

CLS STANDARD · VERSÃO 1.0 · 2026

Classification of Software Litigation

O padrão brasileiro para classificação de litígios de software

Seis categorias (LS-01 a LS-06) · Norma de evidência (Classes A–D)
Graus de confiança · Princípios · Glossário · Modelo de aplicação

Cristiano de Moraes

Engenheiro de software · Assistente técnico em litígios de software

Publicado e mantido por Cepheus Consultoria Ltda · CNPJ 07.037.589/0001-41 · Itatiba–SP

Padrão aberto · Licença CC BY 4.0 · cepheus.com.br/metodologia

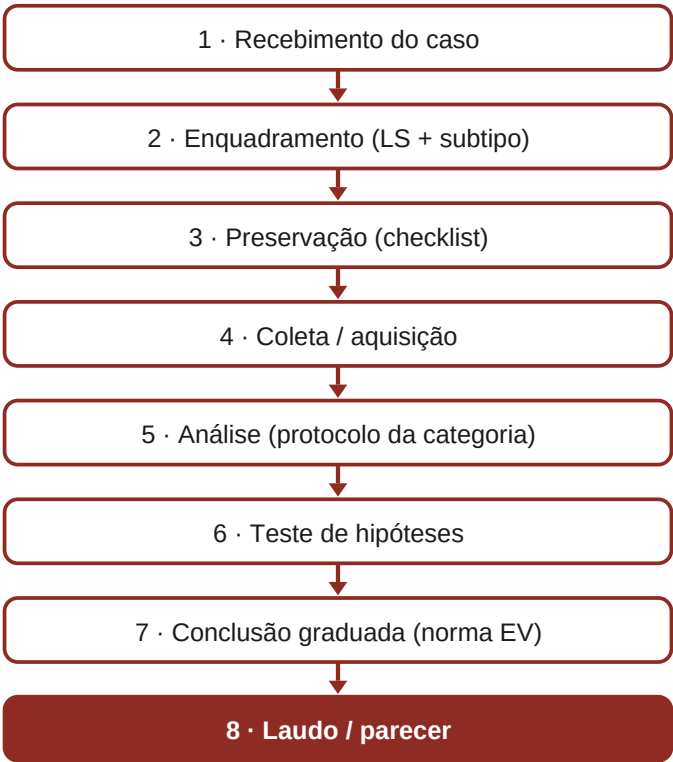
Como usar este padrão

A CLS organiza as disputas envolvendo software em seis categorias e estabelece uma norma transversal de evidência e de confiança, apoiada em seis princípios. **Para o advogado:** enquadre o caso, acione o checklist de preservação e use protocolo e quesitos como roteiro. **Para o perito:** adote o protocolo como método declarado do laudo e emita a conclusão com classes de evidência e grau de confiança.

A CLS em 20 segundos

Categoria	Nome	Produto principal do protocolo
LS-01	ERP e sistemas integrados	Matriz Escopo × Entrega
LS-02	Direito autoral e titularidade	Parecer de Titularidade
LS-03	Código-fonte	Comparação em 5 Camadas
LS-04	APIs e integração	Diagrama Causal + planilha de SLA
LS-05	SaaS e nuvem	Relatório de Portabilidade
LS-06	IA generativa	Matriz de Responsabilidade
EV	Norma transversal	Classes de evidência (A–D) + grau de confiança

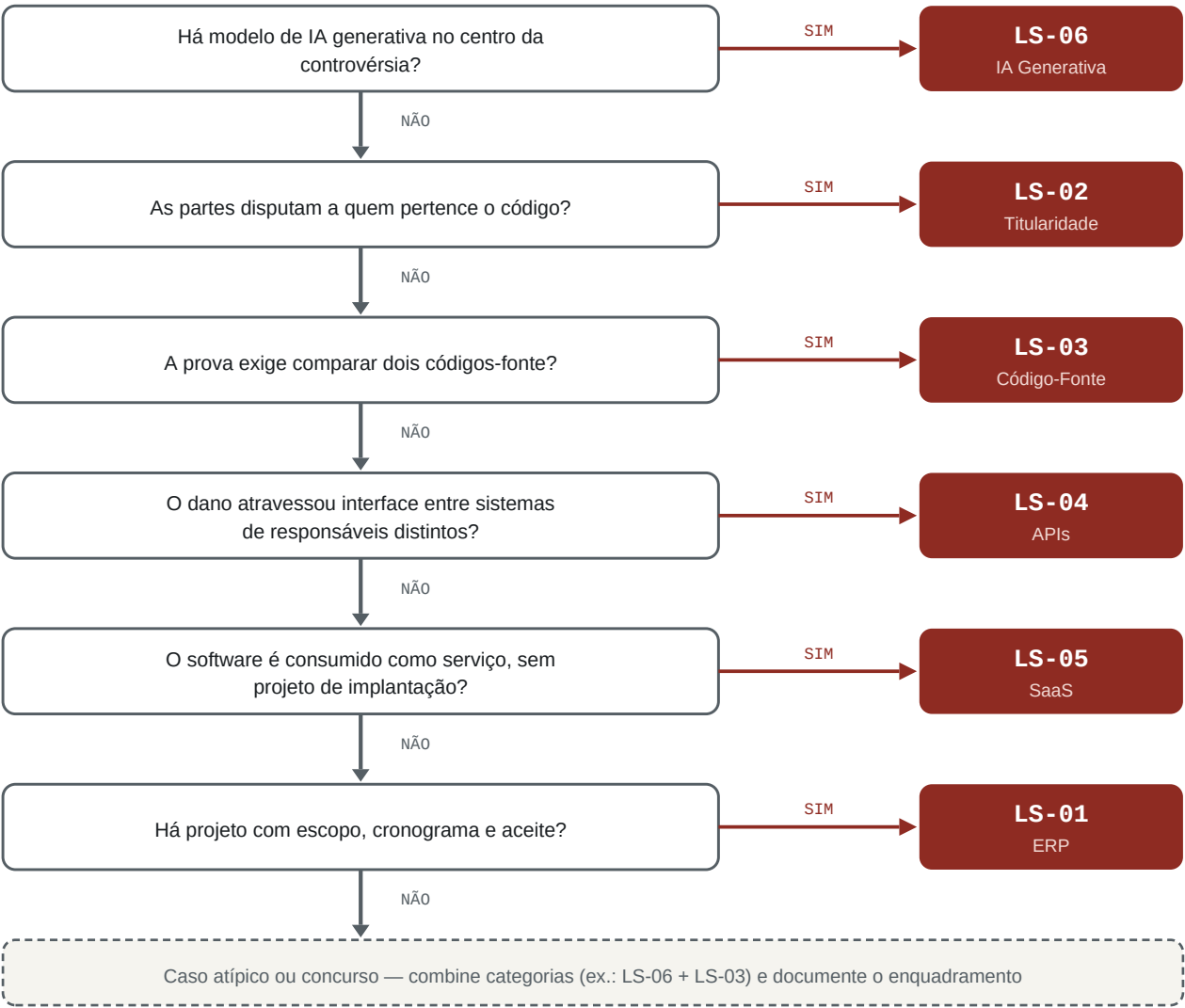
O fluxo pericial padrão



Do recebimento do caso ao laudo, o mesmo caminho em todas as categorias.

Fluxograma de enquadramento

Percorra as perguntas na ordem; a primeira resposta SIM define a categoria de partida.



Princípios

Princípios da CLS

A filosofia que sustenta o padrão — seis compromissos inegociáveis

Antes de qualquer protocolo, a CLS se apoia em seis princípios. Eles não são retórica: cada etapa dos protocolos existe para honrar um deles. Quando um caso concreto obriga a um desvio, é a esses princípios que o perito recorre para decidir.

1. Neutralidade técnica

A análise apura fatos; não defende teses. O assistente técnico e o perito trabalham para a verdade demonstrável, não para a parte que os contratou. Na prática, isso significa que a conclusão **segue a prova** — inclusive quando contraria quem pediu o trabalho. A criação independente (LS-03.3) é uma conclusão tão legítima quanto a cópia; um laudo que só sabe encontrar aquilo que o cliente quer não é técnico, é advocacia disfarçada — e desmorona na contradição.

2. Reprodutibilidade

Um terceiro qualificado, de posse do mesmo material e seguindo o mesmo protocolo, deve chegar à mesma conclusão. Por isso a CLS declara método, insumos e critérios de forma explícita: o laudo não é um oráculo, é um procedimento que qualquer outro perito pode refazer. Método público e repetível é o que separa prova técnica de opinião pessoal.

3. Rastreabilidade

Toda afirmação relevante aponta para a evidência que a sustenta — arquivo, linha, log, documento, com data e origem. Não há salto entre "o que eu vi" e "o que eu concluo": cada elo da cadeia é visível e verificável. A matriz de divergências (LS-01), o mapa de similaridade (LS-03) e a linha do tempo de autoria (LS-02) existem para tornar essa rastreabilidade concreta.

4. Proporcionalidade da conclusão

A força da afirmação nunca ultrapassa a força da evidência. Uma conclusão de alta confiança exige lastro convergente em evidências Classe A; onde só há prints e depoimentos (Classes C e D), o laudo diz "indício", não "prova". A norma EV existe precisamente para impedir o exagero: afirmações são calibradas pelo lastro, e o grau de confiança é declarado.

5. Cadeia de custódia

A evidência digital só vale se sua integridade for demonstrável do momento da coleta até o laudo. Imagem forense, hash, registro de data, origem e responsável — os princípios da ISO/IEC 27037 — garantem que ninguém alterou o material no caminho. Sem cadeia de custódia, mesmo a melhor análise é atacável: a parte contrária apenas pergunta "como sei que este é o código original?" e a prova rui.

6. Falsificabilidade das hipóteses

A CLS não busca confirmar uma tese; busca testá-la contra as alternativas. Em cada caso, as hipóteses concorrentes (cópia, derivação, criação independente; falha do fornecedor, do contratante, da infraestrutura) são formuladas e submetidas à prova: quais a evidência sustenta e quais ela afasta? A conclusão é a hipótese que sobrevive ao confronto com todos os achados — não a primeira que parecia plausível. É o método científico aplicado à perícia.

EV

Norma de evidência e grau de confiança

Norma transversal — aplicável a todas as categorias (LS-01 a LS-06)

Toda análise em conformidade com a CLS classifica cada evidência em uma das quatro classes (A–D) e emite a conclusão com grau de confiança declarado (alto, médio ou baixo), segundo critérios objetivos. É o que aproxima o laudo da prática científica: afirmações proporcionais ao lastro.

EV.1 — Classes de evidência

Classe	O que é	Exemplos	Papel probatório
A	Técnica, íntegra e verificável por terceiros	repositórios com histórico, logs preservados com hash, imagem forense, trilhas de auditoria (ISO/IEC 27037)	Fundante — sustenta conclusões por si
B	Documental datada, de origem rastreável	e-mails com cabeçalhos, atas assinadas, contratos e anexos, exportações de sistemas de chamados, documentação arquivada	Fundante em conjunto — sustenta com corroboração
C	Captura unilateral, sem integridade demonstrável	prints, gravações de tela sem ata, exportações parciais sem hash	Corroborativa — nunca fundamenta sozinha
D	Declaratória / testemunhal	depoimentos, declarações das partes, memória de reuniões	Contextual — orienta a investigação, não a conclusão

EV.2 — Regras de fundamentação

- Todo achado relevante do laudo referencia a classe da evidência que o sustenta — ex.: *"lastreado em evidência Classe A"*;
- Conclusões de **alta confiança** exigem lastro predominante em Classe A, com Classe B em suporte;
- Classes C e D **não fundamentam conclusão técnica isoladamente**;
- A degradação da evidência rebaixa a classe: log sem hash → C; print autenticado por ata notarial → B.

EV.3 — Graus de confiança da conclusão

Grau	Critérios objetivos (todos devem estar presentes)
Alto	evidência Classe A convergente de ao menos duas fontes independentes; hipóteses alternativas testadas e afastadas; análise reproduzível por terceiro com o mesmo material
Médio	Classe A parcial ou predominância de Classe B; ao menos uma hipótese alternativa não totalmente afastada; reprodução possível com ressalvas
Baixo	predominância de Classe C/D, material incompleto ou irreproduzível; conclusão meramente indicativa — o laudo recomenda diligência complementar

EV.4 — Declaração de conformidade (modelo)

"Este laudo observa a CLS v1.0: caso enquadrado na categoria LS-03.2; protocolo LS-03 integralmente cumprido; achados lastreados predominantemente em evidências Classe A (repositórios preservados com hash e histórico de versionamento); conclusão emitida com grau de confiança Alto."

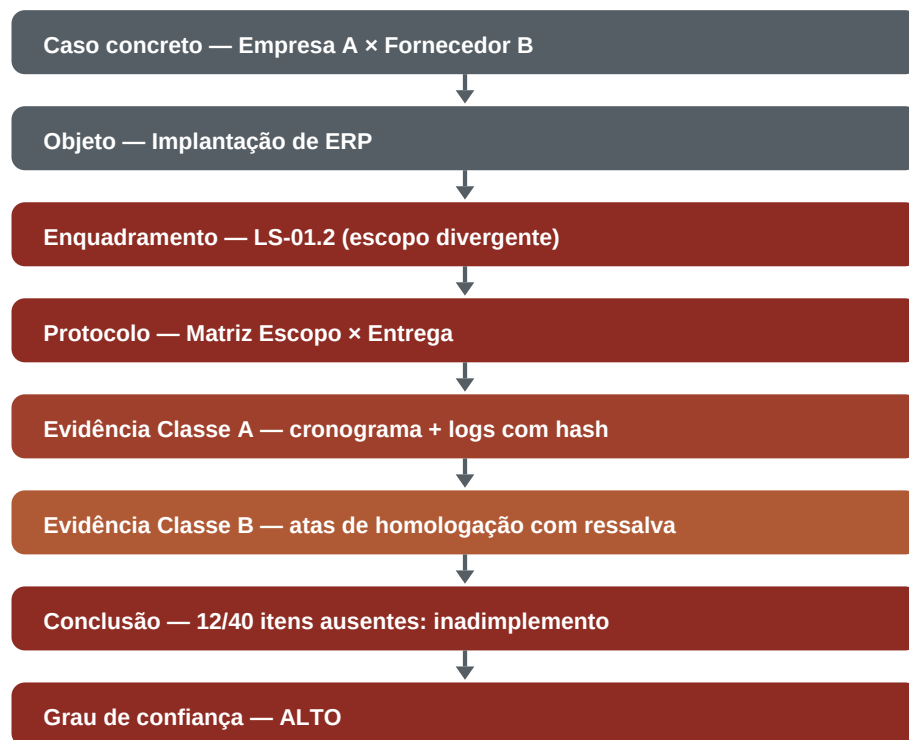
A declaração torna o trabalho auditável ponto a ponto: a parte contrária pode verificar o enquadramento, refazer o protocolo, conferir as classes atribuídas e testar os critérios do grau — e é exatamente essa auditabilidade que dá força ao laudo.

Modelo de aplicação

Modelo de aplicação

Como um caso real percorre o padrão, do enquadramento à confiança

O padrão em movimento: um caso hipotético de ERP atravessando todas as camadas da CLS — enquadramento, protocolo, classes de evidência e grau de confiança. É o mapa que o advogado segue do primeiro dia ao laudo.



Um caso real percorrendo o padrão, do enquadramento ao grau de confiança.

Passo a passo do exemplo

- **Caso.** A Empresa A contratou o Fornecedor B para implantar um ERP; no go-live, parte dos módulos não existia. A disputa é sobre conteúdo da entrega, não sobre atraso.
- **Enquadramento** → **LS-01.2.** Existe projeto com escopo, cronograma e aceite (logo, LS-01), e a controvérsia é o escopo divergente (subtipo .2). A pergunta do fluxograma que fecha aqui: "há projeto com escopo, cronograma e aceite?" → SIM.
- **Protocolo.** Aplica-se a Matriz Escopo × Entrega: reconstrói-se o escopo contratado (anexos), inspeciona-se o sistema entregue e cruzam-se os dois — 12 de 40 itens ausentes ou defeituosos, sendo 3 showstoppers.
- **Evidência Classe A.** Cronograma baseline e logs de produção preservados com hash sustentam objetivamente o que existe e o que não existe no ambiente.
- **Evidência Classe B.** Atas de homologação assinadas com ressalva corroboram que o contratante apontou as ausências no aceite — reforço documental datado.
- **Conclusão.** A entrega não atende ao escopo contratado; as ausências são atribuíveis ao fornecedor (não há mudança de escopo formalizada pelo contratante que as explique).
- **Grau de confiança** → **Alto.** Há evidência Classe A convergente de fontes independentes (cronograma + logs), a hipótese alternativa (scope creep do contratante) foi testada e afastada, e a análise é reproduzível. Todos os critérios de EV.3 para "Alto" estão presentes.

A **declaração de conformidade** que fecha o laudo deste exemplo: *"Caso enquadrado em LS-01.2; protocolo LS-01 cumprido; achados lastreados predominantemente em Classe A; conclusão com grau de confiança Alto, nos termos da norma EV da CLS v1.0."*

Glossário

Glossário oficial da CLS

Definições normativas — para que todos falem a mesma língua

Termos técnicos rendem interpretações divergentes em juízo. Estas são as definições oficiais da CLS v1.0: ao citar o padrão, adota-se este vocabulário. Termos em ordem alfabética.

Termo	Definição oficial
Boilerplate	Código repetitivo e padronizado, necessário mas sem originalidade — configurações iniciais, getters/setters, estrutura gerada por ferramenta. Excluído do corpus comparável na LS-03 por não expressar lógica proprietária.
Cadeia de custódia	Registro documentado e íntegro do percurso de uma evidência digital, da coleta ao laudo, que demonstra que o material não foi alterado. Apoia-se em imagem forense, hash, data, origem e responsável (ISO/IEC 27037).
Classe de evidência	Classificação (A a D) da força probatória de uma evidência conforme sua integridade e verificabilidade, definida na norma EV. Classe A é técnica e íntegra; D é declaratória.
Concurso de categorias	Situação em que um mesmo caso se enquadra em mais de uma categoria da CLS (ex.: output de IA que copia código protegido = LS-06 + LS-03). O laudo documenta o enquadramento múltiplo.
Corpus comparável	O conjunto de código que efetivamente entra na comparação da LS-03, depois de removido o trivial (framework, bibliotecas, código gerado, boilerplate). É onde reside a lógica proprietária.
Criação independente	Cenário em que semelhanças entre dois códigos se explicam por padrões de mercado, mesmo requisito funcional ou uso da mesma biblioteca — e não por cópia. É uma conclusão legítima da LS-03 (subtipo .3).
Derivação	Reprodução não-literal: a lógica, a estrutura e a sequência de um código são aproveitadas em outro, ainda que o texto tenha sido reescrito. Corresponde à obra derivada e ao subtipo LS-03.2.
Filtragem do trivial	Etapla da LS-03 que remove da comparação tudo que é igual por natureza (framework, bibliotecas de terceiros, scaffolding, boilerplate), com critério registrado, para não inflar artificialmente a similaridade.
Grau de confiança	Qualificação objetiva da conclusão do laudo (Alto, Médio ou Baixo) segundo os critérios da norma EV — convergência de evidências Classe A, teste de hipóteses alternativas e reprodutibilidade.
Importação em bloco	Introdução, no histórico de versionamento, de um grande volume de código de uma só vez (um único commit), incompatível com o desenvolvimento incremental. Forte indício de que o código veio pronto de outra origem.
Lógica proprietária	A parcela do software que expressa decisões autorais e regras de negócio específicas — algoritmos, fluxos, cálculos — por oposição ao código padrão. É o objeto central da análise de cópia (LS-03).
Matriz Escopo × Entrega	Quadro que cruza cada obrigação contratada (funcionalidade, integração, relatório) com o estado real da entrega (entregue, parcial, ausente, defeituoso). Produto principal da LS-01.
Norma EV	Norma transversal da CLS que define as classes de evidência (A–D) e os graus de confiança (Alto/Médio/Baixo), aplicável a todas as seis categorias.
Showstopper	Divergência ou defeito de tal gravidade que, sozinho, impede a operação do sistema ou o cumprimento do objeto contratado. Recebe destaque na Matriz Escopo × Entrega (LS-01).
Scope creep	Expansão informal e não formalizada do escopo ao longo do projeto, por acréscimos sucessivos de pedidos do contratante. Hipótese alternativa frequente na atribuição de responsabilidade da LS-01.
Trio (prompt, configuração, output)	O conjunto mínimo que deve ser preservado em litígios de IA generativa (LS-06): o comando dado, os parâmetros do modelo e o resultado, com data e hora — sem ele, o evento é irreproduzível.

LS-01

ERP e Sistemas Integrados

Implantação, escopo contratual e falha de entrega

Litígios de contratação, implantação, customização e manutenção de ERPs — go-live fracassado, escopo não entregue, estouro de prazo e SLA descumprido. O produto central da análise é a matriz escopo × entrega com responsabilidade atribuída.

Quando classificar um caso como LS-01

Use a LS-01 quando a disputa nasce de um **contrato de fornecimento, implantação, customização ou manutenção de sistema integrado de gestão** — ERP de prateleira (SAP, TOTVS, Oracle, Senior, Sankhya), vertical de mercado ou plataforma sob medida vendida como ERP. O núcleo da controvérsia é sempre o mesmo: *o que foi contratado versus o que foi entregue e quem causou a diferença*.

Critério de diferenciação: se a disputa é sobre quem é o dono do código, é LS-02; se é sobre cópia de código, é LS-03; se o sistema é consumido como serviço na nuvem sem projeto de implantação, é LS-05. A LS-01 pressupõe um *projeto* — com escopo, cronograma e aceite.

Tipologia (com critério de enquadramento)

- **LS-01.1 — Go-live fracassado.** O sistema entrou (ou tentou entrar) em produção e a operação falhou: dados migrados com erro, módulos inoperantes, integrações quebradas, paralisação da operação do contratante. *Enquadre aqui quando existir uma data de corte (go-live) e evidência de falha operacional após ela.*
- **LS-01.2 — Escopo divergente.** Funcionalidades da proposta/anexos não existem, existem pela metade ou funcionam diferente do especificado. *Enquadre aqui quando a controvérsia é sobre conteúdo da entrega, não sobre data.*
- **LS-01.3 — Prazo e orçamento estourados.** Cronograma e custo ultrapassados. *Enquadre aqui quando a entrega até existe, mas fora do tempo/custo pactuado — e a disputa é sobre a causa do desvio.*
- **LS-01.4 — Manutenção e SLA.** Contrato continuado com níveis de serviço (tempo de resposta, disponibilidade, prazo de correção) descumpridos de forma reiterada. *Enquadre aqui quando há medição periódica pactuada.*

Protocolo executável — 6 etapas

1. Etapa 1 — Congelamento e aquisição da base documental.

Objetivo: impedir que o material que prova o escopo se perca ou seja alterado.

Insumos: contrato e aditivos; proposta comercial; anexos técnicos e levantamento de requisitos (blueprint/BBP); cronograma baseline e replanejamentos; atas de reunião e de homologação; e-mails e mensagens de decisão; chamados de suporte.

Execução: coletar com hash (SHA-256) e timestamp; exportar sistemas de chamados (Jira, Movidesk, ServiceNow) em formato íntegro; registrar a origem de cada documento.

Entregável: inventário documental com hash — o "anexo zero" do laudo.

2. Etapa 2 — Reconstrução do escopo contratado.

Objetivo: transformar contrato + anexos em uma lista verificável de obrigações.

Execução: extrair cada funcionalidade, integração, relatório, migração e serviço prometido; atribuir a cada item um ID, a cláusula/anexo de origem e o critério de aceite (explícito ou implícito). Onde o contrato for omissivo, registrar a omissão — ela pesa na atribuição de responsabilidade.

Entregável: Tabela de Escopo Contratado (ID × descrição × origem contratual × critério de aceite).

3. Etapa 3 — Inspeção técnica do sistema entregue.

Objetivo: verificar, item a item, o estado real da entrega.

Insumos: acesso ao ambiente (produção e/ou homologação), massa de teste, usuários de perfil adequado.

Execução: testar cada item da Tabela de Escopo em roteiro documentado (vídeo/tela + log); coletar logs de aplicação, banco e integrações; para migração de dados, comparar contagens e amostras origem × destino.

Entregável: evidência datada por item (print/vídeo/log) com resultado: entregue · parcial · ausente · defeituoso.

4. Etapa 4 — Matriz escopo × entrega.

Objetivo: consolidar a divergência em um único quadro legível pelo juízo.

Execução: cruzar Etapa 2 × Etapa 3; quantificar (% de itens entregues, itens críticos ausentes); destacar os itens que impedem a operação (showstoppers).

Entregável: a Matriz de Divergências — peça central do laudo LS-01.

5. Etapa 5 — Atribuição de responsabilidade.

Objetivo: responder "quem causou cada divergência".

Execução: para cada item divergente, rastrear a causa nos documentos: requisito mal definido pelo contratante? mudança não formalizada (scope creep)? falha de execução do fornecedor? homologação assinada com ressalvas? falta de decisão do cliente registrada em ata? Classificar: fornecedor · contratante · concorrente (ambos) · indeterminado.

Entregável: coluna de responsabilidade na Matriz + memória de fundamentação por item.

6. Etapa 6 — Laudo/Parecer em linguagem processual.

Objetivo: traduzir a matriz para o juízo.

Execução: redigir conclusões amarrando cada divergência à cláusula e ao dispositivo legal; responder quesitos; anexar a matriz e o inventário com hash.

Entregável: laudo/parecer + anexos rastreáveis.

Checklist do advogado (pedir logo no início)

- Contrato, aditivos, proposta e TODOS os anexos técnicos (inclusive versões);
- Cronograma baseline + replanejamentos aprovados;
- Atas de homologação/aceite (com ou sem ressalvas);
- Exportação do sistema de chamados do projeto;
- Preservação imediata dos ambientes (produção/homologação) e dos backups da época do go-live.

Erros comuns que destroem o caso

- Deixar o ambiente ser atualizado/corrigido antes da perícia (a evidência do defeito desaparece);
- Confundir aceite formal com aceite tácito — homologação assinada sem ressalva pesa contra o contratante;
- Quesitos genéricos ("o sistema funciona?") que permitem resposta evasiva.

Base legal

Código Civil: arts. 389 (perdas e danos), 422 (boa-fé objetiva), 475 (resolução por inadimplemento), 610 e ss. (empreitada). CDC quando houver relação de consumo. CPC arts. 465–477 (perícia, quesitos e assistente técnico).

Quesitos-modelo

- "Queira o Sr. Perito verificar, para cada item do Anexo Técnico II, se a funcionalidade correspondente está implementada e operacional no ambiente de produção, classificando-a como entregue, parcial,

ausente ou defeituosa, com evidência datada."

- "Queira o Sr. Perito informar se os marcos do cronograma baseline foram cumpridos e, quanto aos descumpridos, se há registro documental (ata, e-mail, chamado) atribuindo a causa do atraso a decisão ou omissão do contratante."

LS-02

Direito Autoral e Titularidade de Software

Autoria, cessão, empregado/prestador e Lei 9.609/98

Disputas sobre quem criou e a quem pertence o software — empregado × empregador, prestador × contratante, sócios em dissolução e produtos com open source. O produto central é o Parecer de Titularidade: autoria técnica correlacionada à titularidade jurídica.

Quando classificar um caso como LS-02

Use a LS-02 quando a controvérsia central é **a quem pertence o software** — não se ele foi copiado (LS-03), nem se foi entregue conforme o contrato (LS-01). A pergunta-chave do enquadramento: "as partes disputam propriedade/autoria de um mesmo código?" Se sim, é LS-02.

Tipologia (com critério de enquadramento)

- **LS-02.1 — Empregado × empregador.** Código criado durante vínculo CLT. Regra do art. 4º da Lei 9.609/98: pertence ao empregador quando decorre do contrato de trabalho — mas a exceção (criação desvinculada, fora do expediente e sem recursos do empregador, §2º) é exatamente o que se prova tecnicamente.
- **LS-02.2 — Prestador (PJ) × contratante.** Sem cláusula expressa de cessão, a titularidade tende ao autor. A análise combina contrato + evidência técnica de quem efetivamente criou.
- **LS-02.3 — Sócios em dissolução.** Quem fica com o produto quando a sociedade acaba. Contrato social + aportes + histórico de desenvolvimento.
- **LS-02.4 — Open source incorporado.** O produto contém componentes GPL/MIT/Apache: a análise verifica conformidade com as licenças e o efeito sobre o conjunto (copyleft).

Protocolo executável — 5 etapas

1. Etapa 1 — Preservação dos repositórios.

Objetivo: garantir que o histórico de quem-fez-o-quê não seja reescrito.

Insumos: acesso aos repositórios (GitHub/GitLab/Bitbucket/SVN) ou cópia completa com histórico (git clone --mirror).

Execução: clonar com histórico integral; gerar hash do bundle; registrar data/hora e origem. Atenção: histórico Git pode ser reescrito (rebase, force-push) — coletar também dos servidores remotos e de máquinas locais quando possível.

Entregável: repositórios preservados com hash e cadeia de custódia.

2. Etapa 2 — Reconstrução da autoria técnica.

Objetivo: estabelecer quem escreveu o quê, quando.

Execução: análise do log (git log, git blame, git shortlog -sne): autores, e-mails, datas, volume e natureza das contribuições; identificação de commits importados em bloco (código trazido pronto de fora); correlação entre identidades (mesma pessoa com e-mails diferentes).

Entregável: Linha do Tempo de Autoria — quem contribuiu, em que período, com que peso.

3. Etapa 3 — Correlação com o vínculo jurídico.

Objetivo: cruzar a autoria técnica com o período e o objeto da relação jurídica.

Insumos: contrato de trabalho/prestação, CTPS, notas fiscais, registro de ponto quando existir.

Execução: sobrepor a Linha do Tempo de Autoria às datas do vínculo; verificar se os commits ocorreram em horário de expediente, com equipamento/infra do contratante (e-mail corporativo nos commits, servidores da empresa); examinar cláusulas de PI, cessão, exclusividade.

Entregável: quadro autoria × vínculo, apontando o que foi criado dentro e fora da relação.

4. Etapa 4 — Auditoria de licenças (quando houver open source).

Objetivo: mapear componentes de terceiros e seus efeitos.

Execução: varredura de dependências e licenças (SPDX; ferramentas como FOSSology/ScanCode); classificação por tipo (permissiva × copyleft); verificação de cumprimento das obrigações (avisos, disponibilização de fonte).

Entregável: BOM (bill of materials) de software com risco por componente.

5. Etapa 5 — Parecer de Titularidade.

Objetivo: unir tudo em uma conclusão jurídica-técnica utilizável.

Execução: para cada módulo relevante, concluir: autor técnico + enquadramento legal (art. 4º caput ou §2º; cessão válida ou ausente) = titular. Registrar limites e incertezas.

Entregável: Parecer de Titularidade com a matriz módulo × autor × fundamento × titular.

Checklist do advogado

- Acesso ou cópia espelho dos repositórios COM histórico (não só o código atual);
- Contratos (trabalho/prestação) com cláusulas de PI destacadas;
- Comprovantes do período do vínculo (CTPS, NFs, e-mails corporativos);
- Se houver risco de reescrita do histórico: pedido liminar de preservação.

Erros comuns

- Analisar só o código atual, sem o histórico — a autoria está no log, não no snapshot;
- Ignorar que git history pode ser adulterado — cruzar com fontes independentes (CI logs, code review, backups);
- Tratar cessão verbal como suficiente — a Lei 9.609/98 pede instrumento.

Base legal

Lei 9.609/98: arts. 1º–6º (proteção, autoria, titularidade; art. 4º e §2º para empregado/prestador). Lei 9.610/98 (subsidiária). CC arts. 104+ (validade da cessão). Art. 342 CP (falsa perícia).

Quesitos-modelo

- "Queira o Sr. Perito reconstituir, pelo histórico de versionamento, os autores e períodos de criação de cada módulo do sistema, indicando o volume e a natureza da contribuição de cada um."
- "Queira o Sr. Perito informar se os commits do Requerente ocorreram dentro do período do vínculo com a Requerida e se há elementos técnicos (e-mail corporativo, horário, infraestrutura) indicando uso de recursos da empresa."

LS-03

Código-Fonte — Cópia, Derivação e Criação Independente

Análise comparativa em 5 camadas, forense de código e prova de reprodução

O núcleo forense da CLS: método de 5 camadas para apurar — e não presumir — se dois códigos guardam relação de cópia, derivação ou criação independente, com filtragem do trivial e cadeia de custódia ISO/IEC 27037.



Comparação em 5 Camadas – a força probatória vem da convergência entre camadas.

Quando classificar um caso como LS-03

Use a LS-03 quando se alega que um software **reproduz outro**, no todo ou em parte — literalmente ou pela lógica. Pergunta de enquadramento: *"a controvérsia exige comparar dois códigos (ou um código contra uma versão anterior dele)?"* Se sim, é LS-03 — ainda que combinada com LS-02 (titularidade) ou com concorrência desleal.

Tipologia (com critério)

- **LS-03.1 — Cópia literal.** Trechos idênticos/quase idênticos além do explicável por framework, template ou geração automática.
- **LS-03.2 — Derivação (cópia não-literal).** Estrutura, sequência e organização reproduzidas com o texto reescrito — o "parafraseado" do código.
- **LS-03.3 — Criação independente.** Semelhanças explicáveis por padrões de mercado, mesmo requisito funcional, mesma biblioteca. *Também é uma conclusão possível e deve ser dita quando for o caso.*

Protocolo executável — 6 etapas

1. Etapa 1 — Aquisição forense dos dois lados.

Objetivo: congelar os artefatos comparandos com integridade demonstrável.

Execução: cópia integral (repositório com histórico, binários, banco, configs) de ambos os códigos; hash SHA-256 por arquivo e do conjunto; registro de data/hora, origem e responsável pela entrega; princípios da ISO/IEC 27037.

Entregável: pacote forense lacrado (hashes + termo de coleta) — sem ele, tudo o que vem depois é atacável.

2. Etapa 2 — Inventário e normalização.

Objetivo: saber exatamente o que será comparado.

Execução: listar arquivos por tipo/linguagem/tamanho; identificar versões de frameworks e dependências (package.json, pom.xml, requirements.txt, copybooks); normalizar formatação para não comparar diferenças cosméticas.

Entregável: inventário lado A × lado B.

3. Etapa 3 — Filtragem do trivial.

Objetivo: remover o que é igual "por natureza" para não inflar similaridade.

Execução: excluir da comparação: código de framework e bibliotecas de terceiros; código gerado (scaffolding, ORMs, protobuf); boilerplate notório. Registrar o critério de exclusão — a parte contrária vai perguntar.

Entregável: corpus comparável = a lógica proprietária de cada lado.

4. Etapa 4 — Análise em 5 camadas.

Objetivo: medir semelhança de formas independentes que se corroboram.

Execução: *Camada 1 · Léxica:* nomes de variáveis, funções, classes, tabelas, constantes. Nomes proprietários coincidentes (WS-CALC-JUROS-X, tbl_cliente_score_v2) valem mais que nomes genéricos (i, total, GetUser).

5. *Camada 2 · Estrutural:* árvore de diretórios, módulos, hierarquia de pacotes, assinatura de funções.

6. *Camada 3 · Semântica:* a lógica — fluxos de decisão, ordem de operações, algoritmos, regras de negócio. É a camada decisiva para derivação (LS-03.2).

7. *Camada 4 · Metadados:* comentários (inclusive erros de grafia repetidos), strings, mensagens de erro, TODOS, cabeçalhos de autoria.

8. *Camada 5 · Cronologia:* histórico de versionamento dos dois lados — quem existia primeiro; commits que importam código em bloco; datas incompatíveis com a versão da criação alegada.

9. Etapa 5 — Classificação e teste de hipóteses.

Objetivo: transformar achados em conclusão honesta.

Execução: para o conjunto de achados, testar as três hipóteses (cópia, derivação, independência): quais achados cada hipótese explica e quais ela não explica? A conclusão é o cenário que melhor explica TODOS os achados — nunca a tese do contratante por default.

Entregável: conclusão fundamentada com grau de convicção e limites declarados.

10. Etapa 6 — Laudo com demonstração lado a lado.

Objetivo: permitir que o juízo VEJA a similaridade.

Execução: quadros comparativos (trecho A × trecho B) dos achados mais fortes, com explicação em linguagem leiga do porquê aquela coincidência não é acaso; anexos técnicos completos para o contraditório.

Entregável: laudo/parecer + quadros comparativos + pacote forense.

Checklist do advogado

- Preservação IMEDIATA dos dois códigos (liminar/ata notarial se necessário) — código muda todo dia;
- Repositórios com histórico, não apenas o código compilado;
- Se só houver acesso ao binário do adversário: avaliar cabimento de busca e apreensão ou exibição judicial.

Erros comuns

- Rodar uma ferramenta de similaridade e tratar o percentual como conclusão — sem filtragem (Etapa 3) o número é lixo;
- Ignorar a hipótese de criação independente — laudo parcial cai na contradita;
- Comparar versões de épocas diferentes sem controlar a cronologia.

Base legal

Lei 9.609/98 (proteção do programa; independe de registro). Lei 9.610/98 (obra derivada). Art. 195 da Lei 9.279/96 quando combinada com concorrência desleal. ISO/IEC 27037 (evidência digital). Art. 342 CP.

Quesitos-modelo

- "Queira o Sr. Perito, após excluir código de frameworks, bibliotecas de terceiros e trechos gerados automaticamente, informar se os códigos apresentam, na camada de regras de negócio, similaridades que não se explicam por padrões de mercado — exemplificando com trechos lado a lado."
- "Queira o Sr. Perito informar se é possível determinar, pelos históricos de versionamento, qual dos códigos é anterior, e se há registros de importação de código em bloco compatíveis com transferência entre os projetos."

LS-04

APIs — Interoperabilidade e Contratos de Integração

Indisponibilidade, alteração unilateral, dados incorretos e responsabilidade em cadeia

Litígios de integração entre sistemas de empresas diferentes: SLA de API descumprido, mudança unilateral de contrato de interface, dados errados propagados e apuração de responsabilidade quando a falha atravessa vários sistemas.

Quando classificar um caso como LS-04

Use a LS-04 quando a disputa nasce da **fronteira entre dois sistemas de partes diferentes** — uma API, um webhook, um arquivo de intercâmbio. Pergunta de enquadramento: *"o dano decorre de algo que atravessou uma interface entre sistemas de responsáveis distintos?"* Se o problema é interno a um único fornecimento, provavelmente é LS-01.

Tipologia (com critério)

- **LS-04.1 — Indisponibilidade × SLA.** A API ficou fora do ar ou degradada além do pactuado. O litígio exige medir disponibilidade real e confrontar com a definição contratual de "indisponível".
- **LS-04.2 — Alteração unilateral da interface.** Endpoint, formato de resposta, autenticação ou rate limit mudaram sem o aviso pactuado, quebrando o consumidor.
- **LS-04.3 — Dados incorretos/incompletos.** A API respondeu, mas errado — e o sistema consumidor decidiu com base no erro.
- **LS-04.4 — Responsabilidade em cadeia.** Falha do provedor A quebra o sistema de B e causa dano ao cliente C. A perícia precisa localizar o elo causal.

Protocolo executável — 5 etapas

1. Etapa 1 — Congelamento do contrato de interface.

Objetivo: fixar o que a API prometia à época dos fatos.

Insumos: documentação da API (OpenAPI/Swagger, GraphQL schema, PDFs), changelogs, termos de uso, SLA contratual.

Execução: obter a versão da documentação vigente NA ÉPOCA (Wayback Machine, repositórios de docs, e-mails com anexos); comparar com a versão atual; registrar diffs.

Entregável: "contrato de interface" da época × atual, com as mudanças destacadas.

2. Etapa 2 — Coleta de logs dos dois lados.

Objetivo: reconstruir as chamadas reais.

Insumos: logs do consumidor (aplicação, gateway, APM) e, quando obtível, do provedor (por acordo ou ordem judicial).

Execução: extrair requisições do período: timestamp, endpoint, status code, latência, payload (com cuidado de dados pessoais/LGPD), headers de rate limit; preservar com hash.

Entregável: dataset de chamadas do período litigioso.

3. Etapa 3 — Reprodução controlada.

Objetivo: demonstrar o comportamento alegado de forma verificável.

Execução: reproduzir as chamadas problemáticas em ambiente controlado, documentando request + response com gravação; quando a API já mudou, demonstrar com os logs da Etapa 2 e o diff da Etapa 1.

Entregável: roteiro de reprodução com evidência audiovisual/log.

4. Etapa 4 — Medição de SLA e análise causal.

Objetivo: quantificar o descumprimento e localizar a causa.

Execução: calcular disponibilidade/latência real com base nos logs e na definição contratual (janelas de manutenção contam? erro 429 conta?); para a cadeia (LS-04.4), montar o diagrama de dependências e

posicionar a primeira falha; separar falha do provedor × implementação incorreta do consumidor (retry ausente, timeout mal configurado) × infraestrutura (DNS, rede).

Entregável: planilha de SLA + diagrama causal com o elo responsável.

5. Etapa 5 — Laudo/Parecer.

Execução: traduzir: "a interface prometia X (doc da época), passou a fazer Y (diff/logs), sem o aviso de Z dias (cláusula), e isso causou a parada do módulo W do contratante (logs correlacionados)".

Entregável: laudo com linha do tempo da falha e responsabilidade fundamentada.

Checklist do advogado

- Guardar a documentação da API da ÉPOCA (print, PDF, Wayback) — ela some;
- Preservar logs do próprio cliente imediatamente (retenção costuma ser de 30–90 dias!);
- Avaliar exibição judicial dos logs do provedor cedo — eles também expiram.

Erros comuns

- Deixar expirar a retenção de logs — o caso morre por falta de prova;
- Medir SLA sem a definição contratual de indisponibilidade;
- Ignorar a hipótese de erro do próprio consumidor (falta de retry/fallback) — a contradita vai explorá-la.

Base legal

CC arts. 389, 422, 475. Marco Civil (Lei 12.965/14) — guarda de registros. LGPD (Lei 13.709/18) quando os payloads contêm dados pessoais. Cláusulas de SLA, aviso prévio de breaking change e limitação de responsabilidade.

Quesitos-modelo

- "Queira o Sr. Perito calcular, com base nos logs preservados e na definição contratual de indisponibilidade, o percentual de disponibilidade da API no período X–Y, confrontando-o com o SLA pactuado."
- "Queira o Sr. Perito informar se houve alteração no formato de resposta do endpoint Z entre as datas A e B, se houve comunicação prévia conforme a cláusula N, e qual o efeito da alteração sobre o sistema do Requerente."

LS-05

SaaS — Software como Serviço e Contratos de Nuvem

Disponibilidade, perda de dados, lock-in, portabilidade e LGPD

Disputas do modelo por assinatura: serviço fora do ar, dados perdidos ou presos no provedor, funcionalidades removidas e descumprimento de LGPD. A particularidade: o contratante não tem acesso ao servidor — a preservação de prova muda completamente.

Quando classificar um caso como LS-05

Use a LS-05 quando o software é **consumido como serviço** — o cliente não possui instalação nem acesso à infraestrutura; acessa pela internet, mediante assinatura. Isso muda a perícia: quase toda a evidência está sob controle do provedor, e a preservação precisa ser imediata e, muitas vezes, judicial. Se houve projeto de implantação com escopo próprio, avalie concurso com LS-01.

Tipologia (com critério)

- **LS-05.1 — Indisponibilidade e perda de dados.** O serviço caiu além do SLA, ou dados sumiram/corromperam sem backup acessível ao cliente.
- **LS-05.2 — Lock-in e portabilidade.** O cliente quer sair e não consegue: exportação inexistente, formato proprietário, dados incompletos, prazos abusivos.
- **LS-05.3 — Alteração unilateral do serviço.** Funcionalidade removida/alterada, plano descontinuado, limite reduzido — sem consentimento ou aviso pactuado.
- **LS-05.4 — LGPD no SaaS.** O provedor (operador) descumpriu segurança, notificação de incidente ou eliminação ao término.

Protocolo executável — 5 etapas

1. Etapa 1 — Preservação do que o cliente alcança.

Objetivo: capturar a evidência acessível ANTES que o provedor mude qualquer coisa.

Execução: gravação de sessão navegando no serviço (tela + horário), com ata notarial quando o valor do caso justificar; exportações disponíveis (relatórios, CSVs); e-mails de status/incidente do provedor; página de status pública (e seu histórico via Wayback).

Entregável: dossiê de evidências do lado cliente, datado e íntegro.

2. Etapa 2 — Requisição formal/judicial do lado provedor.

Objetivo: obter o que só o provedor tem.

Execução: listar objetivamente o que pedir: logs de disponibilidade e de acesso, registros de backup e restauração, trilha de auditoria da conta, comunicações de incidente, DPA e registros do art. 37 da LGPD; fundamentar prazo (retenção expira).

Entregável: rol de exibição de documentos técnico e específico (genérico = negado).

3. Etapa 3 — Medição contra o SLA.

Objetivo: quantificar a indisponibilidade/perda.

Execução: cruzar monitoramento independente (quando o cliente tiver — UptimeRobot, Pingdom, healthchecks internos) com os logs do provedor e a definição contratual; para perda de dados, comparar último estado conhecido (exportações, relatórios, integrações) com o estado atual.

Entregável: planilha de disponibilidade e inventário do dado perdido.

4. Etapa 4 — Teste de portabilidade (LS-05.2).

Objetivo: demonstrar objetivamente se a saída é viável.

Execução: executar a exportação oferecida e auditar: formato (aberto ou proprietário?), completude (todos os campos? anexos? histórico?), integridade (contagens batem?), prazo e custo; documentar cada lacuna.

Entregável: relatório de portabilidade com a lista do que fica preso.

5. Etapa 5 — Conformidade LGPD e laudo.

Execução: verificar DPA, medidas de segurança declaradas × praticadas, notificação de incidente (art. 48), eliminação ao término (art. 16); consolidar tudo em laudo com linha do tempo.

Entregável: laudo/parecer com achados e responsabilidade.

Checklist do advogado

- Preservar HOJE: gravação de sessão, exportações, e-mails do provedor, print da página de status;
- Pedir exibição dos logs do provedor com urgência fundamentada em retenção;
- Rer o contrato: definição de indisponibilidade, limitação de responsabilidade, foro/arbitragem, prazo de devolução de dados no término.

Erros comuns

- Esperar meses para agir — logs do provedor expiram e a página de status muda;
- Rol de exibição genérico ("todos os logs") — especificar sistema, período e tipo;
- Ignorar o monitoramento próprio do cliente como fonte independente de medição.

Base legal

CDC (B2C) e CC (B2B). LGPD: arts. 16 (eliminação), 37 (registros), 46 (segurança), 48 (incidente). Marco Civil (Lei 12.965/14). Contrato de assinatura + SLA + DPA.

Quesitos-modelo

- "Queira o Sr. Perito, com base nos logs exibidos pelo provedor e no monitoramento do Requerente, calcular a disponibilidade do serviço no período X–Y conforme a definição contratual, indicando os eventos de indisponibilidade e sua duração."
- "Queira o Sr. Perito executar o procedimento de exportação de dados oferecido pelo provedor e informar se ele devolve a integralidade dos dados do Requerente, em formato aberto e íntegro, apontando lacunas."

LS-06

IA Generativa — Autoria, Responsabilidade e Viés

Autoria do output, dados de treinamento, decisão automatizada, viés e vazamento por prompt

A categoria emergente da CLS: disputas sobre conteúdo gerado por IA. Como os outputs são probabilísticos e mudam entre sessões, a preservação do trio prompt + configuração + output com timestamp é a espinha dorsal de qualquer caso.

Quando classificar um caso como LS-06

Use a LS-06 quando um **modelo generativo** (LLM, difusão de imagem, geração de código) está no centro da controvérsia — como criador do conteúdo disputado, como causador do dano, ou como veículo de vazamento. Atenção ao concurso de categorias: output de IA que copia código protegido combina LS-06 + LS-03; funcionário que cola segredo no ChatGPT combina LS-06 + concorrência desleal.

Princípio reitor da LS-06: IA generativa é *probabilística* — o mesmo prompt pode dar outputs diferentes. Sem preservar o trio **prompt + configuração + output** com data/hora, o caso se torna indemonstrável.

Tipologia (com critério)

- **LS-06.1 — Autoria do output.** Quem é titular do que a IA gerou? A análise técnica mede o grau de contribuição humana (prompts iterativos, curadoria, edição) versus geração automática.
- **LS-06.2 — Dados de treinamento protegidos.** O modelo reproduz substancialmente conteúdo protegido presente no treinamento? Exame de regurgitação e similaridade output × fonte.
- **LS-06.3 — Decisão automatizada danosa.** Crédito negado, triagem errada, diagnóstico falho: onde no pipeline o erro nasceu (dado de entrada, modelo, threshold, integração)?
- **LS-06.4 — Viés algorítmico.** Tratamento desigual mensurável entre grupos. Auditoria de fairness com inputs controlados.
- **LS-06.5 — Vazamento por prompt.** Dados confidenciais ou pessoais inseridos em ferramentas de IA por colaboradores.

Protocolo executável — 6 etapas

1. Etapa 1 — Preservação do trio prompt/configuração/output.

Objetivo: congelar o fato antes que ele se torne irreproduzível.

Execução: exportar o histórico da conversa/geração (a maioria das plataformas permite); capturar modelo e versão, parâmetros (temperature, top-p, system prompt quando acessível), data/hora e conta; gravar a tela da sessão; hash de tudo.

Entregável: pacote de evidência do evento generativo.

2. Etapa 2 — Identificação do sistema.

Objetivo: saber exatamente O QUE gerou o conteúdo.

Execução: distinguir o modelo base (GPT-x, Claude, Llama), a aplicação sobre ele (chat corporativo, copiloto, SaaS de terceiro), e camadas intermediárias (RAG, fine-tuning, agentes); obter dos fornecedores a documentação da versão vigente na época (model cards, changelogs).

Entregável: ficha técnica do sistema na data dos fatos.

3. Etapa 3 — Reprodução controlada.

Objetivo: testar se o comportamento se repete e em que condições.

Execução: repetir o prompt na mesma versão (quando ainda disponível) N vezes, variando temperatura, documentando a distribuição de outputs; declarar honestamente no laudo o caráter estocástico e o que a reprodução demonstra (tendência, não determinismo).

Entregável: protocolo de reprodução com resultados tabulados.

4. Etapa 4 — Análise de conteúdo (por subtipo).

Execução: LS-06.1: reconstruir a cadeia de prompts e edições humanas; medir a distância entre o output bruto e o resultado final publicado;

5. LS-06.2: comparar o output com a obra supostamente regurgitada usando o método de camadas da LS-03 (léxica/estrutural/semântica);

6. LS-06.3: mapear o pipeline de decisão (input → features → modelo → threshold → ação) e posicionar o erro com dados do caso concreto;

7. LS-06.4: bateria de inputs controlados variando apenas o atributo protegido (nome, gênero, CEP), com métricas de disparidade documentadas;

8. LS-06.5: exame de logs da ferramenta, histórico de prompts e políticas internas; verificação do que o fornecedor declara reter/treinar.

9. Etapa 5 — Cadeia de responsabilidade.

Objetivo: distribuir tecnicamente a causa entre os atores.

Execução: quadro desenvolvedor do modelo × integrador/aplicação × operador/usuário: quem definiu o system prompt? quem calibrou o threshold? quem inseriu o dado? o uso violou os termos da ferramenta?

Entregável: matriz de responsabilidade técnica.

10. Etapa 6 — Laudo com limites declarados.

Execução: conclusões proporcionais à natureza probabilística; distinguir o que é demonstrado (logs, histórico) do que é inferido (tendência estatística); traduzir para o juízo sem antropomorfizar ("o modelo tende a", nunca "o modelo quis").

Entregável: laudo/parecer com metodologia reproduzível e incertezas explícitas.

Checklist do advogado

- Exportar/preservar JÁ o histórico da conversa/geração — contas são apagadas e modelos são descontinuados;
- Identificar a versão do modelo na data do fato (changelogs, faturas, contratos);
- Em decisão automatizada: requerer os logs de input/output do caso concreto e a documentação do modelo (LGPD art. 20 — revisão de decisão automatizada).

Erros comuns

- Tentar reproduzir meses depois em versão diferente do modelo e tratar o resultado como prova do passado;
- Laudo que antropomorfiza ("a IA decidiu enganar") — cai na contradita e desmoraliza a prova;
- Ignorar as camadas intermediárias (RAG, system prompt do integrador) e atribuir tudo ao modelo base.

Base legal (em formação)

Leis 9.609/98 e 9.610/98 (autoralidade — aplicação a outputs de IA em debate). LGPD: art. 20 (revisão de decisão automatizada), arts. 37/46/48. CDC (defeito de serviço com decisão automatizada). Marco Civil. PL 2338/2023 (Marco Legal da IA, em tramitação) e diretrizes da ANPD.

Quesitos-modelo

- "Queira o Sr. Perito informar qual modelo, versão e configuração geraram o conteúdo objeto da lide, e se o histórico preservado (prompt, parâmetros e output) permite reconstituir o evento com data e hora."
- "Queira o Sr. Perito, mediante bateria de testes controlados variando exclusivamente o atributo X, informar se o sistema produz tratamento estatisticamente distinto entre os grupos, quantificando a disparidade."

Sobre esta edição

CLS Standard 1.0 — publicação inicial, 2026. Seis categorias (LS-01 a LS-06), norma de evidência (EV), princípios, glossário oficial e modelo de aplicação.

Licença. CC BY 4.0: uso livre, inclusive comercial, mediante atribuição de autor, título e versão. **Ressalva.** Padrão de análise técnica; não substitui o exame do caso concreto nem constitui parecer jurídico.

CLS — Classification of Software Litigation · versão 1.0 · 2026
Cepheus Consultoria Ltda · CNPJ 07.037.589/0001-41 · contato@cepheus.com.br · (11) 99383-0797
cepheus.com.br/metodologia · CC BY 4.0