**UNIVERSIDADE FEDERAL DO CEARÁ**

**CAMPUS DE RUSSAS**

**CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**JOSÉ EVAMBERTO LIMA OLIVEIRA**

# APLICAÇÕES DE BLOCKCHAIN NA RASTREABILIDADE DE REQUISITOS DE SOFTWARE

**RUSSAS**

**2026**

JOSÉ EVAMBERTO LIMA OLIVEIRA

# APLICAÇÕES DE BLOCKCHAIN NA RASTREABILIDADE DE REQUISITOS DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: Prof. Ms. Pitágoras Graças Martins

RUSSAS

2026

JOSÉ EVAMBERTO LIMA OLIVEIRA

# APLICAÇÕES DE BLOCKCHAIN NA RASTREABILIDADE DE REQUISITOS DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em: 06/07/2026

BANCA EXAMINADORA

---

Prof. Ms. Pitágoras Graças Martins (Orientador)
Universidade Federal do Ceará (UFC)

---

Prof. Ms. Marcus Venicius Virginio de Sousa
INBEC

---

Prof. Dr. Mayrton Dias de Queiroz
Universidade Federal do Ceará (UFC)

## RESUMO

Este trabalho apresenta uma revisão bibliográfica de caráter crítico-analítico, de abordagem qualitativa, sobre a aplicação da tecnologia *blockchain* na rastreabilidade de requisitos de software. O objetivo principal é identificar e analisar criticamente as arquiteturas propostas na literatura e avaliar como lidam com os desafios técnicos inerentes à tecnologia. A busca foi conduzida nas bases IEEE Xplore, Scopus e Google Scholar, resultando na seleção de nove estudos primários publicados entre 2021 e 2025. Os resultados indicam o uso predominante de *smart contracts* na plataforma Ethereum e abordagens inovadoras com blockchains neurais e bancos de dados de grafos, geralmente adotando estratégias de armazenamento híbrido (*on-chain* e *off-chain* via IPFS) para mitigar custos. Apesar de a *blockchain* oferecer imutabilidade, transparência e auditabilidade para o registro de requisitos, o campo de pesquisa ainda se encontra em estágio inicial de maturidade. Constatou-se a predominância de protótipos em ambientes controlados, com lacunas significativas no que tange à interoperabilidade com ferramentas de mercado (como Jira e GitHub), análises insuficientes de escalabilidade em cenários reais e tratamento superficial da privacidade. Conclui-se que, embora o potencial teórico seja expressivo, a adoção industrial da *blockchain* na rastreabilidade de requisitos ainda requer a validação empírica de seu custo-benefício frente às soluções tradicionais.



**Palavras-chave:** *blockchain*; rastreabilidade de requisitos; engenharia de software; revisão crítica; *smart contracts*.

## ABSTRACT

This work presents a critical-analytical literature review, with a qualitative approach, on the application of blockchain technology in software requirements traceability. The main objective is to identify and critically analyze the architectures proposed in the literature and evaluate how they address the inherent technical challenges of the technology. The search was conducted in the IEEE Xplore, Scopus, and Google Scholar databases, resulting in the selection of nine primary studies published between 2021 and 2025. The results indicate a predominant use of smart contracts on the Ethereum platform, alongside innovative approaches using neural blockchains and graph databases, generally adopting hybrid storage strategies (on-chain and off-chain via IPFS) to mitigate costs. Although blockchain offers immutability, transparency, and auditability for requirements tracking, the research field is still in its early stages of maturity. The studies are predominantly prototypes in controlled environments, with significant gaps regarding interoperability with market tools (such as Jira and GitHub), insufficient scalability analyses in real-world scenarios, and superficial treatment of privacy. In conclusion, while the theoretical potential is substantial, the industrial adoption of blockchain in requirements traceability still requires empirical validation of its cost-benefit ratio compared to traditional solutions.



**Keywords:** blockchain; requirements traceability; software engineering; critical review; smart contracts.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE QUADROS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| BC4RT | *Blockchain for Requirements Traceability* |
| CMMI | *Capability Maturity Model Integration* |
| DLT | *Distributed Ledger Technology* |
| IEEE | *Institute of Electrical and Electronics Engineers* |
| IoT | *Internet of Things* |
| IPFS | *InterPlanetary File System* |
| NDL | *Neural Distributed Ledger* |
| PBFT | *Practical Byzantine Fault Tolerance* |
| PoW | *Proof-of-Work* |
| SDLC | *Software Development Life Cycle* |
| SRE | *Software Requirements Engineering* |

**SUMÁRIO**

# 1 INTRODUÇÃO

A *blockchain*, tecnologia originada do desenvolvimento do *bitcoin*, apresenta atributos como imutabilidade, privacidade, transparência e descentralização, além da capacidade de executar contratos automatizados. Tais características despertam interesse em diversas áreas, motivando a investigação de suas aplicações na rastreabilidade de requisitos de software.

A rastreabilidade de requisitos é um conceito central na engenharia de software, especialmente em projetos que demandam alta qualidade, conformidade regulatória e manutenção contínua. Ela pode ser definida como a capacidade de descrever e seguir a vida de um requisito, tanto para frente (desde sua origem até a implementação e testes) quanto para trás (dos artefatos de software até os requisitos que os originaram) (Gotel; Finkelstein, 1994).

Segundo Sayão e Leite (2006), a rastreabilidade de requisitos, um dos pilares da gerência de requisitos, é crucial para assegurar a consistência dos artefatos produzidos ao longo do ciclo de vida do software. Essa prática permite, por exemplo, que mudanças em requisitos sejam propagadas corretamente para o código, testes e documentação, reduzindo riscos e retrabalho.

Além disso, a rastreabilidade é um requisito fundamental em modelos de qualidade como o CMMI (*Capability Maturity Model Integration*) e a norma ISO/IEC 29148, que tratam da engenharia de requisitos. O CMMI, por exemplo, estabelece que a rastreabilidade é necessária para garantir que os requisitos sejam gerenciados de forma eficaz e que estejam alinhados com os objetivos do projeto.

A relevância do tema também é destacada devido a rastreabilidade de requisitos ser fundamental para diversas atividades críticas, incluindo análise de impacto, verificação de conformidade, seleção de testes de regressão e validação de requisitos. Em ambientes ágeis e distribuídos, como demonstrado por Diomedesse e Cunha (2023), o uso de ferramentas como *Jira* e *Confluence* para rastrear requisitos se torna ainda mais importante para manter a coesão e a transparência entre as equipes.

Por fim, estudos como o de Malcher (2015) reforçam que a rastreabilidade é uma prática ainda em evolução, com diversas abordagens e ferramentas sendo propostas para torná-la mais eficaz e menos onerosa. A adoção de sistemas de rastreabilidade de requisitos de software baseados em *blockchain* é uma possibilidade para garantir sua efetividade em projetos reais, devido às suas características intrínsecas.

## 1.1 Justificativa

Projetos de software contemporâneos frequentemente enfrentam desafios como falta de transparência e dificuldades no rastreamento de responsabilidades por mudanças, especialmente quando envolvem equipes distribuídas e a necessidade de proteção contra alterações indevidas. Nesse contexto, a tecnologia *blockchain* apresenta-se como uma solução promissora para aumentar a transparência e a confiança entre a equipe de desenvolvimento e os *stakeholders*.

Além disso, nos últimos anos, a qualidade do software tem sido uma preocupação crescente, impulsionada pela globalização dos mercados e pela terceirização do desenvolvimento de software para equipes distribuídas. As empresas de desenvolvimento procuram obter certificações de qualidade para os seus produtos, enquanto os modelos de maturidade exigem processos de gestão de requisitos bem definidos como CMMI (Sayão; Leite, 2006). Adicionalmente, a *blockchain* aprimora a rastreabilidade ao oferecer um registro histórico completo e imutável das atividades.

## 1.2 Problema de pesquisa

Apesar do crescente interesse nas propriedades da tecnologia *blockchain*, as soluções propostas na literatura variam amplamente em relação às suas arquiteturas e à sua maturidade técnica. Nesse cenário, estabelece-se o seguinte problema de pesquisa: **Como a tecnologia *blockchain* tem sido aplicada na rastreabilidade de requisitos de software e quais são as principais abordagens, desafios técnicos e lacunas de pesquisa remanescentes na literatura?**

## 1.3 Objetivo geral

Identificar e analisar soluções recentes que empregam a tecnologia *blockchain* na rastreabilidade de requisitos de software.

### *1.3.1 Objetivos específicos*

- Apresentar uma revisão bibliográfica das soluções que utilizam a tecnologia *blockchain* na rastreabilidade de requisitos de software.
- Identificar as abordagens de como a *blockchain* é utilizada na rastreabilidade de requisitos de software.

- Identificar possíveis lacunas no uso da *blockchain* na rastreabilidade de requisitos de software.

## 1.4 Organização do trabalho

Este trabalho está organizado em seis seções. A Seção 1 apresenta esta introdução, a justificativa, o problema de pesquisa e os objetivos do estudo. A Seção 2 detalha a fundamentação teórica sobre engenharia de requisitos e a tecnologia *blockchain*. A Seção 3 descreve o método de pesquisa utilizado para a seleção e análise crítica dos artigos. A Seção 4 apresenta a análise sistemática detalhada dos nove estudos selecionados. A Seção 5 consolida e discute os resultados obtidos e as lacunas técnicas identificadas. Por fim, a Seção 6 apresenta as conclusões, limitações do estudo e sugestões para trabalhos futuros.

# 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção visa estabelecer a base conceitual para entender as aplicações da *blockchain* na rastreabilidade de requisitos de software. Serão abordados a definição de Requisitos e da Rastreabilidade de Requisitos no desenvolvimento de software. Em seguida, serão explorados os conceitos fundamentais da tecnologia *blockchain*, incluindo sua arquitetura, funcionamento e atributos relevantes para a gestão de informações. Por fim, serão apresentadas as interfaces e o potencial de integração entre a *blockchain* e a gestão de requisitos, preparando o cenário para a análise detalhada dos estudos identificados na revisão bibliográfica.

## 2.1 Requisitos

Conforme Sommerville (2011), os requisitos de um sistema referem-se às descrições dos serviços que ele deve entregar, bem como às suas restrições operacionais. Os requisitos podem ser classificados em requisitos funcionais e não-funcionais. Requisitos funcionais são requisitos que descrevem uma funcionalidade do sistema e requisitos não-funcionais descrevem como o sistema deve se comportar. A seguir, são apresentados exemplos de cada tipo:

- **RF001** – O software deve emitir um relatório de venda a cada 15 dias.
- **RF002** – O usuário deve obter quais livros ele pegou emprestado.
- **RNF001** – O sistema deve estar disponível 24 horas por dia.
- **RNF002** – Uma consulta aos dados do empregado não pode demorar mais que 100 milissegundos.

## 2.2 Engenharia de Requisitos

De acordo com Kotonya e Sommerville (1998), Engenharia de requisitos é o termo abrangente para as atividades de descoberta, documentação e manutenção dos requisitos de um sistema computacional. A designação de engenharia indica o emprego de técnicas sistemáticas e repetíveis para assegurar que os requisitos do sistema sejam completos, consistentes e relevantes.

### *2.2.1 Documento de Requisitos*

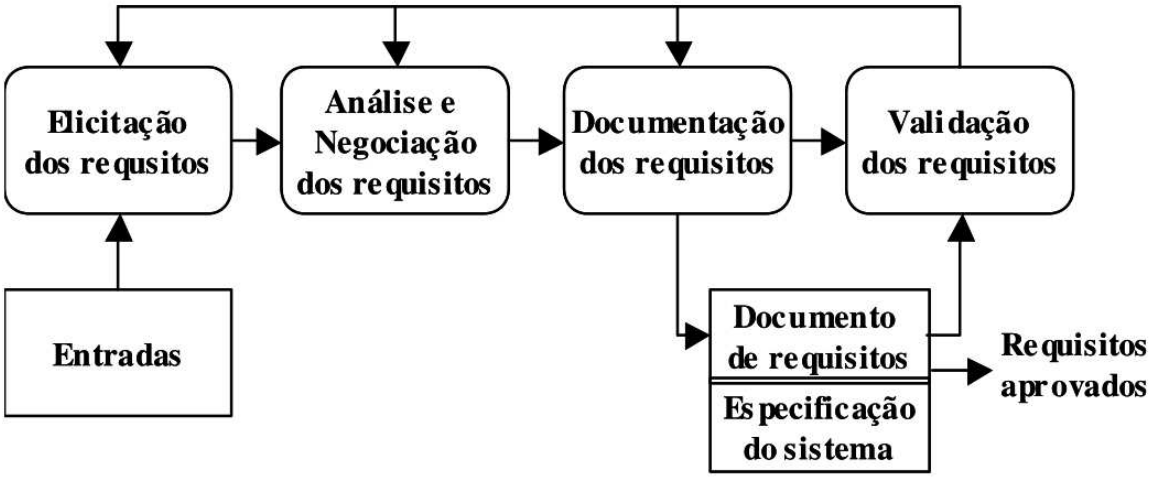
Conforme Sommerville e Sawyer (1997), o documento de requisitos formaliza as necessidades dos stakeholders, como clientes, usuários finais e equipe de desenvolvimento. Em

diferentes organizações, pode ser chamado de especificação funcional, definição de requisitos ou especificação dos requisitos do software. Algumas entidades, como o Departamento de Defesa dos EUA e o IEEE, estabeleceram padrões próprios, sendo o IEEE Std 830-1998 um dos mais conhecidos.

### *2.2.2 As atividades de Engenharia de Requisitos*

De acordo com Sommerville e Sawyer (1997), as atividades de Engenharia de Requisitos geralmente envolvem: levantamento, para identificar as necessidades dos *stakeholders*; análise, para refinar e estruturar esses requisitos; especificação, para documentá-los de forma clara e inequívoca; validação, para assegurar sua correção e completude; e gerenciamento, que trata das alterações e da evolução dos requisitos ao longo do tempo. A Figura 1 esboça um modelo genérico das atividades para o processo de engenharia de requisitos.

Figura 1 – Modelo de atividades genérico para processos de ER




Fonte: Sommerville e Sawyer (1997).

## 2.3 Rastreabilidade de Requisitos

A rastreabilidade de requisitos refere-se à capacidade de descrever e seguir o ciclo de vida de um requisito, tanto para a frente (do requisito para o *design*, código e testes) quanto para trás (do *design*, código ou teste para o requisito original) (Gotel; Finkelstein, 1994). Para gerenciar projetos de forma eficaz, essa capacidade é essencial. Ela possibilita, por exemplo, identificar rapidamente as seções de código ou testes que seriam impactadas pela alteração de um requisito. Além disso, a rastreabilidade permite descobrir quem o sugeriu, o motivo

de sua existência, quais outros requisitos estão relacionados a ele e como ele se conecta com outros artefatos do sistema, como o projeto, a implementação e a documentação do usuário, simplificando assim a análise de impacto e o processo de tomada de decisão.

### *2.3.1 Tipos de Rastreabilidade de Requisitos*

Segundo Sayão e Leite (2006), a rastreabilidade pode ser classificada em duas categorias principais:

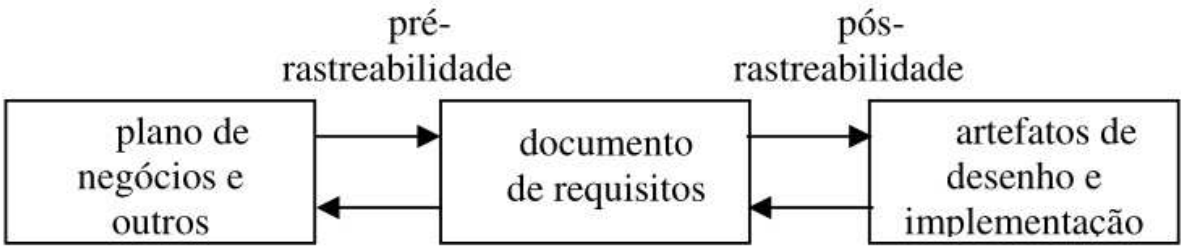
**Quanto à Abrangência dos Artefatos:**

- **Rastreabilidade Horizontal:** Foca em coletar dados sobre as diferentes versões e modificações de requisitos ou outros elementos dentro de uma única etapa do ciclo de vida do projeto.
- **Rastreabilidade Vertical:** Estabelece conexões entre os requisitos e os artefatos gerados em etapas distintas do ciclo de vida do projeto (por exemplo, a ligação entre um requisito e o código ou um teste).

**Quanto ao Momento da Especificação (ilustrado na Figura 2):**

- **Pré-rastreabilidade (Pré-especificação):** Diz respeito aos elementos que servem de base para um requisito, como suas fontes, o ambiente de onde surgem e as interconexões entre os próprios requisitos.
- **Inter-rastreabilidade (Inter-requisitos)**: Mapeia as relações de dependência entre os requisitos.
- **Pós-rastreabilidade (Pós-especificação):** Abrange a rastreabilidade entre os requisitos e os produtos subsequentes desenvolvidos, como o design, o código e os testes.

Figura 2 – Rastreabilidade de requisitos




Fonte: (Sayão; Leite, 2006).

## 2.4 Arquitetura de Software

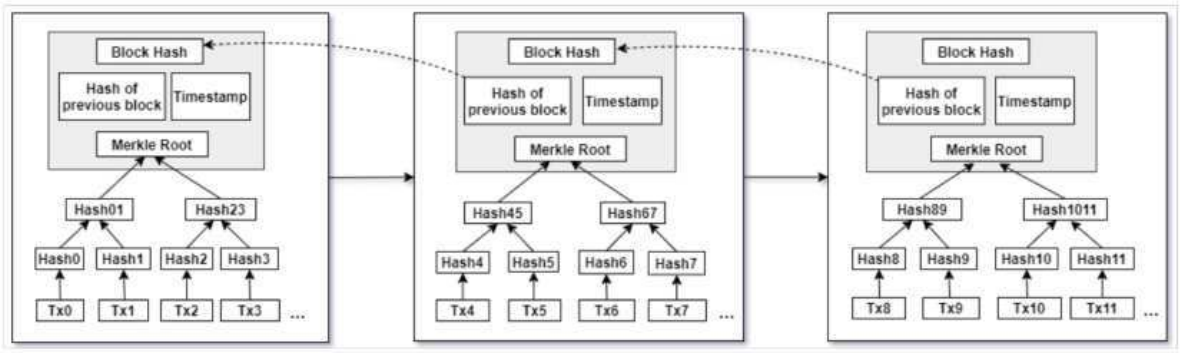
A arquitetura de software refere-se à estrutura fundamental de um sistema computacional, envolvendo seus componentes principais, as propriedades visíveis externamente desses componentes e as relações entre eles (Sommerville, 2011). Ela atua como um projeto de alto nível que orienta o desenvolvimento e facilita a comunicação entre as partes interessadas (*stakeholders*). Decisões arquiteturais são críticas, pois determinam diretamente atributos de qualidade do sistema, como desempenho, segurança, escalabilidade e manutenibilidade.

No contexto do desenvolvimento moderno, compreender a arquitetura é um passo essencial para avaliar o impacto da adoção de novas tecnologias (Sommerville, 2011). A integração de abordagens descentralizadas, como a *blockchain*, em sistemas de gerenciamento de requisitos exige um alinhamento profundo com a arquitetura de software subjacente, garantindo que as propriedades de imutabilidade e distribuição não comprometam a performance e a usabilidade do sistema geral.

## 2.5 *Blockchain*

A *blockchain* é essencialmente um livro-razão digital distribuído e imutável que registra transações de forma segura e transparente. O termo *Blockchain* (Cadeia de blocos) descreve a natureza sequencial da tecnologia. É uma lista crescente de registros, chamados blocos, que são encadeados usando criptografia. Cada bloco contém um *hash* do bloco anterior, um registro de transações, e um *timestamp*, como pode ser observado na Figura 3. Essa ligação garante que, uma vez que um bloco é adicionado à cadeia, ele não pode ser alterado sem invalidar os blocos subsequentes, estabelecendo assim uma ordem cronológica e imutável (Nakamoto, 2008).

Figura 3 – Estrutura do *blockchain*




Fonte: Adaptado de Tschorsch e Scheuermann (2016).

A tecnologia *blockchain*, inicialmente popularizada pelo *bitcoin*, emergiu como um paradigma inovador para o registro e validação de informações de forma distribuída e segura (Tschorsch; Scheuermann, 2016). Sua estrutura subjacente e princípios operacionais a tornam uma ferramenta poderosa para aplicações diversas, incluindo a rastreabilidade de requisitos de software.

### *2.5.1 Consenso*

Em um sistema *blockchain*, não existe uma autoridade central para validar as transações. Em vez disso, a validação é alcançada através de mecanismos de consenso entre os participantes da rede. O algoritmo de consenso mais conhecido é o *Proof of Work* (Prova de Trabalho), utilizado pelo *Bitcoin*, onde os "mineradores"competem para resolver um problema computacional complexo. O primeiro a resolver o problema propõe o próximo bloco, que é então verificado pelos outros nós da rede. Uma vez que a maioria dos nós concorda com a validade do bloco, ele é adicionado à cadeia. Outros mecanismos de consenso incluem *Proof of Stake* (Prova de Participação) e *Delegated Proof of Stake* (Prova de Participação Delegada), que buscam maior eficiência energética (Buterin, 2014).

Enquanto mecanismos como o Proof-of-Work são fundamentais para blockchains públicas e não permissionadas, eles apresentam alto custo computacional. Para contextos de redes privadas ou consorciadas, onde os participantes (nós) já são previamente conhecidos e autorizados, a literatura costuma adotar mecanismos de consenso baseados em votação e identidade. Exemplos notáveis incluem o PBFT (Practical Byzantine Fault Tolerance) e o Proof-of-Authority (PoA), que eliminam a necessidade de mineração intensiva, oferecendo maior escalabilidade, eficiência energética e latência reduzida para a validação das transações (Zheng *et al.*, 2017).

### *2.5.2 Descentralização*

A *Blockchain* é inerentemente descentralizada. Em vez de armazenar todos os dados em um servidor centralizado, cópias do livro-razão são mantidas e atualizadas por múltiplos participantes (nós) na rede. Essa distribuição de dados elimina pontos únicos de falha e torna o sistema mais resistente a ataques cibernéticos e manipulações, pois a alteração de dados em um nó não afeta a integridade do sistema como um todo (Tapscott; Tapscott, 2016).

### *2.5.3 Imutabilidade*

Uma das características mais robustas da *blockchain* é a imutabilidade dos registros. Uma vez que uma transação é validada e adicionada a um bloco, e esse bloco é anexado à cadeia, ela se torna praticamente impossível de ser alterada ou excluída. Isso ocorre devido ao encadeamento dos blocos e aos mecanismos de consenso que exigem a concordância da maioria da rede para qualquer alteração. Essa propriedade é crucial para garantir a integridade e a auditabilidade dos dados ao longo do tempo (Swan, 2015).

### *2.5.4 Criptografia*

A criptografia é a espinha dorsal da segurança da *blockchain*. Chaves criptográficas (pública e privada) são usadas para identificar os usuários e autorizar as transações. As funções de *hash* garantem a integridade dos dados nos blocos e o encadeamento seguro entre eles. Essa aplicação extensiva da criptografia protege as informações contra acesso não autorizado e adulteração (Antonopoulos; Harding, 2023).

## 2.6 Como as Transações são Registradas e Validadas

O processo de registro e validação de transações na *blockchain* segue um fluxo bem definido:

- **Criação da Transação**: Um participante da rede cria uma transação (por exemplo, transferir um requisito de status de "pendente" para "em desenvolvimento" em um sistema de rastreabilidade). A transação é assinada digitalmente com a chave privada do originador.
- **Transmissão para a Rede**: A transação assinada é transmitida para a rede de nós *blockchain*.
- **Validação pelos Nós**: Os nós da rede verificam a validade da transação. Isso inclui checar a assinatura digital, garantir que o originador tenha autorização para realizar a ação, e verificar se a transação está em conformidade com as regras do protocolo.
- **Agregação em um Bloco**: Múltiplas transações validadas são agrupadas em um bloco.
- **Mineração (para *blockchains Proof of Work*)**: No caso de *blockchains* que utilizam *Proof of Work*, os "mineradores"competem para resolver um problema computacional complexo. O primeiro a resolver o problema propõe o novo bloco para a rede.
- **Consenso e Adição à Cadeia**: Os outros nós da rede verificam a validade do bloco

proposto. Se a maioria de nós concordar com sua validade, o bloco é adicionado à cadeia de blocos existente.

- **Propagação e Imutabilidade**: Uma vez adicionado, o novo bloco é propagado por toda a rede, e a transação contida nele se torna parte permanente e imutável do livro-razão distribuído (Nakamoto, 2008; Antonopoulos; Harding, 2023).

## 2.7 Tipos de *Blockchain*

A arquitetura da *blockchain* pode ser adaptada para diferentes cenários, levando a três tipos principais:

*Blockchain* Pública: São redes abertas e transparentes onde qualquer pessoa pode participar, ler as transações, enviar transações e se tornar um validador. Exemplos notáveis incluem *Bitcoin* e *Ethereum*. A principal vantagem é a descentralização máxima e a resistência à censura, mas podem enfrentar desafios de escalabilidade e privacidade para certas aplicações (Swan, 2015).

*Blockchain* Privada: Operadas por uma única entidade, as *Blockchains* privadas têm acesso restrito. Apenas participantes autorizados podem ler, enviar transações ou validar blocos. Embora ofereçam maior controle, velocidade e privacidade, elas são mais centralizadas e, portanto, podem não usufruir de todos os benefícios de segurança e confiança de uma rede pública. São adequadas para cenários internos de uma organização (Antonopoulos; Harding, 2023).

*Blockchain* Consorciada (ou Híbrida): Este tipo combina características das *Blockchains* públicas e privadas. A rede é controlada por um grupo predefinido de organizações, em vez de uma única entidade. O consenso é alcançado pelos membros do consórcio. Elas oferecem um equilíbrio entre descentralização, controle e desempenho, sendo ideais para colaboração entre múltiplas empresas em um setor específico (Tapscott; Tapscott, 2016).

## 2.8 *Smart Contracts* (Contratos Inteligentes) e seu Potencial

Introduzidos e popularizados pela plataforma *Ethereum*, os *Smart Contracts* são programas auto executáveis que definem as regras e condições de um acordo que são armazenados e executados em uma *blockchain*.

Uma vez implantado na *blockchain*, ele é imutável e executa-se automaticamente

quando as condições predefinidas são cumpridas. As regras são codificadas, e a execução é transparente e auditável por todos os participantes da rede (Buterin, 2014). Eles permitem que transações confiáveis sejam realizadas entre partes anônimas sem a necessidade de uma autoridade central, intermediário legal ou sistema de aplicação externa.

O potencial dos *Smart Contracts* é vasto e disruptivo em diversas indústrias, incluindo automação de acordos em processos de negócios, eliminando a necessidade de intermediários e reduzindo custos e tempo (Buterin, 2014). Além disso, a natureza imutável e transparente dos contratos garante que as partes cumprirão seus acordos.

## 2.9 Integração *blockchain* e Gerenciamento de Requisitos

A crescente complexidade dos sistemas de software e a necessidade de maior transparência e auditabilidade no ciclo de vida do desenvolvimento impulsionam a busca por novas abordagens no gerenciamento de requisitos (Dzhalila *et al.*, 2023). A integração da tecnologia *blockchain* nesse domínio surge como uma solução promissora para abordar desafios persistentes, especialmente no que tange à rastreabilidade, imutabilidade e confiança (Farooq *et al.*, 2022a).

### *2.9.1 O Gerenciamento de Requisitos no Desenvolvimento de Software*

O Gerenciamento de Requisitos (GR) é um processo fundamental no desenvolvimento de software que visa garantir que as necessidades dos usuários e as restrições do sistema sejam devidamente identificadas, documentadas, analisadas, priorizadas, rastreadas e comunicadas ao longo de todo o ciclo de vida do projeto. Uma gestão eficaz dos requisitos é crucial para o sucesso de um projeto, minimizando retrabalho, reduzindo custos e aumentando a satisfação do cliente (Pressman; Maxim, 2020). No entanto, o GR tradicional enfrenta desafios significativos, como:

- **Manutenção da Rastreabilidade**: A capacidade de rastrear a origem de um requisito, sua evolução, sua relação com outros requisitos e sua implementação em artefatos de desenvolvimento (código, testes) é muitas vezes complexa e propensa a erros, especialmente em projetos grandes e distribuídos (Gotel; Finkelstein, 1994).
- **Controle de Versão e Histórico de Mudanças**: As alterações nos requisitos são frequentes. Garantir um histórico completo e imutável de todas as modificações, quem as fez e quando, é um desafio.

- **Confiança e Consenso**: Em projetos com múltiplos stakeholders e equipes distribuídas, alcançar consenso e garantir que todas as partes confiam na validade e na versão atual dos requisitos pode ser difícil.
- **Auditoria e Conformidade**: A necessidade de auditar o processo de gerenciamento de requisitos para conformidade com normas regulatórias ou políticas internas exige registros confiáveis e inalteráveis.

### *2.9.2 Possíveis soluções do blockchain no Gerenciamento de Requisitos*

A aplicação da *blockchain* no gerenciamento de requisitos oferece uma abordagem inovadora para mitigar os desafios mencionados, capitalizando nas características intrínsecas da tecnologia, como imutabilidade, descentralização e transparência. A rastreabilidade de requisitos é a capacidade de descrever e seguir o ciclo de vida de um requisito em ambas as direções, para frente e para trás, desde sua concepção até sua implantação e manutenção (Gotel; Finkelstein, 1994). Conforme discutido por Farooq *et al.* (2022a), a *blockchain* pode aprimorar essa rastreabilidade de várias maneiras:

- **Imutabilidade do Histórico de Requisitos**: Cada alteração, aprovação ou rejeição de um requisito pode ser registrada como uma transação em um bloco da *blockchain*. Uma vez que um bloco é adicionado à cadeia, essa informação se torna inalterável e permanente. Isso cria um histórico de auditoria completo e à prova de adulterações, eliminando disputas sobre a versão atual ou anterior de um requisito.
- **Cadeia de Rastreabilidade Verificável**: As ligações entre requisitos, casos de teste, itens de código e defeitos podem ser representadas como transações ou metadados na *blockchain*. Por exemplo, quando um requisito é refinado ou um teste é associado a ele, essas ações podem ser gravadas, formando uma "cadeia de rastreabilidade"verificável criptograficamente.
- **Transparência e Visibilidade**: Em uma *blockchain* pública ou consorciada, todos os participantes autorizados têm acesso à mesma versão do livro-razão distribuído. Isso aumenta a transparência do processo de gerenciamento de requisitos, permitindo que todas as partes interessadas vejam o status, as alterações e as aprovações em tempo real.
- **Descentralização e Consenso Distribuído**: A natureza descentralizada da *blockchain* elimina a necessidade de uma autoridade central para gerenciar os requisitos. As aprovações e validações podem ser realizadas por múltiplos stakeholders de forma distribuída, com o

consenso da rede garantindo a legitimidade das operações. Isso pode ser particularmente útil em projetos de software de código aberto ou em consórcios de empresas.

- **Segurança e Integridade dos Dados**: A criptografia subjacente à *blockchain* protege os dados dos requisitos contra acesso não autorizado e manipulação. Cada transação é assinada digitalmente, garantindo a autenticidade da origem e a integridade da informação.

### *2.9.3 Desafios e Considerações na Integração*

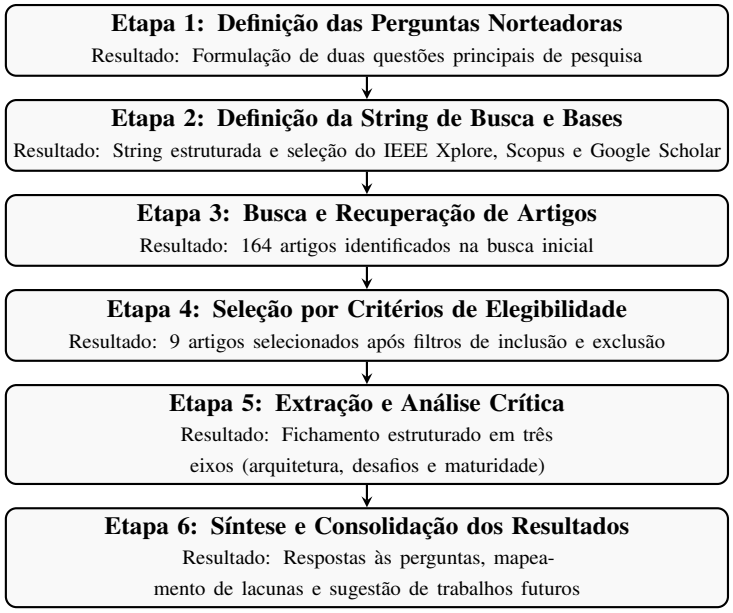
Apesar dos benefícios, a integração da tecnologia *blockchain* no gerenciamento de requisitos pode apresentar desafios notáveis, como a escalabilidade (Dzhalila *et al.*, 2023). O volume de transações e o tamanho dos dados dos requisitos podem ser consideráveis, tornando o desempenho de algumas *blockchains*, especialmente as públicas, uma preocupação (Farooq *et al.*, 2022a). *Blockchains* consorciadas ou privadas podem oferecer um desempenho superior para essa finalidade. Outro desafio é a complexidade de implementação, pois a implementação de uma solução *blockchain* para gerenciamento de requisitos demanda expertise em tecnologias distribuídas, criptografia e desenvolvimento de *Smart Contracts*. A integração com ferramentas existentes também é um ponto a ser considerado, já que a interoperabilidade com ferramentas de gerenciamento de requisitos já em uso (como Jira, Azure DevOps, entre outras) pode exigir o desenvolvimento de APIs específicas. Por fim, há o custo computacional em *blockchains* que utilizam o mecanismo de consenso *Proof of Work*, onde o custo para minerar e manter a rede pode ser elevado, embora isso dependa do tipo de consenso adotado. Em síntese, a integração da *blockchain* no gerenciamento de requisitos representa um avanço significativo. Ao aproveitar a imutabilidade, a descentralização e a automação via *Smart Contracts*, é possível construir sistemas de rastreabilidade de requisitos mais robustos, transparentes e confiáveis, essenciais para o desenvolvimento de software moderno e complexo.

# 3 METODOLOGIA

Esta pesquisa caracteriza-se como uma revisão bibliográfica de caráter crítico-analítico, de abordagem qualitativa, voltada a examinar como a tecnologia *blockchain* vem sendo aplicada na rastreabilidade de requisitos de software. Mais do que descrever individualmente os estudos, a revisão compara conceitos, arquiteturas propostas, métodos de validação, resultados alcançados, limitações apontadas e a utilidade dessas soluções para o problema investigado.

Para a composição do acervo bibliográfico, foram consultadas publicações nas bases de dados IEEE Xplore, Scopus e Google Scholar. Essas plataformas foram selecionadas devido ao seu vasto volume de trabalhos indexados e elevado rigor científico na área de Engenharia de Software. A pesquisa adota um recorte crítico da literatura, estruturando a análise em dimensões específicas: (i) tipos de arquitetura e mecanismos de integração entre *blockchain* e rastreabilidade de requisitos; (ii) desafios técnicos enfrentados (escalabilidade, custos, privacidade, interoperabilidade); e (iii) nível de maturidade e evidência empírica de cada proposta. Essa organização permite avaliar comparativamente as contribuições e fragilidades de cada estudo. A seleção dos estudos adotou critérios de elegibilidade para garantir a transparência, o rigor e a relevância das fontes consultadas, evitando vieses de seleção comuns em revisões tradicionais (Rother, 2007). Na Figura 4 são apresentadas as etapas que compõem o fluxo metodológico desta pesquisa.

Figura 4 – Fluxo das etapas da metodologia da pesquisa




Fonte: Elaborado pelo autor.

## 3.1 Definição das questões norteadoras

As revisões bibliográficas de caráter crítico-analítico são guiadas por questões centrais que direcionam a investigação e a análise crítica da literatura. Para este estudo, foram estabelecidas duas questões principais, que abordam os mecanismos propostos para a integração da *blockchain* e os desafios técnicos inerentes a essa aplicação. As questões definidas foram:

1. Quais são os mecanismos e arquiteturas propostos na literatura para integrar a tecnologia *blockchain* na rastreabilidade de requisitos de software?
2. Como essas soluções arquiteturais lidam com os desafios técnicos inerentes à tecnologia, especialmente escalabilidade e privacidade?

## 3.2 Definição da string de busca

Para compor a base literária, foi elaborada uma estratégia de busca combinando os principais termos do domínio do estudo, conforme detalhado a seguir:

*(("distributed ledger"OR "blockchain"OR "smart contract") AND ("requirements traceability") AND ("software engineering"OR "software development"OR "system engineering"))*

## 3.3 Definição das bases de dados

Conforme detalhado no Quadro 1, as plataformas selecionadas para a extração dos artigos abrangem relevantes bibliotecas digitais da área de computação, garantindo a qualidade técnica e a atualidade dos estudos avaliados.

Quadro 1 – Bases de Dados para Pesquisa

| **Base de dados** | **Endereço Web** |
|---|---|
| IEEE Xplore | https://ieeexplore.ieee.org/Xplore/home.jsp |
| Scopus | https://www.scopus.com/home.uri |
| Google Scholar | https://scholar.google.com |

Fonte: Elaborado pelo autor.

## 3.4 Definição dos critérios de seleção de artigos

Para filtrar os resultados da busca e garantir que a análise crítica recaísse apenas sobre obras pertinentes e atualizadas, foram estabelecidos os critérios de inclusão e exclusão para todos os artigos, apresentados, respectivamente, no Quadro 2 e no Quadro 3.

Quadro 2 – Critérios de inclusão

| Critério | Descrição |
|---|---|
| CI.1 | Estudos que investigam diretamente a aplicação de *blockchain* ou *smart contracts* na rastreabilidade de requisitos de software |
| CI.2 | Disponibilidade do texto integral em inglês ou português |
| CI.3 | Estudos publicados entre 2021 e 2025 |

Fonte: Elaborado pelo autor.

Quadro 3 – Critérios de exclusão

| Critério | Descrição |
|---|---|
| CE.1 | Artigos duplicados entre as bases de dados |
| CE.2 | Publicações que não abordam a tecnologia *blockchain* |
| CE.3 | Artigos que aplicam *blockchain* em contextos alheios à Engenharia e Rastreabilidade de Requisitos |
| CE.4 | Estudos cujos textos integrais não estejam acessíveis para consulta |
| CE.5 | Estudos secundários (tais como revisões sistemáticas e mapeamentos da literatura) |
| CE.6 | Literatura cinza (tais como teses, dissertações, relatórios técnicos e preprints) |

Fonte: Elaborado pelo autor.

## 3.5 Extração e Análise de Dados

Após a seleção dos nove artigos, procedeu-se a uma leitura analítica orientada por três eixos principais: (i) arquitetura e mecanismos propostos; (ii) desafios e limitações técnicas relatados; e (iii) qualidade e nível de maturidade das evidências apresentadas. Esses eixos não se limitaram à extração de informações, mas serviram para avaliar criticamente a adequação das soluções à rastreabilidade de requisitos, bem como identificar lacunas recorrentes na literatura. Para garantir a consistência e o rigor científico da análise, os dados foram extraídos e organizados com foco nos seguintes eixos:

1. **Arquitetura e mecanismos propostos:** identificação do tipo de rede *blockchain* adotado (pública, privada ou de consórcio), plataformas de desenvolvimento, emprego de *smart contracts*, estratégias de armazenamento de dados (*on-chain* ou *off-chain*) e o nível de granularidade da rastreabilidade dos requisitos.
2. **Desafios e limitações técnicas:** investigação de como os trabalhos lidam com problemas de escalabilidade, latência de transações, custos operacionais (*gas*) e o tratamento da privacidade e confidencialidade das informações dos requisitos.
3. **Qualidade e nível de maturidade do estudo:** avaliação de como as soluções propostas foram validadas (se em ambiente de laboratório, simulações ou em projetos corporativos) e o nível de integração com ferramentas de gerência de requisitos consolidadas na indústria (e.g., Jira, GitHub).

## 3.6 Síntese e Consolidação dos Resultados

Na etapa de síntese, os resultados foram organizados em eixos temáticos alinhados às questões de pesquisa, com ênfase na comparação crítica entre os estudos. Foram identificadas convergências (por exemplo, no uso predominante de Ethereum e de armazenamento híbrido), divergências (como o emprego de *blockchains* neurais e bancos de grafos) e inconsistências (como propostas que ignoram custos ou privacidade), o que permitiu discutir a robustez e as limitações de cada abordagem. A síntese dos resultados é apresentada na Seção 5, e as lacunas teóricas e de desenvolvimento identificadas servem de base para a proposição de caminhos para trabalhos futuros na Seção 6.

## 3.7 Considerações finais

Nesta seção, foram descritos os procedimentos e as etapas adotadas para a condução da revisão bibliográfica de caráter crítico-analítico sobre a aplicação da tecnologia *blockchain* na rastreabilidade de requisitos de software. Os passos delineados, desde a formulação das questões norteadoras e da estratégia de busca até a aplicação rigorosa dos critérios de seleção, seguidos pela análise detalhada e síntese crítica dos dados, garantem a estruturação necessária para explorar o tema de forma embasada. Essa abordagem metodológica permitirá não apenas responder às questões de pesquisa levantadas, mas também viabilizará uma análise crítica e uma discussão teórica aprofundada sobre os benefícios práticos e os desafios técnicos inerentes à adoção dessas soluções descentralizadas.

# 4 ANÁLISE DA LITERATURA

## 4.1 Introdução

Nesta seção são apresentados e discutidos criticamente os nove estudos selecionados, à luz das questões de pesquisa. Para cada trabalho, são examinadas a adequação da arquitetura proposta ao problema de rastreabilidade de requisitos, as principais limitações técnicas identificadas e o nível de evidência empírica apresentado, de modo a avaliar a real contribuição de cada solução para o avanço da área.

## 4.2 Artigos analisados

### *4.2.1 A Blockchain-Based Framework for Distributed Agile Software Development*

No que tange à arquitetura e mecanismos, Farooq *et al.* (2022b) propõem o *framework* AgilePlus, que adota uma *blockchain* privada baseada na plataforma Ethereum. A escolha por uma rede privada é justificada pela necessidade de prevenir ataques de 51% e garantir que apenas participantes autorizados, como clientes, gerentes de projeto e desenvolvedores, possam interagir com o sistema. Os *smart contracts* desempenham um papel que vai além do simples registro de logs: eles automatizam regras de negócio como testes de aceitação, verificação de requisitos de pagamento, distribuição automática de *ethers* entre equipes e aplicação de penalidades contratuais caso prazos não sejam cumpridos. Quanto ao armazenamento, o *framework* adota uma estratégia híbrida: os dados das seis camadas operacionais (acordo, elicitação de requisitos, priorização, design e desenvolvimento, testes e por fim pagamento) são armazenados *on-chain*, enquanto dados volumosos, como registros de cadastro, mensagens e videoconferências, são armazenados *off-chain* via IPFS. A granularidade da rastreabilidade situa-se no nível de *user stories*, rastreando o progresso das equipes ágeis ao longo das seis camadas.

Em relação aos desafios e limitações, Farooq *et al.* (2022b) abordam explicitamente a questão da escalabilidade ao utilizar o IPFS como armazenamento secundário descentralizado, reduzindo a carga de dados sobre a *blockchain* e melhorando o desempenho das transações. A privacidade é parcialmente tratada por meio de chaves privadas atribuídas pelo administrador do sistema para autenticação. Contudo, o artigo não aprofunda mecanismos de criptografia fim a fim para a proteção de segredos industriais contidos nos requisitos. O mecanismo de consenso adotado é o *Proof-of-Work* (PoW), o que, em contextos de equipes ágeis distribuídas

com mudanças frequentes de requisitos, pode gerar latências significativas e elevado custo computacional. Embora os autores avaliem a latência para recuperação da cadeia mais longa e o crescimento do tamanho da *chain*, trata-se de uma rede privada, portanto não há custos de *gas* efetivos, mas tampouco há estimativa do volume de transações nem análise de escalabilidade para cenários industriais de desenvolvimento ágil.

Quanto à qualidade e maturidade, o estudo apresenta um protótipo funcional implementado e avaliado em termos de desempenho. Todavia, a avaliação é conduzida em ambiente de laboratório, sem aplicação em projetos práticos da indústria. O *framework* propõe uma interface própria via *DApp* e portal *web*, porém não oferece integração com ferramentas consolidadas do mercado como Jira, Trello ou GitHub. Um ponto crítico reside na adoção do *Proof-of-Work* para uma *blockchain* privada, uma vez que mecanismos como *PBFT* ou *Proof-of-Authority* seriam mais adequados e eficientes para redes permissionadas com participantes conhecidos, levantando a questão de se a complexidade introduzida é realmente necessária para o problema abordado.

Em síntese, o estudo avança ao propor um *framework* detalhado para rastreabilidade em ambientes ágeis distribuídos, mas limita-se a avaliações em laboratório, sem aplicação em ambientes reais. Além disso, a adoção de *Proof-of-Work* em rede privada e a ausência de integração com ferramentas amplamente utilizadas (como Jira ou GitHub) reduzem a viabilidade prática da solução.

### *4.2.2 Blockchain Based Software Engineering Requirements Analysis and Management*

No aspecto da arquitetura e mecanismos, Ghadmi *et al.* (2022) propõem um *framework* semi-centralizado que utiliza a rede Ethereum para validar requisitos de software por meio de *smart contracts*. A solução adota uma arquitetura em que cada requisito proposto por um *vendor* deve ser assinado e confirmado pelo *customer* através de transações na *blockchain*, criando um registro imutável de aprovações e rejeições. Os *smart contracts*, escritos em Solidity, automatizam a verificação e confirmação de requisitos entre as partes, funcionando como um mecanismo de consenso bilateral entre *stakeholders*. O armazenamento segue uma estratégia *off-chain*: dados sensíveis e detalhes dos requisitos são mantidos em um servidor *back-end* centralizado, enquanto apenas os *hashes* de verificação das transações são registrados na *blockchain*. A granularidade da rastreabilidade opera no nível de requisitos individuais, permitindo rastrear cada requisito desde sua criação até sua aprovação ou rejeição.

Em relação aos desafios e limitações, Ghadmi *et al.* (2022) reconhecem explicita-

mente o alto custo de armazenamento na *blockchain* Ethereum, estimando o valor de aproximadamente US$ 76.000 por gigabyte, o que justifica a abordagem *off-chain* para os dados dos requisitos. A privacidade é tratada pela separação entre dados públicos (na *blockchain*) e dados privados (no *back-end* centralizado), porém essa abordagem introduz um ponto único de falha que contradiz os princípios de descentralização da *blockchain*. O mecanismo de consenso é o nativo da rede Ethereum pública, o que implica custos de *gas* para cada transação de validação de requisito. Contudo, não há uma análise quantitativa do custo acumulado para projetos com centenas ou milhares de requisitos, nem da latência envolvida.

Quanto à qualidade e maturidade, o estudo apresenta um protótipo funcional desenvolvido com o *framework* Truffle e Ganache para simulação da *blockchain* Ethereum, caracterizando uma Prova de Conceito (PoC) em ambiente de laboratório. A validação é limitada a cenários simulados, sem aplicação em contextos reais. A solução não propõe nenhuma integração com ferramentas existentes, embora os autores mencionem, como trabalho futuro, a possibilidade de disponibilizar o *framework* como uma API que possa ser embarcada em sistemas já existentes na indústria.

De modo geral, o estudo contribui ao formalizar a validação de requisitos por meio de transações em *blockchain*, porém a adoção da Ethereum como plataforma de referência para um cenário bilateral entre partes conhecidas revela um descompasso entre a complexidade da solução e a simplicidade do problema. A ausência de análise quantitativa de custos acumulados e a dependência de um *back-end* centralizado enfraquecem a proposta de descentralização que a *blockchain* deveria oferecer.

### *4.2.3 Blockchain for Requirements Traceability: A Qualitative Approach*

No que diz respeito à arquitetura e mecanismos, Demi *et al.* (2023) adotam uma abordagem qualitativa baseada em *Grounded Theory* (GT) para investigar os fatores que influenciam a implementação da *blockchain* em contextos organizacionais, com foco na rastreabilidade de requisitos. Diferentemente dos demais estudos analisados, este trabalho não propõe uma arquitetura técnica específica, mas sim uma teoria fundamentada em dados coletados por meio de entrevistas semiestruturadas com dez especialistas em *blockchain* de diversos países e setores (saúde, bancos, energia, administração pública). Os especialistas discutem a aplicabilidade de *blockchains* permissionadas (de consórcio) versus públicas, destacando que a escolha depende do contexto organizacional. Os *smart contracts* são mencionados como mecanismos potenciais

para automatizar políticas de rastreabilidade, mas sem detalhamento de implementação.

Em relação aos desafios e limitações, os especialistas entrevistados por Demi *et al.* (2023) levantam preocupações relevantes sobre a privacidade de dados sensíveis em ambientes de desenvolvimento de software, sugerindo que *blockchains* permissionadas são mais adequadas para esse contexto. A questão da escalabilidade é discutida de forma qualitativa, com especialistas apontando que a performance da *blockchain* pode ser um fator de relutância para profissionais de engenharia de software, conforme ilustrado pela preocupação de que "se eu quiser criar qualquer artefato e o processo de validação e escrita dos blocos levar uma hora, isso poderia ser problemático". Não há, contudo, métricas quantitativas de desempenho. O mecanismo de consenso não é especificado em detalhe, uma vez que o foco do estudo reside nos fatores organizacionais e humanos, não nos aspectos técnicos da implementação.

Quanto à qualidade e maturidade, o estudo destaca-se pela robustez metodológica na aplicação da *Grounded Theory*, com critérios rigorosos de seleção de especialistas baseados em publicações, experiência prática e formação acadêmica. A principal contribuição reside nos constructos teóricos que fundamentam trabalhos futuros de implementação. Contudo, a ausência de qualquer protótipo ou validação técnica limita a contribuição prática imediata do trabalho. Um viés identificável é que a amostra de especialistas é majoritariamente composta por profissionais e acadêmicos europeus (Noruega, Suécia, Espanha), o que pode não refletir desafios específicos de outros contextos geográficos e culturais. O estudo reconhece explicitamente a resistência dos engenheiros de software à adoção de novas tecnologias, um aspecto frequentemente ignorado em propostas puramente técnicas.

Portanto, o estudo avança ao capturar fatores organizacionais e humanos negligenciados pelas propostas puramente técnicas, oferecendo constructos teóricos valiosos para orientar implementações futuras. No entanto, limita-se a percepções qualitativas sem qualquer validação técnica ou prototipação, e o viés geográfico da amostra de especialistas pode restringir a generalização dos achados a outros contextos culturais e industriais.

### *4.2.4 Blockchain Technology for Requirement Traceability in Systems Engineering*

No que tange à arquitetura e mecanismos, Elapolu *et al.* (2024) propõem um *framework* de rastreabilidade de requisitos baseado em *blockchain* com um enfoque inovador em engenharia de sistemas. A solução utiliza uma rede *blockchain* operada por múltiplos nós correspondentes às equipes de desenvolvimento distribuídas, embora o artigo não especifique

explicitamente se a rede é pública, privada ou de consórcio. O design sugere uma rede de consórcio, dado que os nós são controlados por equipes conhecidas. Uma contribuição original deste trabalho é a rastreabilidade em nível dual: tanto no nível de artefatos (documentos de requisitos) quanto no nível de objetos (requisitos individuais dentro de cada artefato). Os *smart contracts* não são empregados; em vez disso, a rastreabilidade é embutida na estrutura dos blocos por meio de metadados e dados de transações. Adicionalmente, a *blockchain* é representada como um grafo de propriedades no banco de dados Neo4J, permitindo consultas via linguagem Cypher para examinar o histórico de requisitos e suas inter-relações.

Em relação aos desafios e limitações, Elapolu *et al.* (2024) não abordam explicitamente questões de privacidade de dados sensíveis contidos nos requisitos. A imutabilidade da *blockchain* é garantida pelo uso de *hashes* criptográficos para identificação de blocos e artefatos, mas não há discussão sobre controle de acesso ou criptografia dos conteúdos. A escalabilidade é parcialmente demonstrada: o estudo apresenta um cenário com 17 blocos que necessitam de 83 nós e 162 relacionamentos no grafo Neo4J, porém projetos comerciais de engenharia de sistemas podem envolver milhares de artefatos e requisitos. Não há análise de custo computacional, taxas de transação ou latência, uma vez que a *blockchain* implementada é customizada e não utiliza uma plataforma pública com custos de *gas*. O mecanismo de consenso utilizado é o *Proof of Work* (PoW); contudo, a viabilidade e o custo computacional de se empregar um algoritmo intensivo como o PoW em uma rede de consórcio entre equipes conhecidas não são criticamente discutidos, representando uma lacuna na avaliação do *framework*.

Quanto à qualidade e maturidade, o estudo apresenta uma implementação funcional aplicada ao processo de engenharia de requisitos de um sistema automotivo autônomo, o que confere maior relevância prática em comparação com protótipos puramente acadêmicos. A integração com Neo4J para visualização gráfica dos relacionamentos de rastreabilidade é um diferencial significativo. Entretanto, a validação é conduzida com um cenário predefinido e controlado, sem a participação de engenheiros de software em projetos industriais. A interoperabilidade é limitada, pois o *framework* utiliza *templates* de aquisição de dados próprios, sem propor integração com ferramentas de gestão de requisitos existentes. Uma lacuna relevante é a ausência de comparação com soluções tradicionais de rastreabilidade, como matrizes de rastreabilidade ou ferramentas como IBM DOORS, dificultando a avaliação do valor agregado real da *blockchain* nesse contexto.

Em resumo, o estudo inova ao propor rastreabilidade em nível dual e ao integrar

a *blockchain* com um banco de dados de grafos para consultas, porém opera com um cenário de escala reduzida (17 blocos) que não demonstra viabilidade para projetos comerciais de engenharia de sistemas. A ausência de discussão sobre privacidade, a falta de análise crítica sobre a adoção do mecanismo de consenso PoW em rede de consórcio e a não comparação com ferramentas tradicionais como IBM DOORS limitam a avaliação do valor agregado efetivo da solução.

### *4.2.5 ChainAgile: A Framework for the Improvement of Scrum Agile Distributed Software Development Based on Blockchain*

No que diz respeito à arquitetura e mecanismos, Qureshi e Farooq (2024) propõem o *framework* ChainAgile, que integra *blockchain* ao processo Scrum ágil distribuído. A solução utiliza uma *blockchain* Ethereum privada, adotando o mesmo modelo do AgilePlus proposto pelo mesmo grupo de pesquisa. Os *smart contracts* são utilizados para formalizar *user stories* após consenso entre *stakeholders*, automatizar testes de aceitação, gerenciar pagamentos e registrar acordos contratuais entre clientes e equipes de desenvolvimento. O armazenamento segue uma estratégia híbrida: dados operacionais das seis camadas são armazenados diretamente na *blockchain*, enquanto registros de clientes, mensagens de grupo e videoconferências são armazenados via IPFS. A rastreabilidade é aplicada no nível de *user stories* ao longo das fases do Scrum: elicitação de requisitos, priorização, design e desenvolvimento, e testes.

Em relação aos desafios e limitações, Qureshi e Farooq (2024) apresentam uma extensa revisão dos desafios em desenvolvimento ágil distribuído, incluindo transparência, segurança, rastreabilidade e confiança. Contudo, a análise de escalabilidade e custo permanece superficial: não há métricas quantitativas sobre custos de *gas* na rede Ethereum, latência de transações ou impacto no desempenho à medida que o volume de *user stories* e iterações cresce. A privacidade de dados sensíveis nos requisitos não é discutida em profundidade, e o armazenamento direto de *posts* e comentários na *blockchain* levanta preocupações sobre exposição de informações estratégicas. O mecanismo de consenso adotado explicitamente pelo *framework* é o *Proof of Work* (PoW), o que introduz limitações severas de custo computacional e latência, uma escolha arquitetural questionável e ineficiente para uma *blockchain* privada onde todos os participantes (clientes e equipes) já são previamente conhecidos e autorizados.

Quanto à qualidade e maturidade, o estudo realiza uma comparação tabelada com trabalhos relacionados, demonstrando que o ChainAgile cobre mais dimensões (escalabilidade,

comunicação, coordenação, testes, segurança e priorização) do que soluções anteriores. Entretanto, a validação permanece no nível teórico-conceitual, sem a apresentação de um protótipo funcional implementado ou de resultados experimentais em ambientes de produção. A interoperabilidade com ferramentas existentes (Jira, Trello, GitHub) não é abordada, o que limita a adoção prática. Um viés observável é a forte sobreposição com o trabalho de Farooq *et al.* (2022b), sendo o ChainAgile uma evolução do AgilePlus proposto pelo mesmo grupo de pesquisa, o que reduz a diversidade de perspectivas na análise.

Como reflexo dessa análise, o ChainAgile amplia o escopo do AgilePlus ao cobrir mais dimensões do desenvolvimento ágil distribuído, mas permanece inteiramente no plano conceitual, sem protótipo funcional ou métricas quantitativas de desempenho. A forte sobreposição com trabalhos anteriores do mesmo grupo de pesquisa e a ausência de integração com ferramentas consolidadas do mercado comprometem a originalidade e a aplicabilidade prática da proposta.

### *4.2.6 Harnessing the Potential of Blockchain in ChainAgilePlus Framework for the Improvement of Distributed Scrum of Scrums Agile Software Development*

No que tange à arquitetura e mecanismos, Qureshi *et al.* (2024) apresentam o *framework* ChainAgilePlus como uma evolução do ChainAgile, estendido para suportar o modelo *Scrum of Scrums* em ambientes distribuídos. A arquitetura adota a *blockchain* Ethereum com uma estrutura de sete camadas (interface, aplicação, lógica de negócio, confiança, transação, infraestrutura e segurança). Os *smart contracts*, escritos em Solidity, automatizam extensivamente regras de negócio: termos e condições contratuais, testes de aceitação, pagamentos e penalidades automáticas. O armazenamento segue a mesma estratégia híbrida do AgilePlus: dados das camadas operacionais são armazenados *on-chain*, enquanto o IPFS é utilizado para armazenamento descentralizado *off-chain* de comunicações. A rastreabilidade abrange o nível de *user stories* e se estende ao longo de seis camadas operacionais: acordo, elicitação de requisitos, priorização, design e desenvolvimento, testes e pagamento.

Em relação aos desafios e limitações, Qureshi *et al.* (2024) mencionam o uso do *Proof-of-Work* como mecanismo de consenso, o que herda as mesmas limitações de custo computacional e latência já identificadas na análise do AgilePlus. O artigo apresenta resultados experimentais de desempenho da *blockchain*, medindo o crescimento do tamanho da cadeia e a latência com a adição de novos blocos, porém essas métricas avaliam o comportamento

técnico da *blockchain* em si, e não a efetividade funcional do *framework* em projetos comerciais de desenvolvimento. A privacidade é abordada por meio de carteiras criptográficas e chaves privadas para autenticação, mas sem mecanismos avançados de controle de acesso granular que protejam requisitos com diferentes níveis de confidencialidade.

Quanto à qualidade e maturidade, o ChainAgilePlus representa uma evolução incremental em relação ao ChainAgile e AgilePlus, com a principal contribuição sendo a extensão para o modelo *Scrum of Scrums*, que envolve coordenação entre múltiplas equipes Scrum. Contudo, assim como seus predecessores, a validação permanece limitada a cenários simulados, sem aplicação em ambientes industriais reais. A interoperabilidade com ferramentas existentes não é abordada. Um viés relevante é que os três trabalhos (AgilePlus, ChainAgile e ChainAgilePlus) originam-se do mesmo grupo de pesquisa, com evoluções incrementais, e a complexidade crescente da arquitetura de sete camadas levanta a questão de se a sobrecarga introduzida pela *blockchain* é proporcional aos benefícios obtidos, considerando que muitas das funcionalidades propostas (comunicação, priorização, pagamentos) já são resolvidas por ferramentas especializadas existentes no mercado.

Assim, o estudo avança ao estender a abordagem para o modelo *Scrum of Scrums*, mas herda as mesmas limitações dos seus predecessores: ausência de validação industrial, uso questionável de *Proof-of-Work* em rede permissionada e falta de integração com ferramentas existentes. A complexidade crescente da arquitetura de sete camadas, sem evidências empíricas proporcionais, levanta dúvidas sobre a real necessidade da *blockchain* para funcionalidades já atendidas por ferramentas consolidadas.

### *4.2.7 Trustworthy and Collaborative Traceability Management: Experts' Feedback on a Blockchain-Enabled Framework*

No que diz respeito à arquitetura e mecanismos, Demi *et al.* (2024) apresentam a validação qualitativa com especialistas da indústria para o protótipo BC4RT (*Blockchain for Requirements Traceability*), cuja implementação técnica foi detalhada no estudo analisado na Seção 4.2.9. O protótipo utiliza a plataforma NDL ArcaNet, uma *blockchain* neural de terceira geração que opera como um *ledger* distribuído ponto a ponto. Os requisitos são modelados como *tokens* digitais únicos e valiosos, e cada *token* emite seu próprio *ledger blockchain* para auditar seu ciclo de vida. A estrutura dos blocos inclui ULID (*Universally Unique Lexicographically Sortable Identifier*), assinaturas dos blocos e conteúdos dos *tokens*. Os *smart contracts* não

são utilizados na implementação atual, embora os especialistas entrevistados recomendem sua incorporação para avaliar automaticamente se os requisitos foram implementados com sucesso. O armazenamento é integralmente *on-chain*, uma vez que a plataforma NDL ArcaNet suporta nativamente o armazenamento seguro de ativos digitais de qualquer tamanho e tipo. A rastreabilidade é implementada no nível de requisitos individuais, ligando cada requisito aos artefatos de código-fonte e casos de teste associados.

Em relação aos desafios e limitações, os especialistas em engenharia de software entrevistados por Demi *et al.* (2024) apontam desafios práticos significativos. A questão da resistência à mudança é destacada: profissionais de software são frequentemente conservadores e relutantes em adotar novas ferramentas que adicionem etapas ao seu fluxo de trabalho. A *performance* da *blockchain* é identificada como um fator preocupante, conforme expresso por especialistas que alertam que tempos de validação de blocos elevados seriam problemáticos em ambientes de desenvolvimento ágeis. A privacidade é tratada pelo mecanismo de certificadores confiáveis da plataforma NDL ArcaNet, que validam operações sobre os *tokens* sem conhecer seu conteúdo. O mecanismo de consenso se baseia nesses certificadores confiáveis, diferindo dos tradicionais PoW ou PoS e, segundo os autores, evitando algoritmos computacionalmente dispendiosos.

Quanto à qualidade e maturidade, este é um dos estudos mais maduros entre os analisados, pois combina a proposta do *framework*, a implementação do protótipo e uma validação qualitativa com dez especialistas em engenharia de software. As recomendações dos especialistas são particularmente valiosas: a integração leve com ferramentas comerciais existentes como Jira, Kanban, GitHub e GitLab é identificada como a recomendação central para viabilizar a adoção. A necessidade de registrar a justificativa das mudanças (*change rationale*), oferecer opções de configuração de papéis e processos, e disponibilizar a solução como aplicação *web* são contribuições práticas relevantes. Uma lacuna reconhecida pelos próprios autores é que o protótipo ainda necessita ser levado a um estágio de produção, e estudos com usuários finais em cenários práticos ainda são necessários.

Diante do exposto, este é o estudo que mais se aproxima de uma solução viável, ao combinar protótipo funcional com validação por especialistas e mecanismos de segurança robustos. Contudo, o protótipo ainda não alcançou estágio de produção, e a dependência da plataforma proprietária NDL ArcaNet pode limitar a reprodutibilidade e a adoção independente por outros grupos de pesquisa e pela indústria.

### *4.2.8 A Blockchain-Enabled Framework for Requirements Traceability*

No que tange à arquitetura e mecanismos, Demi *et al.* (2021) apresentam um *framework* conceitual para rastreabilidade de requisitos habilitado por *blockchain*, composto por quatro fases: estratégia, proposta *blockchain*, implementação e avaliação. Na fase estratégica, destaca-se o modelo de informação de rastreabilidade (TIM – *Traceability Information Model*), que define quais artefatos rastrear e quais relações estabelecer entre eles. Esses elementos podem ser codificados em *smart contracts* para automatizar o registro de artefatos e a identificação semântica dos *trace links*. O *framework* propõe que os *smart contracts* validem a qualidade dos *trace links* por meio de um mecanismo de votação: aprovadores fornecem consentimento ou rejeição registrado como voto no *ledger* distribuído, e o contrato inteligente calcula os votos cumulativos para aceitar ou rejeitar os links. O armazenamento é explicitamente projetado como híbrido: conteúdos de artefatos são armazenados *off-chain* em armazenamento imutável como IPFS, enquanto apenas os metadados e *hashes* permanecem na *blockchain*. A granularidade da rastreabilidade abrange desde requisitos até código-fonte e casos de teste, com *links* primitivos (entre artefatos adjacentes) e compostos (entre artefatos não adjacentes).

Em relação aos desafios e limitações, Demi *et al.* (2021) reconhecem explicitamente os custos de armazenamento em plataformas como Ethereum, que consomem *gas* e podem causar problemas de sincronização de rede e alto consumo de espaço em disco nos nós. A solução proposta de armazenamento *off-chain* via IPFS mitiga parcialmente essa limitação. Um diferencial inovador é o mecanismo de incentivos: políticas de recompensas codificadas em *smart contracts* distribuem *tokens* digitalizados para *stakeholders* que criam *trace links* de qualidade, abordando o problema motivacional da manutenção de rastreabilidade. A privacidade não é discutida em profundidade. A escolha da plataforma *blockchain* é deixada em aberto, com diretrizes gerais baseadas em acessibilidade da rede, suporte a *smart contracts* e necessidade de *tokens*, o que confere flexibilidade mas também imprecisão ao *framework*.

Quanto à qualidade e maturidade, o estudo é essencialmente teórico-conceitual, apresentando o *framework* sem implementação ou validação empírica. A principal contribuição reside na sistematização das fases e componentes necessários para a adoção da *blockchain* na rastreabilidade, e no mecanismo de incentivos para criação de *trace links*, um aspecto geralmente ignorado pela literatura. A interoperabilidade é mencionada conceitualmente: o *framework* sugere o uso de ferramentas de ingestão de dados e *plugins* para capturar artefatos gerados por fontes diversas, como DOORS para gestão de requisitos e Git como sistema de controle

de versão. Contudo, essas integrações não são implementadas nem detalhadas. Embora este trabalho apresente uma lacuna inicial quanto à discussão sobre a resistência organizacional à adoção e a curva de aprendizado para a operação com tecnologias *blockchain*, essa questão seria posteriormente reconhecida e endereçada pela própria autora em estudos qualitativos subsequentes (como o analisado na Seção 4.2.3).

Em perspectiva, o estudo contribui de forma relevante ao sistematizar as fases de adoção da *blockchain* na rastreabilidade e ao propor mecanismos de incentivo e governança para a criação de *trace links*, aspectos frequentemente negligenciados. Porém, a natureza puramente conceitual do trabalho, sem qualquer implementação ou validação empírica, e a indefinição da plataforma *blockchain* a ser utilizada limitam significativamente a contribuição prática imediata.

### *4.2.9 A Neural Blockchain for Requirements Traceability: BC4RT Prototype*

No que diz respeito à arquitetura e mecanismos, Demi *et al.* (2022) apresentam o protótipo BC4RT (*Blockchain for Requirements Traceability*), implementado sobre a plataforma NDL ArcaNet, uma *blockchain* neural de terceira geração. A escolha desta plataforma é motivada por suas características de eficiência, sustentabilidade e escalabilidade, superiores às plataformas convencionais como Ethereum. Os requisitos são modelados como *tokens* digitais valiosos e únicos, cada um gerando seu próprio *ledger blockchain* para auditar seu ciclo de vida completo. Os *smart contracts* não são utilizados na implementação atual; em vez disso, a lógica de estados dos requisitos (criado, alterado, implementado, testado) é gerenciada pela plataforma NDL ArcaNet. O armazenamento é integralmente *on-chain*, uma vez que a plataforma suporta nativamente arquivos de qualquer tamanho e tipo, eliminando a necessidade de armazenamento *off-chain* como IPFS. A granularidade da rastreabilidade é notavelmente refinada e representa a materialização prática da estrutura de rastreabilidade proposta conceitualmente pela própria autora no estudo analisado na Seção 4.2.8: cada requisito é rastreado em seus atributos (versão, descrição, status, contribuidor, *timestamp*) e vinculado diretamente a artefatos de código-fonte e casos de teste.

Em relação aos desafios e limitações, Demi *et al.* (2022) abordam a escalabilidade por meio da própria arquitetura da plataforma NDL ArcaNet, que é internamente estruturada em subconjuntos de grupos que trabalham em paralelo e são interconectados de forma análoga a como grupos de neurônios são agregados em cérebros humanos. A segurança é garantida por criptografia dupla (senha do banco de dados e senha do *token*), AES-256, RSA, e criptografia

de conhecimento zero. O mecanismo de consenso se baseia em certificadores confiáveis que são incentivados a validar operações, diferindo substancialmente dos mecanismos tradicionais e potencialmente evitando os custos computacionais do PoW. Contudo, a dependência de certificadores confiáveis introduz um elemento de centralização que pode comprometer a promessa de descentralização total da *blockchain*. O protótipo implementa quatro papéis de usuário (gerente de requisitos, desenvolvedor, testador e cliente), cada um com permissões específicas, o que mitiga parcialmente questões de privacidade, porém sem controle de acesso granular para diferentes tipos de requisitos.

Quanto à qualidade e maturidade, o BC4RT representa o protótipo mais completo e funcional entre os trabalhos do grupo de Demi *et al.* (2022), com uma interface que permite criar projetos, criar e atualizar requisitos, *upload* de código-fonte e casos de teste, e rastrear o ciclo de vida completo de cada requisito. A utilização de uma *blockchain* neural de terceira geração é inovadora e endereça limitações reais de plataformas convencionais. Todavia, a validação permanece em ambiente controlado, sem testes com engenheiros de software em ambientes reais. A interoperabilidade é limitada: o protótipo é uma aplicação *desktop* independente, sem integração com ferramentas existentes. Os próprios autores reconhecem esta limitação e sugerem, como trabalho futuro, o desenvolvimento de ferramentas de ingestão de dados e *plugins* capazes de capturar artefatos gerados pelas diversas ferramentas utilizadas ao longo do ciclo de vida do desenvolvimento de software. Uma crítica pertinente é que a plataforma NDL ArcaNet é uma solução proprietária desenvolvida pelos coautores do artigo, o que pode limitar a reprodutibilidade e a adoção independente dos resultados.

Por fim, o BC4RT é o protótipo mais completo e funcional entre os analisados, inovando ao adotar uma *blockchain* neural de terceira geração com granularidade refinada de rastreabilidade. Não obstante, a validação restrita a ambiente controlado, a ausência de integração com ferramentas de mercado e o caráter proprietário da plataforma NDL ArcaNet, desenvolvida pelos próprios coautores, comprometem a reprodutibilidade dos resultados e a viabilidade de adoção independente pela indústria.

# 5 RESULTADOS

Esta seção apresenta os resultados da revisão bibliográfica crítica, organizando de forma comparativa as evidências encontradas nos nove estudos analisados. A partir da avaliação das arquiteturas propostas, dos desafios técnicos abordados e do nível de maturidade das soluções, são sintetizadas as principais tendências de uso da *blockchain* na rastreabilidade de requisitos, bem como as lacunas recorrentes que ainda limitam sua adoção prática.

## 5.1 Síntese Comparativa dos Estudos

A Tabela 1 apresenta uma visão consolidada dos nove artigos analisados, permitindo a comparação direta entre as soluções propostas quanto às suas características arquiteturais, estratégias de armazenamento, mecanismos de consenso, nível de validação e grau de interoperabilidade com ferramentas existentes.

Tabela 1 – Comparação das características dos estudos analisados

| **Estudo** | **Plataforma** | **Tipo de Rede** | **Smart Contracts** | **Armazenamento** | **Consenso** | **Validação** |
|---|---|---|---|---|---|---|
| Farooq *et al.* (2022b) | Ethereum | Privada | Sim | Híbrido (IPFS) | PoW | Protótipo |
| Ghadmi *et al.* (2022) | Ethereum | Pública | Sim | *Off-chain* | Nativo Ethereum | PoC |
| Demi *et al.* (2023) | N/A | N/A | N/A | N/A | N/A | Qualitativa (GT) |
| Elapolu *et al.* (2024) | Customizada | Consórcio (implícita) | Não | *On-chain* + Neo4J | PoW | Cenário controlado |
| Qureshi e Farooq (2024) | Ethereum | Privada | Sim | Híbrido (IPFS) | PoW | Conceitual |
| Qureshi *et al.* (2024) | Ethereum | Privada | Sim | Híbrido (IPFS) | PoW | Protótipo |
| Demi *et al.* (2024) | NDL ArcaNet | Permissionada | Não | *On-chain* | Certificadores | Qualitativa |
| Demi *et al.* (2021) | Não especif. | Não especif. | Sim | Híbrido (IPFS) | Não especif. | Conceitual |
| Demi *et al.* (2022) | NDL ArcaNet | Permissionada | Não | *On-chain* | Certificadores | Protótipo |

Fonte: Elaborado pelo autor.
Nota: PoW = *Proof-of-Work*; PoC = Prova de Conceito; GT = *Grounded Theory*; N/A = Não aplicável.

A análise da Tabela 1 revela que a plataforma Ethereum é predominante, sendo adotada em quatro dos nove estudos. Os estudos de Demi *et al.* (2024) e Demi *et al.* (2022) se distinguem ao utilizar a plataforma NDL ArcaNet, uma *blockchain* neural de terceira geração. O armazenamento híbrido com IPFS é a estratégia mais recorrente, adotada em quatro estudos, enquanto dois estudos optam por armazenamento integralmente *on-chain* na plataforma NDL

ArcaNet. Quanto ao nível de validação, observa-se que apenas três estudos apresentam protótipos funcionais, enquanto três permanecem no nível conceitual, evidenciando a imaturidade geral das soluções propostas.

## 5.2 Principais Abordagens, Soluções e Tendências

A partir da análise crítica dos nove estudos, é possível identificar as principais abordagens arquiteturais, soluções técnicas e tendências emergentes na integração da tecnologia *blockchain* com a rastreabilidade de requisitos de software.

### *5.2.1 Abordagens arquiteturais predominantes*

As soluções analisadas podem ser categorizadas em três abordagens arquiteturais distintas. A primeira, e mais frequente, é a abordagem baseada em *smart contracts* sobre a plataforma Ethereum, representada pelos trabalhos de Farooq *et al.* (2022b), Ghadmi *et al.* (2022), Qureshi e Farooq (2024), Qureshi *et al.* (2024) e Demi *et al.* (2021). Nesses estudos, os *smart contracts* automatizam regras de negócio como validação de requisitos, testes de aceitação, distribuição de pagamentos e aplicação de penalidades contratuais. A segunda abordagem utiliza *blockchains* neurais de terceira geração, especificamente a plataforma NDL ArcaNet, conforme proposto por Demi *et al.* (2024) e Demi *et al.* (2022), que modelam requisitos como *tokens* digitais únicos com *ledgers* individuais. A terceira abordagem, representada por Elapolu *et al.* (2024), emprega uma *blockchain* customizada integrada a um banco de dados de grafos (Neo4J) para consultas de rastreabilidade via linguagem Cypher.

### *5.2.2 Estratégias de armazenamento*

Uma tendência clara na literatura é a adoção de estratégias de armazenamento híbrido para mitigar os custos elevados de armazenamento *on-chain*. Conforme estimado por Ghadmi *et al.* (2022), o custo de armazenamento na *blockchain* Ethereum é de aproximadamente US$ 76.000 por gigabyte, o que torna inviável o armazenamento integral de dados volumosos. A solução predominante consiste em manter metadados e *hashes* de verificação na *blockchain*, enquanto os dados completos dos requisitos, comunicações e documentos são armazenados *off-chain* via IPFS (*InterPlanetary File System*). Essa abordagem é adotada por Farooq *et al.* (2022b), Qureshi e Farooq (2024), Qureshi *et al.* (2024) e Demi *et al.* (2021). Em contrapartida, os

trabalhos baseados na plataforma NDL ArcaNet (Demi *et al.*, 2024; Demi *et al.*, 2022) realizam armazenamento integralmente *on-chain*, uma vez que essa plataforma suporta nativamente arquivos de qualquer tamanho e tipo.

### *5.2.3 Granularidade da rastreabilidade*

A granularidade da rastreabilidade varia significativamente entre os estudos. Os *frameworks* voltados para metodologias ágeis, como AgilePlus (Farooq *et al.*, 2022b), ChainAgile (Qureshi; Farooq, 2024) e ChainAgilePlus (Qureshi *et al.*, 2024), operam no nível de *user stories*, rastreando o progresso ao longo das fases do Scrum. Em contraste, Elapolu *et al.* (2024) propõe uma rastreabilidade em nível dual, tanto de artefatos (documentos) quanto de objetos (requisitos individuais), enquanto o protótipo BC4RT (Demi *et al.*, 2022) alcança a granularidade mais refinada, vinculando cada requisito diretamente a artefatos de código-fonte e casos de teste com atributos detalhados (versão, descrição, status, contribuidor, *timestamp*).

### *5.2.4 Mecanismos de incentivo e governança*

Uma tendência inovadora identificada é a proposta de mecanismos de incentivo para estimular a manutenção da rastreabilidade. Demi *et al.* (2021) propõem políticas de recompensas codificadas em *smart contracts* que distribuem *tokens* digitalizados para *stakeholders* que criam *trace links* de qualidade, abordando o problema motivacional frequentemente ignorado pela literatura. Além disso, Demi *et al.* (2021) introduzem um mecanismo de votação para validação de *trace links*, no qual aprovadores registram votos no *ledger* distribuído e o contrato inteligente calcula os votos cumulativos para aceitar ou rejeitar os links. Esses mecanismos de governança descentralizada representam uma contribuição relevante para a sustentabilidade das práticas de rastreabilidade.

### *5.2.5 Evolução incremental de frameworks*

Uma tendência observável é a evolução incremental de soluções dentro de grupos de pesquisa específicos. Os trabalhos de Farooq *et al.* (2022b) (Demi *et al.*, 2021; Demi *et al.*, 2022; Demi *et al.*, 2023; Demi *et al.*, 2024) apresentam uma trajetória que parte de um *framework* conceitual, avança para a implementação de um protótipo (BC4RT) e culmina em validações qualitativas com especialistas. Essa tendência sugere que a área está em um estágio

de amadurecimento progressivo, mas ainda não consolidado.

## 5.3 Discussão das Lacunas

A análise crítica dos nove estudos revela lacunas significativas que limitam a maturidade e a aplicabilidade prática das soluções propostas. As lacunas identificadas são organizadas em seis eixos temáticos, conforme discutido a seguir.

### *5.3.1 Ausência de validação em ambientes de produção*

A lacuna mais recorrente e crítica é a ausência de validação em projetos práticos. Nenhum dos nove estudos analisados apresenta resultados de aplicação em ambientes de produção com equipes atuantes no mercado. Os protótipos de Farooq *et al.* (2022b) e Demi *et al.* (2022) foram avaliados em ambientes de laboratório com cenários controlados. A Prova de Conceito de Ghadmi *et al.* (2022) utiliza Ganache para simulação da *blockchain*. Os estudos qualitativos de Demi *et al.* (2023) e Demi *et al.* (2024) coletam percepções de especialistas, mas não validam as soluções na prática. Essa ausência compromete a avaliação da viabilidade prática, da escalabilidade efetiva e da aceitação pelos engenheiros de software.

### *5.3.2 Falta de interoperabilidade com ferramentas existentes*

Nenhum dos estudos analisados implementa integração efetiva com ferramentas de gestão de requisitos e desenvolvimento já consolidadas no mercado, tais como Jira, Trello, GitHub, GitLab ou IBM DOORS. Os *frameworks* propõem interfaces próprias que exigiriam que equipes abandonassem completamente suas ferramentas habituais, o que representa uma barreira significativa à adoção prática. Embora Demi *et al.* (2024) identifiquem, a partir das recomendações dos especialistas, a necessidade de integração leve com ferramentas comerciais existentes como recomendação central para viabilizar a adoção, essa integração permanece como trabalho futuro. De forma similar, Demi *et al.* (2021) mencionam conceitualmente o uso de ferramentas de ingestão de dados e *plugins*, mas sem implementação ou detalhamento.

### *5.3.3 Análise insuficiente de escalabilidade e custos*

Embora a escalabilidade seja reconhecida como um desafio por diversos autores, as análises quantitativas permanecem superficiais. Nenhum estudo apresenta uma avaliação

completa que inclua simultaneamente: (i) custos de *gas* por transação; (ii) latência de operações em cenários realistas com centenas ou milhares de requisitos; (iii) crescimento do tamanho da cadeia ao longo do tempo; e (iv) impacto no desempenho com múltiplas equipes concorrentes. O cenário demonstrado por Elapolu *et al.* (2024), com apenas 17 blocos, está muito aquém da complexidade de projetos reais de engenharia de sistemas. Os estudos que utilizam *Proof-of-Work* em redes privadas (Farooq *et al.*, 2022b; Qureshi; Farooq, 2024; Qureshi *et al.*, 2024) não justificam adequadamente essa escolha, uma vez que mecanismos como *PBFT* ou *Proof-of-Authority* seriam mais eficientes para redes permissionadas com participantes conhecidos.

### *5.3.4 Tratamento superficial da privacidade*

A privacidade de dados sensíveis contidos nos requisitos de software, como segredos industriais, regras de negócio proprietárias e informações estratégicas, recebe tratamento insuficiente na maioria dos estudos. As soluções se limitam a mecanismos básicos de autenticação por chaves privadas (Farooq *et al.*, 2022b; Qureshi *et al.*, 2024) ou à separação entre dados públicos e privados (Ghadmi *et al.*, 2022). Apenas os trabalhos baseados na plataforma NDL ArcaNet (Demi *et al.*, 2024; Demi *et al.*, 2022) apresentam mecanismos de criptografia mais robustos (AES-256, RSA), porém sem controle de acesso granular para diferentes tipos de requisitos com níveis distintos de confidencialidade. Nenhum estudo aborda técnicas avançadas de preservação de privacidade, como provas de conhecimento zero aplicadas ao conteúdo dos requisitos, criptografia homomórfica ou computação multipartidária segura.

### *5.3.5 Resistência organizacional e curva de aprendizado*

Os estudos predominantemente técnicos ignoram as barreiras organizacionais e humanas à adoção da *blockchain* na rastreabilidade de requisitos. Apenas Demi *et al.* (2023) e Demi *et al.* (2024), por meio de suas abordagens qualitativas, capturam a resistência dos engenheiros de software à adoção de novas ferramentas. Conforme destacado pelos especialistas entrevistados, profissionais de software são frequentemente conservadores e relutantes em adicionar etapas ao seu fluxo de trabalho. A preocupação de que tempos de validação elevados de blocos seriam problemáticos em ambientes ágeis é um exemplo concreto dessa resistência. A ausência de discussão sobre curva de aprendizado, treinamento necessário e gestão de mudança organizacional nos demais estudos representa uma lacuna importante para a viabilidade prática das soluções propostas.

### *5.3.6 Ausência de comparação com soluções tradicionais*

Nenhum dos estudos analisados realiza uma comparação rigorosa entre as soluções baseadas em *blockchain* e as ferramentas tradicionais de rastreabilidade de requisitos, como matrizes de rastreabilidade, IBM DOORS ou sistemas de controle de versão como Git. Essa ausência dificulta a avaliação do valor agregado real da *blockchain* nesse contexto. Conforme observado na análise de Ghadmi *et al.* (2022), o problema de validação de requisitos entre duas partes conhecidas poderia ser resolvido de forma mais simples e econômica com sistemas tradicionais de assinatura digital combinados com controle de versão. Sem essa comparação, permanece a questão fundamental de se a complexidade e a sobrecarga introduzidas pela *blockchain* são proporcionais aos benefícios obtidos em relação a soluções já estabelecidas.

## 5.4 Considerações Finais da Seção

A síntese dos resultados evidencia que a aplicação da tecnologia *blockchain* na rastreabilidade de requisitos de software é um campo de pesquisa em estágio inicial de amadurecimento. As soluções propostas demonstram potencial para endereçar desafios de imutabilidade, transparência e auditabilidade, porém permanecem predominantemente no nível teórico-conceitual ou de protótipos em ambientes controlados. As lacunas identificadas, em especial a ausência de validação industrial, a falta de interoperabilidade com ferramentas existentes e a análise insuficiente de escalabilidade e custos, constituem barreiras significativas que precisam ser superadas para que essas soluções alcancem maturidade e adoção prática na indústria de desenvolvimento de software.

# 6 CONCLUSÕES

Este trabalho teve como objetivo geral identificar e analisar soluções recentes que empregam a tecnologia *blockchain* na rastreabilidade de requisitos de software. Para tanto, foi conduzida uma revisão bibliográfica de caráter crítico-analítico, de abordagem qualitativa, com análise crítica de nove estudos primários selecionados a partir das bases IEEE Xplore, Scopus e Google Scholar, conforme o rigor metodológico estabelecido no Capítulo 3. Os resultados obtidos permitem responder às questões norteadoras definidas e avaliar o cumprimento dos objetivos específicos propostos.

## 6.1 Respostas às Questões Norteadoras

Em relação à primeira questão norteadora: *quais são os mecanismos e arquiteturas propostos na literatura para integrar a tecnologia blockchain na rastreabilidade de requisitos de software?*, a análise revelou três abordagens arquiteturais predominantes. A primeira, e mais frequente, baseia-se em *smart contracts* sobre a plataforma Ethereum, na qual regras de negócio como validação de requisitos, testes de aceitação e penalidades contratuais são automatizadas por contratos inteligentes escritos em Solidity. A segunda abordagem utiliza *blockchains* neurais de terceira geração, especificamente a plataforma NDL ArcaNet, que modela requisitos como *tokens* digitais únicos com *ledgers* individuais para auditoria de ciclo de vida. A terceira emprega uma *blockchain* customizada integrada a bancos de dados de grafos (Neo4J) para consultas de rastreabilidade. Quanto ao armazenamento, a estratégia predominante é híbrida: metadados e *hashes* são registrados *on-chain*, enquanto dados volumosos são mantidos *off-chain* via IPFS, mitigando os custos elevados de armazenamento em plataformas como Ethereum. A granularidade da rastreabilidade varia desde o nível de *user stories* nos *frameworks* ágeis até o nível de requisitos individuais com vínculos diretos a código-fonte e casos de teste nos protótipos mais refinados.

Em relação à segunda questão norteadora: *como essas soluções arquiteturais lidam com os desafios técnicos inerentes à tecnologia, especialmente escalabilidade e privacidade?*, os resultados indicam que o tratamento desses desafios permanece insuficiente na literatura analisada. A escalabilidade é abordada predominantemente pela estratégia de armazenamento *off-chain*, porém nenhum estudo apresenta análises quantitativas completas que incluam custos de *gas*, latência de operações e crescimento da cadeia em cenários realistas com centenas ou

milhares de requisitos. Estudos que adotam *Proof-of-Work* em redes privadas não justificam adequadamente essa escolha frente a mecanismos mais eficientes para redes permissionadas. A privacidade, por sua vez, é tratada por mecanismos básicos de autenticação por chaves privadas ou separação entre dados públicos e privados, sendo que apenas os trabalhos baseados na plataforma NDL ArcaNet apresentam criptografia mais robusta (AES-256, RSA). Técnicas avançadas de preservação de privacidade, como provas de conhecimento zero ou criptografia homomórfica, não são exploradas por nenhum dos estudos.

## 6.2 Cumprimento dos Objetivos Específicos

Quanto aos objetivos específicos, o primeiro: *apresentar uma revisão bibliográfica das soluções que utilizam a tecnologia blockchain na rastreabilidade de requisitos de software*, foi alcançado por meio da análise detalhada de nove artigos publicados entre 2021 e 2025, abrangendo propostas de *frameworks* conceituais, protótipos funcionais e estudos qualitativos com especialistas. O segundo objetivo: *identificar as abordagens de como a blockchain é utilizada na rastreabilidade de requisitos de software*, foi cumprido com a categorização das três abordagens arquiteturais predominantes, das estratégias de armazenamento e dos mecanismos de governança e incentivo identificados. O terceiro objetivo: *identificar possíveis lacunas no uso da blockchain na rastreabilidade de requisitos de software*, foi atendido pela discussão de seis lacunas críticas: a ausência de validação em ambientes de produção, a falta de interoperabilidade com ferramentas consolidadas do mercado, a análise insuficiente de escalabilidade e custos, o tratamento superficial da privacidade, a resistência organizacional à adoção e a ausência de comparação com soluções tradicionais de rastreabilidade.

## 6.3 Considerações Finais

A partir da revisão bibliográfica de caráter crítico-analítico conduzida neste trabalho, conclui-se que a aplicação da tecnologia *blockchain* na rastreabilidade de requisitos de software ainda se encontra em estágio inicial de maturidade. As soluções identificadas concentram-se em *frameworks* conceituais e protótipos avaliados em ambientes controlados, com evidências empíricas limitadas sobre sua efetividade em contextos industriais. As soluções propostas demonstram potencial para endereçar desafios de imutabilidade, transparência e auditabilidade no registro de requisitos, porém nenhum dos nove estudos analisados apresenta validação

em projetos comerciais, e a interoperabilidade com ferramentas já estabelecidas, como Jira, GitHub ou IBM DOORS, permanece como uma lacuna não resolvida. Adicionalmente, a complexidade introduzida pela *blockchain* levanta a questão fundamental, ainda não respondida pela literatura, de se os benefícios obtidos são proporcionais à sobrecarga técnica e organizacional em comparação com soluções tradicionais de rastreabilidade.

## 6.4 Limitações do Estudo

Este trabalho apresenta limitações que devem ser consideradas na interpretação dos resultados. Por se tratar de uma revisão bibliográfica de caráter crítico-analítico, e não de uma revisão sistemática com protocolo reprodutível, a seleção dos estudos pode conter vieses inerentes ao julgamento do pesquisador. A amostra de nove artigos, embora representativa do estado da arte no período de 2021 a 2025, não esgota a totalidade da produção científica sobre o tema. Além disso, a restrição a publicações em inglês e português pode ter excluído contribuições relevantes em outros idiomas.

## 6.5 Sugestões para Trabalhos Futuros

Com base nas lacunas identificadas, sugere-se como direções para trabalhos futuros:

- **Desenvolver integrações com ferramentas consolidadas do mercado:** elaborar conectores, plugins ou interfaces de aplicação (APIs) que permitam a troca de dados entre sistemas baseados em *blockchain* e ferramentas amplamente adotadas, como Jira, GitHub/GitLab, IBM DOORS e Azure DevOps, reduzindo a barreira de adoção e evitando a necessidade de substituição de fluxos de trabalho já estabelecidos.
- **Validar as propostas em ambientes reais:** implementar protótipos funcionais e avaliar seu desempenho, usabilidade e custo-benefício em projetos de desenvolvimento de software efetivos, sejam em empresas do setor, instituições públicas ou projetos de pesquisa, inclusive no contexto do estado do Ceará, gerando evidências empíricas que hoje são escassas na literatura.
- **Aprofundar análises técnicas e econômicas:** investigar mecanismos de consenso mais adequados para redes permissionadas, como *Proof-of-Authority* ou *PBFT*, em substituição a modelos ineficientes como o *Proof-of-Work*; realizar estudos quantitativos comparando custos, escalabilidade e desempenho entre soluções tradicionais e baseadas em *blockchain*;

e desenvolver estratégias avançadas de privacidade e controle de acesso granular para dados sensíveis.

- **Explorar combinações tecnológicas:** avaliar a integração da *blockchain* com outras tecnologias emergentes, como inteligência artificial e bancos de dados de grafos, para automatizar a criação e verificação de links de rastreabilidade, além de ampliar o escopo do rastreamento para incluir a justificativa de alterações e a origem detalhada de cada requisito.
- **Analisar fatores humanos e organizacionais:** elaborar diretrizes e roteiros de adoção que considerem a resistência à mudança, a curva de aprendizado e a governança necessária para implementar essa tecnologia em equipes de desenvolvimento.

# REFERÊNCIAS

ANTONOPOULOS, A. M.; HARDING, D. A. **Mastering Bitcoin**: Programming the open blockchain. [S.l.]: "O'Reilly Media, Inc.", 2023.

BUTERIN, V. Whitepaper, **Ethereum**: A next-generation smart contract and decentralized application platform. 2014. Whitepaper. Disponível em: https://ethereum.org/whitepaper/. Acesso em: 29 jul. 2026.

DEMI, S.; COLOMO-PALACIOS, R.; SÁNCHEZ-GORDÓN, M.; VELASCO, C.; CANO, R. A neural blockchain for requirements traceability: Bc4rt prototype. In: **Systems, Software and Services Process Improvement:** 29th European Conference, EuroSPI 2022, Salzburg, Austria, August 31–September 2, 2022, Proceedings. [S.l.]: Springer, 2022. (Communications in Computer and Information Science), p. 45–59.

DEMI, S.; SÁNCHEZ-GORDÓN, M.; COLOMO-PALACIOS, R. A blockchain-enabled framework for requirements traceability. In: **Systems, Software and Services Process Improvement:** 28th European Conference, EuroSPI 2021, Krems, Austria, September 1–3, 2021, Proceedings. [S.l.]: Springer, 2021. (Communications in Computer and Information Science), p. 3–13.

DEMI, S.; SÁNCHEZ-GORDÓN, M.; KRISTIANSEN, M. Blockchain for requirements traceability: A qualitative approach. **Journal of Software**: Evolution and Process, Wiley Online Library, v. 35, n. 9, p. e2493, 2023.

DEMI, S.; SÁNCHEZ-GORDÓN, M.; KRISTIANSEN, M.; LARRUCEA, X. Trustworthy and collaborative traceability management: Experts' feedback on a blockchain-enabled framework. **Journal of Software**: Evolution and Process, Wiley Online Library, v. 36, n. 11, p. e2707, 2024.

DIOMEDESSE, R. A. P.; CUNHA, L. d. S. B. Proposta de rastreabilidade de requisitos em equipes ágeis distribuídas. **Revista Foco**, v. 16, n. 5, p. e2095–e2095, 2023.

DZHALILA, D.; SIAHAAN, D.; FAUZAN, R.; ASYROFI, R.; KARIMI, M. I. A systematic literature review on blockchain technology in software engineering. **Jurnal Eltikom:** Jurnal Teknik Elektro, Teknologi Informasi Dan Komputer, v. 7, n. 1, p. 38–49, 2023.

ELAPOLU, M. S.; RAI, R.; GORSICH, D. J.; RIZZO, D.; RAPP, S.; CASTANIER, M. P. Blockchain technology for requirement traceability in systems engineering. **Information Systems**, v. 123, p. 102384, 2024. ISSN 0306-4379. Disponível em: https://www.sciencedirect.com/science/article/pii/S0306437924000425. Acesso em: 29 jul. 2026.

FAROOQ, M. S.; AHMED, M.; EMRAN, M. A survey on blockchain acquainted software requirements engineering: model, opportunities, challenges, and future directions. **IEEE Access**, IEEE, v. 10, p. 48193–48228, 2022.

FAROOQ, M. S.; KALIM, Z.; QURESHI, J. N.; RASHEED, S.; ABID, A. A blockchain-based framework for distributed agile software development. **IEEE Access**, IEEE, v. 10, p. 17977–17995, 2022.

GHADMI, B. A. A. A.; ABDULKADER, O. A.; ALWARHI, A. A. Blockchain based software engineering requirements analysis and management: Icict 2022, london, volume

1. In: SPRINGER. **Proceedings of Seventh International Congress on Information and Communication Technology**. [S.l.], 2022. p. 75–83.

GOTEL, O. C.; FINKELSTEIN, C. An analysis of the requirements traceability problem. In: IEEE. **Proceedings of ieee international conference on requirements engineering**. [S.l.], 1994. p. 94–101.

KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering**: Processes and techniques. 1st. ed. [S.l.]: Wiley Publishing, 1998. ISBN 0471972088.

MALCHER, P. R. C. **Catálogo de abordagens de apoio à rastreabilidade de requisitos baseado em uma revisão sistemática da literatura**. Dissertação (Dissertação de Mestrado) — UNIVERSIDADE FEDERAL DO PARÁ, Belém, 2015. Disponível em: https://repositorio.ufra.edu.br/jspui/handle/123456789/344. Acesso em: 10 jun. 2025.

NAKAMOTO, S. **Bitcoin**: A peer-to-peer electronic cash system. 2008. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 29 jul. 2026.

PRESSMAN, R. S.; MAXIM, B. R. **Software engineering**: a practitioner's approach. New York, NY: Mcgraw-Hill Education, 2020.

QURESHI, J. N.; FAROOQ, M. S. Chainagile a framework for the improvement of scrum agile distributed software development based on blockchain. **Plos one**, Public Library of Science San Francisco, CA USA, v. 19, n. 3, p. e0299324, 2024.

QURESHI, J. N.; FAROOQ, M. S.; KHELIFI, A.; ATAL, Z. Harnessing the potential of blockchain in chainagileplus framework for the improvement of distributed scrum of scrums agile software development. **IEEE Access**, IEEE, 2024.

ROTHER, E. T. Revisão sistemática x revisão narrativa. **Acta Paulista de Enfermagem**, Escola Paulista de Enfermagem, Universidade Federal de São Paulo, v. 20, n. 2, p. v–vi, Apr 2007. ISSN 0103-2100. Disponível em: https://doi.org/10.1590/S0103-21002007000200001. Acesso em: 29 jul. 2026.

SAYÃO, M.; LEITE, J. C. S. do P. Rastreabilidade de requisitos. **Revista de Informática Teórica e Aplicada (RITA)**, v. 13, n. 1, p. 57–86, 2006.

SOMMERVILLE, I. **Software Engineering, 9/E**. [S.l.]: Pearson Education India, 2011.

SOMMERVILLE, I.; SAWYER, P. Viewpoints: principles, problems and a practical approach to requirements engineering. **Annals of software engineering**, Springer, v. 3, n. 1, p. 101–130, 1997.

SWAN, M. **Blockchain**: Blueprint for a new economy. [S.l.]: "O'Reilly Media, Inc.", 2015.

TAPSCOTT, D.; TAPSCOTT, A. **Blockchain revolution**: how the technology behind bitcoin is changing money, business, and the world. [S.l.]: Penguin, 2016.

TSCHORSCH, F.; SCHEUERMANN, B. Bitcoin and beyond: A technical survey on decentralized digital currencies. **IEEE Communications Surveys & Tutorials**, IEEE, v. 18, n. 3, p. 2084–2123, 2016.

ZHENG, Z.; XIE, S.; DAI, H.; CHEN, X.; WANG, H. An overview of blockchain technology: Architecture, consensus, and future trends. In: **2017 IEEE International Congress on Big Data (BigData Congress)**. [S.l.: s.n.], 2017. p. 557–564.