Escalabilidade de Dados em Blockchains Permissionadas por Meio de Poda Auditável: Uma Avaliação em Hyperledger Fabric

Arthur G. C. Milanez[1], **Lucas M. Palma**[1], **Jean E. Martina**[1]

[1]Departamento de Informática e Estatística
Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

arthur.crippa@posgrad.ufsc.br

lucas.palma@ufsc.br, jean.martina@ufsc.br

Abstract. *In permissioned blockchains, such as Hyperledger Fabric, full ledger replication among peers increases storage costs as the network grows. Although the state of the art proposes techniques to reduce these costs, few approaches combine storage reduction with continuous mechanisms for checking the custody of removed data. This work proposes a collaborative pruning architecture that integrates on-chain coordination, Reed-Solomon coding, off-chain storage, verifiable metadata, and periodic sample-based custody audits. Although the evaluation was conducted in a local environment, the functional prototype achieved more than 90% local storage reduction in the evaluated scenario while preserving reconstruction from k valid fragments and low-cost probabilistic detection of partial losses.*

Resumo. *Em blockchains permissionadas, como o Hyperledger Fabric, a replicação integral do ledger entre peers aumenta o custo de armazenamento à medida que a rede cresce. Embora o estado da arte proponha técnicas para reduzir esse custo, poucas abordagens combinam redução de armazenamento com mecanismos contínuos de verificação da custódia dos dados removidos. Este trabalho propõe uma arquitetura de poda colaborativa que integra coordenação on-chain, codificação Reed-Solomon, armazenamento off-chain, metadados verificáveis e auditorias amostrais de custódia. Apesar de a avaliação ter sido conduzida em ambiente local, o protótipo funcional indica redução superior a 90% no armazenamento local no cenário avaliado, preservando reconstrução a partir de k fragmentos válidos e detecção probabilística de baixo custo para perdas parciais.*

1. Introdução

Redes *blockchain* mantêm um histórico crescente de blocos, o que torna o armazenamento uma dimensão crítica de escalabilidade. Em redes permissionadas, como o Hyperledger Fabric, cada nó tende a manter uma cópia local do ledger. Esse modelo favorece verificabilidade, mas aumenta progressivamente o custo operacional dos participantes, especialmente em ambientes com recursos limitados [Hyperledger Fabric Contributors, 2023a].

Estratégias como particionamento, codificação por apagamento, armazenamento *off-chain* e poda reduzem a pressão sobre o armazenamento local [Sanka and Cheung, 2021, Wang et al., 2025]. Entretanto, mover dados históricos para fora do peer cria um novo problema: a rede precisa verificar se os fragmentos externos continuam disponíveis após a poda. Uma verificação feita apenas no momento da recuperação é tardia, pois a indisponibilidade pode ser descoberta somente quando os dados já são necessários.

O estado da arte apresenta diferentes estratégias para reduzir o custo de armazenamento em blockchains, incluindo poda seletiva, sumarização, particionamento, codificação por apagamento e armazenamento *off-chain*. Entretanto, essas abordagens normalmente tratam separadamente a remoção física de dados, a preservação redundante dos dados podados e a verificação periódica da custódia externa. Assim, a contribuição deste trabalho está na integração desses mecanismos em um fluxo voltado ao Hyperledger Fabric, e não na criação isolada de um novo protocolo criptográfico de prova de recuperabilidade.

Este trabalho propõe uma arquitetura de poda colaborativa para Hyperledger Fabric que combina coordenação *on-chain*, fragmentação Reed-Solomon, armazenamento *off-chain*, metadados verificáveis e auditoria probabilística de custódia. A auditoria proposta desafia custodiantes a apresentarem *chunks* e provas de Merkle, servindo como mecanismo de detecção antecipada de perdas parciais.

As principais contribuições são: (i) uma arquitetura de poda para reduzir o armazenamento local de peers Fabric; (ii) um mecanismo de auditoria amostral de custódia baseado em *chunks*, árvores de Merkle e rodadas coordenadas por chaincode; e (iii) uma avaliação experimental do custo de fragmentação, recuperação, poda local e auditoria dos fragmentos armazenados *off-chain*.

A avaliação experimental indica que a abordagem proposta reduz substancialmente o armazenamento local dos peers, alcançando redução superior a 90% no cenário avaliado, ou aproximadamente 80% quando considerado o custo de armazenamento de um fragmento *off-chain* pelo próprio participante. Os testes também mostram que a auditoria probabilística consegue detectar perdas parciais dos fragmentos com baixo custo de comunicação, alcançando 98% de detecção ao desafiar apenas 5% dos *chunks* e 100% nos cenários com 10% ou mais. Como limitação, os experimentos foram conduzidos em ambiente local, sem capturar efeitos de latência, variação de banda, concorrência entre organizações e falhas distribuídas típicas de uma implantação real.

1.1. Organização

O restante deste artigo está organizado da seguinte forma. A Seção 2 descreve os conceitos necessários para compreensão da abordagem. A Seção 3 discute os trabalhos relacionados e posiciona a proposta. A Seção 4 detalha a arquitetura, incluindo poda, metadados, auditoria e recuperação. A Seção 5 apresenta os experimentos. Por fim, a Seção 6 apresenta conclusões, limitações e trabalhos futuros.

2. Fundamentação Teórica

Esta seção resume os conceitos diretamente necessários à proposta: codificação Reed-Solomon, armazenamento externo verificável, provas de posse e o contexto de execução

no Hyperledger Fabric.

2.1. Codificação por *Erasure Coding*

A codificação por *erasure coding* é uma técnica de fragmentação e redundância de dados em sistemas distribuídos. Neste trabalho, o esquema adotado é Reed-Solomon, no qual um dado D é dividido em k fragmentos originais e expandido com m fragmentos redundantes, totalizando $n = k + m$ fragmentos.

A principal propriedade dessa técnica é que o dado original pode ser reconstruído a partir de qualquer subconjunto de pelo menos k fragmentos. Formalmente, seja $F = \{f_1, f_2, ..., f_n\}$ o conjunto de fragmentos gerados, então D pode ser recuperado se existir um subconjunto $F' \subseteq F$ tal que $|F'| \geq k$. Essa abordagem permite tolerar falhas ou indisponibilidade de até m fragmentos, reduzindo a dependência de um único ponto de armazenamento e aumentando a robustez do sistema [Li and Li, 2013].

2.2. Armazenamento *Off-chain*

O armazenamento *off-chain* mantém dados fora da *blockchain* e registra no *ledger* apenas referências, metadados e compromissos criptográficos. Essa abordagem é adequada para dados volumosos ou históricos, pois reduz a replicação local sem eliminar a rastreabilidade e a verificação de integridade [Wang et al., 2025]. Ferramentas como o IPFS são exemplos conhecidos de infraestrutura para armazenamento *off-chain*. Neste trabalho, os dados podados são preservados externamente como fragmentos verificáveis, enquanto o ledger mantém o manifesto necessário para auditoria e reconstrução.

2.3. Provas de Posse e Recuperabilidade

Quando dados são transferidos para armazenamento externo, a verificação de integridade no momento da recuperação pode ser insuficiente para detectar perdas antecipadamente. A entidade responsável por armazenar os dados, como um servidor, serviço de nuvem ou outro nó da rede, pode declarar que mantém determinado fragmento, mas removê-lo posteriormente ou conservar apenas parte dele, seja por falha operacional, indisponibilidade do ambiente ou comportamento malicioso. Para lidar com esse problema, protocolos de verificação remota de armazenamento permitem que um verificador desafie periodicamente um provedor a demonstrar que ainda possui os dados, sem exigir o download completo do arquivo.

PDP [Ateniese et al., 2007] e PoR [Juels and Kaliski, 2007, Shacham and Waters, 2013] oferecem garantias distintas: PDP verifica posse de dados amostrados, enquanto PoR busca demonstrar recuperabilidade sob modelos adversariais mais fortes. A auditoria deste trabalho deve ser entendida como uma verificação probabilística de custódia baseada em *chunks* e provas de Merkle, não como um PoR completo.

2.4. *Hyperledger Fabric*

O Hyperledger Fabric é uma plataforma de blockchain permissionada, na qual os participantes da rede são identificados e autorizados antes de interagir com o sistema. Diferente de blockchains públicas, como *Bitcoin* ou *Ethereum*, o Fabric permite maior controle sobre governança, privacidade, desempenho e regras de participa-

ção. Ele organiza a rede em componentes como organizações, *peers*, ordenadores, canais e contratos inteligentes, chamados de *chaincodes*. Por meio dos canais, é possível criar ambientes isolados de comunicação entre grupos específicos de participantes, permitindo que determinadas transações sejam visíveis apenas para as partes envolvidas. [Hyperledger Fabric Contributors, 2023b].

3. Revisão da Literatura

A literatura sobre redução de armazenamento em blockchains pode ser organizada em três linhas complementares. A primeira reduz o armazenamento local por poda, sumarização ou remoção controlada de transações. Palm et al. [Palm et al., 2018] propõem poda seletiva em Hyperledger Fabric preservando derivabilidade de estado, enquanto MOF-BC [Dorri et al., 2019] permite remover ou sumarizar transações, mantendo auditabilidade parcial e reduzindo consumo de memória. Essas propostas reduzem o ledger, mas não distribuem fragmentos redundantes nem auditam periodicamente a custódia externa dos dados removidos.

A segunda linha usa particionamento, codificação ou reorganização da replicação. BFT-Store [Qi et al., 2020] e PartitionChain [Du et al., 2021] exploram codificação Reed-Solomon em blockchains permissionadas, enquanto CUB [Xu et al., 2018], BC-Store [Chou et al., 2020], GCBlock [Qu et al., 2020] e Dynamic-EC [Zhang et al., 2025] investigam formas de reduzir a replicação total do ledger. No contexto do Hyperledger Fabric, SASLedger [Sun et al., 2021] e Ring-Overlap [Liu et al., 2022] reduzem a pressão de armazenamento ao reorganizar a permanência dos dados entre participantes ou anéis de armazenamento. Em geral, essas propostas focam disponibilidade e economia de espaço, mas não incluem uma etapa explícita de auditoria contínua dos dados podados.

A terceira linha trata da verificação remota de armazenamento. PDP [Ateniese et al., 2007] e PoR [Juels and Kaliski, 2007, Shacham and Waters, 2013] permitem verificar posse ou recuperabilidade por desafios probabilísticos. Sistemas como Permacoin [Miller et al., 2014], Filecoin [Labs, 2017] e IntegrityChain [Li et al., 2020] mostram a utilidade dessas provas em redes de armazenamento distribuído. A proposta deste artigo combina ideias dessas linhas no Fabric: coordenação *on-chain*, poda física, Reed-Solomon, manifesto verificável e auditoria amostral dos fragmentos *off-chain*. A Tabela 1 resume esse posicionamento.

Como sintetizado na Tabela 1, a integração proposta é relevante porque cada mecanismo atua sobre uma limitação diferente. A poda reduz o armazenamento local, o armazenamento *off-chain* preserva os dados removidos fora dos peers, Reed-Solomon fornece tolerância à indisponibilidade parcial dos fragmentos e a auditoria amostral permite verificar continuamente se os custodiantes mantêm os dados desafiados. O objetivo não é substituir protocolos formais de PoR, mas aproximar redução de armazenamento e manutenção preventiva da recuperabilidade em um fluxo aplicável ao Fabric.

4. Proposta

Esta seção apresenta a arquitetura proposta para mitigação do problema de escalabilidade de armazenamento em redes *blockchain*. A proposta parte da premissa de que a manutenção integral do *ledger* por todos os nós, embora favoreça redundância e verificabilidade, torna-se progressivamente mais custosa à medida que a rede cresce. Em consequência, o

Tabela 1. Comparação resumida com trabalhos relacionados.

Trabalho	Estratégia principal	Fabric	EC	Poda física	Auditoria
Palm et al.	Poda seletiva e derivabilidade de estado [Palm et al., 2018]	Sim	Não	Sim	Não
MOF-BC	Remoção/sumarização para otimização de memória [Dorri et al., 2019]	Genérico	Não	Sim	Parcial
BFT-Store	Armazenamento confiável com Reed-Solomon [Qi et al., 2020]	Não	Sim	Não	Não
PartitionChain	Particionamento com Reed-Solomon [Du et al., 2021]	Não	Sim	Não	Não
SASLedger	Redução de armazenamento em redes Fabric [Sun et al., 2021]	Sim	Não	Parcial	Não
Ring-Overlap	Escalonamento de armazenamento para Fabric [Liu et al., 2022]	Sim	Não	Parcial	Não
Filecoin/PDP	Verificação remota de armazenamento [Labs, 2017, Ateniese et al., 2007]	Não	Sim	Não	Sim
Este trabalho	Poda Fabric, Reed-Solomon, manifesto e auditoria amostral	Sim	Sim	Sim	Sim

aumento contínuo do volume de dados pode restringir a participação de nós com recursos limitados e comprometer, na prática, a descentralização do sistema.

Com o objetivo de tratar esse problema, propõe-se um *framework* de gerenciamento de dados que combina um protocolo consensual de poda, Reed-Solomon, armazenamento *off-chain* e auditoria amostral de custódia. A abordagem busca reduzir o volume de dados mantido localmente pelos nós sem eliminar a possibilidade de verificação e reconstrução posterior dos dados removidos. As decisões de poda, formação do grupo e registro dos metadados são coordenadas por chaincode, enquanto cada organização mantém autonomia para escolher o provedor externo onde armazenará os fragmentos sob sua responsabilidade, desde que o provedor atenda às regras de confiança e diversidade definidas pela rede. A Figura 1 ilustra, de forma resumida, o fluxo geral da proposta.

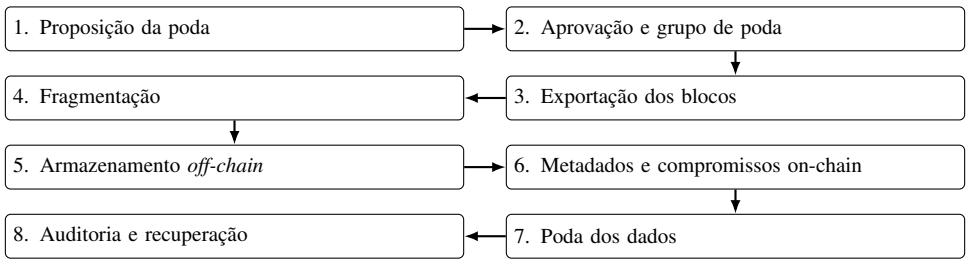

Figura 1. Fluxo compacto do processo de poda colaborativa.

A abordagem foi arquitetada para ambientes permissionados, nos quais os participantes da rede são identificáveis e mantêm inicialmente uma cópia completa dos dados registrados no *ledger*. No contexto deste trabalho, onde os testes foram realizados no Hyperledger Fabric, os peers armazenam localmente o histórico de blocos do canal. Essa característica permite que os nós possam realizar as operações propostas a partir do

mesmo conjunto de dados original, sem necessidade de redistribuição do conteúdo entre os participantes.

4.1. Modelo de Ameaças e Escopo das Garantias

Assume-se uma rede permissionada, com organizações identificadas, assinaturas digitais verificáveis e registros *on-chain* imutáveis após consenso. O adversário considerado é um custodiante que pode apagar parte de um fragmento, responder com dados incorretos, omitir resposta ou tentar reduzir seu custo mantendo apenas parte dos *chunks*. A proposta não considera, nesta versão, colusão ampla entre auditores e custodiantes nem provedores capazes de prever desafios antes da geração da rodada. Assim, a garantia fornecida é probabilística: uma auditoria bem-sucedida mostra que os *chunks* desafiados pertencem ao fragmento comprometido pela raiz de Merkle registrada, mas não prova sozinha a posse integral do fragmento nem a recuperabilidade forte do intervalo podado. A recuperabilidade depende da disponibilidade de pelo menos k fragmentos válidos.

4.2. Visão Geral da Arquitetura

Inicialmente, os dados são mantidos integralmente na *blockchain*. A política de governança do canal define um limiar de ocupação local $X\%$ e os parâmetros mínimos da poda. Quando o limiar é atingido, um nó submete ao chaincode uma proposta contendo intervalo de blocos, parâmetros k e m, política de auditoria e prazo de adesão. A proposta é aprovada apenas se satisfizer a política de governança do canal, por exemplo maioria das organizações ou política previamente configurada. Após a aprovação, o manifesto da poda é registrado no *ledger*; somente então os participantes executam exportação, fragmentação, armazenamento externo, auditoria e remoção local dos dados.

Uma vez aprovada, a poda desencadeia a fase de codificação e armazenamento, na qual os dados são fragmentados por Reed-Solomon e armazenados em provedores externos validados pela rede. Após essa etapa, a cópia integral dos dados pode ser removida localmente pelos nós da rede, permanecendo no *ledger* apenas os metadados necessários para verificabilidade, auditoria e reconstrução. Por fim, os dados podados podem ser auditados periodicamente e recuperados sob demanda.

4.3. Poda dos Dados e Nota de Confiança

O grupo de poda é formado pelas organizações que aderem à proposta aprovada e satisfazem requisitos mínimos de confiança e diversidade. A atribuição de fragmentos é derivada de forma determinística a partir do manifesto, por exemplo usando uma ordenação pública baseada em nota de confiança, organização, identificador da poda e índice do fragmento. Assim, todos os peers conseguem recalcular qual organização deve custodiar cada fragmento, sem depender de uma decisão externa ao ledger. A nota de confiança registra disponibilidade declarada, durabilidade esperada, falhas de auditoria e diversidade administrativa dos provedores, orientando a seleção de custodiantes e a redistribuição em caso de falhas.

4.4. Codificação e Distribuição dos Dados

Na etapa de execução, o conjunto de dados que será podado D é submetido ao esquema Reed-Solomon. Inicialmente, D é dividido em k fragmentos originais e, em seguida, são gerados m fragmentos redundantes, totalizando $n = k + m$ fragmentos:

$$n = k + m \tag{1}$$

Seja $F = \{f_1, f_2, \ldots, f_n\}$ o conjunto de fragmentos gerados. O dado original pode ser reconstruído a partir de qualquer subconjunto $F' \subseteq F$ tal que:

$$|F'| \geq k \tag{2}$$

Essa propriedade permite tolerar a indisponibilidade simultânea de até m fragmentos sem perda de recuperabilidade.

Na proposta, a geração dos fragmentos é distribuída entre os participantes do grupo de poda. Como os nós mantêm inicialmente o conteúdo integral no *ledger*, cada participante do grupo pode derivar localmente o fragmento sob sua responsabilidade a partir do mesmo conjunto de dados original. Para que esse processo seja consistente, a fragmentação deve ser determinística: todos os participantes devem utilizar o mesmo intervalo de blocos, a mesma ordenação dos dados exportados, os mesmos parâmetros k e m.

Antes da execução da poda, a rede registra um manifesto de poda contendo as informações necessárias para reproduzir e verificar o processo de fragmentação em cada um dos nós. Esse manifesto inclui o intervalo de blocos selecionado, os parâmetros do esquema Reed-Solomon, o índice de cada fragmento, a organização responsável por sua custódia e os compromissos criptográficos associados ao conjunto original e aos fragmentos gerados. Dessa forma, *peers* distintos podem gerar ou verificar fragmentos compatíveis sem depender de um coordenador central para produzir todo o conjunto de fragmentos e distribuir entre os nós.

O manifesto determina qual fragmento cada participante deve gerar e custodiar. Cada participante utiliza essa informação para derivar localmente seu fragmento a partir dos mesmos blocos exportados e, em seguida, armazená-lo no provedor externo indicado, observadas as restrições de confiabilidade e diversidade. O uso de múltiplos locais *off-chain* não exige um provedor distinto por fragmento, mas busca evitar concentração excessiva em um único serviço ou domínio administrativo.

Essa escolha evita a concentração do processo de codificação em um único nó e reduz a necessidade de transferência adicional de grandes volumes de dados entre os participantes. Além disso, como os fragmentos são derivados localmente a partir de dados já validados pela própria rede, reduz-se o risco de que um participante isolado manipule todo o conteúdo antes da distribuição.

Embora Reed-Solomon ofereça economia de espaço e tolerância a falhas, ele também introduz custos computacionais e de comunicação. A codificação e a decodificação exigem operações matemáticas adicionais, enquanto a recuperação requer leitura coordenada de múltiplos fragmentos distribuídos. Desse modo, a proposta assume o *trade-off* entre redução de armazenamento local e aumento do custo de recuperação.

Caso um participante gere ou disponibilize um fragmento incompatível com o manifesto registrado no *ledger*, a inconsistência pode ser detectada por meio da comparação entre o *hash* do fragmento recebido e o compromisso criptográfico correspondente.

Assim, a verificabilidade do processo não depende apenas da confiança no participante responsável, mas também dos metadados imutáveis associados à operação de poda.

4.5. Metadados e Verificabilidade

Para cada intervalo podado, o *ledger* mantém um manifesto com os metadados resumidos na Tabela 2: identificador da poda, intervalo de blocos, compromisso criptográfico do dado original $C(D)$, parâmetros k e m, identificadores e hashes dos fragmentos, responsáveis pela custódia, referências de localização *off-chain*, raízes de Merkle dos fragmentos e política de auditoria. O compromisso $C(D)$ valida o dado reconstruído, os hashes verificam fragmentos individuais e as raízes de Merkle permitem auditorias parciais sem recuperar o fragmento completo. Assim, mesmo após a remoção local dos dados históricos, a blockchain permanece como âncora de integridade, localização e responsabilização.

Tabela 2. Metadados registrados para recomposição dos dados podados.

Dado registrado	Função na recomposição
Identificador da poda	Associa a operação de recuperação ao manifesto correto.
Intervalo de blocos	Define o conjunto histórico removido e posteriormente reconstruído.
Compromisso do dado original $C(D)$	Permite verificar se o dado reconstruído corresponde ao conjunto podado.
Parâmetros k e m	Indicam quantos fragmentos são necessários para reconstruir o dado e o nível de redundância adotado.
Algoritmo de codificação	Define o procedimento usado para decodificar os fragmentos recuperados.
Identificadores e hashes dos fragmentos	Permitem selecionar fragmentos válidos e detectar corrupção ou substituição indevida.
Custodiante e localização off-chain	Indicam quem deve fornecer cada fragmento e onde ele pode ser recuperado.

4.6. Auditoria Amostral de Custódia

Para verificar periodicamente se os provedores mantêm os fragmentos atribuídos, a arquitetura incorpora uma auditoria amostral de custódia. O objetivo não é recuperar o dado completo a cada verificação, mas obter evidências de posse dos *chunks* desafiados e detectar perdas parciais antes de uma recuperação completa.

Após a geração de cada fragmento f_i, o responsável por sua custódia o divide de forma determinística em blocos menores de tamanho fixo, denominados *chunks*. Seja $B_i = \{b_{i,1}, b_{i,2}, \ldots, b_{i,t}\}$ o conjunto de *chunks* de f_i. Sobre esses blocos é construída uma árvore de Merkle, cuja raiz R_i é registrada no manifesto on-chain. O tamanho dos *chunks* é um parâmetro da política de auditoria e deve equilibrar custo e granularidade: blocos menores reduzem o tráfego por desafio e aumentam a precisão da auditoria, enquanto blocos maiores reduzem o número de folhas e o tamanho relativo das estruturas auxiliares.

As rodadas de auditoria são registradas e parametrizadas pela chaincode, mas sua ativação depende de agentes externos executados pelas organizações participantes. Cada organização pode executar uma aplicação denominada *agente auditor*, responsável por consultar periodicamente a rede, verificar se há intervalos elegíveis para nova auditoria e submeter uma transação de criação de rodada quando a política temporal permitir.

Para evitar desafios previsíveis, a semente da rodada não deve depender apenas de dados públicos conhecidos antes da auditoria. Uma implementação compatível com Fabric pode usar *commit-reveal*: organizações publicam compromissos $H(s_j)$, revelam os valores s_j após o encerramento da fase de compromisso, e a semente final é calculada como $H(s_1||...||s_r||id_{\text{poda}}||id_{\text{rodada}})$. Se ao menos uma organização honesta contribuir com uma semente imprevisível, o custodiante não conhece antecipadamente os *chunks* que serão desafiados. A Tabela 3 resume os dados usados em cada rodada.

Durante a execução da rodada, o agente auditor selecionado envia ao provedor responsável um desafio contendo os índices dos *chunks* sorteados. O provedor responde com os blocos solicitados, as provas de Merkle correspondentes e uma assinatura digital vinculada à sua identidade na rede. O auditor então executa quatro verificações: valida a assinatura do provedor, calcula o *hash* de cada *chunk* recebido, confere se as provas de Merkle conduzem à raiz R_i registrada no manifesto e registra o resultado na chaincode. Em caso de ausência de resposta ou prova inválida, o auditor registra uma falha acompanhada das evidências coletadas.

O resultado das auditorias alimenta o mecanismo de nota de confiança descrito anteriormente. Auditorias bem-sucedidas mantêm ou elevam a confiança do provedor, enquanto falhas reiteradas reduzem sua pontuação e podem impedir sua participação em novas operações de poda.

A escolha dos parâmetros da auditoria deve refletir o risco tolerado pela rede. Seja t o total de *chunks* do fragmento; o parâmetro q representa a fração desafiada e $r = \lceil qt \rceil$ representa o número efetivo de *chunks* sorteados na rodada.

Uma política inicial de resposta a falhas pode ser gradual. A primeira falha isolada reduziria a nota de confiança e exigiria nova auditoria em menor intervalo; falhas consecutivas poderiam suspender temporariamente o provedor de novas podas e acionar a replicação do fragmento em outro local. Em casos recorrentes ou quando houver evidência de descarte deliberado dos dados, a rede pode remover o provedor da lista de custodiantes autorizados e, se o custodiante for uma organização participante, aplicar penalidades.

Quando uma falha é confirmada, a rede não deve aguardar uma recuperação futura para agir. O manifesto permite identificar quais fragmentos ainda possuem compromissos válidos e quais custodiantes devem ser desconsiderados. Se ainda houver pelo menos k fragmentos disponíveis, um agente autorizado pode reconstruir o dado podado, gerar novamente o fragmento perdido e registrar sua redistribuição para um novo provedor aprovado. Assim, a auditoria atua como mecanismo preventivo de manutenção da redundância, reduzindo a chance de que perdas graduais comprometam a recuperabilidade do intervalo podado.

Esse mecanismo complementa, mas não substitui, a recuperação. Ele aumenta a chance de detectar custodiantes que removeram parte dos dados, enquanto a garantia de recuperação continua dependente da obtenção de pelo menos k fragmentos íntegros.

4.7. Recuperação dos Dados Podados

Quando um nó deseja recuperar dados previamente podados, ele inicia o procedimento consultando o *ledger* para localizar os metadados associados ao dado D. A partir dessas informações, o nó identifica os provedores responsáveis pelos fragmentos e requisita um

Tabela 3. Dados utilizados nas rodadas de auditoria amostral de custódia.

Dado	Origem	Uso na auditoria
Raiz de Merkle R_i	Manifesto	Ancora a verificação dos *chunks* respondidos pelo custodiante.
Tamanho dos *chunks*	Política/manifesto	Define a granularidade do desafio e o custo de transferência.
Identificador da rodada	Chaincode	Distingue execuções de auditoria sobre a mesma operação de poda.
Semente da rodada	*Commit-reveal*/chaincode	Permite selecionar os *chunks* desafiados de forma recomputável.
Índices dos *chunks*	Auditor	Formam o desafio enviado ao custodiante ou provedor responsável.
Chunks e provas de Merkle	Resposta do custodiante	Demonstram que os blocos solicitados pertencem ao fragmento comprometido.
Assinatura da resposta	Custodiante	Vincula a resposta à identidade responsável pela custódia do fragmento.
Resultado e evidências	Auditor/chaincode	Registram sucesso, atraso, ausência de resposta ou prova inválida.

subconjunto mínimo de pelo menos k fragmentos válidos.

Uma vez recebidos os fragmentos, o nó executa o processo de reconstrução de D conforme os parâmetros do esquema Reed-Solomon. Em seguida, a integridade do dado reconstruído é verificada por meio da comparação entre o comprometimento criptográfico calculado localmente e o valor $C(D)$ registrado no *ledger*.

Formalmente, a recuperação de D é bem-sucedida se:

$$\text{Recover}(D) = 1 \iff \big(|F'| \geq k\big) \wedge \big(C(D') = C(D)\big) \tag{3}$$

em que D' representa o dado reconstruído a partir do subconjunto de fragmentos F'.

Essa elaboração torna a etapa de recuperação consistente com o mecanismo de fragmentação adotado. Não se assume, portanto, a existência de um único local externo contendo o dado completo, mas sim a reconstrução a partir de múltiplos fragmentos distribuídos.

5. Experimentos

Esta seção apresenta a avaliação experimental da proposta, com o objetivo de analisar seu comportamento sob diferentes configurações e validar sua efetividade frente ao problema de escalabilidade em *blockchains* permissionadas. Para isso, foram definidos cenários de teste que consideram variações no número de participantes da rede, no volume de dados processados e nos parâmetros utilizados nas etapas de fragmentação, recuperação e auditoria das informações. Os experimentos realizados abrangem a geração de grupos de poda, a fragmentação dos dados, a restauração dos fragmentos, a auditoria amostral de custódia e a poda local. Todos os experimentos foram conduzidos em um ambiente totalmente local, no qual os nós da rede foram executados na mesma infraestrutura. Dessa forma, os resultados não contemplam efeitos relacionados à latência de rede entre os

participantes, como atrasos de comunicação, variações de banda ou instabilidades típicas de ambientes distribuídos reais.

Os testes foram organizados em experimentos isolados, voltados a medir etapas específicas, e experimentos de fluxo completo, que consideram coordenação, recuperação e poda. Antes de avaliar a auditoria amostral de custódia, foram analisados três aspectos da arquitetura base: o custo de formação do grupo de poda, o comportamento da fragmentação e reconstrução com diferentes parâmetros de codificação, e o custo da recuperação completa a partir dos fragmentos armazenados *off-chain*.

Na formação do grupo de poda, o tempo observado foi de 12,6 s com 5 organizações, 23,6 s com 10 organizações e 37,7 s com 15 organizações. Esses resultados mostram que o custo de coordenação cresce com o número de participantes envolvidos, pois a etapa depende da coleta e validação das respostas enviadas pelas organizações. No cenário avaliado, os tempos representam o pior caso sem paralelismo entre respostas, de modo que implementações com chamadas concorrentes tendem a reduzir o tempo observado na prática.

Tabela 4. Tempo de fragmentação e reconstrução isolada.

Dados	(k, m)	**Fragmentação**	**Reconstrução**	**Validação**
500 MB	(8,2)	1,2 s	0,44 s	k recupera; $k-1$ falha
500 MB	(6,4)	1,4 s	0,47 s	k recupera; $k-1$ falha
500 MB	(5,5)	1,7 s	0,53 s	k recupera; $k-1$ falha
1 GB	(8,2)	3,2 s	0,994 s	k recupera; $k-1$ falha
1 GB	(6,4)	6,0 s	0,997 s	k recupera; $k-1$ falha
1 GB	(5,5)	6,3 s	1,11 s	k recupera; $k-1$ falha

A Tabela 4 detalha os testes de codificação. A fragmentação cresce com o volume de dados e também é afetada pela configuração de redundância: configurações com maior número de fragmentos redundantes tendem a exigir mais processamento. Já a reconstrução permaneceu com valores muito próximos independente da variação dos valores k e m, indicando que o custo computacional local da decodificação é baixo em comparação ao custo de transferência dos fragmentos externos.

Para uma análise mais abrangente, foi avaliado o processo completo de recuperação com $(k, m) = (6, 4)$ e 1 GB de dados. O tempo total médio foi de 911,998 s, dos quais 911 s foram gastos no download dos fragmentos externos. A consulta aos metadados consumiu 0,008 s e a reconstrução local 0,99 s. Esse resultado indica que, na conexão utilizada nos testes, o gargalo esteve associado à transferência dos fragmentos externos, e não às operações internas de consulta ou decodificação.

5.1. Auditoria Amostral de Custódia

Foram realizados testes complementares para avaliar o comportamento do mecanismo de auditoria amostral de custódia. O objetivo desses experimentos foi verificar se o agente auditor consegue detectar a remoção parcial de dados armazenados off-chain sem realizar o download completo do fragmento. Para isso, cada fragmento foi dividido em *chunks* de

tamanho fixo e, em seguida, foi construída uma árvore de Merkle sobre esses *chunks*. Em cada rodada, o auditor selecionou um subconjunto de *chunks*, enviou o desafio ao provedor e verificou a resposta a partir das provas de Merkle e da raiz registrada no manifesto.

O primeiro grupo de testes avaliou o impacto da quantidade de *chunks* desafiados na taxa de detecção. Para evitar ambiguidade, q representa a fração do fragmento desafiada e $c = \lceil qN \rceil$ representa o número de *chunks* sorteados, em que N é o total de *chunks*. O cenário considerou um fragmento de 500 MB, *chunk size* de 1 MB, $N = 500$ e perda artificial de 10% dos *chunks*. Como os desafios são sorteados sem reposição, a detecção teórica foi calculada pelo modelo hipergeométrico:

$$P_{\text{det}} = 1 - \frac{\binom{N-L}{c}}{\binom{N}{c}}, \tag{4}$$

em que $L = pN$ é o número de *chunks* perdidos e p é a fração removida. A aproximação independente $1-(1-p)^c$ é próxima nesse cenário, mas o modelo hipergeométrico reflete melhor o procedimento adotado.

Tabela 5. Detecção de perda parcial em um fragmento de 500 MB com *chunk size* de 1 MB e perda de 10%

q	c **chunks**	**Perda**	**Detecção prática**	**Detecção teórica**	**Transferência média**
5%	25	10%	98,000%	93,300%	1,356%
10%	50	10%	100,000%	99,615%	1,618%
15%	75	10%	100,000%	99,981%	1,869%
20%	100	10%	100,000%	99,999%	1,802%

Os resultados da Tabela 5 indicam que uma política com 5% dos *chunks* desafiados detectou a perda parcial em 98% das rodadas, valor compatível com a estimativa teórica de 93,3% considerando variação amostral. A partir de 10%, a detecção prática atingiu 100% no cenário avaliado. A perda artificial de 10% foi escolhida como cenário de falha parcial severa, suficiente para afetar a confiança no custodiante sem representar perda total do fragmento.

O segundo grupo de testes avaliou o custo operacional do mecanismo de auditoria em função do tamanho do fragmento e do tamanho dos *chunks*. Nesse cenário, não foi introduzida perda de dados, pois o objetivo foi medir o custo do fluxo normal de auditoria. O parâmetro q foi fixado em 20% dos *chunks* de cada fragmento. A métrica *chunk/hash* representa o tempo para o auditor ou agente ler o fragmento, dividi-lo em *chunks* e calcular o hash de cada *chunk*. No benchmark atual, essa etapa foi executada em cada repetição para reconstruir a base da árvore de Merkle, razão pela qual seu custo cresce com o tamanho total do fragmento. A métrica *provider* representa o tempo gasto pelo provedor para montar a resposta ao desafio, incluindo localizar e ler os *chunks* solicitados, buscar as provas de Merkle correspondentes e assinar a resposta. A métrica *verify* representa o tempo gasto pelo auditor para validar a assinatura do provedor e verificar, para cada *chunk* recebido, se a prova de Merkle conduz à raiz registrada no manifesto.

Como apresentado na Tabela 6, o custo de preparação dos hashes cresce aproximadamente de acordo com o tamanho do fragmento, passando de cerca de 389–413

Tabela 6. Custo operacional da auditoria amostral de custódia com $q = 20\%$ e sem perda de dados

Fragmento	*chunk size*	Chunks	c chunks	*chunk/hash* ms	*provider* ms	*verify* ms	Transferência
500 MB	1 MB	500	100	389,220	103,530	135,330	20,006%
500 MB	4 MB	125	25	397,270	102,980	134,590	20,001%
500 MB	16 MB	32	7	413,010	109,170	146,730	21,728%
1 GB	1 MB	1024	205	792,080	212,330	275,990	20,026%
1 GB	4 MB	256	52	812,300	214,380	280,230	20,314%
1 GB	16 MB	64	13	825,030	208,320	277,150	20,313%

ms para fragmentos de 500 MB para cerca de 792–825 ms em fragmentos de 1 GB. Esse comportamento era esperado, pois a etapa percorre o fragmento inteiro para gerar os compromissos dos *chunks*. Já os tempos de montagem da resposta pelo provedor e de verificação pelo auditor permanecem inferiores a 300 ms em todos os cenários avaliados. O custo de transferência acompanhou o valor definido para q, ficando próximo de 20% do fragmento, com pequena elevação nos cenários de *chunks* de 16 MB devido ao arredondamento do número de *chunks* desafiados.

Esses resultados reforçam a utilidade da auditoria como complemento à recuperação completa. Enquanto a reconstrução dos dados podados depende do download de pelo menos k fragmentos válidos, a auditoria permite verificar periodicamente a custódia de cada fragmento com custo controlado e parametrizável. Na prática, a rede pode ajustar q conforme o nível de confiança desejado: valores menores reduzem transferência e custo operacional, enquanto valores maiores aumentam a probabilidade de detectar provedores que removeram parte dos dados sob sua responsabilidade.

Os resultados também indicam que q pode ser tratado como variável de política da rede, e não como valor fixo para todos os custodiantes. No cenário com perda artificial de 10%, $q = 10\%$ já alcançou detecção prática total, enquanto $q = 5\%$ reduziu ainda mais a quantidade de dados transferidos, mas apresentou pequena redução na taxa de detecção. Assim, provedores com histórico confiável podem ser auditados com desafios mais leves, ao passo que custodiantes recém-adicionados, instáveis ou penalizados podem receber desafios maiores e rodadas mais frequentes até restabelecerem sua nota de confiança.

5.2. Ganho de Espaço

Após realizar todas as etapas do protocolo, é possível avaliar o ganho de espaço obtido com a poda local dos dados em cada peer, tomando como referência o Fabric nativo sem poda, no qual o peer mantém localmente os *blockfiles* completos. Como mostrado na Tabela 7, no cenário com aproximadamente 1 GB de dados, a poda física defensiva liberou cerca de 92,2% do espaço ocupado no peer. Nesse processo, não são removidos dados necessários para a configuração e continuidade operacional da rede: blocos iniciais, blocos finais e cabeçalhos dos blocos intermediários podados são preservados localmente. O restante dos dados é removido do peer, fragmentado por Reed-Solomon e armazenado em servidores externos.

Outra métrica importante é o ganho real de espaço. O ganho apresentado anteriormente considera apenas a redução local no peer. No entanto, se o peer participar do grupo de poda, ele também deverá armazenar pelo menos um fragmento off-chain. Portanto, o

ganho real corresponde ao espaço removido do peer menos o espaço ocupado pelo fragmento que esse peer passa a armazenar fora da blockchain. Nesse cenário, considerando Reed-Solomon com k=6 e m=4, ou seja, 10 fragmentos no total, e assumindo que cada peer armazene apenas um fragmento, o ganho real estimado cai para aproximadamente 80,07%. Esse valor pode variar de acordo com o número de organizações no grupo e com a distribuição dos fragmentos, pois esses fatores influenciam diretamente o tamanho e a quantidade de fragmentos armazenados por cada peer.

Tabela 7. Ganho de armazenamento obtido com a poda local

Métrica	Valor	Comentário
Redução local no peer	92,2%	Representa o espaço liberado diretamente no armazenamento local do peer após a remoção dos *blockfiles*.
Ganho real estimado	80,07%	Considera que o peer também passa a armazenar um fragmento off-chain, reduzindo o ganho líquido obtido.

6. Conclusão

Este trabalho apresentou uma arquitetura de poda colaborativa para Hyperledger Fabric que combina coordenação *on-chain*, fragmentação Reed-Solomon, armazenamento *off-chain*, manifesto verificável e auditoria amostral de custódia. Os resultados indicam redução local de 92,2% no armazenamento do peer, com ganho líquido estimado de 80,07% quando o participante também armazena um fragmento externo.

A auditoria detectou uma perda artificial de 10% dos *chunks* em 98% das rodadas ao desafiar 5% dos *chunks*, e em 100% das rodadas com desafios de 10% ou mais. Esses resultados mostram baixo custo de detecção probabilística, mas não constituem prova de recuperabilidade forte. As principais limitações são a avaliação local, a ausência de comparação experimental com implantações distribuídas reais, o modelo inicial de nota de confiança e a necessidade de uma geração de desafios imprevisível contra custodiantes estratégicos. Como trabalhos futuros, pretende-se avaliar redes de maior escala, comparar com Fabric nativo, implementar sementes por *commit-reveal* ou VRF e automatizar reparo e penalização.

Referências

Ateniese, G., Burns, R., Curtmola, R., Herring, J., Kissner, L., Peterson, Z., and Song, D. (2007). Provable data possession at untrusted stores. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 598–609. ACM.

Chou, I.-T., Su, H.-H., Hsueh, Y.-L., and Hsueh, C.-W. (2020). Bc-store: A scalable design for blockchain storage. In *Proceedings of the 2nd International Electronics Communication Conference*, pages 33–38.

Dorri, A., Kanhere, S. S., and Jurdak, R. (2019). MOF-BC: A memory optimized and flexible blockchain for large scale networks. *Future Generation Computer Systems*, 92:357–373.

Du, Z., Pang, X., and Qian, H. (2021). Partitionchain: A scalable and reliable data storage strategy for permissioned blockchain. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):4124–4136.

Hyperledger Fabric Contributors (2023a). Hyperledger Fabric Documentation: Ledger. `https://hyperledger-fabric.readthedocs.io/en/release-2.5/ledger/ledger.html`. Accessed: 2025-05-1.

Hyperledger Fabric Contributors (2023b). Hyperledger Fabric Documentation: What is Hyperledger Fabric? `https://hyperledger-fabric.readthedocs.io/en/release-2.5/whatis.html`. Accessed: 2025-05-1.

Juels, A. and Kaliski, B. S. (2007). PORs: Proofs of retrievability for large files. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, pages 584–597. ACM.

Labs, P. (2017). Filecoin: A decentralized storage network. `https://filecoin.io/filecoin.pdf`. [Accessed 29-04-2026].

Li, J. and Li, B. (2013). Erasure coding for cloud storage systems: A survey. *Tsinghua Science and Technology*, 18(3):259–272.

Li, Y., Yu, Y., Chen, R., Du, X., and Guizani, M. (2020). Integritychain: Provable data possession for decentralized storage. *IEEE Journal on Selected Areas in Communications*, 38(6):1205–1217.

Liu, W., Zhang, D., Mu, C., Zhao, X., and Zhao, J. (2022). Ring-overlap: A storage scaling mechanism for hyperledger fabric. *Applied Sciences*, 12(19):9568.

Miller, A., Juels, A., Shi, E., Parno, B., and Katz, J. (2014). Permacoin: Repurposing bitcoin work for data preservation. In *2014 IEEE Symposium on Security and Privacy*, pages 475–490. IEEE.

Palm, E., Schelén, O., and Bodin, U. (2018). Selective blockchain transaction pruning and state derivability. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 31–40. IEEE.

Qi, X., Zhang, Z., Jin, C., and Zhou, A. (2020). A reliable storage partition for permissioned blockchain. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):14–27.

Qu, B., Wang, L.-E., Liu, P., Shi, Z., and Li, X. (2020). Gcblock: A grouping and coding based storage scheme for blockchain system. *IEEE Access*, 8:48325–48336.

Sanka, A. I. and Cheung, R. C. (2021). A systematic review of blockchain scalability: Issues, solutions, analysis and future research. *Journal of Network and Computer Applications*, 195:103232.

Shacham, H. and Waters, B. (2013). Compact proofs of retrievability. *Journal of Cryptology*, 26(3):442–483.

Sun, H., Pi, B., Sun, J., Miyamae, T., and Morinaga, M. (2021). Sasledger: A secured, accelerated scalable storage solution for distributed ledger systems. *Future Internet*, 13(12):310.

Wang, X., Li, H., Yi, L., Ning, Z., Tao, X., Guo, S., and Zhang, Y. (2025). A survey on off-chain networks: Frameworks, technologies, solutions and challenges. *ACM Computing Surveys*, 57(12).

Xu, Z., Han, S., and Chen, L. (2018). Cub, a consensus unit-based storage scheme for blockchain system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 173–184. IEEE.

Zhang, M., Wu, C., Li, J., and Guo, M. (2025). Dynamic-ec: an efficient dynamic erasure coding method for permissioned blockchain systems. *Frontiers of Computer Science*, 19(1):1–14.