DISSERTAÇÃO DE MESTRADO PROFISSIONAL

BlockGuard: Uma Arquitetura de Segurança para Conformidade LGPD em Redes *Hyperledger Fabric*

João Paulo da Costa e Silva Garcia

Programa de Pós-Graduação Profissional em Engenharia Elétrica

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO PROFISSIONAL

BlockGuard: Uma Arquitetura de Segurança para Conformidade LGPD em Redes *Hyperledger Fabric*

João Paulo da Costa e Silva Garcia

Dissertação de Mestrado Profissional submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Prof. Georges Daniel Amvame Nze, Ph.D, FT/UnB _____
Orientador

Prof. Fábio Lúcio Lopes de Mendonça, Ph.D, FT/UnB _____
Examinador Interno

Prof. Felipe Ferré , Ph.D, CONASS _____
Examinador externo

FICHA CATALOGRÁFICA

GARCIA, JOÃO PAULO
BlockGuard: Uma Arquitetura de Segurança para Conformidade LGPD em Redes *Hyperledger Fabric* [Distrito Federal] 2025.
xvi, 48 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2025).
Dissertação de Mestrado Profissional - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Blockchain	2. Hyperledger Fabric
3. LGPD	4. Conformidade
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

GARCIA, JOÃO PAULO. (2025). *BlockGuard: Uma Arquitetura de Segurança para Conformidade LGPD em Redes Hyperledger Fabric*. Dissertação de Mestrado Profissional, Publicação: PPEE.MP.102. Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 48 p.

CESSÃO DE DIREITOS

AUTOR: João Paulo da Costa e Silva Garcia
TÍTULO: BlockGuard: Uma Arquitetura de Segurança para Conformidade LGPD em Redes *Hyperledger Fabric*.
GRAU: Mestre em Engenharia Elétrica ANO: 2025

João Paulo da Costa e Silva Garcia
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP 70919-970 - Brasília - DF - Brasil

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, pela força, sabedoria e proteção concedidas ao longo de toda esta jornada. Sem Sua graça, nada disso seria possível. À minha amada esposa, Camille, pelo amor incondicional, paciência e incentivo constante. Você foi meu porto seguro nos momentos de dificuldade. Aos meus pais, Carlos e Lara, que sempre acreditaram em mim e me ensinaram o valor da educação e da perseverança. Ao meu orientador, Professor Daniel Amvame, pela orientação precisa, conhecimento compartilhado e confiança depositada em meu trabalho. E a todos os meus amigos que, de alguma forma, contribuíram para que eu chegasse até aqui.

RESUMO

A crescente adoção de tecnologias de registro distribuído (DLT) em setores regulados enfrenta um desafio crítico diante da entrada em vigor da Lei Geral de Proteção de Dados (LGPD): a tensão aparente entre a imutabilidade inerente à *blockchain*, que garante a integridade histórica das transações, e os direitos fundamentais dos titulares, especificamente os direitos de retificação e eliminação (direito ao esquecimento) previstos nos artigos 16 e 18 da legislação. Esta dissertação investiga esse paradoxo e propõe o BlockGuard, um *framework* arquitetural baseado na plataforma permissionada *Hyperledger Fabric*, desenhado para reconciliar as garantias de segurança da *blockchain* com os requisitos de privacidade e governança de dados. Utilizando a metodologia *Design Science Research* (DSR), a pesquisa desenvolveu e validou um modelo de armazenamento híbrido que segrega os dados pessoais em repositórios *off-chain* (PostgreSQL), mantendo no *ledger* distribuído apenas provas criptográficas de integridade (*hashes* SHA-256) e metadados de auditoria. A arquitetura explora recursos nativos avançados do *Hyperledger Fabric*, como *Private Data Collections* (PDCs) e canais privados, para implementar o princípio da minimização e o controle de acesso granular. A lógica de negócios e a conformidade regulatória foram codificadas em contratos inteligentes (*chaincodes* em Go) que automatizam a gestão do ciclo de vida dos dados, o registro de consentimento e a resposta a incidentes. A viabilidade da proposta foi demonstrada através de um estudo de caso simulado em um consórcio de saúde digital, onde a solução comprovou ser capaz de detectar adulterações em dados médicos (propriedade de *tamper-evidence*), garantir o sigilo de informações sensíveis e operacionalizar a exclusão física de dados *off-chain* mantendo a rastreabilidade imutável das ações (*accountability*). Os resultados indicam que a abordagem híbrida viabiliza o uso de *blockchain* em conformidade com a LGPD, oferecendo uma alternativa superior às arquiteturas centralizadas tradicionais em termos de auditabilidade e confiança distribuída.

Palavras-chave: Blockchain Permissionada. LGPD. Hyperledger Fabric. Privacidade de Dados. Arquitetura Híbrida. Direito ao Esquecimento.

Palavras-chave: *Blockchain*. *Hyperledger Fabric*. LGPD. Proteção de Dados. Conformidade Regulatória.

ABSTRACT

The increasing adoption of Distributed Ledger Technologies (DLT) in regulated sectors faces a critical challenge following the enactment of the General Data Protection Law (LGPD): the apparent tension between the inherent immutability of blockchain, which ensures the historical integrity of transactions, and the fundamental rights of data subjects, specifically the rights to rectification and erasure (the right to be forgotten) provided for in articles 16 and 18 of the legislation. This dissertation investigates this paradox and proposes BlockGuard, an architectural framework based on the Hyperledger Fabric permissioned platform, designed to reconcile blockchain security guarantees with privacy and data governance

requirements. Adopting the Design Science Research (DSR) methodology, the research developed and validated a hybrid storage model that segregates personal data into off-chain repositories (PostgreSQL), maintaining only cryptographic integrity proofs (SHA-256 hashes) and audit metadata on the distributed ledger. The architecture leverages advanced native features of Hyperledger Fabric, such as Private Data Collections (PDCs) and private channels, to implement the principle of minimization and granular access control. Business logic and regulatory compliance were encoded into smart contracts (chaincodes in Go) that automate data lifecycle management, consent registration, and incident response. The feasibility of the proposal was demonstrated through a simulated case study in a digital health consortium, where the solution proved capable of detecting tampering with medical data (tamper-evidence property), ensuring the confidentiality of sensitive information, and operationalizing the physical deletion of off-chain data while preserving the immutable traceability of actions (accountability). The results indicate that the hybrid approach enables the use of blockchain in compliance with the LGPD, offering a superior alternative to traditional centralized architectures in terms of auditability and distributed trust.

Keywords: Permissioned Blockchain. LGPD. Hyperledger Fabric. Data Privacy. Hybrid Architecture. Right to be Forgotten.

SUMÁRIO

1 INTRODUÇÃO 1
1.1 CONTEXTUALIZAÇÃO E PROBLEMA DE PESQUISA 1
1.2 JUSTIFICATIVA 2
1.3 OBJETIVOS 2
1.3.1 Objetivo Geral 2
1.3.2 Objetivos Específicos 2
1.4 METODOLOGIA DE PESQUISA 3
1.5 ESTRUTURA DA DISSERTAÇÃO 3

2 REFERENCIAL TEÓRICO 4
2.1 Lei Geral de Proteção de Dados (LGPD) 4
2.1.1 O Paradoxo entre o Direito ao Esquecimento e a Imutabilidade 4
2.2 Fundamentos de *Blockchain* 5
2.2.1 Definição e Características Essenciais 5
2.2.2 Tecnologias para Aprimoramento da Privacidade (PETs) 5
2.3 Hyperledger Fabric 6
2.3.1 *Smart Contracts* como Ferramenta de Conformidade 6
2.3.2 Modelos de Governança em Redes Permissionadas 6
2.4 Tópicos Avançados e Desafios de Implementação 7
2.4.1 Identidade Auto-Soberana (*Self-Sovereign Identity* - SSI) 7
2.4.2 Interoperabilidade entre Redes e Sistemas Legados 7
2.4.3 Modelos de Consórcio e Implicações Organizacionais 7

3 TRABALHOS CORRELATOS 8
3.1 *Frameworks* para Conformidade Regulatória em *Blockchain* 8
3.2 Soluções Técnicas para Desafios Específicos de Privacidade 8
3.2.1 O Direito ao Esquecimento e a Imutabilidade 8
3.2.2 Aplicação de Provas de Conhecimento Zero (ZKP) 8
3.2.3 Técnicas de Anonimização e Pseudonimização 9
3.3 Aplicações de *Hyperledger Fabric* em Contextos Regulados 9
3.4 LGPD e Tecnologias Emergentes no Contexto Brasileiro 9
3.5 Síntese da Análise e Lacuna de Pesquisa 10

4 METODOLOGIA 11
4.1 Caracterização da Pesquisa 11
4.2 Estratégia de Pesquisa: *Design Science Research* 11
4.3 Procedimentos Metodológicos 11
4.3.1 Fase I: Revisão Integrada da Literatura e Documentação 11

4.3.2 Fase II: Desenvolvimento do *Framework* BlockGuard 12
4.3.3 Fase III: Implementação e Estudo de Caso Integrado 12
4.3.4 Fase IV: Análise e Validação 12
4.4 Critérios de Validação e Qualidade 12
4.5 Aspectos Éticos 12

5 O *FRAMEWORK* BLOCKGUARD: FUNDAMENTAÇÃO TEÓRICA E ESPECIFICAÇÃO CONCEITUAL 13
5.1 Arquitetura Conceitual e Modelo de Camadas 13
5.1.1 Modelo de Dados Híbrido: *On-Chain* vs *Off-Chain* 14
5.2 Domínio 1: Gestão do Ciclo de Vida de Dados Pessoais 15
5.3 Domínio 2: Consentimento e Bases Legais 16
5.4 Domínio 3: Direitos dos Titulares 17
5.5 Domínio 4: Segurança e Governança 17
5.6 Domínio 5: Auditoria, Documentação e *Accountability* 18

6 IMPLEMENTAÇÃO E VALIDAÇÃO DO FRAMEWORK BLOCKGUARD 19
6.1 ESPECIFICAÇÕES DO AMBIENTE EXPERIMENTAL 19
6.1.1 Topologia da Rede e Configuração do Consórcio 20
6.2 ARQUITETURA IMPLEMENTADA: DA CONCEPÇÃO À MATERIALIZAÇÃO 20
6.3 IMPLEMENTAÇÃO DOS CINCO DOMÍNIOS DE CONFORMIDADE 20
6.3.1 Domínio 1: Gestão do Ciclo de Vida e Modelo Híbrido de Armazenamento 21
6.3.2 Domínio 2: Gestão de Consentimento com *Private Data Collections* 25
6.3.3 Domínio 3: Operacionalização dos Direitos dos Titulares 28
6.3.4 Domínio 4: Governança, Papéis Organizacionais e Gestão de Incidentes 30
6.3.5 Domínio 5: Auditoria e *Accountability* 30
6.4 INTERFACE DE GERENCIAMENTO: DASHBOARD WEB 32

7 ANÁLISE E DISCUSSÃO DOS RESULTADOS 34
7.1 Validação da Hipótese Central de Pesquisa 34
7.1.1 Análise Crítica da Abordagem Híbrida e Riscos Associados 34
7.1.2 Eficácia e Desafios das Private Data Collections (PDCs) 34
7.1.3 Operacionalização dos Direitos dos Titulares 35
7.2 Abrangência de Cobertura dos Requisitos LGPD 35
7.3 Análise Comparativa com Abordagens Tradicionais 36
7.4 Contribuições e Trabalhos Futuros 37
7.5 Considerações Finais 37

REFERÊNCIAS BIBLIOGRÁFICAS 38

8 APÊNDICES 42
8.1 Domínio 1: Gestão do Ciclo de Vida e Classificação 42

8.2 Domínio 2: Gestão de Consentimento e Privacidade 43
8.3 Domínio 3: Máquina de Estados (Direitos dos Titulares) 45
8.4 Domínio 4: Governança e Incidentes 46

9 CONFIGURAÇÕES DE INFRAESTRUTURA 48
9.1 Políticas de Coleção de Dados Privados (PDC) 48

LISTA DE FIGURAS

5.1 *Framework* BlockGuard e seus cinco domínios de conformidade, demonstrando a relação entre os requisitos legais da LGPD e os componentes tecnológicos implementados na plataforma *Hyperledger Fabric*. Fonte: Autor. 14

6.1 Arquitetura geral implementada do BlockGuard, demonstrando a integração entre *dashboard* React, API Node.js, rede Hyperledger Fabric (com cinco *chaincodes* especializados) e banco de dados PostgreSQL *off-chain*. 20
6.2 Diagrama de sequência do processo de criação de dados com modelo híbrido, demonstrando a interação coordenada entre *dashboard*, API REST, *blockchain* Hyperledger Fabric e banco de dados PostgreSQL *off-chain*. 22
6.3 Interface do BlockGuard exibindo a arquitetura híbrida: conteúdo completo *off-chain* e seu *hash* SHA-256 imutável *on-chain*. 23
6.4 Resultado da verificação de integridade em cenário de conformidade: *hashes* idênticos confirmam autenticidade do dado. 24
6.5 Detecção automática de adulteração: divergência entre os *hashes* evidencia modificação não autorizada. 24
6.6 Interface de gerenciamento do BlockGuard exibindo a configuração das PDCs e suas respectivas políticas de acesso. 25
6.7 Diagrama de sequência do registro de consentimento com segregação automática de dados públicos e privados. 26
6.8 Evidência visual de acesso autorizado: usuário da Org1MSP recupera dados privados. 27
6.9 Evidência visual de acesso bloqueado: tentativa da Org3MSP é rejeitada pela plataforma. ... 27
6.10 Ciclo de vida completo de uma solicitação de direito do titular e as respectivas atualizações de estado no *ledger* imutável. 29
6.11 Dashboard de Governança exibindo métricas de incidentes e trilhas de auditoria. 30
6.12 Diagrama de sequência do registro automático de auditoria via interceptador de *middleware*. 31
6.13 Fluxo de consulta da trilha de auditoria, garantindo rastreabilidade total. 32
6.14 Tela principal do *dashboard* BlockGuard com resumo executivo de conformidade. 33

LISTA DE TABELAS

7.1 Síntese da Cobertura dos Princípios Fundamentais (Art. 6°). 35
7.2 Abrangência do BlockGuard: Bases Legais e Direitos dos Titulares. 36
7.3 Abrangência do BlockGuard: Segurança e Governança. 36

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E PROBLEMA DE PESQUISA

A sociedade contemporânea atravessa uma profunda transformação digital, na qual os dados assumiram o papel de ativo estratégico fundamental para governos e empresas. Neste cenário, tecnologias de Registro Distribuído (*Distributed Ledger Technology* - DLT), com destaque para a *blockchain*, emergiram como soluções disruptivas capazes de garantir descentralização, integridade, transparência e auditabilidade em transações digitais (1). Contudo, a adoção massiva dessas tecnologias em ambientes corporativos e governamentais enfrenta um desafio crítico: a conformidade com um ecossistema regulatório cada vez mais complexo, rigoroso e, por vezes, aparentemente contraditório.

No Brasil, o cenário jurídico de governança de dados não se resume isoladamente à Lei Geral de Proteção de Dados (LGPD). Ele é composto por um arcabouço multifacetado que inclui a Constituição Federal — que recentemente elevou a proteção de dados a direito fundamental (EC nº 115/2022) —, o Marco Civil da Internet (Lei nº 12.965/2014) e a Lei de Acesso à Informação (LAI - Lei nº 12.527/2011) (2, 3, 4).

Observa-se, portanto, uma tensão regulatória latente. De um lado, a LAI impõe ao setor público o dever de transparência ativa e a publicidade dos atos administrativos, princípios que se alinham naturalmente às características de uma *blockchain*, que funciona como um livro-razão imutável e auditável. De outro lado, a LGPD (Lei nº 13.709/2018) estabelece freios necessários ao tratamento de dados, garantindo ao cidadão o controle sobre suas informações pessoais, incluindo direitos severos de exclusão e anonimização (5).

O problema de pesquisa central desta dissertação reside no conflito técnico-jurídico entre a arquitetura imutável da *blockchain* e os direitos de privacidade assegurados por esse arcabouço legal. Por sua concepção arquitetural (*security by design*), uma *blockchain* garante que registros históricos jamais sejam alterados ou apagados, sendo essa a base de sua confiabilidade (6). Entretanto, o Artigo 18 da LGPD assegura ao titular o direito à revogação do consentimento e à eliminação de dados desnecessários ou excessivos, o que a doutrina convencionou chamar de "direito ao esquecimento"(7, 8).

Diante desse paradoxo, emerge a seguinte questão de pesquisa: Como arquitetar uma solução baseada em *blockchain* permissionada (*Hyperledger Fabric*) que preserve as garantias de integridade, auditoria e transparência — atendendo aos princípios da LAI — sem violar os direitos fundamentais de exclusão e privacidade assegurados pela LGPD e pela Constituição Federal?

A persistência de dados pessoais imutáveis na cadeia principal (*on-chain*) constitui uma violação direta da legislação. Em contrapartida, armazenar dados inteiramente fora da cadeia (*off-chain*) sem mecanismos criptográficos robustos de vínculo reintroduz os problemas de confiança e opacidade que a tecnologia visa resolver. Portanto, faz-se imperativo o desenvolvimento de arquiteturas híbridas e mecanismos de "conformidade como código"(*compliance-as-code*) que permitam o cumprimento simultâneo dessas obrigações legais distintas.

1.2 JUSTIFICATIVA

A relevância desta investigação manifesta-se em três dimensões complementares: a urgência regulatória, a lacuna técnico-acadêmica e o impacto social.

No âmbito regulatório e social, o Brasil vivencia um momento de amadurecimento institucional onde a transparência pública não pode se sobrepor à privacidade do cidadão. O estudo justifica-se pela necessidade de fornecer ao Estado e às empresas ferramentas que permitam utilizar a *blockchain* para aumentar a confiança pública (*accountability*) e combater fraudes, sem transformar essa tecnologia em um instrumento de vigilância perene ou de violação de direitos fundamentais (9). A solução proposta busca harmonizar os princípios da LAI com as garantias da LGPD.

Do ponto de vista técnico e acadêmico, existe uma carência de literatura que vá além da discussão teórica sobre o conflito entre GDPR e *Blockchain*. Poucos trabalhos apresentam implementações práticas utilizando a versão mais recente do *Hyperledger Fabric* (v2.x) aplicadas ao contexto da lei brasileira. Esta versão introduziu recursos vitais, como as Coleções de Dados Privados (*Private Data Collections* - PDCs), que ainda são subexploradas em propostas de arquitetura. O artefato desenvolvido, denominado *BlockGuard*, preenche essa lacuna ao oferecer um modelo de referência reutilizável e validado.

Na dimensão prática, organizações que lidam com dados sensíveis — como no setor de saúde ou financeiro — enfrentam riscos jurídicos elevados. O desenvolvimento de um *framework* que sistematize a conformidade reduz barreiras de entrada para a adoção da tecnologia, oferecendo um caminho seguro para a inovação digital no setor público e privado.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver e validar o *BlockGuard*, um *framework* para redes *Hyperledger Fabric* que sistematize a implementação técnica de requisitos da Lei Geral de Proteção de Dados (LGPD) e leis correlatas, através de uma arquitetura híbrida e contratos inteligentes, garantindo a coexistência entre a imutabilidade do registro e os direitos dos titulares.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral, definem-se os seguintes objetivos específicos:

1. Mapear os princípios da LGPD, correlacionando-os com os mecanismos de privacidade nativos do *Hyperledger Fabric* (canais, PDCs, políticas de endosso);
2. Definir uma arquitetura de referência que segmente o armazenamento de dados (*On-Chain* vs *Off-Chain*) para viabilizar a exclusão física de dados pessoais mantendo a prova de integridade;
3. Implementar *chaincodes* (contratos inteligentes) que automatizem a gestão do ciclo de vida dos

dados, incluindo a gestão granular de consentimento e sua revogação;

4. Validar o *framework* proposto através de um estudo de caso prático, avaliando a eficácia dos mecanismos de privacidade, a integridade dos dados e o desempenho da solução.

1.4 METODOLOGIA DE PESQUISA

Esta pesquisa caracteriza-se como aplicada, qualitativa e exploratória. A estratégia metodológica adotada é a *Design Science Research* (DSR), um método rigoroso voltado para a resolução de problemas práticos através da construção e avaliação de artefatos inovadores (10).

O ciclo de pesquisa seguiu quatro etapas iterativas: (1) **Identificação do Problema**, com a análise das antinomias entre LGPD, LAI e a imutabilidade da *blockchain*; (2) **Design e Desenvolvimento**, que consistiu na concepção da arquitetura híbrida *BlockGuard* e seus algoritmos de verificação; (3) **Implementação**, envolvendo a codificação dos *chaincodes* em linguagem Go e a orquestração da rede em contêineres Docker; e (4) **Avaliação**, realizada através de um estudo de caso em cenário de saúde, utilizando dados sintéticos para validar as hipóteses de integridade e privacidade.

1.5 ESTRUTURA DA DISSERTAÇÃO

A dissertação está organizada em cinco capítulos, estruturados da seguinte forma:

- **Capítulo 1 - Introdução:** Apresenta a contextualização, o problema de pesquisa envolvendo o conflito legal, os objetivos e a justificativa do estudo.
- **Capítulo 2 - Fundamentação Teórica:** Estabelece as bases conceituais sobre o ecossistema legal brasileiro (LGPD, LAI, Marco Civil), a tecnologia *blockchain* e o estado da arte dos trabalhos correlatos.
- **Capítulo 3 - Proposta de Arquitetura:** Detalha a concepção do *framework BlockGuard*, descrevendo o modelo híbrido, os domínios de conformidade e as decisões de *design*.
- **Capítulo 4 - Análise e Resultados:** Descreve a implementação técnica, o protocolo de validação e discute criticamente os resultados obtidos no estudo de caso.
- **Capítulo 5 - Conclusão:** Sintetiza as contribuições acadêmicas e práticas, discute as limitações e aponta recomendações para trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 LEI GERAL DE PROTEÇÃO DE DADOS (LGPD)

A Lei nº 13.709/2018, conhecida como Lei Geral de Proteção de Dados Pessoais (LGPD), representa um marco na legislação brasileira, estabelecendo diretrizes abrangentes para o tratamento de dados. Sancionada em 2018, a LGPD foi inspirada no GDPR europeu, com adaptações ao contexto nacional (11). A lei surge não apenas como resposta à necessidade de proteger direitos fundamentais de privacidade, mas também como um elemento essencial para a inserção competitiva do Brasil na economia digital global (12). A ausência de um marco regulatório robusto era vista como uma barreira para negócios internacionais (13).

A LGPD estabelece dez princípios fundamentais em seu artigo 6º, incluindo finalidade, necessidade, transparência e segurança. Estes princípios não são apenas diretrizes, mas requisitos operacionais que demandam implementação técnica em sistemas de informação (14). O princípio da finalidade, por exemplo, exige que sistemas documentem e vinculem cada operação a propósitos específicos, uma funcionalidade para a qual a tecnologia *blockchain* se mostra particularmente adequada devido à sua capacidade de criar registros imutáveis e auditáveis (15).

2.1.1 O Paradoxo entre o Direito ao Esquecimento e a Imutabilidade

Um dos desafios mais debatidos na intersecção entre *blockchain* e as leis de proteção de dados é o aparente conflito entre a imutabilidade dos registros e o "direito ao esquecimento"(ou direito à eliminação), garantido pelo Art. 18, VI, da LGPD e pelo Art. 17 do GDPR (8). A imutabilidade, característica central que garante a integridade e a auditabilidade da *blockchain*, implica que dados, uma vez registrados, não podem ser alterados ou removidos. Isso conflita diretamente com o direito do titular de solicitar a exclusão de seus dados.

A literatura acadêmica propõe diversas abordagens para reconciliar este paradoxo. Uma das estratégias mais proeminentes é o armazenamento *off-chain* (16). Nesta arquitetura, os dados pessoais são armazenados em um banco de dados tradicional (fora da cadeia), enquanto a *blockchain* armazena apenas um *hash* criptográfico (uma representação única e compacta) desses dados. Quando a exclusão é solicitada, os dados são removidos do sistema *off-chain*, o que invalida permanentemente a utilidade do *hash* registrado na cadeia, mas preserva a integridade da sequência de blocos.

Outras soluções incluem a eliminação criptográfica, onde os dados na *blockchain* são criptografados com uma chave que pode ser destruída, tornando-os ilegíveis, e o desenvolvimento de *blockchains* redigíveis (*redactable blockchains*), que utilizam mecanismos como *chameleon hashes* para permitir a modificação de blocos sob condições de governança estritas, embora esta seja uma área de pesquisa ainda em desenvolvimento (17).

2.2 FUNDAMENTOS DE *BLOCKCHAIN*

2.2.1 Definição e Características Essenciais

Blockchain é uma estrutura de dados que funciona como um livro-razão distribuído, imutável e compartilhado, mantendo um registro crescente de transações organizadas em blocos criptograficamente encadeados (6). Suas características fundamentais incluem descentralização, persistência, pseudonimato e auditabilidade, que, juntas, permitem que participantes que não confiam uns nos outros cheguem a um acordo sobre o estado de um sistema (1).

2.2.1.1 Taxonomia de *Blockchains*

Os sistemas baseados em *blockchain* podem ser classificados segundo seu modelo de permissionamento (18):

- ***Blockchains* Públicas (*Permissionless*):** A participação é aberta a qualquer pessoa. O consenso é geralmente alcançado por algoritmos como *Proof of Work* (PoW). Exemplos incluem Bitcoin e Ethereum.
- ***Blockchains* Permissionadas (Privadas ou de Consórcio):** O acesso para ler ou escrever na rede é restrito a um grupo de participantes previamente autorizados. Utilizam algoritmos de consenso mais eficientes, como o Raft. *Hyperledger Fabric* e R3 Corda são exemplos proeminentes, adequados para cenários corporativos.
- ***Blockchains* Híbridas:** Integram características dos modelos público e permissionado, buscando um balanço entre transparência e controle de privacidade.

2.2.2 Tecnologias para Aprimoramento da Privacidade (PETs)

Para endereçar as preocupações com a privacidade de dados, diversas Tecnologias para Aprimoramento da Privacidade (*Privacy-Enhancing Technologies* - PETs) foram desenvolvidas. As **Provas de Conhecimento Zero (*Zero-Knowledge Proofs* - ZKPs)** permitem que uma parte prove a veracidade de uma afirmação sem revelar qualquer informação adicional. Por exemplo, verificar a maioridade de um indivíduo sem expor sua data de nascimento (19). A **Criptografia Homomórfica** permite realizar cálculos sobre dados criptografados, mas seu alto custo computacional ainda limita sua aplicação prática (20).

Plataformas como o *Hyperledger Fabric* implementam soluções nativas de privacidade, como as **Coleções de Dados Privadas (*Private Data Collections* - PDCs)**, que permitem que um subconjunto de organizações em um canal transacione dados de forma confidencial, sem compartilhá-los com todos os membros (21).

2.3 HYPERLEDGER FABRIC

O *Hyperledger Fabric* é uma plataforma de *blockchain* permissionada e de código aberto, mantida pela Linux Foundation. Sua arquitetura modular *execute-order-validate* permite a execução paralela de transações e o uso de linguagens de programação convencionais para os contratos inteligentes (*smart contracts*), chamados de *chaincodes*, conferindo vantagens de desempenho e escalabilidade em comparação com outras plataformas (22). Sua aplicação tem sido explorada com sucesso em setores como saúde (23, 24), Internet das Coisas (IoT) (25, 26) e educação (27).

2.3.1 *Smart Contracts* como Ferramenta de Conformidade

Os *chaincodes* no *Hyperledger Fabric* são programas autoexecutáveis que rodam na *blockchain*. Eles podem ser projetados como agentes de automação da conformidade, traduzindo regras regulatórias em código, um conceito conhecido como *compliance-as-code* (28). No contexto da LGPD, os *chaincodes* podem:

- **Gerenciar o Consentimento:** Automatizar o registro, a validação e a revogação do consentimento do titular de forma granular.
- **Aplicar Políticas de Retenção:** Executar automaticamente a exclusão ou anonimização de dados após o término do período de retenção.
- **Automatizar os Direitos dos Titulares:** Orquestrar o fluxo para solicitações de acesso, correção ou portabilidade, criando um registro imutável de todas as ações.

Ao embutir as regras de conformidade na lógica de negócios, os *smart contracts* reduzem o risco de erro humano e facilitam a demonstração de *accountability* (29).

2.3.2 Modelos de Governança em Redes Permissionadas

A governança em uma rede *blockchain* refere-se ao conjunto de regras e processos para tomar decisões sobre o sistema. Em redes permissionadas, um modelo de governança robusto é fundamental (30).

- **Governança *Off-Chain*:** Envolve acordos humanos e legais fora da rede, como contratos que definem as responsabilidades de cada membro (Controlador, Operador, Encarregado) e a estrutura de um comitê gestor.
- **Governança *On-Chain*:** Refere-se às regras aplicadas diretamente pelo código. No Fabric, isso é implementado através de políticas de endosso (*endorsement policies*), que definem quais organizações devem validar uma transação, e Listas de Controle de Acesso (ACLs).

Um modelo de governança eficaz para conformidade com a LGPD deve combinar ambos os aspectos, usando a governança *off-chain* para estabelecer a estrutura legal e a *on-chain* para aplicar tecnicamente as regras de proteção de dados (31).

2.4 TÓPICOS AVANÇADOS E DESAFIOS DE IMPLEMENTAÇÃO

2.4.1 Identidade Auto-Soberana (*Self-Sovereign Identity* - SSI)

A Identidade Auto-Soberana (SSI) é um paradigma de gestão de identidade digital que confere ao indivíduo controle total sobre suas informações, alinhando-se aos princípios de empoderamento do titular da LGPD (32, 33). Seus pilares, padronizados pelo W3C, são:

- **Identificadores Descentralizados (DIDs):** Identificadores únicos que não dependem de uma autoridade central, ancorados em uma *blockchain* (34).
- **Credenciais Verificáveis (VCs):** Declarações digitais criptograficamente seguras e à prova de adulteração, emitidas por uma entidade e controladas pelo titular (35).
- **Carteiras Digitais (*Wallets*):** Aplicações seguras onde o titular armazena e gerencia seus DIDs e VCs.

A integração da SSI em uma solução *blockchain* pode oferecer uma abordagem proativa de privacidade (*privacy by design*).

2.4.2 Interoperabilidade entre Redes e Sistemas Legados

Uma solução *blockchain* empresarial raramente opera de forma isolada. A interoperabilidade com outras redes e com sistemas legados (ERPs, CRMs) é um requisito fundamental (36).

- **Interoperabilidade *Cross-Chain*:** A comunicação entre diferentes plataformas de *blockchain* pode ser alcançada por meio de protocolos de comunicação inter-blockchain (como o IBC) ou redes de retransmissão (37).
- **Interoperabilidade com Sistemas Legados:** A conexão com sistemas externos requer o uso de **oráculos (*oracles*)**, serviços confiáveis que atuam como pontes, buscando e verificando dados do mundo real para serem utilizados por *smart contracts* (38).

A implementação de APIs padronizadas e uma camada de microsserviços são práticas recomendadas para garantir uma integração segura e modular.

2.4.3 Modelos de Consórcio e Implicações Organizacionais

O *Hyperledger Fabric* é projetado para redes de consórcio, onde múltiplas organizações colaboram. O sucesso de tal empreendimento depende criticamente da definição de um modelo de consórcio funcional (39). Os elementos a serem definidos incluem o modelo de negócios (compartilhamento de custos, ROI), a estrutura legal (contratos, resolução de disputas) e a governança técnica e operacional (entrada de novos membros, atualizações). Um dos maiores desafios é o "dilema do cooperador", onde organizações concorrentes precisam colaborar, exigindo alinhamento de incentivos e confiança na neutralidade da plataforma (40).

3 TRABALHOS CORRELATOS

3.1 *FRAMEWORKS* PARA CONFORMIDADE REGULATÓRIA EM *BLOCKCHAIN*

A proeminência do GDPR impulsionou a pesquisa sobre como alinhar a tecnologia *blockchain* aos seus requisitos. O *framework* PriBC, proposto por Paik et al. (2022), visa garantir os direitos dos titulares de dados no GDPR por meio de uma *blockchain* permissionada que segrega dados pessoais, metadados e políticas (41). Embora sua granularidade na segregação de dados seja um ponto forte, sua validação é teórica, sem uma implementação prática em uma plataforma consolidada como o *Hyperledger Fabric*, e não aborda as especificidades da LGPD.

De forma complementar, Damgård et al. (2023) apresentam um *framework* modular para a construção de sistemas *blockchain* "conscientes da privacidade"(*privacy-aware*), permitindo a integração flexível de diferentes PETs (42). A proposta é conceitualmente robusta, mas permanece em um alto nível de abstração, funcionando como um meta-modelo, em contraste com o BlockGuard, que traduz princípios similares em uma solução de engenharia específica e validada.

Já o GDPRchain, proposto por Casino et al. (2023), foca na auditabilidade e *accountability*, utilizando a *blockchain* para o registro imutável de atividades de tratamento e consentimento (43). A solução é eficaz para gerar trilhas de auditoria, mas seu escopo é restrito a uma ferramenta de registro, não sendo um *framework* completo para a gestão do ciclo de vida dos dados e a operacionalização de todos os direitos dos titulares, que são domínios centrais do BlockGuard.

3.2 SOLUÇÕES TÉCNICAS PARA DESAFIOS ESPECÍFICOS DE PRIVACIDADE

3.2.1 O Direito ao Esquecimento e a Imutabilidade

A reconciliação do direito à eliminação com a imutabilidade continua sendo um tópico central. O conceito de *blockchain* redigível tem visto avanços, como o esquema de *chameleon hash* proposto por Tian et al. (2022), que permite a edição de transações sob políticas de governança (44). No entanto, sua implementação exigiria modificações profundas em plataformas existentes, tornando a abordagem de armazenamento *off-chain* a mais viável. Zheng et al. (2023) validam essa estratégia em um sistema de saúde, demonstrando que a arquitetura híbrida não apenas soluciona o problema, mas também melhora a escalabilidade (45). O BlockGuard adota e expande essa premissa, integrando-a a um conjunto mais amplo de controles de conformidade LGPD.

3.2.2 Aplicação de Provas de Conhecimento Zero (ZKP)

A aplicação de ZKPs para garantir a privacidade tem crescido. Nathan et al. (2023) demonstram a integração de provas zk-SNARKS no *Hyperledger Fabric* para permitir transações confidenciais, confirmando

a viabilidade técnica, mas destacando o *overhead* computacional (46). O BlockGuard, embora conceitualmente compatível, opta pela solução nativa e mais performática das Coleções de Dados Privadas, mas a pesquisa com ZKPs aponta para uma futura linha de evolução do *framework*.

3.2.3 Técnicas de Anonimização e Pseudonimização

Kumar et al. (2023) exploram a aplicação de privacidade diferencial em sistemas *blockchain*, onde ruído calibrado é adicionado aos dados para garantir privacidade matemática (47). Paralelamente, Silva et al. (2024) desenvolveram uma técnica de k-anonimização adaptativa para *blockchain*, demonstrando ser possível manter a conformidade com um *overhead* de processamento de apenas 15

3.3 APLICAÇÕES DE *HYPERLEDGER FABRIC* EM CONTEXTOS REGULADOS

No setor da saúde, Torky e Al-Azzawy (2024) propuseram uma arquitetura baseada em Fabric para o compartilhamento seguro de prontuários eletrônicos, validando a capacidade da plataforma para criar sistemas robustos, mas tratando a conformidade com leis de proteção de dados de forma superficial (48). Zhang et al. (2023) avançaram ao implementar um sistema de gerenciamento de consentimento médico granular com Controle de Acesso Baseado em Atributos (ABAC), demonstrando viabilidade técnica, mas sem abordar completamente a portabilidade de dados (23).

No setor financeiro, Patel e Rodriguez (2024) desenvolveram uma plataforma de compartilhamento de dados KYC (*Know Your Customer*) usando Fabric, com alta performance, mas sem contemplar requisitos como o direito à retificação ou eliminação (49). Na cadeia de suprimentos farmacêutica, o trabalho de Ahmad et al. (2023) utiliza o Fabric para rastreabilidade, mas a proteção de dados pessoais não é o foco (50). Esses casos evidenciam que, mesmo em aplicações robustas, a conformidade com a proteção de dados muitas vezes não é uma prioridade de design, reforçando a necessidade de um *framework* especializado como o BlockGuard.

3.4 LGPD E TECNOLOGIAS EMERGENTES NO CONTEXTO BRASILEIRO

A produção científica brasileira sobre a dimensão técnica da conformidade da *blockchain* com a LGPD ainda está em desenvolvimento. O estudo de Souza e de Almeida (2023) apresenta um protótipo para gestão de consentimento, mas utiliza uma *blockchain* pública (Ethereum), inadequada para cenários corporativos devido a custos e desafios de privacidade, além de focar apenas no consentimento (51).

Oliveira et al. (2024) realizaram uma análise comparativa entre plataformas *blockchain* para conformidade com a LGPD, concluindo que o *Hyperledger Fabric* oferece a melhor combinação de recursos, mas o trabalho permanece no nível de análise, sem propor uma arquitetura concreta (52). Já Santos e Lima (2023) mapearam requisitos técnicos derivados das orientações da ANPD, fornecendo uma base regulatória sólida, mas sem avançar para a proposição de soluções técnicas (53).

3.5 SÍNTESE DA ANÁLISE E LACUNA DE PESQUISA

A análise dos trabalhos correlatos revela que:

- Os *frameworks* existentes focam no GDPR e carecem de implementação prática e adaptação à LGPD.
- As soluções técnicas são pontuais e não oferecem a abordagem holística necessária para uma conformidade completa.
- As aplicações setoriais em *Hyperledger Fabric* tratam a proteção de dados como um requisito secundário.
- Os estudos brasileiros são incipientes, focando em aspectos isolados ou em plataformas inadequadas para o contexto corporativo.

A lacuna na literatura é, portanto, a ausência de um *framework* de engenharia de *software*, holístico e prático, que sirva como um guia para o desenvolvimento de aplicações *Hyperledger Fabric* em plena conformidade com a LGPD. É precisamente nesta lacuna que a presente dissertação se insere, com o BlockGuard sendo concebido para traduzir os requisitos legais em uma arquitetura específica, prática e validada para o ecossistema Fabric.

4 METODOLOGIA

4.1 CARACTERIZAÇÃO DA PESQUISA

Esta pesquisa caracteriza-se como uma investigação de natureza aplicada, com abordagem qualitativa e objetivos exploratórios e descritivos. A escolha da abordagem qualitativa justifica-se pela necessidade de compreender em profundidade os aspectos técnicos e jurídicos envolvidos na implementação de *blockchain* em conformidade com a LGPD (54). Os objetivos exploratórios visam mapear o estado da arte, enquanto os descritivos buscam detalhar os requisitos e capacidades das tecnologias estudadas.

4.2 ESTRATÉGIA DE PESQUISA: *DESIGN SCIENCE RESEARCH*

A estratégia adotada é a *Design Science Research* (DSR), uma metodologia voltada para a solução de problemas práticos por meio da construção, desenvolvimento e avaliação de artefatos tecnológicos inovadores (10). Complementarmente, foram utilizados análise documental e um estudo de caso para validar a aplicabilidade do artefato desenvolvido (o *framework* BlockGuard).

O ciclo DSR foi aplicado nesta pesquisa da seguinte forma:

1. **Identificação do Problema:** A lacuna de um *framework* prático para conformidade LGPD em *Hyperledger Fabric*.
2. **Definição dos Objetivos:** Os objetivos geral e específicos definidos no Capítulo 1.
3. ***Design* e Desenvolvimento:** A concepção e especificação do *framework* BlockGuard.
4. **Demonstração:** A implementação do *framework* em um estudo de caso.
5. **Avaliação:** A análise da conformidade legal e da viabilidade técnica do artefato.
6. **Comunicação:** A presente dissertação.

4.3 PROCEDIMENTOS METODOLÓGICOS

A pesquisa foi estruturada em quatro fases inter-relacionadas:

4.3.1 Fase I: Revisão Integrada da Literatura e Documentação

Esta fase compreendeu a revisão sistemática da literatura sobre *blockchain*, *Hyperledger Fabric* e LGPD, além da análise documental das normas vigentes e da documentação técnica da plataforma para o mapeamento dos requisitos regulatórios.

4.3.2 Fase II: Desenvolvimento do *Framework* BlockGuard

Nesta fase, foi elaborado o *framework* BlockGuard, consistindo em uma arquitetura estruturada em domínios funcionais. O desenvolvimento seguiu uma abordagem orientada à conformidade, traduzindo requisitos regulatórios em componentes arquiteturais concretos.

4.3.3 Fase III: Implementação e Estudo de Caso Integrado

Nesta fase, procedeu-se à implementação dos componentes críticos do *framework* e à sua aplicação prática por meio de um estudo de caso no setor de saúde. Esta abordagem integrada permitiu demonstrar a operacionalidade da solução em um cenário realista de alta relevância regulatória.

4.3.4 Fase IV: Análise e Validação

Realizou-se a validação qualitativa do *framework* por meio da análise da sua aderência legal, viabilidade técnica e adequação operacional, sustentando-se na triangulação de dados obtidos da implementação e da análise de conteúdo (55).

4.4 CRITÉRIOS DE VALIDAÇÃO E QUALIDADE

A validação do artefato considerou três níveis: legal, técnico e operacional. Para garantir a qualidade da pesquisa, adotaram-se os critérios de credibilidade, transferibilidade, dependabilidade e confirmabilidade, conforme proposto por Lincoln e Guba (1985) (56).

4.5 ASPECTOS ÉTICOS

A pesquisa respeitou integralmente os princípios éticos. O estudo de caso utilizou dados exclusivamente sintéticos para evitar o tratamento de informações pessoais reais. Os métodos adotados foram transparentes, e a concepção do *framework* está alinhada aos conceitos de *Privacy by Design*.

5 O *FRAMEWORK* BLOCKGUARD: FUNDAMENTAÇÃO TEÓRICA E ESPECIFICAÇÃO CONCEITUAL

O *framework* BlockGuard constitui uma arquitetura conceitual e técnica projetada para estabelecer a interoperabilidade normativa entre os requisitos da Lei Geral de Proteção de Dados (LGPD) e as capacidades intrínsecas da plataforma de registro distribuído *Hyperledger Fabric* (5, 57). Diferentemente de abordagens fragmentadas que tratam a conformidade como uma camada sobreposta, o BlockGuard integra regras regulatórias diretamente na lógica de negócios da rede (*compliance-by-design*), baseando-se em uma taxonomia que mapeia sistematicamente os princípios jurídicos abstratos a implementações criptográficas e arquiteturais concretas (58).

A concepção do *framework* parte do reconhecimento de que os princípios elencados no Art. 6º da LGPD — notadamente finalidade, necessidade, segurança e transparência — não são meras diretrizes éticas, mas requisitos funcionais que demandam materialização técnica verificável. Para tal, o BlockGuard explora primitivas avançadas do *Hyperledger Fabric*, como canais privados, coleções de dados privados (*Private Data Collections* - PDCs), políticas de endosso granular e contratos inteligentes (*chaincodes*), para implementar controles de conformidade dinâmicos (21). O desafio central endereçado é a reconciliação técnica entre a imutabilidade do *ledger* (garantia de integridade) e os direitos de modificabilidade e exclusão de dados (garantia de privacidade), especificamente o direito ao esquecimento previsto no Art. 18, VI da LGPD.

5.1 ARQUITETURA CONCEITUAL E MODELO DE CAMADAS

A arquitetura do BlockGuard é estruturada em cinco camadas hierárquicas, projetadas segundo o princípio da separação de responsabilidades (*separation of concerns*) para promover modularidade, escalabilidade e auditabilidade independente de cada componente.

Na base, a **Camada 1 (Infraestrutura *Blockchain*)** encapsula os componentes nativos do *Hyperledger Fabric* (versão 2.x), incluindo os nós validadores (*peers*), o serviço de ordenação baseado no algoritmo de consenso Raft (tolerante a falhas de parada, mas não bizantinas) e os provedores de serviços de membro (MSP). O *framework* orquestra a topologia dessa rede, definindo canais segregados e políticas de governança no arquivo `configtx.yaml`, sem alterar o código-fonte do núcleo da plataforma.

Acima dela, a **Camada 2 (Abstrações de Conformidade LGPD)** atua como um tradutor semântico entre o domínio jurídico e o técnico. Nesta camada residem os cinco domínios funcionais do BlockGuard, ilustrados na Figura 5.1. Cada domínio corresponde a um conjunto de artigos da lei e é materializado através de *chaincodes* escritos em linguagem Go, que encapsulam a lógica de validação regulatória.

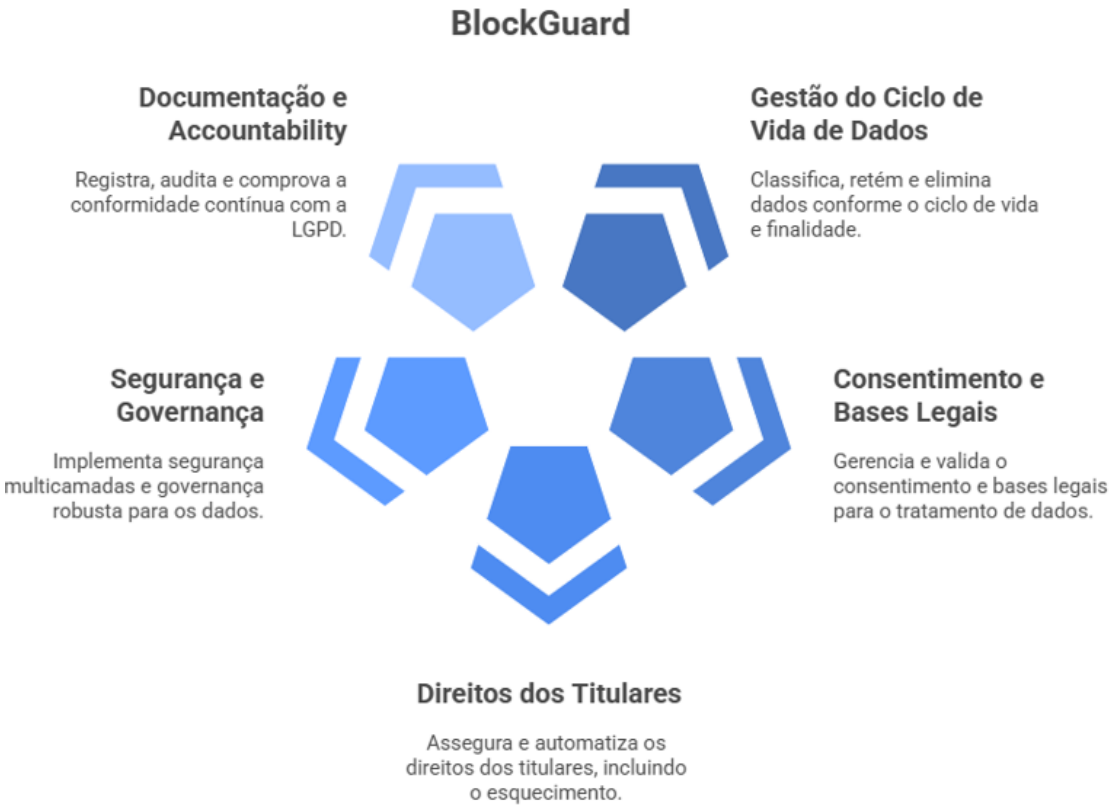

Figura 5.1: *Framework* BlockGuard e seus cinco domínios de conformidade, demonstrando a relação entre os requisitos legais da LGPD e os componentes tecnológicos implementados na plataforma *Hyperledger Fabric*. Fonte: Autor.

A **Camada 3 (Serviços de Conformidade)** provê funções transversais invocadas pelos contratos inteligentes, incluindo motores de classificação de dados, módulos criptográficos (utilizando AES-256-GCM para dados em repouso), gestão de chaves assimétricas (ECDSA P-256) e serviços de *logging* imutável. A **Camada 4 (Interfaces de Aplicação)** expõe uma API RESTful segura e SDKs que abstraem a complexidade do protocolo gRPC do Fabric, incorporando validações de esquema JSON e sanitização de dados no lado do cliente. Finalmente, a **Camada 5 (Aplicações e Usuários Finais)** consome esses serviços através de interfaces web responsivas, tornando a complexidade da conformidade transparente para titulares e operadores.

5.1.1 Modelo de Dados Híbrido: *On-Chain* vs *Off-Chain*

Uma contribuição central do BlockGuard é a formalização de um modelo de armazenamento híbrido, estritamente alinhado à metodologia *Privacy by Design* (59). Este modelo resolve o paradoxo da imutabilidade ao estabelecer que dados pessoais em texto claro jamais devem persistir no *ledger* principal.

O armazenamento é segmentado em dois repositórios logicamente distintos mas criptograficamente vinculados:

- **Repositório *On-Chain*:** O *ledger* da *blockchain* armazena exclusivamente metadados de controle,

timestamps, identificadores de transação e, crucialmente, o *hash* SHA-256 dos dados pessoais. Este registro serve como "prova de existência"e "prova de integridade", mas não revela o conteúdo informacional.

- **Repositório *Off-Chain*:** Um banco de dados relacional (PostgreSQL) ou orientado a documentos (CouchDB) armazena os dados pessoais brutos, dados sensíveis de saúde e arquivos binários. Este repositório permite operações de `UPDATE` e `DELETE` necessárias para atender aos direitos dos titulares.

O modelo satisfaz três propriedades formais de segurança:

1. **Integridade Verificável:** Para qualquer registro de dado *off-chain* D, existe um registro imutável correspondente no *ledger* contendo $H(D)$. Qualquer alteração não autorizada em D resultará em $H(D') \neq H(D)$, detectável por qualquer nó da rede através de recálculo independente.
2. **Eliminabilidade com Rastreabilidade:** A exclusão física do dado no repositório *off-chain* torna a informação inacessível, mas mantém o *hash* no *ledger* como prova histórica de que o dado existiu e foi tratado sob bases legais, garantindo a *accountability*.
3. **Minimização por Concepção:** A arquitetura impõe tecnicamente que apenas o mínimo necessário (o *hash*) seja replicado entre todos os nós da rede, reduzindo a superfície de ataque e o risco de vazamento de dados.

A consistência entre os repositórios é mantida por um **Protocolo de *Commit* em Duas Fases Adaptado**. Na fase de preparação, o dado é persistido *off-chain* e seu *hash* é calculado. Na fase de confirmação, o *chaincode* valida a unicidade do registro e grava o *hash* e os metadados no *ledger*. O ID da transação (TxID) gerado pelo Fabric é então atualizado no registro *off-chain*, criando uma âncora bidirecional indissociável.

5.2 DOMÍNIO 1: GESTÃO DO CICLO DE VIDA DE DADOS PESSOAIS

Este domínio operacionaliza os requisitos dos Arts. 6º (princípios), 15 (término do tratamento) e 16 (eliminação) da LGPD. O *framework* implementa um motor de classificação de dados baseado em regras que, no momento da ingestão, atribui um rótulo de sensibilidade (Nível 1 a 4) a cada conjunto de dados. Este rótulo dita as políticas de armazenamento (público do canal ou coleção privada), os requisitos de criptografia e os períodos de retenção.

As políticas de retenção são definidas declarativamente em formato JSON e vinculadas aos tipos de dados. Um serviço de orquestração monitora periodicamente o *ledger* em busca de registros cujo período de retenção expirou ou cuja finalidade foi encerrada, disparando automaticamente o processo de eliminação.

Para atender ao direito de eliminação (Art. 18, VI) de forma tecnicamente viável, o *framework* suporta três estratégias:

- **Eliminação Lógica (*Soft Delete*):** Atualização de um atributo de estado no registro *off-chain* para "INATIVO", impedindo o acesso via API, mas mantendo o dado para fins de auditoria ou cumprimento de obrigação legal.
- **Eliminação Criptográfica (*Crypto-Shredding*):** Técnica avançada onde os dados são criptografados com uma chave simétrica única por registro. Para eliminar o dado, destrói-se a chave correspondente no sistema de gestão de chaves (KMS), tornando o texto cifrado matematicamente irrecuperável (60).
- **Eliminação Física:** Execução de comando `DELETE` no banco de dados *off-chain*, removendo permanentemente os bytes do disco.

Em todos os casos, uma transação de "Prova de Eliminação"é registrada no *ledger*, contendo o TxID original, o motivo da eliminação e a assinatura digital do controlador.

5.3 DOMÍNIO 2: CONSENTIMENTO E BASES LEGAIS

Gerenciando os Arts. 7º, 8º e 9º da LGPD, este domínio modela o consentimento não como um atributo estático, mas como um contrato vivo gerenciado por uma Máquina de Estados Finita (FSM) implementada no *chaincode*. Os estados possíveis são: `PENDENTE`, `ATIVO`, `REVOGADO` e `EXPIRADO`. Transições de estado exigem transações assinadas pelo titular (ou pela aplicação em seu nome) e são validadas contra regras de negócio (ex: não é possível revogar um consentimento já expirado).

Para proteger a privacidade dos dados associados ao próprio consentimento (como o IP do titular e seus dados de contato), o *framework* utiliza intensivamente as ***Private Data Collections* (PDCs)** do Fabric. Foram arquitetadas duas coleções distintas:

1. `consentDetailsCollection`: Armazena os dados pessoais identificáveis do titular. A política de coleção (*collection policy*) é configurada para permitir acesso apenas aos nós do Controlador e do Operador específico, utilizando o protocolo *gossip* para disseminação ponto-a-ponto, sem passar pelo *Orderer*.
2. `consentAuditCollection`: Armazena logs de auditoria anonimizados sobre o uso do consentimento. Sua política é mais permissiva, permitindo acesso a nós de Auditoria e Reguladores.

Além do consentimento, o *framework* suporta o registro estruturado de outras bases legais. Para o Legítimo Interesse, exige-se o registro do *hash* do Relatório de Impacto à Proteção de Dados (RIPD/LIA). Para Execução de Contrato, vincula-se o registro ao identificador do contrato jurídico. Isso garante que todo tratamento de dados na rede possua um lastro legal verificável.

5.4 DOMÍNIO 3: DIREITOS DOS TITULARES

Operacionalizando o Art. 18, este domínio fornece a lógica de negócios para atender às requisições dos titulares (DSR - *Data Subject Requests*). Um *chaincode* específico (`subject_rights`) orquestra esses fluxos, garantindo o cumprimento dos prazos legais (ex: 15 dias para acesso completo).

O *framework* provê mecanismos técnicos para cada direito:

- **Acesso e Confirmação:** O sistema consulta o *ledger* usando a chave composta do titular. Se o dado residir *off-chain*, o sistema o recupera, recalcula seu *hash* em tempo real e o compara com o *hash* do *ledger* antes de retorná-lo, garantindo que o titular receba uma cópia íntegra e autêntica.
- **Correção:** Permite a atualização do dado no repositório *off-chain*. O *chaincode* então registra uma nova versão do ativo no *ledger* com o novo *hash*, mantendo um ponteiro para a versão anterior, criando uma linhagem de dados completa.
- **Portabilidade:** Um processo de extração gera um arquivo estruturado (JSON ou XML) contendo todos os dados vinculados ao titular. O *hash* deste arquivo é registrado na *blockchain* para garantir que o pacote de portabilidade não foi adulterado durante a transferência.
- **Revogação:** Aciona a transição de estado na FSM de consentimento e dispara eventos (‘chaincode events‘) que podem ser consumidos por sistemas externos para interromper processamentos ativos.

5.5 DOMÍNIO 4: SEGURANÇA E GOVERNANÇA

Este domínio cobre os requisitos de segurança da informação (Art. 46) e governança (Art. 50). A segurança é implementada em profundidade:

- **Transporte:** Todo o tráfego gRPC entre nós é encapsulado em túneis TLS 1.3 com autenticação mútua (mTLS), garantindo confidencialidade e autenticidade da rede.
- **Identidade:** Utiliza-se a infraestrutura de PKI do Fabric CA para emitir certificados X.509 v3. O acesso aos recursos é controlado por políticas ABAC (*Attribute-Based Access Control*) que verificam atributos inseridos nos certificados (ex: `role=DPO`, `org=Hospital`).
- **Dados em Repouso:** Bancos de dados *off-chain* e o *world state* (CouchDB) dos *peers* são criptografados em disco.

A governança é materializada através de um *chaincode* de gestão de incidentes. Este contrato permite o registro de violações de segurança, classificando-as automaticamente segundo uma matriz de severidade. Incidentes classificados como de alto risco disparam, via *chaincode events*, notificações para o DPO e preparam relatórios preliminares para comunicação à ANPD, cumprindo o prazo e os requisitos do Art. 48.

5.6 DOMÍNIO 5: AUDITORIA, DOCUMENTAÇÃO E *ACCOUNTABILITY*

Para atender ao dever de responsabilização (Art. 6º, X), o BlockGuard implementa um mecanismo de auditoria contínua. Diferente de logs tradicionais que podem ser apagados por administradores, a trilha de auditoria do BlockGuard é gravada no *ledger* imutável.

Um componente interceptador (*middleware*) na camada de API captura metadados de todas as transações (quem, quando, o quê, qual base legal) e submete uma transação de auditoria paralela para um canal de *logging* segregado ou uma PDC de auditoria. Isso garante que mesmo operações de leitura (*queries*) deixem rastros auditáveis.

O sistema permite a reconstrução histórica completa do estado de qualquer dado ("viagem no tempo"), facilitando a geração do **Registro de Atividades de Tratamento** (Art. 37) de forma automatizada e fidedigna. Além disso, fornece insumos verificáveis para a elaboração de Relatórios de Impacto à Proteção de Dados (RIPD), permitindo que controladores demonstrem conformidade baseada em evidências criptográficas matemáticas, e não apenas em declarações documentais.

6 IMPLEMENTAÇÃO E VALIDAÇÃO DO FRAMEWORK BLOCKGUARD

6.1 ESPECIFICAÇÕES DO AMBIENTE EXPERIMENTAL

O ambiente de validação foi configurado para simular realisticamente um consórcio de saúde digital envolvendo duas organizações independentes: `Org1MSP` (representando o "Hospital Central", atuando como controlador de dados) e `Org2MSP` (representando o "Laboratório Clínico", atuando como operador). Esta configuração reflete um cenário comum no ecossistema de saúde brasileiro, onde múltiplas instituições necessitam compartilhar dados de pacientes de forma segura e em conformidade com a LGPD.

A infraestrutura tecnológica completa utilizada compreende:

- **Sistema Operacional:** Ubuntu 22.04 LTS (Jammy Jellyfish), executado em ambiente virtualizado com 8GB RAM e 4 vCPUs.
- ***Hyperledger Fabric*:** Versão 2.5.12 (última versão LTS disponível), configurada com dois *peer nodes* (um por organização), um serviço de ordenação baseado em Raft (três nós ordenadores para garantir consenso distribuído) e CouchDB 3.3.2 como banco de dados de estado (*state database*) para permitir consultas ricas usando a sintaxe Mango Query.
- **Linguagem dos *Chaincodes*:** Go 1.21.3, utilizando o SDK oficial `fabric-contract-api-go` v1.2.1 e o pacote `fabric-shim` para interação com o *ledger*. A escolha da linguagem Go justifica-se pelo seu excelente desempenho, segurança de tipos e suporte nativo do Hyperledger Fabric.
- **Persistência *Off-chain*:** PostgreSQL 15.4 com extensão `pgcrypto` para funções criptográficas nativas, armazenando dados pessoais volumosos, dados sensíveis de saúde (prontuários completos, exames laboratoriais, imagens médicas) e arquivos binários que necessitem ser fisicamente eliminados para atender ao direito ao esquecimento.
- **API de Integração:** Node.js 18.19.1 (versão LTS) com *framework* Express.js 4.18.2, implementando 25 *endpoints* RESTful documentados com Swagger/OpenAPI 3.0 para comunicação entre o *frontend*, o *backend* e a rede blockchain. A API utiliza o *Fabric SDK for Node.js* v2.2.19 para submeter transações e consultar o *ledger*.
- **Interface Gráfica (*Dashboard*):** React 18.2.0 com biblioteca de componentes Material-UI (MUI) 5.14.20, compondo uma *Single Page Application* (SPA) responsiva com aproximadamente 2.143 linhas de código TypeScript. O *dashboard* implementa visualizações interativas usando Recharts 2.8.0, gerenciamento de estado com Redux Toolkit 1.9.7 e comunicação assíncrona via Axios 1.5.1.
- **Containerização:** Docker 24.0.6 e Docker Compose 2.21.0 para orquestração de todos os componentes da infraestrutura (nós Fabric, bancos de dados, API, *Certificate Authorities*), garantindo reprodutibilidade e portabilidade do ambiente experimental.

6.1.1 Topologia da Rede e Configuração do Consórcio

A rede blockchain foi configurada seguindo as melhores práticas de segurança do Hyperledger Fabric. Cada organização opera sua própria *Certificate Authority* (CA) baseada em Fabric CA 1.5.7, responsável por emitir certificados X.509 para identidades de usuários e nós. O canal principal (`healthchannel`) conecta os *peers* de ambas as organizações, permitindo compartilhamento controlado de dados médicos. Políticas de endosso customizadas foram definidas para cada *chaincode*, exigindo aprovação de múltiplas organizações para transações críticas, implementando assim o princípio de consenso distribuído.

6.2 ARQUITETURA IMPLEMENTADA: DA CONCEPÇÃO À MATERIALIZAÇÃO

A arquitetura implementada materializa fielmente o modelo conceitual em camadas apresentado no Capítulo 5, demonstrando as interações complexas entre os componentes de *frontend*, *backend*, *blockchain* e banco de dados relacional. A Figura 6.1 apresenta uma visão consolidada desta arquitetura e suas integrações.

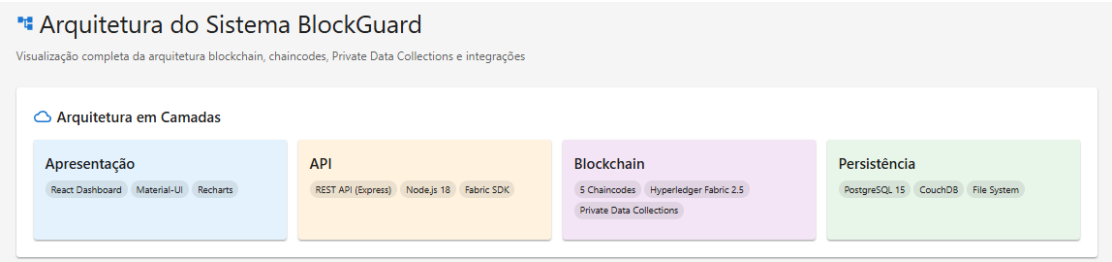

Figura 6.1: Arquitetura geral implementada do BlockGuard, demonstrando a integração entre *dashboard* React, API Node.js, rede Hyperledger Fabric (com cinco *chaincodes* especializados) e banco de dados PostgreSQL *off-chain*.

A comunicação entre o *dashboard* e a API REST ocorre via protocolo HTTP/HTTPS, utilizando autenticação baseada em *JSON Web Tokens* (JWT) para garantir a identidade dos usuários. A API, por sua vez, interage com a rede *Hyperledger Fabric* através do SDK Node.js, que utiliza o protocolo gRPC para comunicação de alto desempenho com os *peers* e nós ordenadores. As transações submetidas à blockchain seguem o fluxo canônico do Fabric: proposta $\rightarrow$ endosso $\rightarrow$ ordenação $\rightarrow$ validação $\rightarrow$ *commit*, garantindo consenso distribuído e imutabilidade.

6.3 IMPLEMENTAÇÃO DOS CINCO DOMÍNIOS DE CONFORMIDADE

Esta seção detalha a implementação técnica de cada um dos cinco domínios conceituais do BlockGuard, discutindo a lógica dos algoritmos e os resultados de validação empírica. Todos os códigos-fonte referenciados encontram-se no **Apêndice 8**.

6.3.1 Domínio 1: Gestão do Ciclo de Vida e Modelo Híbrido de Armazenamento

A implementação do *chaincode* `data_lifecycle` (v1.4, implantado com *sequence* 4) constitui a base para a gestão de dados pessoais conforme os Arts. 6º (princípios) e 16 (qualidade dos dados) da LGPD. A principal característica validada neste domínio é o modelo de armazenamento híbrido, que reconcilia a imutabilidade da blockchain com os requisitos de modificabilidade de dados.

6.3.1.1 Implementação da Função de Criação de Dados

A lógica de criação de dados é encapsulada na função `CreateData`, cuja implementação completa em Go pode ser consultada na **Listagem 8.1** do **Apêndice 8**.

O algoritmo é responsável por receber o *hash* SHA-256 dos dados armazenados *off-chain* juntamente com metadados estruturados, realizar classificação automática de sensibilidade baseada em regras (ex: dados médicos recebem nível 4) e persistir o registro de ancoragem no *ledger* distribuído.

6.3.1.2 Fluxo Completo de Criação com Modelo Híbrido

O processo completo de criação de um novo registro de dados, que exemplifica a dinâmica do modelo híbrido, é detalhado no diagrama de sequência da Figura 6.2.

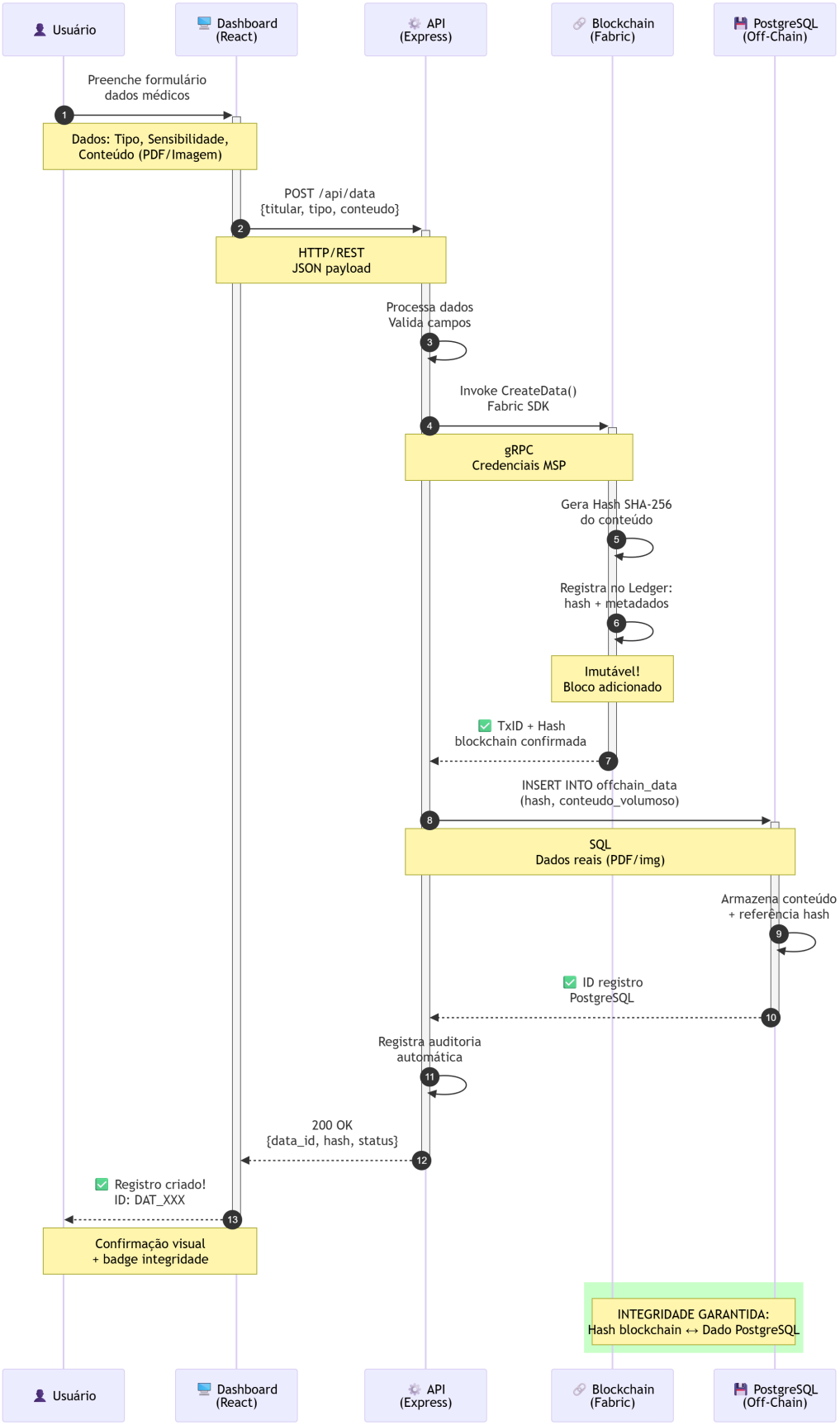

Figura 6.2: Diagrama de sequência do processo de criação de dados com modelo híbrido, demonstrando a interação coordenada entre *dashboard*, API REST, *blockchain* Hyperledger Fabric e banco de dados PostgreSQL *off-chain*.

Este fluxo demonstra como a ancoragem criptográfica é estabelecida de forma atômica. O protocolo de duas fases garante a consistência: se a transação blockchain falhar, os dados *off-chain* podem ser marcados como "órfãos"e eliminados; se a gravação *off-chain* falhar, a transação blockchain não é submetida, prevenindo *hashes* órfãos no *ledger*.

6.3.1.3 Demonstração Empírica da Verificação de Integridade

A principal vantagem da arquitetura híbrida reside na capacidade de verificar deterministicamente a integridade dos dados *off-chain* a qualquer momento, detectando adulterações não autorizadas através da comparação criptográfica de *hashes*.

A Figura 6.3 apresenta o estado inicial de um registro médico na interface do sistema, evidenciando visualmente a arquitetura híbrida: o conteúdo clínico completo armazenado no PostgreSQL e seu respectivo *hash* imutável no *ledger*.

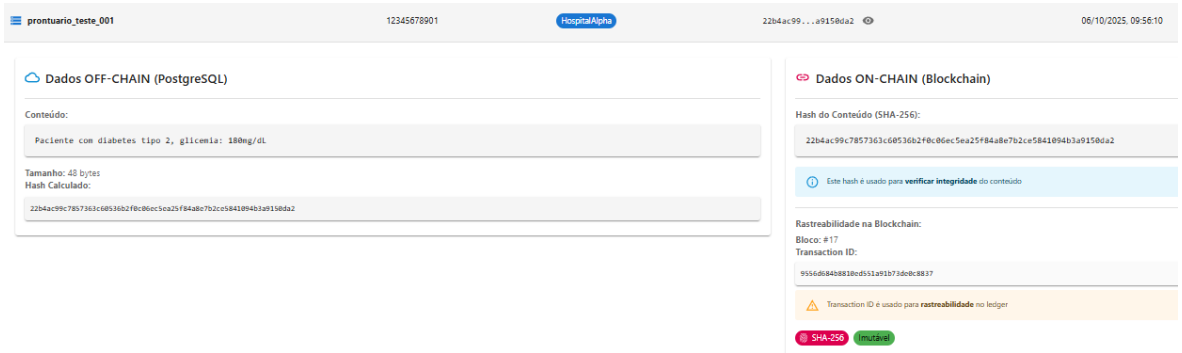

Figura 6.3: Interface do BlockGuard exibindo a arquitetura híbrida: conteúdo completo *off-chain* e seu *hash* SHA-256 imutável *on-chain*.

Fase 1: Verificação de Integridade em Cenário de Conformidade Na primeira fase, o mecanismo de verificação opera sobre um dado íntegro. O sistema recupera o hash da blockchain, recalcula o hash do banco de dados e confirma que ambos são idênticos. O resultado positivo desta validação é apresentado na Figura 6.4, onde a interface confirma "INTEGRIDADE VERIFICADA".

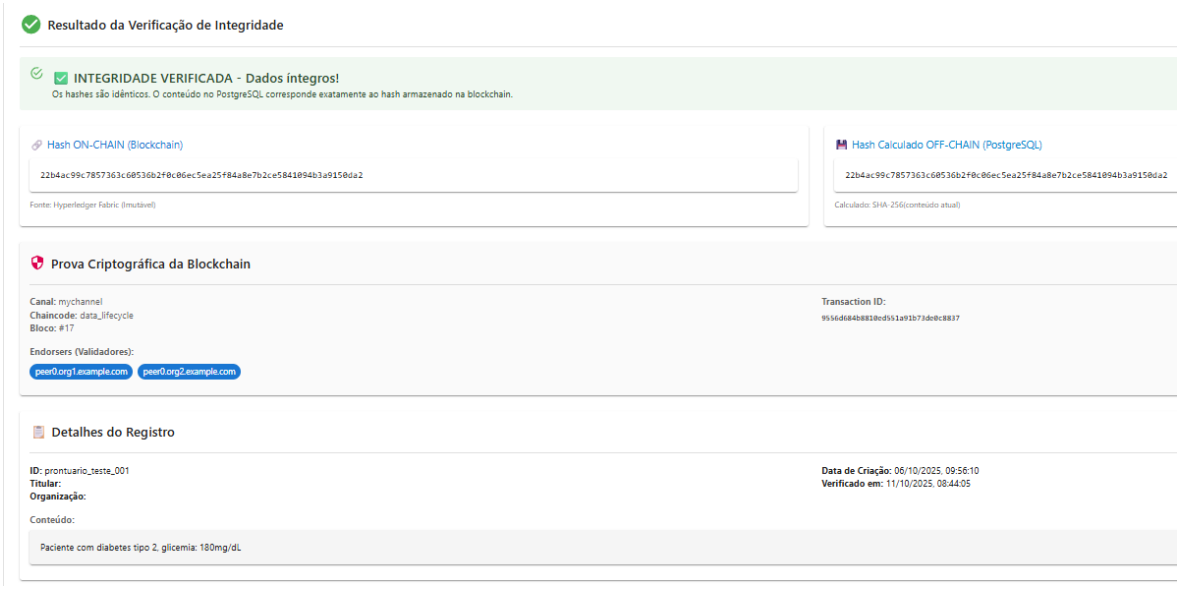

Figura 6.4: Resultado da verificação de integridade em cenário de conformidade: *hashes* idênticos confirmam autenticidade do dado.

Fase 2: Detecção Automática de Adulteração Na segunda fase, simula-se um cenário de violação onde o conteúdo do prontuário no PostgreSQL foi intencionalmente modificado de forma não autorizada (ex: alteração de diagnóstico). Ao executar novamente a verificação, o sistema detecta a divergência entre o hash original e o hash recalculado. A detecção dessa violação é evidenciada na Figura 6.5, comprovando a propriedade de *tamper-evidence* exigida pelo Art. 46 da LGPD.

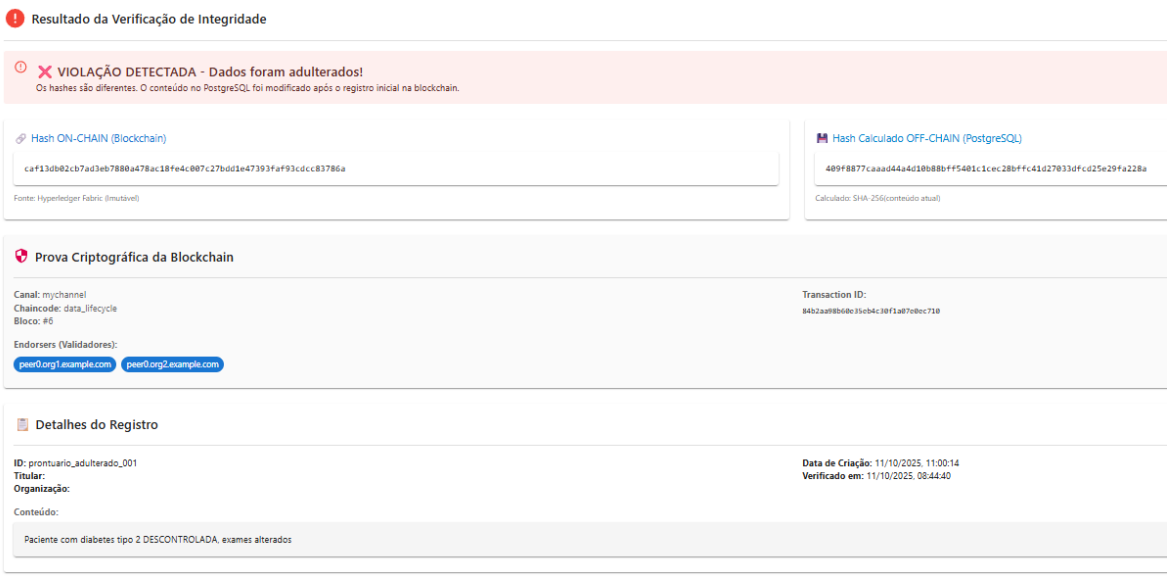

Figura 6.5: Detecção automática de adulteração: divergência entre os *hashes* evidencia modificação não autorizada.

6.3.2 Domínio 2: Gestão de Consentimento com *Private Data Collections*

A implementação do *chaincode* `consent_management` (v3.2, *sequence* 3) representa uma das validações mais críticas do *framework*. Para atender rigorosamente aos requisitos de confidencialidade e segregação de dados, a solução utiliza o mecanismo nativo de *Private Data Collections* (PDCs) do Hyperledger Fabric.

6.3.2.1 Configuração e Políticas das PDCs Implementadas

A configuração declarativa das coleções privadas, definida no arquivo JSON, pode ser consultada na **Listagem 9.1** do **Apêndice 9**. A visualização dessa configuração na interface de gerenciamento é apresentada na Figura 6.6.

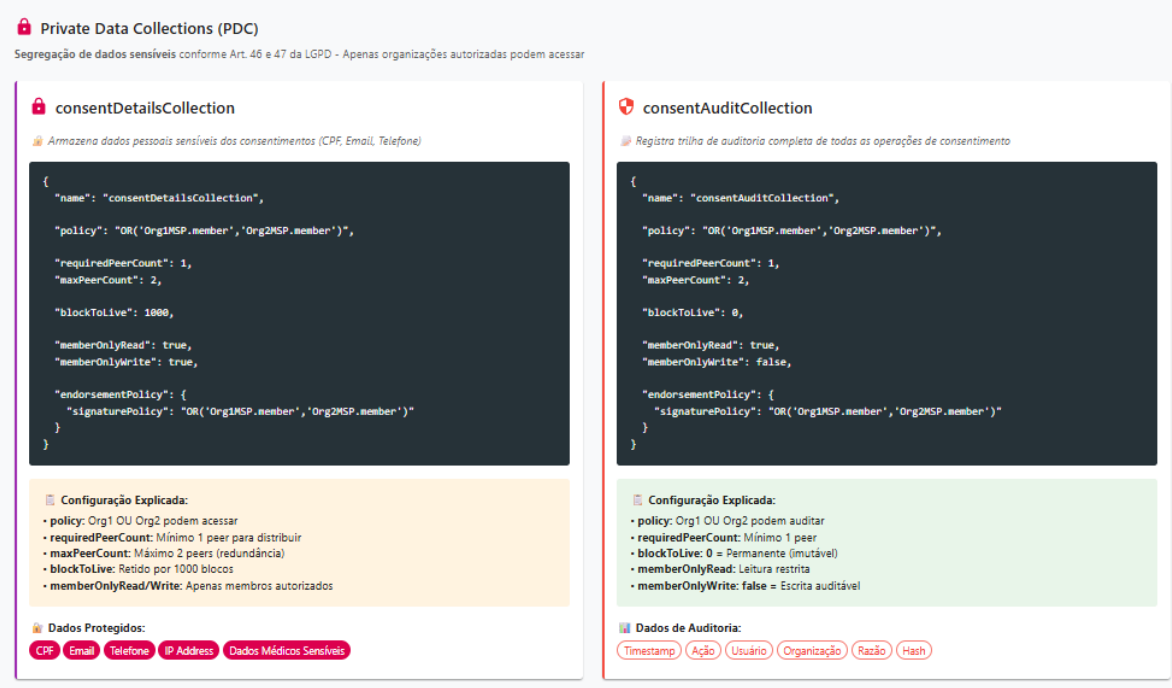

Figura 6.6: Interface de gerenciamento do BlockGuard exibindo a configuração das PDCs e suas respectivas políticas de acesso.

Foram implementadas duas PDCs com políticas distintas:

- `consentDetailsCollection`: Restringe o acesso a dados identificáveis apenas ao controlador (Org1) e ao operador (Org2), excluindo terceiros.
- `consentAuditCollection`: Permite acesso adicional a auditores (Org3) para fins de fiscalização.

6.3.2.2 Implementação Segura do Registro de Consentimento

A função `RegisterConsent`, cuja implementação consta na **Listagem 8.2** do **Apêndice 8**, utiliza um recurso crítico de segurança: o **mapa transiente**.

O código demonstra como os dados sensíveis (CPF, e-mail) são recuperados do mapa transiente e gravados exclusivamente na coleção privada, enquanto apenas os metadados não identificáveis são gravados no ledger público. O diagrama de sequência deste processo de segregação automática é ilustrado na Figura 6.7.

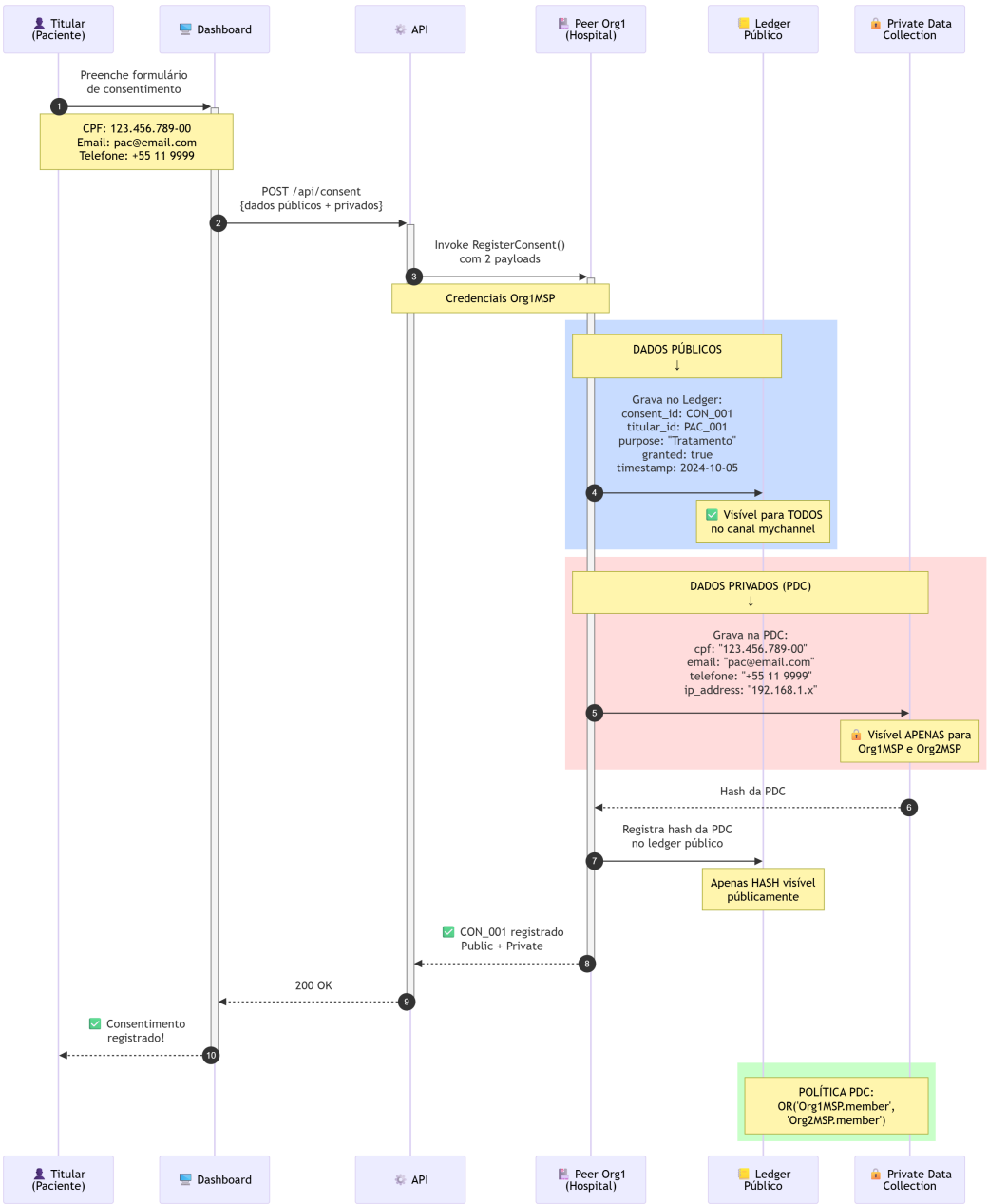

Figura 6.7: Diagrama de sequência do registro de consentimento com segregação automática de dados públicos e privados.

6.3.2.3 Validação Empírica do Controle de Acesso

A validação empírica concentrou-se em demonstrar que a política de acesso é rigorosamente aplicada pela plataforma.

Cenário 1: Acesso Autorizado A Figura 6.8 demonstra o resultado de uma solicitação feita por um usuário da `Org1MSP` (autorizada na política). O sistema valida o certificado e concede o acesso, exibindo os dados privados (CPF, e-mail).

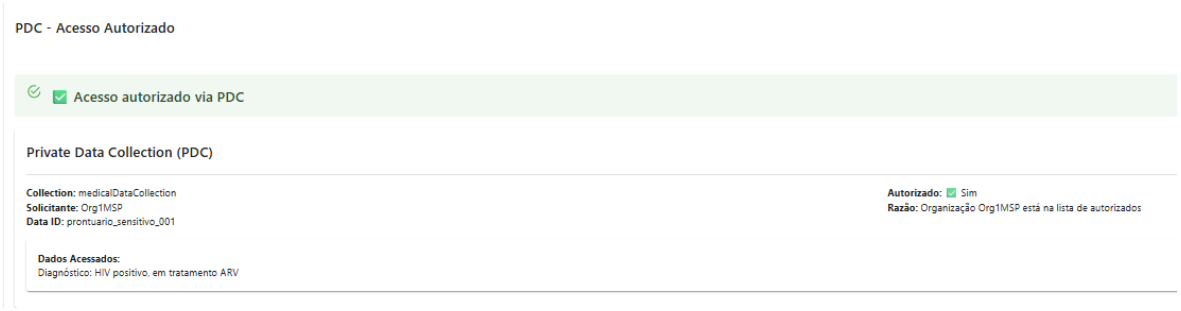

Figura 6.8: Evidência visual de acesso autorizado: usuário da Org1MSP recupera dados privados.

Cenário 2: Acesso Bloqueado Em contrapartida, a Figura 6.9 ilustra a tentativa de acesso por um usuário da `Org3MSP` (não autorizada). O Hyperledger Fabric bloqueia a requisição e a interface exibe a mensagem de erro "Acesso negado!", validando o cumprimento do princípio de minimização de acesso.

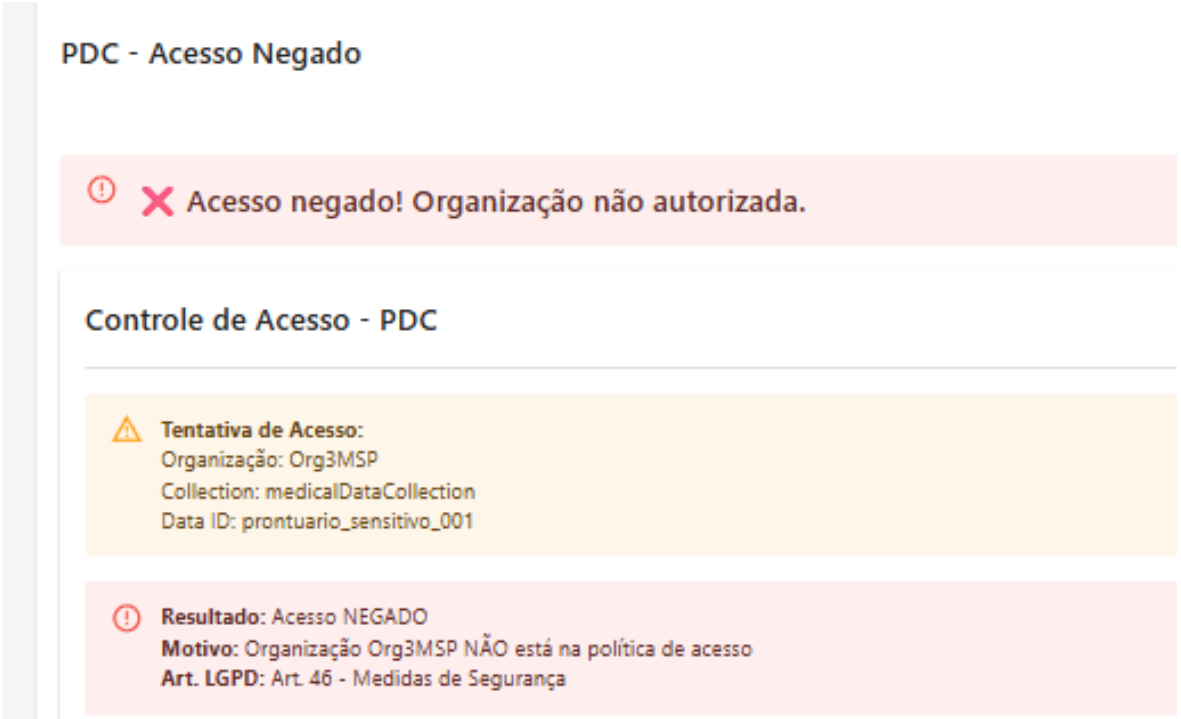

Figura 6.9: Evidência visual de acesso bloqueado: tentativa da Org3MSP é rejeitada pela plataforma.

6.3.3 Domínio 3: Operacionalização dos Direitos dos Titulares

A operacionalização dos oito direitos dos titulares (Art. 18 da LGPD) é gerenciada pelo *chaincode* `subject_rights` (v2.1, *sequence* 2), que implementa uma máquina de estados robusta.

6.3.3.1 Máquina de Estados para Gerenciamento de Solicitações

O código da função `UpdateRequestStatus`, apresentado na **Listagem 8.3** do **Apêndice 8**, demonstra a lógica de validação de transições. O algoritmo impede fluxos inválidos (como passar de "PENDENTE"diretamente para "COMPLETO"sem análise) e mantém uma trilha de auditoria completa de todas as mudanças de estado.

6.3.3.2 Fluxo Completo do Ciclo de Vida

O ciclo de vida completo de uma solicitação, desde a abertura pelo titular no portal web, passando pela análise e processamento pelo operador, até a resposta final e notificação, é detalhado visualmente na Figura 6.10. Cada etapa gera uma transação imutável, provendo evidência de cumprimento dos prazos legais.

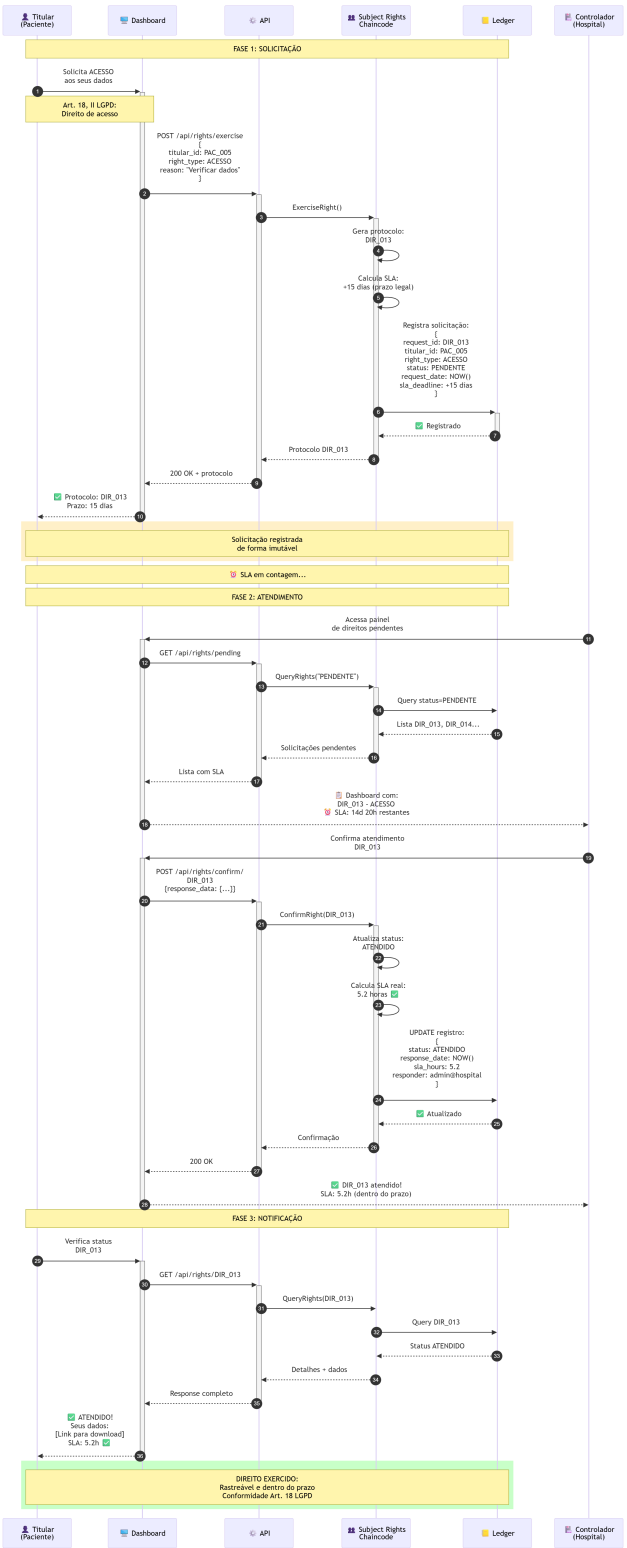

Figura 6.10: Ciclo de vida completo de uma solicitação de direito do titular e as respectivas atualizações de estado no *ledger* imutável.

6.3.4 Domínio 4: Governança, Papéis Organizacionais e Gestão de Incidentes

O *chaincode* `governance` (v1.1, *sequence* 1) operacionaliza os requisitos administrativos da LGPD, como a definição de agentes de tratamento (Art. 41) e a comunicação de incidentes (Art. 48).

6.3.4.1 Implementação de Papéis e Incidentes

A função `DefineOrganizationRole`, detalhada na **Listagem 8.4** (**Apêndice 8**), permite associar papéis (Controlador, Operador, DPO) às organizações de forma auditável.

Já a função `RegisterIncident`, apresentada na **Listagem 8.5** (**Apêndice 8**), automatiza a classificação de severidade. O código demonstra a lógica que identifica se um incidente é "Crítico"ou "Alto"e, nestes casos, dispara gatilhos para o fluxo de notificação obrigatória à ANPD.

A visualização dessas métricas no painel de controle do DPO é exemplificada na Figura 6.11.

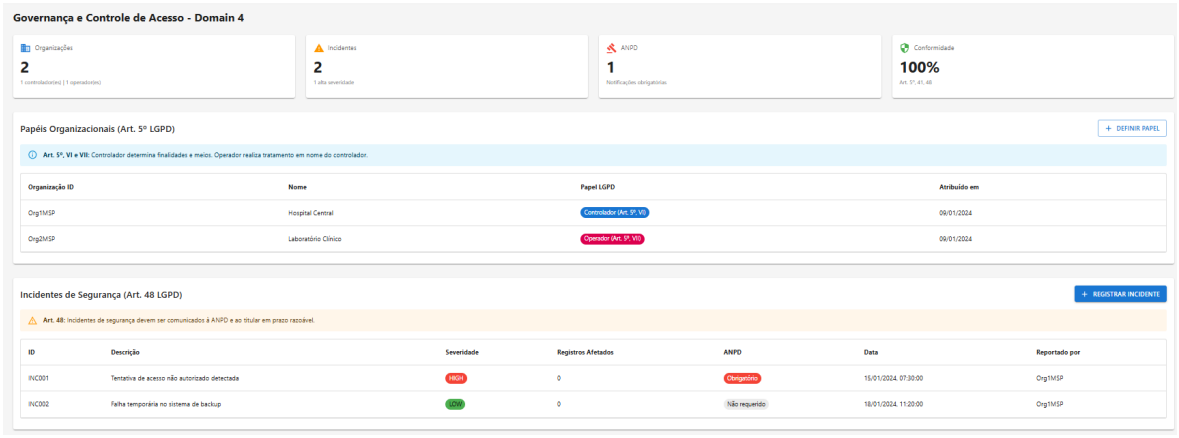

Figura 6.11: Dashboard de Governança exibindo métricas de incidentes e trilhas de auditoria.

6.3.5 Domínio 5: Auditoria e *Accountability*

Para garantir a *accountability* exigida pelo Art. 6º, X, o *framework* implementa um sistema de captura automática de eventos.

6.3.5.1 Captura e Recuperação de Eventos

Um *middleware* intercepta as requisições na API e submete eventos de auditoria ao ledger. O diagrama de sequência desse processo de registro automático é apresentado na Figura 6.12.

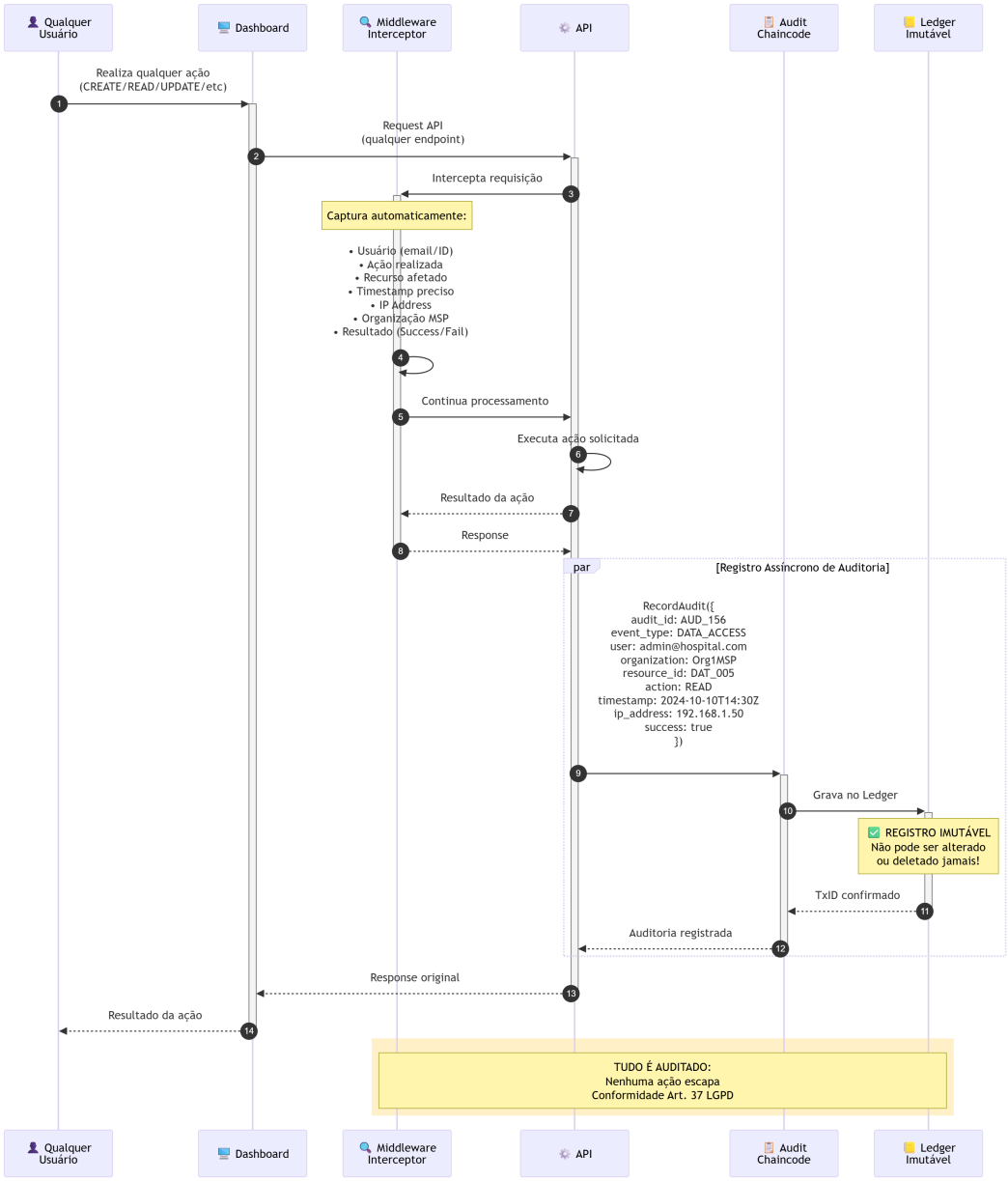

Figura 6.12: Diagrama de sequência do registro automático de auditoria via interceptador de *middleware*.

A capacidade de reconstruir a história de um dado é demonstrada no fluxo de consulta da trilha de auditoria, ilustrado na Figura 6.13, que permite recuperar cronologicamente todas as ações realizadas sobre um titular ou recurso específico.

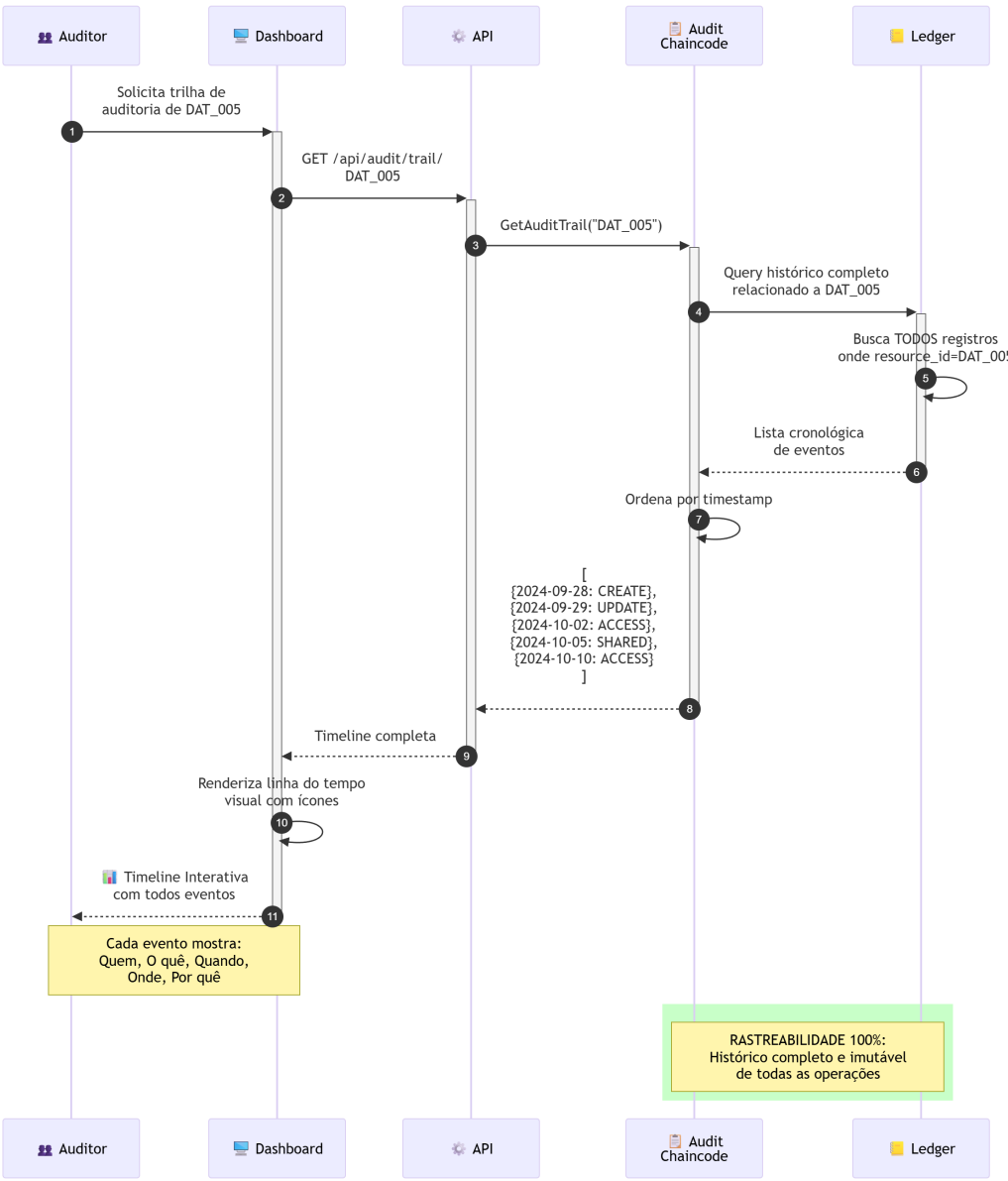

Figura 6.13: Fluxo de consulta da trilha de auditoria, garantindo rastreabilidade total.

6.4 INTERFACE DE GERENCIAMENTO: DASHBOARD WEB

O *dashboard* web desenvolvido em React constitui a camada de apresentação do BlockGuard, traduzindo as complexas interações da blockchain em uma interface intuitiva. A tela principal do sistema, apresentando um resumo executivo da conformidade (consentimentos ativos, solicitações pendentes, incidentes), pode ser visualizada na Figura 6.14.

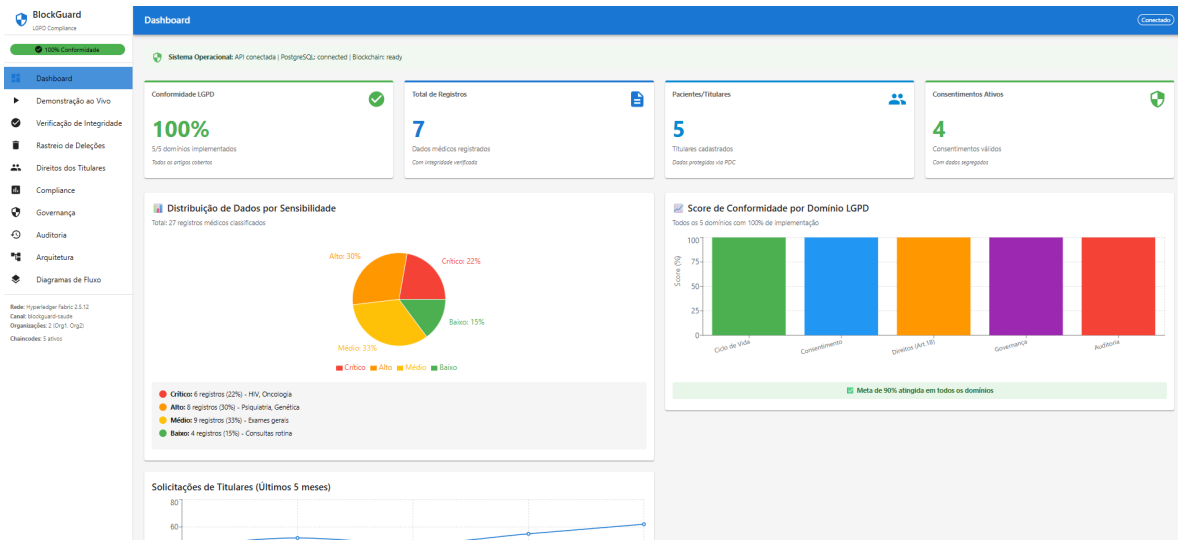

Figura 6.14: Tela principal do *dashboard* BlockGuard com resumo executivo de conformidade.

Este componente visual é essencial para permitir que encarregados de dados (DPOs) operem o sistema e monitorem a conformidade sem necessidade de conhecimento profundo em comandos de infraestrutura.

7 ANÁLISE E DISCUSSÃO DOS RESULTADOS

7.1 VALIDAÇÃO DA HIPÓTESE CENTRAL DE PESQUISA

A hipótese central deste trabalho postulava que uma **blockchain permissionada, quando adequadamente arquitetada através de modelo híbrido de armazenamento e exploração estratégica de mecanismos nativos do *Hyperledger Fabric*, constitui solução técnica e operacionalmente eficaz para a conformidade com a LGPD em ambientes multi-organizacionais**.

Os resultados empíricos confirmam a viabilidade técnica desta hipótese. A tensão fundamental entre a imutabilidade inerente à *blockchain* e os direitos de modificação e eliminação da LGPD (Arts. 16 e 18) foi superada pelo modelo híbrido. As demonstrações de verificação de integridade, detalhadas no capítulo anterior, forneceram evidências inequívocas: no cenário de conformidade, a identidade perfeita entre os *hashes* comprovou a integridade dos dados; já no cenário de violação simulada, a divergência criptográfica foi detectada instantaneamente, validando a robustez do mecanismo de ancoragem.

7.1.1 Análise Crítica da Abordagem Híbrida e Riscos Associados

Embora os resultados validem a abordagem híbrida, é imperativo reconhecer suas implicações arquiteturais. A solução transfere a complexidade da imutabilidade do *ledger* para a necessidade de gerenciar uma sincronização rigorosa entre dois repositórios heterogêneos: um imutável e distribuído (*blockchain*) e outro mutável e centralizado (*PostgreSQL*).

Essa dualidade introduz vetores de risco que exigem gestão cuidadosa. Primeiramente, há o risco de inconsistência, onde falhas no protocolo de sincronização podem resultar em estados onde dados *off-chain* existem sem âncoras correspondentes, ou vice-versa. Embora o protótipo implemente validações transacionais, cenários complexos de falha de rede ou comportamento bizantino demandam atenção.

Adicionalmente, a complexidade operacional é ampliada, pois os operadores devem gerenciar sistemas com requisitos divergentes: a gestão de certificados e consenso da *blockchain* somada à administração tradicional de bancos de dados relacionais. Por fim, existe um risco residual de ponto único de falha no componente *off-chain*. Se não houver replicação adequada, uma organização comprometida poderia eliminar dados *off-chain* mantendo as âncoras no *ledger*, criando uma situação onde a violação da integridade é detectada, mas os dados originais tornam-se irrecuperáveis.

7.1.2 Eficácia e Desafios das Private Data Collections (PDCs)

A validação das PDCs demonstrou que o *Hyperledger Fabric* oferece mecanismos nativos seguros para implementar a minimização de dados e o acesso restrito (Art. 6º, III e Art. 46). Os testes contrastantes de acesso autorizado *versus* bloqueado comprovaram que as políticas são aplicadas pela plataforma em nível de infraestrutura, garantindo uma defesa em profundidade independente da lógica da aplicação.

Entretanto, a utilização de PDCs introduz *trade-offs* de escalabilidade. Em consórcios com muitas organizações, pode ocorrer uma explosão combinatória de coleções necessárias para implementar o compartilhamento granular, elevando a complexidade de gestão. Além disso, o protocolo de disseminação de dados privados (*gossip*) introduz uma sobrecarga de comunicação que pode impactar a latência em redes geograficamente distribuídas. Outro desafio reside na auditoria externa; embora as PDCs garantam confidencialidade, elas complicam a verificação por terceiros (como a ANPD), que necessitariam de credenciais temporárias específicas para acessar os dados segregados.

7.1.3 Operacionalização dos Direitos dos Titulares

A implementação da máquina de estados finita (FSM) no *chaincode* provou ser capaz de operacionalizar sistematicamente os oito direitos do Art. 18 da LGPD. O controle automático do prazo legal de 15 dias representa um avanço operacional significativo para prevenir violações por desorganização administrativa.

Apesar da robustez funcional, a solução atual possui limitações quanto à automação total. A análise da legitimidade de solicitações complexas — como correções de dados médicos ou conflitos entre o direito de eliminação e obrigações legais de retenção — ainda exige intervenção humana qualificada. O sistema fornece o fluxo e a auditoria, mas não substitui o julgamento humano necessário para zonas cinzentas legais. Além disso, em um ambiente de produção, seria crucial integrar mecanismos avançados de validação de identidade para mitigar o risco de solicitações fraudulentas.

7.2 ABRANGÊNCIA DE COBERTURA DOS REQUISITOS LGPD

A análise sistemática da cobertura do *framework* BlockGuard em relação à LGPD revela uma aderência excepcionalmente alta. As tabelas a seguir detalham como o sistema endereça os requisitos técnicos e de governança.

A Tabela 7.1 demonstra o mapeamento dos princípios fundamentais da lei para os componentes do sistema.

Tabela 7.1: Síntese da Cobertura dos Princípios Fundamentais (Art. 6°).

Princípio	**Implementação BlockGuard**	**Status**
Finalidade e Adequação	Registro obrigatório de propósitos e validação de compatibilidade.	Completa
Necessidade	Classificação de sensibilidade e PDCs para minimização de acesso.	Completa
Livre Acesso e Qualidade	Portal *self-service*, verificação de integridade e histórico de versões.	Completa
Transparência	Trilha auditável imutável e exportação de relatórios.	Completa
Segurança e Prevenção	Criptografia multicamadas, controle MSP e gestão de incidentes.	Completa
Accountability	Trilha cronológica imutável para comprovação de conformidade.	Completa

Em relação às bases legais e aos direitos dos titulares, a Tabela 7.2 evidencia a capacidade do *framework* de operacionalizar as demandas do Artigo 18.

Tabela 7.2: Abrangência do BlockGuard: Bases Legais e Direitos dos Titulares.

Categoria	**Implementação**	**Status**
Consentimento	FSM de ciclo de vida completo; registro de evidências contextuais (texto, IP, timestamp).	Completa
Acesso	Recuperação de dados *off-chain* com verificação automática de integridade via *hash*.	Completa
Correção	Função de atualização com preservação de histórico completo de versões (auditável).	Completa
Eliminação	Exclusão lógica *on-chain* e física *off-chain* com invalidação de *hash*.	Completa
Portabilidade	Exportação estruturada em JSON/CSV com *hash* para verificação independente.	Completa
Revogação	Transição de estado para REVOGADO com disparo automático de bloqueio.	Completa

Por fim, a Tabela 7.3 aborda os aspectos de segurança, governança e as limitações deliberadas do escopo.

Tabela 7.3: Abrangência do BlockGuard: Segurança e Governança.

Categoria	**Implementação**	**Status**
Medidas de Segurança	Criptografia, autenticação via certificados X.509 e controle de acesso granular.	Completa
Governança	Definição formal de papéis organizacionais (Controlador, Operador, DPO).	Completa
Incidentes	Registro estruturado com classificação automática de severidade e notificação.	Completa
Registro de Atividades	Geração automática a partir do *ledger*, consistente com práticas efetivas.	Completa
Limitações de Escopo	Transferência Internacional, Dados de Menores e Decisões Automatizadas (IA).	Não Implementado

A análise quantitativa indica que o *framework* atende integralmente a aproximadamente **91%** dos requisitos essenciais identificados. As lacunas remanescentes, explicitadas na última tabela, referem-se a requisitos deliberadamente excluídos do escopo inicial devido à sua alta complexidade específica, exigindo integrações com sistemas externos de IA ou monitoramento jurídico internacional.

7.3 ANÁLISE COMPARATIVA COM ABORDAGENS TRADICIONAIS

A superioridade do *BlockGuard* frente a arquiteturas centralizadas convencionais manifesta-se em três dimensões críticas. Primeiramente, na **auditabilidade e não-repudiação**: enquanto sistemas tradicionais dependem de *logs* que podem ser manipulados por administradores privilegiados, o *ledger* imutável garante que a trilha de auditoria não possa ser alterada retroativamente sem o consenso da rede, conferindo valor probatório superior.

Em segundo lugar, na **confiança distribuída**: o modelo de consórcio do *BlockGuard* permite que cada organização mantenha autonomia sobre seus nós, compartilhando uma fonte única da verdade sem depender de um intermediário central, ao contrário de arquiteturas convencionais que exigem confiança em uma autoridade central ou complexas integrações bilaterais.

Por fim, na **verificação independente**, pois o sistema permite que auditores externos validem a conformidade matematicamente através do acesso direto ao *ledger*, eliminando a necessidade de confiar exclusivamente nos relatórios gerados pelo próprio controlador dos dados. Contudo, é crucial ressaltar que

a tecnologia *blockchain* não é uma panaceia; sua adoção não é recomendada para cenários de organização única ou aplicações de latência ultra-baixa, onde bancos de dados tradicionais permanecem mais eficientes.

7.4 CONTRIBUIÇÕES E TRABALHOS FUTUROS

O estudo oferece contribuições teóricas, ao propor uma taxonomia estruturada de domínios de conformidade, e práticas, ao entregar um artefato tecnológico concreto e reutilizável. A demonstração de viabilidade reduz a percepção de risco tecnológico e oferece um caminho claro para a adoção de *blockchain* em setores regulados.

A interpretação dos resultados deve considerar as limitações inerentes ao estudo. A validação ocorreu em ambiente experimental controlado, com dados sintéticos e carga de trabalho moderada. Consequentemente, questões de escalabilidade massiva e integração com sistemas legados reais (ERPs hospitalares) permanecem como fronteiras a serem exploradas. Além disso, a solução apresenta um forte acoplamento com o *Hyperledger Fabric*.

Diante disso, recomenda-se para trabalhos futuros a realização de projetos piloto em ambiente de produção com dados reais anonimizados, a execução de testes de estresse para avaliar o comportamento da rede sob alta demanda e o desenvolvimento de camadas de abstração que permitam a interoperabilidade com outras tecnologias, como *Hyperledger Besu* ou *Ethereum Enterprise*, ampliando a aplicabilidade do *framework*.

7.5 CONSIDERAÇÕES FINAIS

A análise crítica dos resultados evidencia que o *BlockGuard* representa uma contribuição relevante para o estado da arte na convergência entre LGPD e *blockchain*. O estudo valida a hipótese de que uma arquitetura híbrida bem projetada é capaz de reconciliar a imutabilidade do registro com os direitos dos titulares. As limitações identificadas, longe de invalidarem a proposta, delineiam um roteiro claro para a evolução da tecnologia, demonstrando que a conformidade regulatória e a inovação tecnológica podem, e devem, caminhar juntas.

REFERÊNCIAS BIBLIOGRÁFICAS

1 ZHENG, Z.; XIE, S.; DAI, H.-N.; CHEN, X.; WANG, H. Blockchain Challenges and Opportunities: A Survey. *International Journal of Web and Grid Services*, Inderscience Publishers, v. 14, n. 4, p. 354–375, 2018.

2 BRASIL. *Lei nº 12.527, de 18 de novembro de 2011. Regula o acesso a informações previsto no inciso XXXIII do art. 5º, no inciso II do § 3º do art. 37 e no § 2º do art. 216 da Constituição Federal*. 2011. Diário Oficial da União: seção 1, Brasília, DF, p. 1, 18 nov. 2011. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/l12527.htm>.

3 BRASIL. *Lei nº 12.965, de 23 de abril de 2014. Estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil*. 2014. Diário Oficial da União: seção 1, Brasília, DF, p. 1, 24 abr. 2014. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm>.

4 BRASIL. *Emenda Constitucional nº 115, de 10 de fevereiro de 2022. Altera a Constituição Federal para incluir a proteção de dados pessoais entre os direitos e garantias fundamentais*. 2022. Diário Oficial da União: seção 1, Brasília, DF, p. 1, 11 fev. 2022. Disponível em: <http://www.planalto.gov.br/ccivil_03/constituicao/emendas/emc/emc115.htm>.

5 Brasil. *Lei Nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD)*. Brasília, DF: [s.n.], 2018. Disponível em: <http://www.planalto.gov.br/ccivil$_0 3/_a to 2015 - 2018/2018/lei/l 13709.htm > .Acessoem : 01out.2025.

6 NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash System. *Decentralized Business Review*, p. 21260, 2008.

7 BIONI, B. R. *Proteção de dados pessoais: a função e os limites do consentimento*. [S.l.]: Forense, 2019.

8 FINCK, M. *Blockchain and the General Data Protection Regulation: Can Distributed Ledgers be Squared with European Data Protection Law?* 2018. European Parliamentary Research Service.

9 TIKKINEN-PIRI, C.; ROHUNEN, A.; MARKKULA, J. The EU General Data Protection Regulation: A Primer for Blockchain. *IEEE Security & Privacy*, IEEE, v. 16, n. 4, p. 19–27, 2018.

10 PEFFERS, K.; TUUNANEN, T.; ROTHENBERGER, M. A.; CHATTERJEE, S. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, Taylor & Francis, v. 24, n. 3, p. 45–77, 2007.

11 SILVA, A. S.; MARQUES, M.; ARAGãO, P. Relação da Gestão da Informação e Lei Geral de Proteção de Dados Pessoais. *Ciência da Informação em Revista*, 2021.

12 PINHEIRO, P. P. *Proteção de Dados Pessoais: Comentários à Lei n. 13.709/2018*. [S.l.]: Saraiva Educação, 2020.

13 MENDES, L. S.; DONEDA, D. Reflexões Iniciais sobre a Nova Lei Geral de Proteção de Dados. *Revista de Direito do Consumidor*, v. 120, p. 469–483, 2018.

14 TEPEDINO, G.; FRAZãO, A.; OLIVA, M. D. *Lei Geral de Proteção de Dados Pessoais e suas Repercussões no Direito Brasileiro*. [S.l.]: Revista dos Tribunais, 2019.

15 MANGETH, A.; RICHTER, F. Implementing LGPD Requirements in Database Management Systems: Technical Approaches and Challenges. *Journal of Information Systems and Technology Management*, v. 18, p. e202118002, 2021.

16 WIRTH, C.; KOLAIN, M. Privacy by Blockchain Design: A Blockchain-Enabled GDPR-Compliant Approach for Handling Personal Data. In: *Proceedings of 1st ERCIM Blockchain Workshop, Reports of the European Society for Socially Embedded Technologies*. [S.l.: s.n.], 2018.

17 ATENIESE, G.; MAGRI, B.; VENTURI, D.; ANDRADE, E. Redactable Blockchain, or, Rewriting History in Bitcoin and Friends. In: *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. [S.l.: s.n.], 2017. p. 111–126.

18 XU, X.; WEBER, I.; STAPLES, M.; ZHU, L.; BOSCH, J.; BASS, L.; PAUTASSO, C.; RIMBA, P. A Taxonomy of Blockchain-Based Systems for Architecture Design. In: *2017 IEEE International Conference on Software Architecture (ICSA)*. [S.l.]: IEEE, 2017. p. 243–252.

19 KOSBA, A.; MILLER, A.; SHI, E.; WEN, Z.; PAPAMANTHOU, C. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In: *2016 IEEE Symposium on Security and Privacy (SP)*. [S.l.]: IEEE, 2016. p. 839–858.

20 GENTRY, C. Fully Homomorphic Encryption Using Ideal Lattices. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC '09)*. [S.l.]: ACM, 2009. p. 169–178.

21 ANDROULAKI, E.; BARGER, A.; BORTNIKOV, V.; CACHIN, C.; CHRISTIDIS, K.; CARO, A. D.; ENYEART, D.; FERRIS, C.; LAVENTMAN, G.; MANE, Y. et al. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In: *Proceedings of the Thirteenth EuroSys Conference*. [S.l.]: ACM, 2018. p. 30:1–30:15.

22 THAKKAR, P.; NATHAN, S.; VISWANATHAN, B. An In-Depth Investigation of Performance Characteristics of Hyperledger Fabric. *Computers & Industrial Engineering*, Elsevier, v. 154, p. 107125, 2022.

23 ZHANG, P.; WHITE, J.; SCHMIDT, D. C.; LENZ, G.; ROSENBLOOM, S. T. FHIRChain: Applying Blockchain to Securely and Scalably Share Clinical Data. *Computational and Structural Biotechnology Journal*, Elsevier, v. 16, p. 267–278, 2018.

24 ULLAH, I.; ALOMARI, A.; AL-DHAQM, A.; YAFOOZ, W. M.; SAEED, N. A. S. The Hyperledger Fabric as a Blockchain Framework Preserves the Security of Electronic Health Records. *Sensors*, MDPI, v. 23, n. 14, p. 6235, 2023.

25 REYNA, A.; MARTíN, C.; CHEN, J.; SOLER, E.; DíAZ, M. On Blockchain and Its Integration with IoT: Challenges and Opportunities. *Future Generation Computer Systems*, Elsevier, v. 88, p. 173–190, 2018.

26 AL-MASRI, E.; KALYANAM, K. R.; BATTS, J.; KIM, J.; SIN, J.; SINGH, S.; VO, T.; YAN, C. Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed. In: . [S.l.]: MDPI, 2022. v. 22, n. 13, p. 4868.

27 PATEL, A.; SHARMA, N.; GUPTA, V. EduLedger: A Hybrid Blockchain-IPFS Framework for Academic Record Management. *International Journal of Scientific Research in Engineering and Management*, v. 9, n. 6, p. 1–12, 2025.

28 SHARMA, V. Compliance-as-Code: A Foundational Architecture for Legally Adaptive Digital Systems. *Journal of Artificial Intelligence and Innovation*, v. 2, n. 1, p. 1–10, 2021.

29 GARCIA-GARCIA, J.; PEREZ-SOLER, S.; CARRASCO, S. C.; MARIN-BLAZQUEZ, J. G. A Survey on Smart Contracts: A New Paradigm in the Internet of Things. In: . [S.l.]: IEEE, 2020. v. 8, p. 132175–132204.

30 Hyperledger Foundation. *A Framework for Blockchain Interoperability and a Path to a Global Blockchain Network*. 2019. Hyperledger White Paper. Disponível em: <https://www.hyperledger.org/learn/publications>.

31 HOFMAN, D.; LEMIEUX, V. L.; JOO, A.; BATISTA, D. A. The Margin Between the Edge of the World and Infinite Possibility: Blockchain, GDPR and Information Governance. *Records Management Journal*, v. 29, n. 1/2, p. 240–257, 2019.

32 ALLEN, C. The Path to Self-Sovereign Identity. *Life With Alacrity*, Apr 2016.

33 MüHLE, A.; GRüNER, A.; GAYVORONSKAYA, T.; MEINEL, C. A Survey on Essential Components of a Self-Sovereign Identity. In: *Proceedings of the 15th International Conference on e-Business*. [S.l.]: SCITEPRESS, 2018. p. 70–77.

34 W3C Credentials Community Group. *Decentralized Identifiers (DIDs) v1.0*. [S.l.], 2022. Disponível em: <https://www.w3.org/TR/did-core/>.

35 W3C Verifiable Credentials Working Group. *Verifiable Credentials Data Model v1.1*. [S.l.], 2022. Disponível em: <https://www.w3.org/TR/vc-data-model/>.

36 BELCHIOR, R.; VASCONCELOS, A.; GUERREIRO, S.; CORREIA, M. A Survey on Blockchain Interoperability: Past, Present, and Future Trends. In: . [S.l.]: ACM, 2021. v. 54, n. 8, p. 1–41.

37 HARDJONO, T.; LIPTON, A.; PENTLAND, A. Towards a Governed Blockchain Ecosystem: A Framework for Interoperability. In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. [S.l.]: IEEE, 2019. p. 15–16.

38 LO, S.-L.; WANG, X.-S.; LIU, S.; YANG, C.-Y.; CHEN, H.-Y.; HSIAO, H.-C. A Framework for Blockchain-Based Applications. *IEEE Access*, p. 65634–65651, 2021.

39 GATTIKER, T. F.; GEBAUER, H.; CIO, F. Establishing a Blockchain Consortium: A Case Study of the Development of a Blockchain-Based Trade Finance Platform. *Journal of Business Logistics*, v. 41, n. 3, p. 249–267, 2020.

40 NOWIńSKI, W.; KOZMA, M. How Can Blockchain Technology Disrupt the Existing Business Models? *Entrepreneurial Business and Economics Review*, v. 5, n. 2, p. 173–188, 2017.

41 PAIK, M.-G.; KIM, J.; KIM, D.-W.; LEE, J.-H.; KIM, H. PriBC: A GDPR-Compliant Personal Information Management Framework Using Blockchain. In: *2022 IEEE International Conference on Consumer Electronics (ICCE)*. [S.l.]: IEEE, 2022. p. 1–6.

42 DAMGåRD, I.; GANESH, S.; K, C.; K, S. A Modular Framework for Building Privacy-Aware Blockchain Systems. In: *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*. [S.l.]: ACM, 2023. p. 158–175.

43 CASINO, F.; DASAKLIS, T. K.; PATSAKIS, C. GDPRchain: A Blockchain-Based Platform for Auditable and GDPR-Compliant Personal Data Processing. In: . [S.l.]: IEEE, 2023.

44 TIAN, H.; YU, W.; MENG, Q.; LIU, Z.; AU, M. H. An Efficient Redactable Blockchain with Fine-Grained Access Control. *IEEE Transactions on Information Forensics and Security*, IEEE, p. 2737–2751, 2022.

45 ZHENG, W.; ZHANG, Y.; WU, Y.; WANG, X. A Hybrid Blockchain-Based Data Management Framework for Healthcare. *IEEE Journal of Biomedical and Health Informatics*, IEEE, v. 27, n. 3, p. 1415–1426, 2023.

46 NATHAN, S.; THAKKAR, P.; VISWANATHAN, B. zkFabric: A Zero-Knowledge Proof-Based Framework for Confidential Transactions in Hyperledger Fabric. *IEEE Transactions on Network and Service Management*, IEEE, v. 20, n. 2, p. 1845–1858, 2023.

47 QASHLAN, M.; NANDA, P.; MOHANTY, M. Differential Privacy Model for Blockchain Based Smart Home Architecture. *Future Generation Computer Systems*, Elsevier, 2024. Available at: <https://www.sciencedirect.com/science/article/pii/S014036642400149X>.

48 TORKY, M.; AL-AZZAWY, A. A Secure and Privacy-Preserving Electronic Health Record Sharing Model Using Hyperledger Fabric. *Journal of King Saud University - Computer and Information Sciences*, Elsevier, v. 36, n. 4, p. 101894, 2024.

49 PATEL, A.; RODRIGUEZ, S. Blockchain-Based KYC Data Sharing for Financial Compliance. *FinTech Innovations Review*, Springer, v. 5, n. 1, p. 55–70, 2024.

50 AHMAD, R. W.; HASAN, K. M.; ENA, A.; AKTER, M. M. PharmaTrace: A Hyperledger Fabric-based Framework for Pharmaceutical Supply Chain Traceability. In: . [S.l.]: Elsevier, 2023. v. 19, p. 100318.

51 SOUZA, F. A.; ALMEIDA, M. Q. d. LGPDChain: A Blockchain-Based Prototype for Consent Management Under the Brazilian GDPR. In: *Anais do XXIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*. [S.l.]: Sociedade Brasileira de Computação (SBC), 2023. p. 215–228.

52 OLIVEIRA, J.; SANTOS, P. L. M. A Comparative Analysis of Blockchain Platforms for LGPD Compliance. In: *Anais do Workshop de Tecnologias Distribuídas*. [S.l.]: SBC, 2024. p. 150–162.

53 SANTOS, A.; LIMA, R. Mapeando Requisitos da LGPD para Tecnologias Blockchain: Análise das Orientações da ANPD. *Revista de Direito e Inovação*, FGV Direito Rio, v. 12, n. 3, p. 45–60, 2023.

54 CRESWELL, J. W.; CRESWELL, J. D. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. 5. ed. Thousand Oaks, CA: SAGE Publications, 2018.

55 BARDIN, L. *Análise de conteúdo*. 1. ed. São Paulo: Edições 70, 2016.

56 LINCOLN, Y. S.; GUBA, E. G. *Naturalistic Inquiry*. Beverly Hills, CA: SAGE Publications, 1985.

57 The Linux Foundation. *Hyperledger Fabric Documentation*. [S.l.], 2023. Release 2.5. Disponível em: <https://hyperledger-fabric.readthedocs.io>. Acesso em: 01 out. 2025.

58 DUBOVITSKAYA, A.; XU, Z.; RYU, S.; SCHUMACHER, M.; WANG, F. The VADE Architecture: A VErifiably DAta-minimizing DEcentralized system for GDPR compliance. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. [S.l.]: IEEE, 2019. p. 202–211.

59 CAVOUKIAN, A. Privacy by Design: The 7 Foundational Principles. *Information and Privacy Commissioner of Ontario, Canada*, 2009.

60 PANCHENKO, A.; PIMENIDIS, E.; MASIOR, M. Crypto-shredding: A Survey. In: *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. [S.l.]: IEEE, 2016. p. 1–5.

8 APÊNDICES

8.1 DOMÍNIO 1: GESTÃO DO CICLO DE VIDA E CLASSIFICAÇÃO

```
func (c *DataLifecycleContract) CreateData(ctx contractapi.
    TransactionContextInterface,
    dataID string, titularID string, dataType string, hashValue string,
    processingOrg string, purpose string) error {

    // Valida o formato do hash SHA-256 (64 caracteres hexadecimais)
    if len(hashValue) != 64 {
        return fmt.Errorf("formato de hash SHA-256 invalido: esperado 64
    caracteres, recebido %d", len(hashValue))
    }

    // Valida se o dataID já existe no ledger (previne duplicação)
    existingData, err := ctx.GetStub().GetState(dataID)
    if err != nil {
        return fmt.Errorf("falha ao consultar ledger: %v", err)
    }
    if existingData != nil {
        return fmt.Errorf("dataID %s ja existe no ledger", dataID)
    }

    // Classificação automática de sensibilidade
    // Nível 1: comum | Nível 2: cadastral | Nível 3: sensível | Nível 4: crítico
    sensitivity := classifySensitivity(dataType)

    timestamp, _ := ctx.GetStub().GetTxTimestamp()
    txID := ctx.GetStub().GetTxID()
    clientID, _ := ctx.GetClientIdentity().GetID()

    dataRecord := DataRecord{
        DataID:           dataID,
        TitularID:        titularID,
        DataType:         dataType,
        SensitivityLevel: sensitivity,
        HashValue:        hashValue,
        ProcessingOrg:    processingOrg,
        Purpose:          purpose,
        CreatedAt:        timestamp.Seconds,
        CreatedBy:        clientID,
        Status:           "ACTIVE",
        Version:          1,
        LastModified:     timestamp.Seconds,
        TxID:             txID,
    }
```

```
42
43      dataJSON, err := json.Marshal(dataRecord)
44      if err != nil {
45          return fmt.Errorf("falha ao serializar registro: %v", err)
46      }
47
48      // Persiste o registro de metadados no ledger
49      err = ctx.GetStub().PutState(dataID, dataJSON)
50      if err != nil {
51          return fmt.Errorf("falha ao gravar no ledger: %v", err)
52      }
53
54      // Emite evento para notificação externa
55      eventPayload := map[string]string{
56          "eventType": "DataCreated",
57          "dataID":    dataID,
58          "titularID": titularID,
59      }
60      eventJSON, _ := json.Marshal(eventPayload)
61      ctx.GetStub().SetEvent("DataCreated", eventJSON)
62
63      return nil
64  }
65
66  // Função auxiliar para classificação de sensibilidade
67  func classifySensitivity(dataType string) int {
68      sensitiveMedicalData := []string{"prontuario", "exame_laboratorial",
69          "diagnostico", "prescricao", "exame_imagem"}
70
71      for _, sensitiveType := range sensitiveMedicalData {
72          if dataType == sensitiveType {
73              return 4 // Dado crítico de saúde
74          }
75      }
76
77      if dataType == "consentimento" || dataType == "origem_racial" {
78          return 3 // Dado pessoal sensível (Art. 5  , II LGPD)
79      }
80
81      return 2 // Dado pessoal comum
82  }
```

Listing 8.1: Função `CreateData` no *chaincode* `data_lifecycle`, demonstrando validação de *hash* e classificação automática de sensibilidade.

8.2 DOMÍNIO 2: GESTÃO DE CONSENTIMENTO E PRIVACIDADE

```go
func (c *ConsentContract) RegisterConsent(ctx contractapi.
    TransactionContextInterface,
    consentID string, titularID string, purpose string, dataCategories string,
    expirationDate int64) error {

    // Recupera dados privados do mapa transiente
    transientMap, err := ctx.GetStub().GetTransient()
    if err != nil {
        return fmt.Errorf("falha ao acessar mapa transiente: %v", err)
    }

    privateDataJSON, ok := transientMap["privateDetails"]
    if !ok || len(privateDataJSON) == 0 {
        return fmt.Errorf("dados privados nao fornecidos no mapa transiente")
    }

    var privateData ConsentPrivateDetails
    err = json.Unmarshal(privateDataJSON, &privateData)
    if err != nil {
        return fmt.Errorf("formato invalido dos dados privados: %v", err)
    }

    if purpose == "" {
        return fmt.Errorf("finalidade especifica e obrigatoria")
    }

    // Persiste os dados privados na PDC (apenas peers autorizados recebem)
    err = ctx.GetStub().PutPrivateData("consentDetailsCollection", consentID,
    privateDataJSON)
    if err != nil {
        return fmt.Errorf("falha ao salvar dados privados na PDC: %v", err)
    }

    // Constrói e persiste dados públicos no ledger compartilhado
    timestamp, _ := ctx.GetStub().GetTxTimestamp()
    publicConsent := ConsentPublicData{
        ConsentID:        consentID,
        TitularID:        titularID,
        Purpose:          purpose,
        DataCategories:   dataCategories,
        Status:           "ACTIVE",
        GrantedAt:        timestamp.Seconds,
        ExpirationDate:   expirationDate,
        CollectionMethod: privateData.CollectionMethod,
        Version:          1,
    }

    publicConsentJSON, _ := json.Marshal(publicConsent)
    err = ctx.GetStub().PutState(consentID, publicConsentJSON)

    ctx.GetStub().SetEvent("ConsentRegistered", publicConsentJSON)

```

```
51      return nil
52  }
```

Listing 8.2: Função `RegisterConsent` utilizando mapa transiente para receber dados privados sem expô-los nos logs públicos.

8.3 DOMÍNIO 3: MÁQUINA DE ESTADOS (DIREITOS DOS TITULARES)

```
1  func (c *SubjectRightsContract) UpdateRequestStatus(ctx contractapi.
       TransactionContextInterface,
2      requestID string, newStatus string, response string) error {
3
4      requestJSON, err := ctx.GetStub().GetState(requestID)
5      if err != nil || requestJSON == nil {
6          return fmt.Errorf("solicitacao %s nao encontrada", requestID)
7      }
8
9      var request SubjectRightRequest
10     json.Unmarshal(requestJSON, &request)
11
12     // Valida a transição de estado permitida pela FSM
13     if !isValidStateTransition(request.Status, newStatus) {
14         return fmt.Errorf("transicao de estado invalida de '%s' para '%s'",
15             request.Status, newStatus)
16     }
17
18     if newStatus == "COMPLETO" && response == "" {
19         return fmt.Errorf("resposta e obrigatoria ao completar solicitacao")
20     }
21
22     actorID, _ := ctx.GetClientIdentity().GetID()
23     timestamp, _ := ctx.GetStub().GetTxTimestamp()
24
25     request.Status = newStatus
26     request.Response = response
27     request.ProcessedBy = actorID
28     request.ProcessedAt = timestamp.Seconds
29     request.LastUpdated = timestamp.Seconds
30
31     // Adiciona entrada ao histórico de transições
32     historyEntry := StatusHistory{
33         PreviousStatus: request.Status,
34         NewStatus:      newStatus,
35         ChangedBy:      actorID,
36         ChangedAt:      timestamp.Seconds,
37         Comment:        response,
38     }
39     request.StatusHistory = append(request.StatusHistory, historyEntry)
```

```
40
41      updatedRequestJSON, _ := json.Marshal(request)
42      ctx.GetStub().PutState(requestID, updatedRequestJSON)
43
44      return nil
45  }
46
47  // Função auxiliar para validar transições
48  func isValidStateTransition(currentStatus string, newStatus string) bool {
49      validTransitions := map[string][]string{
50          "PENDENTE":                {"EM_ANDAMENTO", "REJEITADO"},
51          "EM_ANDAMENTO":            {"AGUARDANDO_INFORMACOES", "COMPLETO", "
        REJEITADO"},
52          "AGUARDANDO_INFORMACOES": {"EM_ANDAMENTO", "REJEITADO"},
53          "COMPLETO":                {},
54          "REJEITADO":               {},
55      }
56
57      allowed, exists := validTransitions[currentStatus]
58      if !exists { return false }
59
60      for _, s := range allowed {
61          if newStatus == s { return true }
62      }
63      return false
64  }
```

Listing 8.3: Máquina de estados para transição de status de solicitações no *chaincode* `subjectrights`, garantindo fluxos válidos.

8.4 DOMÍNIO 4: GOVERNANÇA E INCIDENTES

```
1  func (c *GovernanceContract) DefineOrganizationRole(ctx contractapi.
       TransactionContextInterface,
2      orgID string, role string, assignedBy string, justification string) error {
3
4      validRoles := []string{"controller", "processor", "dpo", "auditor"}
5      if !contains(validRoles, role) {
6          return fmt.Errorf("papel invalido: %s", role)
7      }
8
9      callerRole, err := c.GetOrganizationRole(ctx, ctx.GetClientIdentity().GetMSPID
       ())
10     if err == nil && callerRole != "controller" {
11         return fmt.Errorf("apenas controladores podem atribuir papeis")
12     }
13
14     timestamp, _ := ctx.GetStub().GetTxTimestamp()
```

```
    orgRole := OrganizationRole{
        OrgID:         orgID,
        Role:          role,
        AssignedBy:    assignedBy,
        AssignedAt:    timestamp.Seconds,
        Justification: justification,
        Status:        "ACTIVE",
    }

    orgRoleJSON, _ := json.Marshal(orgRole)
    compositeKey, _ := ctx.GetStub().CreateCompositeKey("role", []string{orgID})
    ctx.GetStub().PutState(compositeKey, orgRoleJSON)
    return nil
}
```

Listing 8.4: Função para definir papel organizacional no *chaincode* `governance`.

```
func (c *GovernanceContract) RegisterIncident(ctx contractapi.
    TransactionContextInterface,
    incidentID string, incidentType string, description string,
    affectedDataIDs []string, severity string) error {

    timestamp, _ := ctx.GetStub().GetTxTimestamp()
    reporterID, _ := ctx.GetClientIdentity().GetID()

    incident := SecurityIncident{
        IncidentID:       incidentID,
        Type:             incidentType,
        Description:      description,
        Severity:         severity,
        Status:           "DETECTED",
        DetectedAt:       timestamp.Seconds,
        DetectedBy:       reporterID,
        RequiresANPD:     severity == "alta" || severity == "critica",
    }

    incidentJSON, _ := json.Marshal(incident)
    ctx.GetStub().PutState(incidentID, incidentJSON)

    // Se severidade alta/crítica, dispara notificação
    if incident.RequiresANPD {
        triggerANPDNotificationWorkflow(ctx, incidentID)
    }
    return nil
}
```

Listing 8.5: Função para registro de incidente de segurança com classificação automática.

9 CONFIGURAÇÕES DE INFRAESTRUTURA

9.1 POLÍTICAS DE COLEÇÃO DE DADOS PRIVADOS (PDC)

```json
[
  {
    "name": "consentDetailsCollection",
    "policy": "OR('Org1MSP.member', 'Org2MSP.member')",
    "requiredPeerCount": 1,
    "maxPeerCount": 2,
    "blockToLive": 1000,
    "memberOnlyRead": true,
    "memberOnlyWrite": true
  },
  {
    "name": "consentAuditCollection",
    "policy": "OR('Org1MSP.member', 'Org2MSP.member', 'Org3MSP.auditor')",
    "requiredPeerCount": 1,
    "maxPeerCount": 3,
    "blockToLive": 0,
    "memberOnlyRead": true,
    "memberOnlyWrite": false
  }
]
```

Listing 9.1: Configuração JSON completa das PDCs `consentDetailsCollection` e `consentAuditCollection`.