábrica e o catálogo nativo do Great Expectations, em configuração padrão, enxergam esse defeito? Não se trata de descobrir um fenômeno, mas de auditar um instrumento: medir a cobertura de ferramentas de prateleira diante de uma classe de defeito cujos registros ausentes elas não apontam sozinhas. O próprio Fonseca (2026, p. 1) fornece a formulação do problema ao observar que o estado da arte é correto apenas canal a canal, e que "um detector relata os vazamentos que por acaso aciona e nada certifica com o seu silêncio" (FONSECA, 2026, p. 2, tradução nossa)[7].

A hipótese central é a seguinte: ferramentas genéricas de qualidade de dados, em configuração padrão, detectam defeitos sintáticos, como nulos, duplicatas e violações de esquema, mas não detectam um conjunto *point-in-time* contaminado por viés de sobrevivência, e é essa a lacuna; apontar as entidades que faltam ao conjunto exige uma referência temporal externa, que o paradigma genérico não fornece.

Parte da hipótese decorre da própria forma das verificações, e a contribuição empírica está no que só o experimento mostra.

Da hipótese decorrem três questões de pesquisa (QP), numeradas de QP1 a QP3:

QP1. As verificações genéricas, em configuração padrão, distinguem um conjunto contaminado de seu gêmeo correto?

QP2. Qual o custo, em linhas de código, de comparar o conjunto com a referência *point-in-time* dada, por verificações sob medida?

QP3. Qual a diferença entre a capacidade das ferramentas, em configuração padrão, de detectar defeitos sintáticos e defeitos *point-in-time* (a lacuna de capacidade, ou *capability gap*), e em quais verificações esse resultado já é determinado pela própria lógica da verificação?

O objetivo geral é projetar e avaliar um método de injeção de defeitos *point-in-time* e de verificações sob medida que meça e caracterize esse ponto cego, instanciado com dados oficiais do mercado acionário brasileiro. Os objetivos específicos são seis. Os três primeiros são: propor

[7] No original: "a detector reports the leaks it happens to trigger and certifies nothing by its silence".

um método de avaliação que não dependa de conceitos de um mercado específico; reconstruir, a partir de fontes oficiais, o universo de papéis negociados em cada mês e a relação das companhias que saíram; e gerar o conjunto correto e o seu gêmeo contaminado. Os outros três são: implementar as verificações nos três níveis de configuração; medir a detecção, os falsos positivos e o custo de configuração; e caracterizar a lacuna e discutir a sua generalização a outras ferramentas.

1.3 Relevância, método e contribuições

A relevância do tema está no custo da má qualidade de dados, que Redman (1996, p. 9) estima, de forma que ele mesmo chama de conservadora, em cerca de dez por cento da receita de empresas de médio e grande porte sem programa ativo de qualidade de dados, e no uso amplo das ferramentas avaliadas (PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026, p. 1:2, 1:5). Se a aprovação por essas ferramentas for lida como garantia de correção, um conjunto com viés de sobrevivência pode sustentar decisões e modelos sem que ninguém perceba o defeito.

Trata-se de pesquisa aplicada, de abordagem quantitativa, que combina pesquisa bibliográfica, pesquisa documental em bases públicas da B3 e da Comissão de Valores Mobiliários (CVM) e um experimento controlado, conduzido segundo o *Design Science Research* (seção 3).

A contribuição é dupla. Metodologicamente, propõe-se uma estratégia de avaliação, desenhada sem conceito próprio de nenhum mercado e instanciada aqui uma vez: um protocolo de conjuntos gêmeos em que o defeito *point-in-time* é injetado por remoção controlada, acompanhado de um protocolo de métricas, que lê o resultado de cada verificação ao lado do veredito dela sobre o conjunto correto.

Empiricamente, entrega-se uma instanciação brasileira, construída apenas sobre fontes oficiais de acesso público e gratuito. São as séries históricas e o banco de códigos internacionais de valores mobiliários (*International Securities Identification Number*, ISIN) da B3 (B3 S.A. – BRASIL, BOLSA, BALCÃO [B3], 2026a, 2026b) e os dados abertos da CVM (COMISSÃO DE VALORES MOBILIÁRIOS [CVM], 2026a, 2026b). Essa característica a distingue de Fonseca (2026), o trabalho mais próximo no eixo da detecção, cujos números principais vêm de dados de mercado proprietários, que o autor não tem permissão para redistribuir (FONSECA, 2026, p. 1, 21).

2 REFERENCIAL TEÓRICO

2.1 Qualidade de dados: dimensões e completude

A partir de levantamento com consumidores de dados, o arcabouço de Wang e Strong (1996) organiza qualidade de dados em quatro categorias: intrínseca, contextual, representacional e de acessibilidade. Pipino, Lee e Wang (2002) dão o passo seguinte, operacionalizando dimensões em métricas calculáveis por três formas funcionais recorrentes: razão simples, operação de mínimo ou máximo sobre indicadores normalizados, e média ponderada.

Os tratados de Batini e Scannapieco (2006) e Batini e Scannapieco (2016) consolidam metodologias de avaliação e melhoria, e o DAMA International (2017, p. 44) se propõe a estabelecer o vocabulário comum da gestão de dados. As seis dimensões nucleares da seção britânica da *Data Management Association* (DAMA), o DAMA UK, reproduzidas nesse compêndio, o *Data Management Body of Knowledge* (DMBOK), incluem acurácia, completude, consistência, unicidade e validade (DAMA INTERNATIONAL, 2017, p. 457).

O que interessa a este trabalho é a forma dessas taxonomias. As métricas propostas são, em sua maioria, funções do conteúdo presente no conjunto: proporção de células preenchidas, proporção de valores dentro do domínio, proporção de chaves não repetidas. Há exceções: a métrica de *timeliness* que Pipino, Lee e Wang (2002, p. 214-215) reproduzem de trabalho anterior consome metadados de processo que não estão nas linhas do conjunto, e a *population completeness* dos mesmos autores compara uma coluna contra uma população de referência, detectando justamente os valores que deveriam ocorrer nela e faltam (PIPINO; LEE; WANG, 2002, p. 213).

O objeto avaliado, porém, é sempre o estado presente do conjunto. Quando o passado entra, o faz como linha de base para detectar variação (DAMA INTERNATIONAL, 2017, p. 458-459), e a referência externa é um domínio fixo ou um cadastro tomado no estado corrente, nunca o universo de entidades vigente em uma data passada.

Redman (1996, p. 256-257) separa duas completudes: a de atributo, violada quando falta o valor de um atributo obrigatório num registro presente, e a de entidade, que exige que estejam no conjunto todas e somente as entidades devidas. A de atributo se verifica no próprio registro; a de entidade depende de um padrão externo ao conjunto, que o autor reconhece difícil de obter.

O defeito de ausência tratado neste trabalho é uma falha de completude de entidade, em que o padrão é o universo de entidades vigente em cada data passada. No modelo relacional sem

valores nulos, sob a hipótese de mundo aberto, Batini e Scannapieco (2006, p. 24-25) caracterizam a completude de uma relação pela relação de referência, a que contém todas as tuplas que deveriam estar nela, e a medem pela razão entre o tamanho da relação e o da referência. Observam que, na prática, a relação de referência raramente está disponível. Que a ausência só se mede contra uma referência é, portanto, conhecido, e este trabalho não o apresenta como descoberta. O que ele acrescenta é a execução: o que duas ferramentas de prateleira fazem diante de uma falha de completude cuja referência é o universo de uma data passada.

A dimensão que soa temporal nesse vocabulário, a *timeliness*, não é o construto que falta. Wang e Strong (1996, p. 32) a definem pela idade do dado, e o verbete do DMBOK reúne nela volatilidade, atualidade e latência, sendo esta última o intervalo entre a criação do dado e sua disponibilização para uso (DAMA INTERNATIONAL, 2017, p. 459). Ela mede se o dado está atualizado e se chegou a tempo, não se ele descreve corretamente uma data passada. A evidência empírica vai no mesmo sentido.

Papastergios, Ehrlinger e Gounaris (2026) revisaram documentação e código-fonte de dbt Core, Deequ, Evidently, Great Expectations, Griffin, MobyDQ e Soda Core. Nessas ferramentas, classificam uma única funcionalidade como avaliação direta da dimensão *currentness* da norma *International Organization for Standardization/International Electrotechnical Commission* (ISO/IEC) 25012: a verificação de recência de *timestamps*, que registram como verificação pronta em apenas três das sete ferramentas, e não no Great Expectations (PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026, p. 1:4, 1:28). Todas as suas variantes se resolvem contra o relógio do sistema ou contra *timestamps* de referência. A leitura que se segue é deste trabalho, não dos autores: nenhuma dessas variantes toma por referência uma data de corte passada.

2.2 Ferramentas de teste de dados

Os dois instrumentos avaliados neste trabalho compartilham uma arquitetura: um catálogo declarativo de verificações prontas, mais um mecanismo de extensão para o que o catálogo não cobre. No dbt, a distinção é explícita na própria nomenclatura (DBT LABS, 2026b). *Generic tests* são verificações parametrizáveis, declaradas em arquivos *YAML Ain't Markup Language* (YAML) e aplicadas a colunas ou modelos, como not_null, unique, accepted_values e relationships. *Singular tests*, em contraste, são consultas em *Structured Query*

Language (SQL) escritas sob medida (DBT LABS, 2026b), em que se exprime qualquer propriedade que se consiga formular em SQL, inclusive lógica temporal, ao preço de escrevê-la por inteiro.

No Great Expectations, a estrutura se repete. Há um catálogo extenso de *expectations* nativas, que se podem filtrar pelo problema de qualidade de dados que cada uma trata (GREAT EXPECTATIONS, c2026a). Para a semântica que o catálogo não cobre, a versão 1.19.1 da ferramenta, inspecionada neste trabalho, oferece duas saídas: uma *Custom Expectation* cuja lógica é uma consulta SQL (GREAT EXPECTATIONS, c2026c) e a autoria de uma classe de *expectation* em Python. A seção de customização da documentação corrente cobre a adaptação de uma *expectation* existente por herança (GREAT EXPECTATIONS, c2026b) e a rota por consulta, mas não descreve a autoria de lógica nova em Python, embora o pacote a permita.

O sistema de Schelter et al. (2018), o Deequ, completa o quadro com restrições estruturais e estatísticas. Sua única restrição sob a dimensão temporal é a detecção de anomalia sobre o histórico de métricas (SCHELTER et al., 2018, p. 1783), voltada a desvio de métrica ao longo do tempo, e não à correção de um estado passado. A divisão entre catálogo e extensão não nasceu com essas ferramentas. Na engenharia clássica de *data warehouse*, cada verificação da limpeza é, em Kimball e Caserta (2004, cap. 4), uma *data quality screen*: de um lado ficam as verificações genéricas, de coluna e de estrutura, que o livro lista; do outro, as regras de dados e de valor, que, segundo os autores, dependem do assunto e não se deixam listar.

Nenhuma das *screens* descritas traz a referência temporal externa: as estatísticas comparam a carga com a série das cargas anteriores, e não com o que o conjunto deveria conter numa data passada (KIMBALL; CASERTA, 2004, cap. 4).

A alegação de classe apoia-se num argumento analítico sobre a forma das verificações, e não na contagem de ferramentas. A verificação que julga apenas as linhas presentes não tem como apontar a linha ausente, em qualquer ferramenta que a execute, e o defeito *point-in-time* só é apontado contra uma referência temporal externa, que o paradigma declarativo genérico não fornece. Esse paradigma não produz o que o conjunto deveria conter em cada data.

Ter o Great Expectations no desenho, ao lado do dbt, não prova a alegação. Mostra, em duas implementações, até onde o argumento vale e onde ele para: nas verificações que julgam o conjunto como um todo. As duas ferramentas convergem na rota documentada para semântica que o catálogo não cobre, uma consulta SQL escrita à mão, e diferem no invólucro: no dbt, é

um arquivo autônomo; no Great Expectations, é o código Python que cria o objeto com a consulta e o executa.

2.3 Correção point-in-time, defeito de ausência e viés de sobrevivência

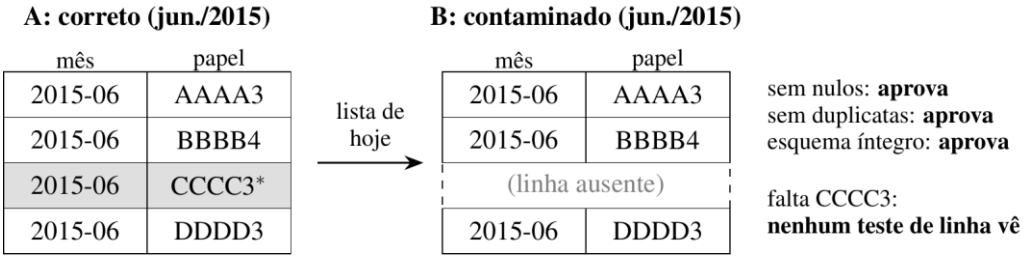
A correção *point-in-time* tem formulação rigorosa na literatura de bancos de dados temporais, de que as taxonomias da subseção 2.1 não se servem. Jensen e Snodgrass (1996, p. 313, 319-322) formalizam, no modelo conceitual bitemporal que desenvolveram, uma distinção já conhecida: tempo de validade (*valid time*), quando um fato é verdadeiro na realidade modelada, e tempo de transação (*transaction time*), quando ele consta da base. Um banco bitemporal registra os dois tempos em cada fato, e esse registro duplo lhe permite responder a duas perguntas distintas: "o que a base afirmava, em 2019, sobre o estado do mundo em 2019?" e "o que a base afirma hoje sobre 2019?".

O defeito estudado aqui nasce quando a fonte não guarda história nenhuma (JENSEN; SNODGRASS, 1999, p. 40; SNODGRASS, 2000, p. 114), ou expurga as entidades que deixaram de existir (SNODGRASS, 2000, p. 269-270): sobra apenas a lista de hoje, e ela passa a responder pelo passado em silêncio.

O problema tem outros nomes fora dessa literatura. Croushore e Stark (2001, p. 111) documentam o problema de *vintage*: séries macroeconômicas são revisadas, e avaliar uma previsão passada com a série revisada atribui a quem a fez informação que ele não possuía. As finanças quantitativas o chamam *look-ahead bias*, o viés de antecipação: o uso de informação que ainda não estava disponível na data da decisão (DUARTE et al., 2026; LÓPEZ DE PRADO, 2018).

A tradução do problema para o cotidiano de um *pipeline* analítico é direta e é o núcleo conceitual deste trabalho. Considere-se um conjunto que pretende descrever, para cada data t de um histórico, quais entidades eram elegíveis em t. Se esse conjunto for construído a partir da lista de entidades hoje existentes, ele omitirá silenciosamente toda entidade que existia em t e desapareceu antes de hoje. O conjunto resultante tem a seguinte propriedade desconfortável: todo registro nele contido é verdadeiro; o defeito está inteiramente no que não está lá. A Figura 1 ilustra o defeito.

Figura 1 – O defeito de ausência: o conjunto contaminado passa nos testes que só leem as linhas presentes



Fonte: Elaborado pelos autores (2026), com códigos fictícios.

Nota: * Papel de companhia que deixou a bolsa depois de 2015 e, por isso, não está na lista de hoje.

Daí decorre a assimetria que o experimento mede: todos os registros presentes são válidos, e o defeito está apenas nos ausentes. Apontar o que falta exige uma referência temporal externa, que o paradigma declarativo genérico não fornece.

A omissão, além disso, não precisa nascer de uma lista tirada do presente. Basta um filtro de limpeza que decida a inclusão de um registro em *t* com informação posterior a *t*. Duarte et al. (2026) mostram, na literatura empírica de opções, que filtros de seleção amostral adotados para mitigar erros de medida produzem viés de *look-ahead* exatamente quando se apoiam em informação indisponível no momento de formação da carteira. O procedimento de qualidade, nesse caso, é a origem do defeito de ausência, e não o seu detector.

O caso em que o que vaza é a própria composição do universo aparece entre as formas de *look-ahead* (FONSECA, 2026, p. 1) e tem nome próprio: o viés de sobrevivência (*survivorship bias*). Diz-se que um conjunto de dados tem viés de sobrevivência quando retrata o passado de uma população apenas pelos membros que nela permaneceram até uma data posterior, como a de hoje (BROWN; GOETZMANN; ROSS, 1995, p. 853-854, 856, 871; ELTON; GRUBER; BLAKE, 1996, p. 1098-1099).

Esse condicionamento à sobrevivência produz um erro sistemático nas estimativas feitas com o conjunto sempre que a permanência depende do que se mede, como o desempenho (BROWN et al., 1992, p. 559-560, 576). Brown et al. (1992) mostraram que dele emerge aparência de previsibilidade de desempenho onde não há; Brown, Goetzmann e Ross (1995, p. 853-856) estenderam o argumento a mercados inteiros; Elton, Gruber e Blake (1996) estimaram o viés em fundos mútuos norte-americanos; e Carhart (1997, p. 58-59) o controla incluindo na base todos os fundos de ações diversificados conhecidos, inclusive os que deixaram de existir.

O mecanismo é simples de enunciar e fácil de reproduzir por acidente. Para avaliar uma estratégia entre 2010 e 2025, um analista precisa saber quais ativos eram negociáveis em cada data do período. Se ele parte da lista de ativos negociáveis hoje e recupera o histórico de preços de cada um, obtém uma base que exclui, por construção, toda empresa cujas ações deixaram a bolsa desde 2010. Em cada data, se as ações que depois saíram rendiam menos que as outras, o retorno médio da base fica acima do retorno médio do universo naquela data (DUARTE et al., 2026; ELTON; GRUBER; BLAKE, 1996, p. 1119).

Assim definido, o viés de sobrevivência é uma violação *point-in-time*: o conjunto contaminado compõe o universo de cada data passada com informação que nela não existia, a de quem permaneceria até a data posterior. A saída de uma entidade do universo negociável, porém, tem causas heterogêneas, e nem toda ausência de um código de negociação corresponde à morte de uma empresa. Dos treze fundos que haviam sumido do diretório usado como fonte por Elton, Gruber e Blake (1996, p. 1102-1103), doze tinham deixado de existir, e um tinha apenas mudado de nome.

Ranse (2026), num *preprint* que reconstrói por aproximação a composição histórica de um índice de menor capitalização em mercado emergente, estima que, com carteiras que dão o mesmo peso a cada papel, o *backtest*[8] restrito aos sobreviventes superestima o retorno anualizado em 4,94 pontos percentuais. Segundo o autor, apenas parte dessa diferença vem de fracassos propriamente ditos, porque a maioria dos papéis que deixaram o índice continua negociando. No próprio mercado estudado, há precedente em Barros (2026, p. 2-3): um estudo revisado por pares, do período de 2002 a 2024, sobre 751 emissores, tirados de um universo de 1.097 códigos de ações listadas e deslistadas das séries históricas da B3 (COTAHIST), a mesma fonte de cotações deste trabalho.

2.4 Trabalhos relacionados e lacuna

A vizinhança deste trabalho se organiza melhor por método do que por tema, e divide-se em quatro famílias.

A primeira família é o mapeamento de capacidade sem execução (ALEXAKIS et al., 2025; PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026; REHBERGER et al., 2027; ZHOU et al., 2024). A avaliação de Rehberger et al. (2027, p. 322), por exemplo, apoia-se

[8] Simulação histórica de como uma estratégia teria se saído num período passado (LÓPEZ DE PRADO, 2018, cap. 11).

exclusivamente em informação publicamente disponível, como sítios de fornecedores, documentação e repositórios, e Alexakis et al. (2025, p. 21) declaram entre suas limitações não incluir validação empírica nem *benchmark* de ferramentas em ambiente operacional. Por construção, essa família reporta capacidade e jamais mede detecção.

A segunda é a medição por injeção de defeitos e contagem de detecções. Martins et al. (2025) executam cinco ferramentas de limpeza, incluindo Great Expectations, sobre classes de defeito inteiramente estáticas, e nos *preprints* de Chen, Phani e Schelter (2026) e de Glock et al. (2026) tampouco há tipo temporal. Schinninger et al. (2025) medem com Great Expectations erros temporais injetados em fluxo, todos capturados por *expectations* do catálogo nativo (SCHINNINGER et al., 2025, p. 799-800). A leitura que se segue é deste trabalho: cada um desses erros deixa marca no conjunto verificado, e o registro ausente não deixa. Redyuk et al. (2021) e Shankar et al. (2023) propõem detectores novos e tratam o tempo como eixo de comparação distribucional entre lotes datados, não como validade numa data passada.

A terceira é a comparação pareada com uma referência limpa. Zhang et al. (2026) mantêm tudo constante e alteram exatamente uma convenção de avaliação por rodada, isolando o efeito de cada uma de cinco formas de vazamento sobre o desempenho preditivo e de carteira. A lógica é a mesma adotada aqui, com outra variável manipulada; nesse trabalho e no de Benhenda (2026), ambos *preprints*, a variável dependente é o desempenho do modelo ou da carteira, e não a detecção por uma ferramenta de qualidade de dados.

A quarta é a garantia formal. Fonseca (2026), em *preprint*, identifica a ausência de *look-ahead* com uma propriedade formal e entrega um verificador que examina o *pipeline* sem executá-lo. É o único trabalho desta subseção que mede a detecção de vazamento temporal plantado de propósito, e o faz em *pipelines*, não em conjuntos de dados. Entre os cenários com que valida o verificador está o de um universo filtrado por sobrevivência (FONSECA, 2026, p. 23-24, 28), mas em nenhum ponto do artigo aparece ferramenta genérica de qualidade de dados. A garantia formal responde "é possível certificar a ausência de vazamento?", enquanto este trabalho responde "o que o ferramental já instalado deixa passar hoje?".

Pelo método, pertence à segunda família o *preprint* de Gargouri e Reza (2026), o único trabalho desta subseção que executa o dbt: um arcabouço de teste multicamada, avaliado por injeção controlada de anomalias. As anomalias, porém, são genéricas, e a fonte não reporta o veredito dos testes sobre os dados íntegros nem taxa de falsos positivos. Os próprios autores

registram, entre as limitações, que famílias de falha mais ricas, incluindo o que chamam de *drift* temporal, permanecem inexploradas (GARGOURI; REZA, 2026, p. 10).

Fora dessas famílias, Breck et al. (2019) nomeiam um defeito *point-in-time*, o *time travel*, em que o valor de uma variável de entrada absorve eventos ocorridos entre o instante em que o dado foi servido e aquele em que o dado de treino foi gerado (BRECK et al., 2019, p. 6). O termo ocorre uma única vez no artigo, e o defeito não ganha restrição própria no esquema nem é quantificado em nenhum dos estudos de caso (BRECK et al., 2019, p. 4-6, 9-10, 13).

Nenhum dos trabalhos acima satisfaz a conjunção de quatro condições: medir detecção, por ferramentas genéricas de prateleira, de defeito *point-in-time* por viés de sobrevivência, sob desenho gêmeo com referência oficial e reproduzível. Das quatro, a que carrega o interesse é a execução de ferramentas de prateleira contra um defeito de ausência, com o veredito sobre o conjunto correto como controle negativo. É importante enunciá-la com precisão: não se alega primazia em medir detecção de defeito *point-in-time*, medição que Fonseca (2026) já faz. Este trabalho audita o que duas ferramentas genéricas de qualidade de dados, de prateleira e em configuração padrão, enxergam de um conjunto *point-in-time* contaminado por viés de sobrevivência.

3 METODOLOGIA

3.1 Caracterização da pesquisa

Pela classificação de Prodanov e Freitas (2013, p. 51-57, 69), a pesquisa é aplicada quanto à natureza, pois busca resposta para um problema prático: saber se ferramentas de uso corrente detectam um tipo específico de defeito. Quanto à abordagem, é quantitativa: os resultados são contagens de detecções, de falsos positivos e de linhas de código. Quanto aos objetivos, é explicativa: isola, por experimento controlado, a única diferença entre os conjuntos comparados e observa o efeito dela sobre o veredito de cada verificação. Quanto aos procedimentos, combina pesquisa bibliográfica (seção 2), pesquisa documental em bases públicas da B3 e da CVM e pesquisa experimental.

O experimento serve de avaliação a um artefato, e o método que legitima o artefato como resultado de pesquisa é o *Design Science Research*. Hevner et al. (2004) estabeleceram as sete diretrizes que distinguem pesquisa de design de mero desenvolvimento, entre as quais a exigência de avaliação rigorosa e de contribuição clara e verificável, e Peffers et al. (2007) formalizaram o processo em seis atividades: identificação do problema e motivação, definição

dos objetivos para uma solução, projeto e desenvolvimento, demonstração, avaliação e comunicação.

Na classificação de Gregor e Hevner (2013), este trabalho é uma *exaptation*, a extensão de uma solução conhecida num campo a um problema novo noutro (GREGOR; HEVNER, 2013, p. 345, 347). A solução conhecida é a distinção entre tempo de validade e tempo de transação, dos bancos de dados temporais (subseção 2.3); o problema novo é avaliar o que o ferramental genérico de qualidade de dados enxerga de um defeito *point-in-time*. O artefato compreende um método abstrato de avaliação e a sua instanciação no mercado acionário brasileiro.

3.2 Fontes de dados, universo e amostra

Quatro fontes oficiais, de acesso público e gratuito, sustentam a instanciação: da B3, as séries históricas de cotações e o banco de ISIN; da CVM, o Cadastro de Companhias Abertas e o Formulário Cadastral. As duas da CVM são publicadas sob a *Open Database License* (ODbL) (CVM, 2026a, 2026b). As séries históricas de cotações da B3, o COTAHIST, registram os papéis com negócio em cada pregão e incluem, por construção, os que depois desapareceram; os arquivos cobrem papéis listados e deslistados (BARROS, 2026, p. 9). Nelas, nenhum campo do registro tem granularidade inferior ao dia, e nenhum informa o instante em que o dado passou a estar disponível (B3, 2020). Numa fonte que registra somente tempo de validade, o tempo de transação é irrecuperável (subseção 2.3). Por isso este trabalho reconstrói e verifica o que era verdadeiro em cada data, e não o que se sabia em cada uma delas.

O universo é definido pela negociação, e não pela composição histórica de índice, para que se reconstrua a partir dessas fontes. A janela vai de janeiro de 2010 a dezembro de 2025, em anos-calendário completos, com data de corte em 31 de dezembro de 2025. O painel tem 192 retratos mensais (*snapshots*), e o de cada mês reúne os papéis com ao menos um negócio em algum pregão do mês: são 901 papéis e 85.033 papel-meses, isto é, pares de papel e mês. Não há amostragem: o experimento usa todos os papéis do universo. Por esse critério, 279 papéis ficam fora do universo em meses sem negócio entre o primeiro e o último negócio de cada um, 9.652 meses ao todo; esses meses ficam fora de A e de B e não entram na diferença entre os dois.

A definição do universo é, ela própria, uma decisão *point-in-time*, e chegar a ela exigiu três rodadas de medição, cada uma corrigindo um erro na mesma direção. A primeira versão do artefato filtrava os papéis pelo código com que a B3 classifica cada papel no boletim diário (B3, 2020), aqui chamado *livro*, e empresas em recuperação judicial migram de livro sem sair do

mercado. Testados os 137 rótulos da versão inicial contra o exercício de 2025, verificaram-se oito falsos positivos, papéis classificados como desaparecidos que negociaram sob outro livro naquele ano, e onze falsos negativos, papéis que negociaram exclusivamente fora do livro filtrado e nunca chegaram a ser candidatos.

Os dois erros somam 19 códigos, número equivalente a 13,9% dos 137 rótulos. A correção evidente, ampliar a lista de livros, também falhou: um dos livros envolvidos é a categoria residual do arquivo, rotulada "OUTROS", e outro não consta da tabela publicada, cuja revisão vigente é de 2020, embora apareça nos dados a partir de 2022 (B3, 2020). Nenhuma enumeração sobrevive às duas coisas.

A regra adotada corta pelo tipo de papel, que os caracteres 7 a 9 do ISIN identificam (B3, 2026a), admite o mercado à vista e o fracionário, e vem com uma verificação que interrompe a construção se surgir valor desconhecido. Ficam fora, por decisão declarada, os leilões, dois papéis de um segmento cuja especificação não consta no layout e duas *units*, a PPLA11 e a BBTG12. Segundo a descrição da B3 (B3, 2026a), elas reúnem só *Brazilian Depositary Receipts* (BDRs), certificados que representam valores mobiliários emitidos no exterior, e não correspondem a companhia aberta brasileira.

O episódio tem valor além da correção. Um filtro de universo aparentemente inócuo removeu em silêncio empresas em dificuldade e injetou um defeito de ausência no próprio artefato construído para medir a detecção de outro. Nenhuma verificação sintática sobre o arquivo resultante o acusaria, pois ele tinha 137 linhas, chaves completas, nenhuma duplicata e esquema íntegro.

Para responder à objeção de que o filtro adotado, mais amplo, teria sido escolhido para inflar o efeito medido, o experimento é repetido sobre o universo restrito ao livro 02, que o layout do COTAHIST descreve como lote padrão (B3, 2020, p. 7), e reportado como análise de sensibilidade. Esse universo reintroduz de propósito as omissões narradas acima.

A ligação entre cada papel e a companhia é feita por um *crosswalk*[9] determinístico, com duas pontes: a do Formulário Cadastral, pelo código de negociação, que o formulário traz a partir do arquivo de 2018, e a do banco de ISIN, pelo ISIN que o COTAHIST traz para o papel. Nenhuma etapa emprega casamento por similaridade de nome, uma vez que razões sociais parecidas produzem ligações falsas entre companhias distintas. Com as duas pontes, 42 dos 901

[9] Tabela de correspondência entre identificadores de fontes diferentes; aqui, entre o código de negociação do papel e o Cadastro Nacional da Pessoa Jurídica (CNPJ) da companhia.

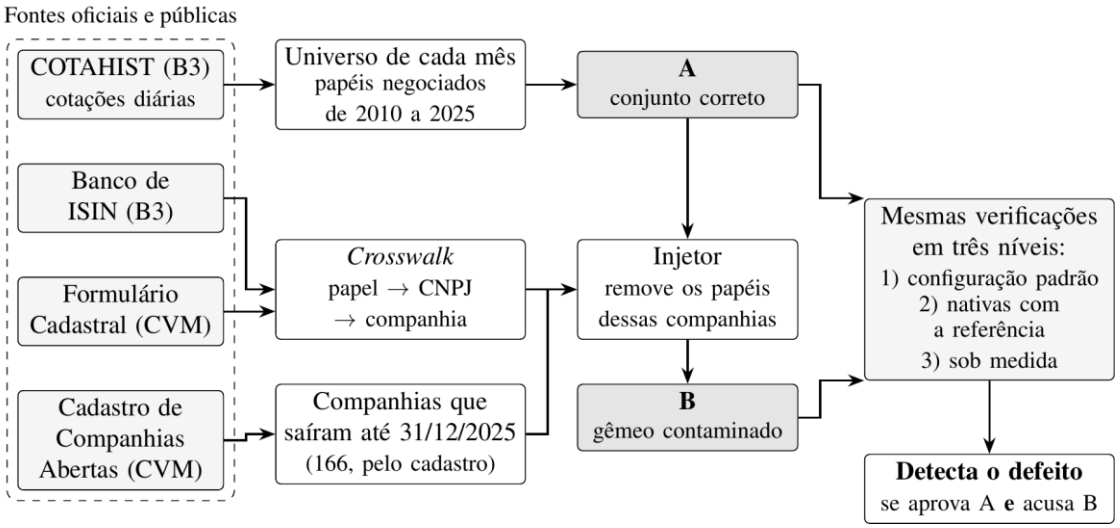
papéis do painel ficam sem ligação; eles permanecem em A e em B, fora do defeito injetado e da régua.

O *crosswalk* sustenta uma distinção que o dado bruto não oferece: o código de negociação desaparecer não equivale à saída da companhia. Parte dos códigos que somem pertence a companhias que apenas trocaram de sigla, mudaram de segmento ou passaram por reorganização societária, mas sem deixar de existir, e só o cadastro da CVM separa a saída real da mera renomeação.

3.3 Desenho experimental: conjuntos gêmeos, injeção e controles

O desenho central é um par de conjuntos que diferem em exatamente uma variável (Figura 2). O conjunto A é o universo *point-in-time* correto: para cada mês do painel, os papéis negociados naquele mês, incluídos por construção os que depois desapareceram. O conjunto B é o seu gêmeo contaminado, obtido de A pela remoção dos papéis das companhias que, segundo o cadastro da CVM, não sobreviveram até a data de corte. Todo o resto é idêntico: esquema, tipos, período, granularidade e procedência.

Figura 2 – Desenho experimental dos conjuntos gêmeos



Fonte: Elaborado pelos autores (2026).

O isolamento de uma única variável é o que dá força causal ao resultado. Se uma verificação aprova A e aprova B, não houve detecção, porque o defeito é a única coisa que os distingue. Se aprova A e acusa B, houve detecção. Se acusa A, não serve de detector, quer acuse, quer aprove B. A mesma reconstrução desempenha três papéis: é a verdade (A), a receita do

defeito (a relação das companhias que saíram) e a régua contra a qual se pontua o que cada verificação detectou.

O conjunto B opera em nível de companhia. Retém-se de A o papel cuja companhia, na data de corte, tinha registro não cancelado no Cadastro de Companhias Abertas e não estava, segundo esse cadastro, falida nem em liquidação; o papel sem ligação também fica. Remove-se, portanto, só a saída de companhia que o cadastro registra até a data de corte, 166 companhias ao todo, e nunca a mera renomeação. O injetor apenas remove: não sintetiza valores, não altera o esquema e não muda as linhas que ficam, e por isso toda linha de B é linha de A.

Uma segunda variante, B′, opera em nível de código de negociação: retém de A apenas os códigos negociados no último mês do painel. É o procedimento de partir da lista de códigos de hoje e levá-la ao passado. B′ remove saídas reais e também renomeações, além do papel que apenas ficou sem negócio no fim do painel. Por isso, a diferença entre B′ e B, tomada sobre os papéis ligados, quantifica quanto do que a abordagem por código descarta é de companhia sem saída registrada no cadastro até a data de corte.

As verificações se organizam em três níveis de configuração. Em configuração padrão, aplicam-se ao conjunto verificado os quatro *generic tests* que o dbt traz de fábrica (DBT LABS, 2026b) e as *expectations* nativas do Great Expectations: as que julgam cada linha, as que julgam o conjunto e as de esquema, que conferem as colunas da tabela e o tipo de cada uma. Nas verificações que comparam o conjunto com uma lista ou uma tabela, o argumento de comparação é obrigatório, e a ferramenta não traz valor para ele.

O critério, registrado antes da execução, foi dar a cada uma só o que tem quem não reconstruiu o passado: os códigos negociados no mês de corte e no último pregão dele, e a cópia do Cadastro de Companhias Abertas, em três leituras. Cada um descreve o mercado numa data só. Duas dessas escolhas não são independentes da construção do experimento, e isso se declara: a lista do mês de corte coincide com os códigos de A no último mês, que é a regra de B′, e a leitura do cadastro por datas é a regra que define B.

No nível com a referência, três verificações nativas recebem a referência *point-in-time*, que é o universo correto de cada mês e, neste desenho, o próprio A: no dbt, o relationships, com a referência como tabela testada; no Great Expectations, a contenção de códigos e a contagem mensal de linhas.

No nível sob medida, implementam-se um *singular test*, que lista o que está na referência e falta no conjunto, e as duas rotas de extensão do Great Expectations descritas na subseção 2.2.

Ficam fora do desenho os testes que o dbt permite instalar por pacote (DBT LABS, 2026b), os contratos de modelo, que a documentação corrente descreve como garantias sobre a forma do modelo (DBT LABS, 2026c), e a verificação de atualização das fontes (DBT LABS, 2026a). Sem os pacotes, nenhuma verificação do dbt em configuração padrão julga o conjunto como um todo, e a comparação entre as duas ferramentas é entre catálogos de escopos diferentes. Todas as verificações são executadas contra A, B, B′ e as cópias do controle positivo, e também contra os conjuntos do universo restrito ao livro 02.

Dois controles protegem a interpretação do resultado. O controle positivo injeta três tipos de defeito sintático deliberado, os nulos, as duplicatas e o erro de esquema, cada um numa cópia de A. Se as verificações genéricas de uma ferramenta capturam um defeito sintático e não capturam o temporal, o silêncio diante do temporal não pode ser atribuído a verificações que não funcionam. O controle negativo submete A, íntegro, a todas as verificações e mede a taxa de falsos positivos: uma verificação que acusasse também o conjunto correto não estaria detectando o defeito, e sim disparando indiscriminadamente.

O desenho inclui ainda um caso adversarial: março de 2020, mês em que a Organização Mundial da Saúde classificou a COVID-19 como pandemia e em que o preço do petróleo foi particularmente afetado (RELATÓRIO..., 2020, p. 28, 31). Naquele mês, o *circuit breaker*, mecanismo de segurança que interrompe a negociação, foi acionado seis vezes (B3, [2021], p. 32), mas a contagem de papéis negociados por pregão permaneceu entre 341 e 372. A verificação que acusa B precisa permanecer em silêncio diante de A nesse mês: a que dispara nos dois casos não está detectando ausência *point-in-time*, e sim anomalia.

O método não usa conceito próprio de nenhum mercado. De qualquer domínio de aplicação, ele pede três condições. A primeira é existirem registros primários datados dos quais se reconstrua o universo de cada data, incluídas as entidades que depois desapareceram. A segunda é existir uma fonte autoritativa da qual se leia, sob um critério de sobrevivência declarado, que entidades não sobreviveram até a data de corte. A terceira é existir uma ligação determinística entre as entidades dos registros e as da fonte, sem casamento por similaridade de nome. As três condições são proposição deste trabalho, e não se examinou se outros domínios as cumprem.

3.4 Instrumentos, métricas e análise dos dados

Os instrumentos são as duas ferramentas avaliadas, executadas localmente com o DuckDB como motor de execução[10], e *scripts* em Python, que adquirem os arquivos, constroem os conjuntos e calculam as métricas. São 106 verificações, que, com as escolhas de parâmetros, dão 144 combinações: em configuração padrão, 16 do dbt e 122 do Great Expectations; no nível com a referência, três; sob medida, três. Antes da execução, registraram-se com data os parâmetros e a referência de cada verificação e a regra que liga a cada companhia o que se aponta.

A análise usa estatística descritiva: contagens e proporções, reportadas por classe de defeito. A classe sintática e a classe *point-in-time* ficam separadas, porque essa separação permite exibir a assimetria que a hipótese prevê. O desfecho primário, que responde à QP1, é a detecção pelas ferramentas em configuração padrão. Ela se reporta com a verificação e a escolha de parâmetros que a produzem, e a detecção numa escolha não se estende às demais. A unidade da régua é a companhia, e não a linha, porque B remove de A as linhas de todos os papéis ligados a uma companhia, ou de nenhum.

Uma combinação é uma verificação com uma escolha de parâmetros. A verificação detecta o defeito quando aprova A e acusa o conjunto contaminado, e a ferramenta o detecta, num nível de configuração, quando ao menos uma das suas verificações o detecta. Apontar é listar as linhas ou os valores em que a verificação falha; a que não aponta papéis nem companhias fica só com o veredito. Cada saída de companhia que B remove é um evento, que a verificação detecta quando, ao verificar o conjunto contaminado, aponta a companhia ou um papel ligado a ela, sem contar o que aponta também ao verificar A.

Parte dos resultados não depende dos dados. Como toda linha de B também está em A, uma verificação que julga cada linha presente por si e aprova A aprova necessariamente B; são dessa forma os quatro testes do dbt em configuração padrão. Do mesmo modo, uma verificação que lista o que falta no conjunto em relação à referência *point-in-time* aprova A e acusa necessariamente B. Esses resultados, chamados aqui de fixados pela forma, são consequência lógica e se reportam como ponto de partida, e não como achado. Os demais, que só a execução revela, são chamados de dependentes dos dados. Os intervalos de confiança previstos no registro foram calculados; como o zero observado é fixado pela forma, eles não carregam informação estatística.

[10] Versões executadas: dbt-core 1.11.16, dbt-duckdb 1.11.1, DuckDB 1.5.6 e Great Expectations 1.19.1.

O custo de configuração, que responde à QP2, se mede em linhas de código não vazias que não são só comentário, segregadas por linguagem e por papel e contadas por um *script* fixado antes de as verificações existirem. As verificações seguem o registro feito antes da execução, e as linhas medem o tamanho da configuração, e não o esforço de quem a escreve. A construção da referência fica fora dessa medida, porque é comum aos níveis comparados e precede as verificações.

3.5 Ameaças à validade, reprodutibilidade e aspectos éticos

Quanto ao construto, o defeito injetado não é sintetizado, porque se removem entidades reais, identificadas em registro oficial, e a régua, tirada de um só registro, foi conferida por amostra. Quanto à validade interna, o controle positivo mostra que as verificações de linha funcionam. Os papéis sem ligação e as companhias que pararam de negociar sem ter o registro cancelado ficam em B, ameaça de sinal conhecido: B fica menos contaminado do que deveria, o que dificulta a detecção e favorece a hipótese.

A régua foi conferida fora do cadastro. Numa amostra de 30 das 166 companhias, sorteada com semente fixa e estratificada pelo motivo da saída, com os critérios gravados antes do sorteio, leram-se os documentos que as companhias entregam à CVM. Das 30, 21 concordam com o cadastro, 6 discordam e 3 ficam indeterminadas.

Quatro das seis são canceladas de ofício com entregas no ano anterior ao cancelamento, o que o critério conta como discordância sem mostrar que o cancelamento esteja errado. A amostra não comprova que todas as 166 sejam saídas. Os resultados fixados pela forma não dependem da régua: valem para qualquer relação de companhias que B remova. Dependem dela os vereditos das verificações de conjunto e a diferença entre B′ e B, e nada disso foi recalculado sem as seis companhias que discordam.

A validade externa se limita a um defeito, um mercado e duas ferramentas, numa versão e num motor de execução. Quanto à conclusão, o veredito das verificações de conjunto depende dos limites e das listas com que se comparam; por isso, o resultado se reporta para cada escolha prevista, e o que depende das escolhas se lê como exploratório.

Declaram-se os desvios em relação ao registro feito antes da execução. O registro previa medir também o tempo de autoria das verificações, sob cronômetro; a medida não foi feita, e a QP2, que perguntava o custo em linhas de código e em tempo de autoria, ficou só com as linhas. Três regras de reporte e quatro análises que o registro não pedia se fixaram depois de vistos os

resultados. Nenhum critério de comparação e nenhuma verificação mudou depois de vistos os vereditos.

Quanto à reprodutibilidade, a execução é integralmente local e roda sem rede, a partir de arquivos públicos. Cada arquivo adquirido entra num manifesto com resumo criptográfico da família *Secure Hash Algorithm* (SHA), na variante SHA-256, e carimbo de tempo de obtenção, e as versões das dependências são fixadas. O cuidado é necessário porque o arquivo de cadastro da CVM é sobrescrito diariamente e não é versionado, e o Formulário Cadastral e o banco de ISIN também são regravados pela fonte: sem o congelamento das cópias, o conjunto de sobreviventes e as ligações poderiam mudar entre execuções.

As verificações de integridade interrompem a construção quando falham, em vez de avisar, porque um aviso ignorado numa etapa intermediária se propaga em silêncio até os resultados. As medições do universo, o registro prévio e os seus desvios, a saída de cada fase da construção, os intervalos de confiança e as faixas de tolerância estão congelados no repositório do trabalho, e cada fase foi reproduzida byte a byte a partir de um clone limpo, sempre na mesma máquina. As reproduções mostram que o resultado não depende do estado do diretório de trabalho nem de rede; não mostram que outra máquina dê os mesmos bytes.

Sem as cópias congeladas, um terceiro reproduz menos: os arquivos do COTAHIST são estáveis na fonte, mas o *crosswalk*, a relação das companhias que saíram, o conjunto B, os vereditos e as métricas só se reproduzem sobre as cópias. Além das reproduções, cada contagem das métricas sai igual por uma segunda rota, escrita só com a biblioteca padrão da linguagem. Essa rota não é independente na origem, porque veio da mesma leitura do registro que o cálculo conferido, e as duas podem errar do mesmo modo; o código dela foi lido, e três dos números, refeitos à mão.

O código do *pipeline* e as verificações foram gerados por inteligência artificial (IA), a partir de planos e do registro aprovados antes, e revisados depois; por isso, o tempo de autoria previsto no registro não foi medido. O autor definiu problema, hipótese e desenho, obteve e leu as fontes, tomou as decisões e conferiu alegações e números. Os modelos Claude Fable, Claude Opus e Claude Sonnet, da Anthropic, e Gemini Neon, da Google, geraram códigos, buscaram trabalhos, extraíram trechos, conferiram referências e simularam revisão por pares. A partir das decisões do autor, redigiram versões, editaram a forma e ajustaram à diagramação as três figuras, concebidas pelo autor e pela orientadora. O autor revisou o conteúdo e assume responsabilidade total por este trabalho.

Quanto aos aspectos éticos, a pesquisa não envolve seres humanos, não usa dados pessoais e trabalha apenas com informações de acesso público, da B3 e da CVM, estas sob a licença ODbL. Essa situação dispensa o registro e a avaliação pelo sistema dos Comitês de Ética em Pesquisa e da Comissão Nacional de Ética em Pesquisa (CEP/Conep) (BRASIL, 2016, art. 10, parágrafo único, II).

4 RESULTADOS

4.1 Conjuntos construídos

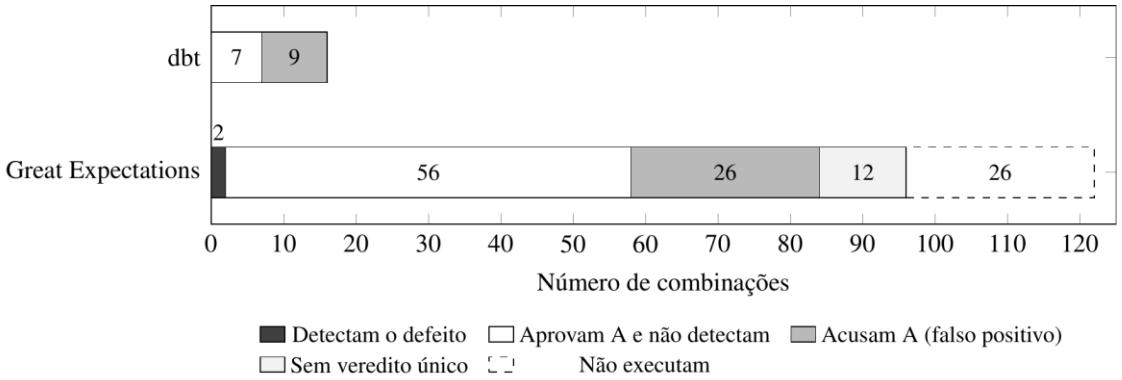
O conjunto A tem 901 papéis e 85.033 papel-meses, e B é A sem os papéis ligados às 166 companhias que saíram até a data de corte. B′ descarta 473 dos 901 papéis de A: 431 ligados a uma companhia e os 42 sem ligação. Dos 431 ligados, 172 (39,9%), de 119 companhias, são de companhia sem saída registrada no cadastro até a data de corte. Desses 172, 107 são de companhia que segue negociada no último mês do painel por outro papel, 14 de companhia cujo último negócio em A é de 2025, antes de dezembro, e 51 de companhia cujo último negócio em A é anterior a 2025, sem saída registrada no cadastro até a data de corte.

Na direção contrária, B′ retém um papel que B remove, a PORT3, de companhia que negociou em dezembro de 2025 e teve o registro cancelado no mesmo mês. No universo restrito ao livro 02, B′ descarta 451 dos 877 papéis, 409 deles ligados, e a diferença é de 165 papéis (40,3% dos ligados). Os 42 papéis sem ligação somam 1.702 dos 85.033 papel-meses de A, e quarenta deles têm o último negócio antes de 2018.

4.2 Detecção e controle negativo (QP1)

Executaram-se as 144 combinações registradas antes da execução contra todos os conjuntos, e a Figura 3 resume a configuração padrão. As 16 combinações do dbt são de linha. Nove acusam A (56,3%), e as sete que o aprovam aprovam também B e B′: o dbt não detecta o defeito em nenhuma escolha de parâmetros. Das 122 do Great Expectations, 26 não executam, e a detecção delas não se define; 12, as de contagem por mês comparadas com um centro, não têm veredito único. Das 84 restantes, 26 acusam A (31,0%): 16 das 48 de conjunto, 10 das 17 de linha e nenhuma das 19 de esquema. Das 58 que aprovam A, duas acusam B, e uma delas acusa também B′.

Figura 3 – Combinações de verificação e parâmetros em configuração padrão, por resultado e por ferramenta



Fonte: Elaborado pelos autores (2026).

As duas detecções são de verificações de conjunto do Great Expectations, que dão só o veredito, sem apontar o que falta (Quadro 1). A primeira exige que os códigos distintos do conjunto contenham uma lista, aqui a dos 428 códigos negociados em dezembro de 2025, que coincide com os códigos de A nesse mês. Ela aprova A e B′, que retém exatamente esses códigos, e acusa B, em que falta a PORT3, o único papel do último mês de A que B remove.

A segunda exige que o valor mais comum da coluna esteja num conjunto de valores, aqui o CNPJ contra o cadastro inteiro. Em A, uma só companhia tem a maior contagem de papel-meses, 868, e a verificação aprova. Essa companhia não está em B nem em B′, e neles três companhias empatam no topo, com 576 papel-meses cada; sem o parâmetro que admite empate, a verificação só aprova com um único valor no topo. No universo restrito ao livro 02, nenhuma das duas detecta, e nenhuma verificação em configuração padrão detecta o defeito.

Quadro 1 – Verificações que aprovam A e acusam B, por nível de configuração, no mercado acionário brasileiro, de 2010 a 2025

Nível	Ferramenta	Verificação e escolha	B	B′
Padrão	Great Expectations	Lista contida nos códigos distintos do conjunto, com os códigos de dezembro de 2025	Detecta	Não detecta
Padrão	Great Expectations	CNPJ mais comum contido num conjunto de valores, com o cadastro inteiro	Detecta	Detecta
Com a referência	dbt	relationships, com a referência como tabela testada	Detecta	Detecta
Com a referência	Great Expectations	Códigos da referência contidos nos códigos distintos do conjunto	Detecta	Detecta
Com a referência	Great Expectations	Contagem de linhas igual à da referência, mês a mês	Detecta	Detecta
Sob medida	dbt	Singular test	Detecta	Detecta
Sob medida	Great Expectations	Custom Expectation por consulta	Detecta	Detecta
Sob medida	Great Expectations	Expectation em Python, com métrica própria	Detecta	Detecta

Fonte: Elaborado pelos autores (2026).

No nível com a referência e sob medida, as seis verificações, todas de conjunto, aprovam A e acusam B e B′, também no universo restrito ao livro 02. Esse resultado é fixado pelo que elas comparam (subseção 3.4). As oito verificações do Quadro 1 aprovam A no painel inteiro, março de 2020 incluso. Das seis desses dois níveis, só o *singular test* aponta o que falta: detecta os 166 eventos em B e 165 em B′. O que falta é o da companhia da PORT3: o papel está no último mês do painel, B′ não tira nenhum papel-mês dele, e o teste não tem o que apontar.

Nenhuma das 11 combinações do dbt que apontam o que falha detecta evento algum: o de 166, em B e em B′, resultado fixado pela forma. O zero se mantém sem as 29 companhias canceladas por incorporação (o de 137). Pelo motivo que o cadastro registra, 99 das 166 companhias saíram por cancelamento voluntário, 29 por cancelamento de ofício e 29 por incorporação; as demais, por outros motivos ou com situação de falida.

As 12 combinações sem veredito único são duas verificações de contagem por mês do Great Expectations, a do número de linhas e a do número de companhias distintas, cada uma com seis centros tirados da série do próprio conjunto. Sem limite, que é o padrão literal da ferramenta, e com o envelope do mínimo e do máximo do próprio conjunto, as duas aprovam A, B e B′, por construção. Com um centro, existe uma faixa de tolerâncias em que a verificação aprova A e acusa o conjunto contaminado, em B nas 12 combinações e, em B′, em 11. Com a média do painel como centro, por exemplo, a contagem de linhas separa A de B para desvios de 13,80% a 18,95%. A faixa, porém, só se conhece depois de visto o conjunto contaminado.

A fração das combinações em configuração padrão que acusam A, a taxa de falsos positivos do controle negativo, depende da escolha de parâmetros. Sem parâmetro a escolher, acusa A uma das seis do dbt e uma das 50 do Great Expectations, e nas duas ferramentas é a que exige o CNPJ não nulo: nos 42 papéis sem ligação, o CNPJ é nulo. Com as companhias de situação ativa no cadastro e com o cadastro lido por datas, todas acusam A; com o cadastro inteiro, nenhuma das duas do dbt e três das seis do Great Expectations. Com cada lista de códigos, acusam A as duas do dbt e cinco das seis do Great Expectations. Quatro combinações, duas de cada ferramenta, acusam A e aprovam B: comparam o CNPJ do papel com o cadastro lido por datas na data de corte, a leitura que define B.

As 26 combinações não executadas do Great Expectations são as mesmas em todos os conjuntos, e cada uma para com a mesma exceção. Em 20, a versão 1.19.1 não implementa a

métrica para o motor de execução usado; em cinco, a métrica pede a expressão regular a uma função que não trata o DuckDB; e a última o pacote exporta sem a métrica que a validação pede.

4.3 Custo de configuração (QP2)

O custo de configuração sai em linhas de código, contadas na versão executada das verificações. Em configuração padrão, os 16 testes do dbt ocupam 59 linhas de YAML, e as 122 combinações do Great Expectations, em 108 arquivos, ocupam 486 linhas de Python e 150 de SQL. Sob medida, o *singular test* do dbt tem 8 linhas de SQL e nenhum invólucro. A rota por consulta do Great Expectations tem também 8 linhas de SQL e 5 de Python em torno delas. A rota em Python não tem SQL: tem 5 linhas na lógica da métrica e 18 de invólucro. No nível com a referência, o relationships do dbt ocupa 6 linhas de YAML, e as duas *expectations* nativas, 13 e 14 linhas de Python, com 2 de SQL cada. A rota em Python carrega, executa e dá os mesmos vereditos da rota por consulta em todos os conjuntos. A construção da referência *point-in-time* fica fora da contagem.

4.4 Lacuna de capacidade (QP3)

No controle positivo, o dbt em configuração padrão detecta duas das três cópias, e o Great Expectations, as três. A cópia com nulos no código e a cópia com linhas duplicadas são detectadas pelas duas ferramentas. A cópia com erro de esquema, em que a data do *snapshot* vem como texto, é detectada por duas verificações de tipo do Great Expectations e por nenhuma do dbt, cujos quatro testes não verificam esquema.

Na classe *point-in-time*, o dbt detecta o de 166 eventos, em B e em B′, em todas as suas seis escolhas de parâmetros. No Great Expectations, a proporção de eventos detectados não se define em nenhuma das 14 escolhas, porque as suas verificações em configuração padrão ficam todas só com o veredito, também as duas que detectam o defeito. A lacuna de capacidade sai, assim, como um par de contagens por ferramenta: assimétrico no dbt e, no Great Expectations, sem número do lado *point-in-time*.

5 DISCUSSÃO

A hipótese, na letra da subseção 1.2, é uma conjunção, e sob essa leitura não se confirma: duas combinações do Great Expectations detectam o defeito em configuração padrão, no painel. A leitura oração por oração, que se segue, veio depois dos resultados e é exploratória. Ela separa

o que a forma lógica das verificações já fixava, que se demonstra sem os dados, do que a execução testou.

Quanto à QP1, as verificações do dbt em configuração padrão não distinguem o conjunto contaminado do seu gêmeo correto, e o resultado é fixado pela forma: como toda linha de B é linha de A, o teste que só lê as linhas presentes e aprova A aprova também B. A cegueira estrutural de que fala a subseção 1.1 vale, assim, para as verificações de linha, e nesse nível o dbt não tem outras. O resultado é do teste aplicado ao conjunto verificado, e não do teste em si: com a referência *point-in-time* como tabela testada, o relationships detecta.

No Great Expectations, a resposta não é a mesma, porque duas verificações de conjunto acusam B. Cada detecção, porém, vale para a escolha de parâmetros que a produz, e nenhuma das duas resiste à troca do universo: uma decorre da coincidência de uma lista com o último mês de A, e a outra, de um empate. O resultado mostra que, sem a referência temporal externa, uma verificação de conjunto pode acusar o conjunto contaminado, numa escolha, sem dizer o que falta. A escolha do universo mudou o que dependia dos dados e não mudou o que a forma fixava.

A oração da hipótese sobre o apontar se confirma no dbt e não foi testada no Great Expectations, cujas verificações, no formato de resultado usado, ficam só com o veredito. Com a referência *point-in-time*, verificações nativas das duas ferramentas detectam no veredito, sem código sob medida: o limite é de informação, e não de mecanismo. O resultado coincide com o que o referencial descrevia. A *population completeness* de Pipino, Lee e Wang (2002) detecta a ausência contra uma referência externa, e a comparação com uma tabela só avalia completude quando existe uma fonte tomada como verdade (PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026, p. 1:27). Nenhuma das duas ferramentas fornece essa fonte; neste desenho, ela é o próprio A, reconstruído de fora.

Nas verificações de contagem por mês, há sinal, e a cegueira estrutural não se estende a elas. O que falta é o limite: a faixa de tolerâncias que separa os conjuntos só se conhece depois de visto o conjunto contaminado. Nenhuma tolerância convencional, fixada sem vê-lo, foi testada, e não houve controle de remoção do mesmo tamanho, com companhias sobreviventes sorteadas. Comparar o conjunto com a série dele mesmo não substituiu, assim, a referência. Na leitura deste trabalho, é o limite que o referencial atribuía à detecção de anomalia, que confronta uma métrica com a série dos seus valores anteriores (SCHELTER et al., 2018, p. 1783), e às *screens* estatísticas de Kimball e Caserta (2004, cap. 4). Esses resultados dependem das escolhas de centro e são exploratórios.

O controle negativo é resultado próprio da QP1. Na leitura deste trabalho, a acusação de A depende mais da escolha da referência do que da ferramenta. Cada lista e cada leitura do cadastro dadas às verificações descreve o mercado numa data só, e a escolha que confere se o conjunto cabe numa referência de data única acusa A sempre que algum papel do painel saiu antes dessa data. Com uma referência assim, parte das verificações acusa o conjunto correto e aprova o contaminado.

O resultado converge com o caso de Duarte et al. (2026), em que o próprio procedimento de qualidade origina o defeito; os autores tratam de filtros de amostra em opções, e a ligação com as ferramentas é deste trabalho. As combinações não executadas, por sua vez, delimitam o alcance do resultado do Great Expectations: o catálogo que a ferramenta anuncia é, na versão e no motor de execução usados, maior do que o conjunto das verificações que executam.

Quanto à QP3, a lacuna de capacidade é assimétrica no dbt e fica sem número no Great Expectations, o que não é o mesmo que zero: a indefinição vem do desenho, e não da ferramenta. A oração da hipótese sobre os defeitos sintáticos se confirma por inteiro só no Great Expectations, e para três cópias de defeito injetado por regra, não para a classe sintática em geral.

Há aqui um limite do que o experimento mostrou. O registro feito antes da execução classificou como dependente dos dados o resultado das combinações do dbt que apontam o que falha, e a classificação estava errada: a execução revelou que o zero é fixado pela forma. Nas verificações de linha, portanto, o experimento confirmou uma consequência lógica, que serve de linha de base. O conteúdo empírico do trabalho está noutro lugar: nos vereditos das verificações de conjunto, nas faixas de tolerância, na magnitude do controle negativo por escolha, nas combinações não executadas e na diferença entre B′ e B.

Quanto à QP2, a verificação sob medida é pequena em qualquer das três rotas, e, no veredito, o que detecta é a referência, e não o mecanismo de extensão. A resposta é mais estreita do que a pergunta de partida, que incluía o tempo de autoria. Ela não mede esforço, tempo nem descoberta, e tampouco o custo da referência que essas linhas consultam, que teve de ser reconstruída de fora. A alegação de classe da subseção 2.2 apoia-se apenas no argumento analítico, e a execução não lhe acrescenta base empírica de duas ferramentas. No veredito, as duas divergem, mas a divergência compara catálogos de escopos diferentes: o dbt foi medido pelos quatro testes de fábrica, sem verificação de conjunto, e o Great Expectations, pelo catálogo nativo, que as tem. Nada aqui se estende a ferramenta não executada.

No nível do código de negociação, a lista do último mês descarta mais do que as saídas (subseção 4.1), mas o total de papéis ligados sem saída registrada mede um critério misto, pois o universo se define pela negociação e a saída, pelo registro. Só a classe das companhias que seguem negociadas no último mês por outro papel se lê sem ambiguidade, e ela dá medida, neste mercado, à observação de que nem toda ausência de um código de negociação corresponde à morte de uma empresa, que Elton, Gruber e Blake (1996, p. 1102-1103) documentaram em fundos. A unidade da régua, a companhia, é escolha do desenho: para quem calcula retornos, a unidade é o papel, e nessa leitura parte do que B′ descarta pode ser saída real do papel.

Diante dos trabalhos relacionados, a relação mais próxima é com Gargouri e Reza (2026, p. 7), em que a linha de base de testes estruturais captura as quatro anomalias de integridade de chave, três das seis mistas e nenhuma das seis de domínio semântico. Aqui, a configuração padrão detecta os nulos e as duplicatas e, no dbt, não aponta as companhias que faltam. O padrão é o mesmo, e as taxas não se comparam, porque as unidades são outras. O que este desenho acrescenta é o controle negativo: a fonte não reporta o veredito dos testes sobre os dados íntegros, e aqui ele é parte do método.

O desenho de injetar defeitos e contar detecções, que Martins et al. (2025) aplicam a classes estáticas, estendeu-se a uma classe temporal e a um conjunto gêmeo. Com Schinninger et al. (2025), o resultado converge: os erros que os autores injetam deixam marca no conjunto verificado e são capturados pelo catálogo nativo (SCHINNINGER et al., 2025, p. 799-800), e o registro ausente, que não deixa marca, não é apontado por nenhuma verificação de linha. Com os trabalhos que mapeiam capacidade sem executar as ferramentas (ALEXAKIS et al., 2025; PAPASTERGIOS; EHRLINGER; GOUNARIS, 2026; REHBERGER et al., 2027; ZHOU et al., 2024), a divergência é de método, e os resultados se completam: a execução mostrou as combinações que não executam e o veredito que muda com a escolha de parâmetros, que o mapeamento não alcança.

A frase de Fonseca (2026, p. 2) citada na subseção 1.2 é sobre detectores de vazamento em *pipelines*, e não sobre ferramentas de qualidade de dados; por analogia, as verificações que aprovam os três conjuntos nada certificam sobre eles com o seu silêncio.

Para quem publica esses dados, o resultado indica que a referência *point-in-time* depende de versões datadas do cadastro, que a fonte sobrescreve e não versiona; para quem os consome, que a aprovação por verificações genéricas não diz que o conjunto esteja livre de viés de sobrevivência. Nenhum desses efeitos foi medido.

6 CONSIDERAÇÕES FINAIS

Este trabalho auditou se o dbt e o Great Expectations, em configuração padrão, detectam um conjunto *point-in-time* contaminado por viés de sobrevivência, usado como sonda. Na leitura literal da hipótese registrada antes da execução, a conjunção não se confirma, porque duas combinações do Great Expectations detectam o defeito em configuração padrão, embora sejam verificações de conjunto que dão só o veredito, valham para escolhas específicas de parâmetros e falhem no universo restrito ao lote padrão. Nas verificações de linha, a cegueira é consequência lógica demonstrável sem os dados: toda linha do conjunto contaminado é linha do correto, e a verificação que aprova um também aprova o outro.

Quanto à QP1, as verificações genéricas do dbt não distinguem o conjunto contaminado do seu gêmeo correto em nenhuma escolha de parâmetros, e as do Great Expectations só o fazem nas duas combinações de conjunto já apontadas. O controle negativo integra a resposta: parte das combinações das duas ferramentas acusa o conjunto correto, em proporção dependente dos parâmetros, e parte do catálogo do Great Expectations não executa na versão e no motor usados.

Quanto à QP2, a verificação sob medida é pequena em qualquer rota, inclusive no *singular test* do dbt, que lista o que falta; a contagem não mede esforço, tempo nem descoberta e exclui a construção da referência. Quanto à QP3, a lacuna de capacidade forma um par assimétrico: no controle positivo, o dbt detecta apenas nulos e duplicatas e o Great Expectations detecta também o erro de esquema, enquanto na classe *point-in-time* o dbt não aponta saídas e o Great Expectations se atém apenas ao veredito. Com a referência *point-in-time*, verificações nativas das duas ferramentas detectam sem código sob medida: no veredito, falta-lhes a referência temporal externa.

Dos seis objetivos específicos, os quatro primeiros foram atingidos: o método de avaliação foi proposto, o universo de cada mês e a relação das companhias que saíram foram reconstruídos de fontes oficiais, os conjuntos gêmeos foram gerados, e as verificações, implementadas nos três níveis de configuração. O quinto foi atingido em parte: mediram-se a detecção e os falsos positivos, e o custo de configuração saiu só em linhas de código, sem o tempo de autoria. O sexto também foi atingido em parte: a lacuna está caracterizada, mas a generalização a outras ferramentas apoia-se só no argumento analítico sobre a forma das verificações de linha.

A contribuição metodológica é o protocolo de conjuntos gêmeos por remoção controlada e suas métricas, que incorporam o controle negativo e a separação entre o que a forma lógica

fixa e o que depende dos dados. A contribuição empírica é a instanciação brasileira com fontes oficiais públicas e gratuitas, cujo conteúdo empírico reúne os vereditos das verificações de conjunto, o controle negativo por escolha de parâmetros, as combinações não executadas, o episódio do filtro de universo e a diferença entre B′ e B. Não se alega primazia em medir detecção de defeito *point-in-time*: o trabalho entrega uma auditoria de duas ferramentas de prateleira em configuração padrão.

As limitações se dizem pelo que tiram da conclusão. O resultado é de um só defeito, de um só mercado acionário e de duas ferramentas, numa versão e num motor de execução, sem os testes que o dbt permite instalar por pacote. O universo é o dos papéis negociados em cada mês, não o dos listados nem a composição de índice. O custo de construir a referência não foi medido. A régua vem de um só registro, o cadastro, e foi conferida fora dele por amostra, em que nem todas as companhias concordam com ele. A diferença de linhas entre as rotas não se generaliza a quem configura as ferramentas; o artefato também não foi posto em uso por equipes de dados, em ambiente operacional.

Os trabalhos futuros saem desses limites. O primeiro é verificar se outro domínio de aplicação cumpre as três condições exigidas pelo método. O segundo é estender o desenho a outras formas do problema *point-in-time*, como o problema de *vintage* em séries macroeconômicas revisadas.

O terceiro é executar o mesmo protocolo em outras ferramentas, com os testes instalados por pacote, e em outra versão ou outro motor de execução. O quarto é testar, nas verificações de contagem, tolerâncias fixadas sem ver o conjunto contaminado, com um controle de remoção do mesmo tamanho. O quinto é pôr o artefato em uso por equipes de dados, em ambiente operacional, e medir, com implementadores humanos, o tempo e a descoberta, que a contagem de linhas não alcança.

REFERÊNCIAS

ALEXAKIS, Theodoros *et al.* Evaluating data quality: Comparative insights on standards, methodologies, and modern software tools. *Electronics*, Basel, v. 14, n. 15, 3038, p. 1-28, 2025. DOI: https://doi.org/10.3390/electronics14153038. Disponível em: https://www.mdpi.com/2079-9292/14/15/3038. Acesso em: 18 nov. 2026.

B3 S.A. – BRASIL, BOLSA, BALCÃO. *Layout do arquivo – Cotações históricas*: COTAHIST.AAAA.TXT. [São Paulo]: B3, 2020. Revisão 02, 5 out. 2020. Capítulo I. Disponível em:

https://www.b3.com.br/data/files/33/67/B9/50/D84057102C784E47AC094EA8/SeriesHistoricas_Layout.pdf. Acesso em: 28 out. 2026.

B3 S.A. – BRASIL, BOLSA, BALCÃO. *Relatório anual 2020*. [São Paulo]: B3, [2021]. Disponível em: https://www.rad.cvm.gov.br/ENET/frmExibirArquivoIPEExterno.aspx?NumeroProtocoloEntrega=857700. Acesso em: 26 out. 2026.

B3 S.A. – BRASIL, BOLSA, BALCÃO. *Códigos ISIN: pesquisa*. [São Paulo]: B3, 2026a. Banco de dados completo, gerado em 16 out. 2026; detalhe por ISIN consultado em 21 out. 2026. Disponível em: https://sistemaswebb3-listados.b3.com.br/isinPage/. Acesso em: 18 out. 2026.

B3 S.A. – BRASIL, BOLSA, BALCÃO. *Cotações históricas*. [São Paulo]: B3, 2026b. Disponível em: https://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/historico/mercado-a-vista/cotacoes-historicas/. Acesso em: 29 out. 2026.

BARROS, Thiago de Sousa. Lottery-type stocks in Brazil: Evidence from a survivorship-bias-corrected universe. *Revista Brasileira de Finanças*, Rio de Janeiro, v. 24, e202613, p. 1-34, 2026. DOI: https://doi.org/10.12660/rbfin.v24n1.2026.98265. Disponível em: https://periodicos.fgv.br/rbfin/article/view/98265. Acesso em: 18 nov. 2026.

BATINI, Carlo; SCANNAPIECO, Monica. *Data Quality*: Concepts, methodologies and techniques. Berlin: Springer, 2006. (Data-Centric Systems and Applications). ISBN 978-3-540-33172-8. DOI: https://doi.org/10.1007/3-540-33173-5. Disponível em: https://link.springer.com/book/10.1007/3-540-33173-5. Acesso em: 18 nov. 2026.

BATINI, Carlo; SCANNAPIECO, Monica. *Data and Information Quality*: Dimensions, principles and techniques. Switzerland: Springer, 2016. (Data-Centric Systems and Applications). ISBN 978-3-319-24106-7. DOI: https://doi.org/10.1007/978-3-319-24106-7. Disponível em: https://link.springer.com/book/10.1007/978-3-319-24106-7. Acesso em: 18 nov. 2026.

BENHENDA, Mostapha. *Look-Ahead-Bench*: a standardized benchmark of look-ahead bias in point-in-time LLMs for finance. 20 Jan. 2026. arXiv:2601.13770v1. Preprint não revisado por pares. Disponível em: https://arxiv.org/abs/2601.13770. Acesso em: 14 nov. 2026.

BRASIL. Conselho Nacional de Saúde. Resolução no 510, de 7 de abril de 2016. *Diário Oficial da União: seção 1*, Brasília, DF, n. 98, p. 44-46, 24 maio 2016. Disponível em: https://www.gov.br/conselho-nacional-de-saude/pt-br/atos-normativos/resolucoes/2016/resolucao-no-510.pdf. Acesso em: 17 nov. 2026.

BRECK, Eric *et al.* Data validation for machine learning. *In*: CONFERENCE ON SYSTEMS AND MACHINE LEARNING, 2., 2019, Palo Alto. *Proceedings of Machine Learning and Systems*. [*S. l.: s. n.*], 2019. v. 1, p. 334-347. PDF sem fólio impresso; as páginas citadas são as do PDF, de 1 a 14. Disponível em: https://proceedings.mlsys.org/paper_files/paper/2019/hash/928f1160e52192e3e0017fb63ab65391-Abstract.html. Acesso em: 29 out. 2026.

BROWN, Stephen J. *et al.* Survivorship bias in performance studies. *The Review of Financial Studies*, [Cary, NC], v. 5, n. 4, p. 553-580, 1992. DOI: https://doi.org/10.1093/rfs/5.4.553. Disponível em: https://academic.oup.com/rfs/article-abstract/5/4/553/1590264. Acesso em: 18 nov. 2026.

BROWN, Stephen J.; GOETZMANN, William N.; ROSS, Stephen A. Survival. *The Journal of Finance*, [*S. l.*], v. 50, n. 3, p. 853-873, July 1995. DOI: https://doi.org/10.1111/j.1540-6261.1995.tb04039.x. Disponível em: https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1995.tb04039.x. Acesso em: 18 nov. 2026.

CARHART, Mark M. On persistence in mutual fund performance. *The Journal of Finance*, [*S. l.*], v. 52, n. 1, p. 57-82, Mar. 1997. DOI: https://doi.org/10.1111/j.1540-6261.1997.tb03808.x. Disponível em: https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1997.tb03808.x. Acesso em: 18 nov. 2026.

CHEN, Hao; PHANI, Arnab; SCHELTER, Sebastian. *PrismaDV*: Automated task-aware data unit test generation. 23 Apr. 2026. arXiv:2604.21765v1. Preprint não revisado por pares. Disponível em: https://arxiv.org/abs/2604.21765. Acesso em: 14 nov. 2026.

COMISSÃO DE VALORES MOBILIÁRIOS. *Cadastro de Companhias Abertas*. [Rio de Janeiro]: Comissão de Valores Mobiliários, 2026a. Portal de Dados Abertos da CVM. Licença: *Open Database License* (ODbL) v1.0, https://opendatacommons.org/licenses/odbl/1-0/. Disponível em: https://dados.cvm.gov.br/dados/CIA_ABERTA/CAD/DADOS/cad_cia_aberta.csv. Acesso em: 18 out. 2026.

COMISSÃO DE VALORES MOBILIÁRIOS. *Formulário Cadastral (FCA): valores mobiliários*. [Rio de Janeiro]: Comissão de Valores Mobiliários, 2026b. Portal de Dados Abertos da CVM. Licença: *Open Database License* (ODbL) v1.0, https://opendatacommons.org/licenses/odbl/1-0/. Disponível em: https://dados.cvm.gov.br/dados/CIA_ABERTA/DOC/FCA/DADOS/. Acesso em: 18 out. 2026.

CROUSHORE, Dean; STARK, Tom. A real-time data set for macroeconomists. *Journal of Econometrics*, [Amsterdam], v. 105, n. 1, p. 111-130, Nov. 2001. DOI: https://doi.org/10.1016/S0304-4076(01)00072-0. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0304407601000720. Acesso em: 18 nov. 2026.

DAMA INTERNATIONAL. *DAMA-DMBOK*: Data management body of knowledge. 2nd ed. Basking Ridge: Technics Publications, 2017. *E-book*. ISBN 978-1-63462-236-3.

DBT LABS. *About dbt source command*. [*S. l.*]: dbt Labs, 16 Sept. 2026a. Disponível em: https://docs.getdbt.com/reference/commands/source. Acesso em: 14 nov. 2026.

DBT LABS. *Add data tests to your DAG*. [*S. l.*]: dbt Labs, 16 Sept. 2026b. Disponível em: https://docs.getdbt.com/docs/build/data-tests. Acesso em: 14 nov. 2026.

DBT LABS. *Model contracts*. [*S. l.*]: dbt Labs, 29 Oct. 2026c. Disponível em: https://docs.getdbt.com/docs/mesh/govern/model-contracts. Acesso em: 14 nov. 2026.

DUARTE, Jefferson *et al.* Too good to be true: Look-ahead bias in empirical options research. *The Review of Financial Studies*, [*S. l.*], 2026. Artigo em acesso antecipado; volume e paginação ainda não atribuídos. Versão consultada: manuscrito aceito, SSRN 4590083, 5 ago. 2025. DOI: https://doi.org/10.1093/rfs/hhag061. Disponível em: https://academic.oup.com/rfs/advance-article-abstract/doi/10.1093/rfs/hhag061/8722307. Acesso em: 18 nov. 2026.

ELTON, Edwin J.; GRUBER, Martin J.; BLAKE, Christopher R. Survivorship bias and mutual fund performance. *The Review of Financial Studies*, [Cary, NC], v. 9, n. 4, p. 1097-1120, Winter 1996. DOI: https://doi.org/10.1093/rfs/9.4.1097. Disponível em: https://academic.oup.com/rfs/article-abstract/9/4/1097/1580100. Acesso em: 18 nov. 2026.

FONSECA, Xavier. *Look-Ahead-Freedom as Temporal Non-Interference*: A verifiable correctness property for backtesting and agentic trading pipelines. 6 July 2026. arXiv:2607.04958v1. Preprint não revisado por pares. Disponível em: https://arxiv.org/abs/2607.04958. Acesso em: 14 nov. 2026.

GARGOURI, Ismail; REZA, Hassan. *A Multi-Layer Testing Framework for Automated Data Quality Assurance in Cloud-Native ELT Pipelines*. 2026. arXiv:2605.20500v1. Preprint não revisado por pares. Disponível em: https://arxiv.org/abs/2605.20500. Acesso em: 14 nov. 2026.

GLOCK, Anna-Christina *et al. Data Quality Rule Generation with LLMs*. 5 Sept. 2026. arXiv:2609.06053v1. Preprint não revisado por pares. Disponível em: https://arxiv.org/abs/2609.06053. Acesso em: 14 nov. 2026.

GREAT EXPECTATIONS. *Create an Expectation*. [*S. l.*]: Great Expectations, c2026a. Disponível em: https://docs.greatexpectations.io/docs/core/define_expectations/create_an_expectation. Acesso em: 14 nov. 2026.

GREAT EXPECTATIONS. *Customize an Expectation Class*. [*S. l.*]: Great Expectations, c2026b. Disponível em: https://docs.greatexpectations.io/docs/core/customize_expectations/define_a_custom_expectation_class/. Acesso em: 14 nov. 2026.

GREAT EXPECTATIONS. *Use SQL to define a custom Expectation*. [*S. l.*]: Great Expectations, c2026c. Versão exibida na consulta: GX Core 1.25.0. Disponível em: https://docs.greatexpectations.io/docs/core/customize_expectations/use_sql_to_define_a_custom_expectation/. Acesso em: 14 nov. 2026.

GREGOR, Shirley; HEVNER, Alan R. Positioning and presenting design science research for maximum impact. *MIS Quarterly*, [Minneapolis], v. 37, n. 2, p. 337-355, June 2013. DOI: https://doi.org/10.25300/MISQ/2013/37.2.01. Disponível em: https://misq.umn.edu/misq/article-abstract/37/2/337/103/Positioning-and-Presenting-Design-Science-Research. Acesso em: 18 nov. 2026.

HEVNER, Alan R. *et al.* Design science in information systems research. *MIS Quarterly*, [Minneapolis], v. 28, n. 1, p. 75-105, Mar. 2004. DOI: https://doi.org/10.2307/25148625. Disponível em: https://misq.umn.edu/misq/article-abstract/28/1/75/261/Design-Science-in-Information-Systems-Research1. Acesso em: 18 nov. 2026.

JENSEN, Christian S.; SNODGRASS, Richard T. Semantics of time-varying information. *Information Systems*, [Oxford], v. 21, n. 4, p. 311-352, June 1996. DOI: https://doi.org/10.1016/0306-4379(96)00017-8. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/0306437996000178. Acesso em: 18 nov. 2026.

JENSEN, Christian S.; SNODGRASS, Richard T. Temporal data management. *IEEE Transactions on Knowledge and Data Engineering*, [New York], v. 11, n. 1, p. 36-44, Jan./Feb. 1999. DOI: https://doi.org/10.1109/69.755613. Disponível em: https://ieeexplore.ieee.org/document/755613. Acesso em: 18 nov. 2026.

KIMBALL, Ralph; CASERTA, Joe. *The Data Warehouse ETL Toolkit*: Practical techniques for extracting, cleaning, conforming, and delivering data. Indianapolis: Wiley, 2004. *E-book*. Edição eletrônica (2011) do texto de 2004. ISBN 978-1-118-07968-3.

LÓPEZ DE PRADO, Marcos. *Advances in Financial Machine Learning*. Hoboken: Wiley, 2018. *E-book*. ISBN 978-1-119-48210-9.

MARTINS, Pedro *et al.* Performance and scalability of data cleaning and preprocessing tools: A benchmark on large real-world datasets. *Data*, Basel, v. 10, n. 5, 68, p. 1-22, 2025. DOI: https://doi.org/10.3390/data10050068. Disponível em: https://www.mdpi.com/2306-5729/10/5/68. Acesso em: 18 nov. 2026.

PAPASTERGIOS, Vasileios; EHRLINGER, Lisa; GOUNARIS, Anastasios. Unfolding data quality dimensions in practice: A survey. *Journal of Data and Information Quality*, [New York], v. 18, n. 1, p. 1:1-1:37, Mar. 2026. DOI: https://doi.org/10.1145/3786328. Disponível em: https://dl.acm.org/doi/10.1145/3786328. Acesso em: 18 nov. 2026.

PEFFERS, Ken *et al.* A design science research methodology for information systems research. *Journal of Management Information Systems*, [Armonk], v. 24, n. 3, p. 45-77, Winter 2007. DOI: https://doi.org/10.2753/MIS0742-1222240302. Disponível em: https://www.tandfonline.com/doi/abs/10.2753/MIS0742-1222240302. Acesso em: 18 nov. 2026.

PIPINO, Leo L.; LEE, Yang W.; WANG, Richard Y. Data quality assessment. *Communications of the ACM*, [New York], v. 45, n. 4, p. 211-218, Apr. 2002. DOI: https://doi.org/10.1145/505248.506010. Disponível em: https://dl.acm.org/doi/10.1145/505248.506010. Acesso em: 18 nov. 2026.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. *Metodologia do trabalho científico*: métodos e técnicas da pesquisa e do trabalho acadêmico. 2. ed. Novo Hamburgo: Feevale, 2013. ISBN 978-85-7717-158-3. Disponível em: https://www.feevale.br/institucional/editora-feevale/metodologia-do-trabalho-cientifico---2-edicao. Acesso em: 17 nov. 2026.

RANSE, Harjot Singh. *Survivorship Bias in Emerging Market Small-Cap Indices*: Evidence from India's NIFTY Smallcap 250. 2026. arXiv:2603.19380v1. Preprint não revisado por pares. Disponível em: https://arxiv.org/abs/2603.19380. Acesso em: 14 nov. 2026.

REDMAN, Thomas C. *Data Quality for the Information Age*. Boston: Artech House, 1996. *E-book*. ISBN 0-89006-883-6.

REDYUK, Sergey *et al.* Automating data quality validation for dynamic data ingestion. *In*: INTERNATIONAL CONFERENCE ON EXTENDING DATABASE TECHNOLOGY, 24., 2021, Nicosia. *Proceedings [...]*. Konstanz: OpenProceedings.org, 2021. p. 61-72. ISBN 978-3-89318-084-4. DOI: https://doi.org/10.5441/002/edbt.2021.07. Disponível em: https://openproceedings.org/2021/conf/edbt/p79.pdf. Acesso em: 18 nov. 2026.

REHBERGER, Tobias *et al.* Evaluating data quality tools: Measurement capabilities and LLM integration. *In*: INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS, 37., 2026, Graz. *Database and Expert Systems Applications [...]*. Cham: Springer Nature Switzerland, 2027. p. 319-325. (Lecture Notes in Computer Science, v. 16859). ISBN 978-3-032-34374-1. DOI: https://doi.org/10.1007/978-3-032-34374-1_24. Disponível em: https://link.springer.com/chapter/10.1007/978-3-032-34374-1_24. Acesso em: 18 nov. 2026.

RELATÓRIO DE INFLAÇÃO. Brasília: Banco Central do Brasil, v. 22, n. 1, mar. 2020. ISSN 1517-6576. Disponível em: https://www.bcb.gov.br/content/ri/relatorioinflacao/202003/ri202003p.pdf. Acesso em: 25 out. 2026.

SCHELTER, Sebastian *et al.* Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, [*S. l.*], v. 11, n. 12, p. 1781-1794, Aug. 2018. DOI: https://doi.org/10.14778/3229863.3229867. Disponível em: https://dl.acm.org/doi/10.14778/3229863.3229867. Acesso em: 18 nov. 2026.

SCHINNINGER, Christoph *et al.* Icewafl: A configurable data stream polluter. *In*: INTERNATIONAL CONFERENCE ON EXTENDING DATABASE TECHNOLOGY, 28., 2025, Barcelona. *Proceedings [...]*. Konstanz: OpenProceedings.org, 2025. p. 796-802. ISBN 978-3-89318-099-8. Artigo curto. DOI: https://doi.org/10.48786/edbt.2025.64. Disponível em: https://openproceedings.org/2025/conf/edbt/paper-240.pdf. Acesso em: 18 nov. 2026.

SHANKAR, Shreya *et al.* Automatic and precise data validation for machine learning. *In*: ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 32., 2023, Birmingham, United Kingdom. *Proceedings [...]*. New York: ACM, 2023. p. 2198-2207. ISBN 979-8-4007-0124-5. DOI: https://doi.org/10.1145/3583780.3614786. Disponível em: https://dl.acm.org/doi/10.1145/3583780.3614786. Acesso em: 18 nov. 2026.

SNODGRASS, Richard T. *Developing Time-Oriented Database Applications in SQL*. San Francisco: Morgan Kaufmann, 2000. ISBN 1-55860-436-7. Disponível em: https://www2.cs.arizona.edu/~rts/tdbbook.pdf. Acesso em: 29 out. 2026.

WANG, Richard Y.; STRONG, Diane M. Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, [Armonk], v. 12, n. 4, p. 5-34, Spring 1996. DOI: https://doi.org/10.1080/07421222.1996.11518099. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/07421222.1996.11518099. Acesso em: 18 nov. 2026.

ZHANG, Fan *et al. When Alpha Disappears*: A one-switch benchmark for decision-time leakage in financial backtests. 12 May 2026. arXiv:2605.23959v1. Preprint não revisado por pares. Disponível em: https://arxiv.org/abs/2605.23959. Acesso em: 14 nov. 2026.

ZHOU, Yuhan *et al.* A survey on data quality dimensions and tools for machine learning: Invited paper. *In*: IEEE INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE TESTING, 6., 2024, Shanghai. *Proceedings [...]*. Los Alamitos: IEEE Computer Society, 2024. p. 120-131. ISBN 979-8-3503-6505-4. DOI: https://doi.org/10.1109/AITest62860.2024.00023. Disponível em: https://ieeexplore.ieee.org/document/10685186. Acesso em: 18 nov. 2026.