INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DESENVOLVIMENTO DE UMA AUTORIDADE CERTIFICADORA BASEADA EM BLOCKCHAIN NO HYPERLEDGER FABRIC

ALANA CRISTINA FELIX DA SILVA

Cajazeiras
2025

ALANA CRISTINA FELIX DA SILVA

DESENVOLVIMENTO DE UMA AUTORIDADE CERTIFICADORA BASEADA EM BLOCKCHAIN NO HYPERLEDGER FABRIC

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dsc. Francisco Daladier Marques Júnior.

Cajazeiras

2025

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

S586d Silva, Alana Cristina Felix da.

Desenvolvimento de uma autoridade certificadora baseada em blockchain no hyperledger fabric / Alana Cristina Felix da Silva. – Cajazeiras, 2025.
41f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas.) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Dr. Francisco Daladier Marques Júnior.

1. Desenvolvimento de sistemas. 2. Certificado digital. 3. Autoridade certificadora. 4. Blockchain. I. Instituto Federal da Paraíba. II. Título.

IFPB/CZ CDU: 004.4(043.2)

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

ALANA CRISTINA FELIX DA SILVA

DESENVOLVIMENTO DE UMA AUTORIDADE CERTIFICADORA BASEADA EM BLOCKCHAIN NO HYPERLEDGER FABRIC

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dr. Francisco Daladier Marques Júnior

Aprovada em: **20 de Março de 2025.**

Prof. Dr. Francisco Daladier Marques Júnior - Orientador

Prof. Me. Fábio Abrantes Diniz - Avaliador

IFPB - Campus Cajazeiras

Prof. Esp. João Igor Barros Rocha - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Daladier Marques Junior**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 25/03/2025 09:15:03.
- **Joao Igor Barros Rocha**, **PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO**, em 25/03/2025 09:40:18.
- **Fabio Abrantes Diniz**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 25/03/2025 11:10:41.

Este documento foi emitido pelo SUAP em 21/03/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código 685765
Verificador: 5f5940f87f
Código de Autenticação:

Dedico este trabalho aos meus pais com muito orgulho, família, professores e amigos.

AGRADECIMENTOS

O desenvolvimento deste projeto de curso incluiu muitas pessoas importantes para mim, nas quais de inicio agradeço verdadeiramente.

Aos meus pais Alcirene e Luis Junior, que sempre apoiaram nos meus sonhos e na loucura de sair de uma cidade pequena no Piauí e tentar meus estudos em outro estado sem conhecer nada e nem ninguém, apenas me incentivaram a ir em busca do propósito que tanto busco em mim mesma sem perder o ânimo em momentos que pensei em largar tudo e só voltar para casa.

Agradeço e dedico a minha tia Catarina Acelina "Díinha"(*In memoriam*) que infelizmente perdir quase na finalização do meu curso, mas esteve sempre comigo me apoiando e sentindo um orgulho que me deixava ainda mais confiante - Ao meu querido amigo que esteve também comigo feliz com minhas pequenas conquistas e perdir muito cedo Maylon Winicius (*In memoriam*).

Ofereço também a todos os meus familiares, principalmente os que me apoiaram tanto na escolha de sair da minha cidade quanto os que sempre estavam ali quando precisei, me mantendo firme e forte, em especial ao meu adorado tio Neimario que foi um dos grandes suportes nessa minha caminhada. Mas enfim, tias, primas, primos, entre outros familiares que me incentivaram ao longo do meu processo construtivo.

Aos meus amigos que encontrei ao longo da minha caminhada pelo curso pois me ajudaram bastante - em particular a Jônathas Batista sem ele nao teria me erguido em outra cidade, e vários outros que foram de grande apoio neste trabalho Franks Patrício e Álvaro Renan, citaria sem sombra de dúvidas.

Ao meu orientador Francisco Daladier por ter uma paciência única, além de ter me ajudado e apoiado em qualquer dúvida ou confusão que tive e esteve sempre disposto a responder minhas mensagens independente se estivesse ocupado, foi de grande ajuda todo o conhecido alcançado.

”A primeira regra de qualquer tecnologia utilizada nos negócios é que a automoção aplicada a uma operação eficiente aumenta a eficiência. A segunda é que a automação aplicada a uma operação ineficiente aumenta a ineficiência. ”

Bill Gates - A estrada do futuro

RESUMO

A variedade de aplicações, que incluem desde contratos inteligentes até credenciais acadêmicas criptografadas por *Blockchain*, evidenciam a versatilidade e profundidade dessa tecnologia na atualidade, redefinindo modelos tradicionais de segurança e competência. Para compreender essa revolução, é crucial entender o que é um *Blockchain* e sua capacidade transformadora. Este trabalho propõe a adoção de uma rede *Blockchain*, utilizando a plataforma *Hyperledger Fabric*, para o gerenciamento de ativos organizacionais por meio da emissão de certificados digitais de uma Autoridade Certificadora (AC). As ACs desempenham um papel fundamental na autenticação de identidades digitais, mas enfrentam desafios como centralização, vulnerabilidades a ataques cibernéticos e dificuldades na verificação e revogação de certificados. No contexto educacional, o *Hyperledger Fabric*, reconhecido por sua eficácia em ambientes empresariais, será implementado como o *framework* central, propondo segurança, descentralização e transparência no processo de certificação digital. A tecnologia *Blockchain* permitirá um ambiente confiável para armazenar e autenticar certificados digitais.

Palavras-chave: Blockchain. Hyperledger Fabric. Certificados Digitais. Autoridade Certificadora.

ABSTRACT

The variety of applications, ranging from smart contracts to blockchain-encrypted academic credentials, highlights the versatility and depth of this technology today, redefining traditional models of security and competence. To understand this revolution, it is crucial to comprehend what Blockchain is and its transformative potential. This work proposes the adoption of a Blockchain network, using the Hyperledger Fabric platform, for the management of organizational assets through the issuance of digital certificates by a Certification Authority (CA). CAs play a fundamental role in authenticating digital identities but face challenges such as centralization, vulnerabilities to cyberattacks, and difficulties in certificate verification and revocation. In the educational context, Hyperledger Fabric, recognized for its effectiveness in business environments, will be implemented as the central framework, offering security, decentralization, and transparency in the digital certification process. Blockchain technology will provide a reliable environment for storing and authenticating digital certificates.

Keywords: Blockchain. Hyperledger Fabric. Digital Certificates. Certificate Authority

LISTA DE FIGURAS

Figura 1 – Figura adaptada da 101blockchain e da PwC 18
Figura 2 – Arquitetura de proposta 30
Figura 3 – Imagem da interface dos Certificados 31
Figura 4 – Tela de Revogação de Certificação 32
Figura 5 – Tela do Administrador 32

LISTA DE TABELAS

Tabela 1 – Comparação entre *Hyperledger e Ethereum* 21
Tabela 2 – Requisitos Funcionais 27
Tabela 3 – Requisitos Não Funcionais 28

LISTA DE CÓDIGOS

Algoritimo 1 – Função do código de *RegisterUser* 38

Algoritimo 2 – Função do código *verifySignature* 39

Algoritimo 3 – Função do código *verifyCertificate* 40

Algoritimo 4 – Função do código *sign* 41

Algoritimo 5 – Função do código *revokeCertificate* 42

LISTA DE ABREVIATURAS E SIGLAS

AC	Autoridade Certificadora
API	*Application Programming Interface*
CRL	Lista de Revogação de Certificados
DApps	Aplicativos Descentralizados
DeFi	Finanças Descentralizadas
IBM	*International Business Machines Corporation*
ICP	Infraestrutura de Chave Pública
JSON	JavaScript Object Notation
MSP	*Membership Service Provider*
NFT	Token não fungível
PBFT	*Practical Byzantine Fault Tolerance*
PKI	*Public Key Infrastructure*
PoA	*Proof of Authority*
PoS	*Proof of Stake*
PoW	*Proof of Work*
REST	*Representational State Transfer*
RSA	*Rivest-Shamir-Adleman*
SHA	*Secure Hash Algorithm*

SUMÁRIO

1 INTRODUÇÃO 15

1.1 JUSTIFICATIVA 16

1.2 OBJETIVOS 17

1.2.1 Objetivo Geral 17

1.2.2 Objetivos Específicos 17

1.3 METODOLOGIA 17

2 FUNDAMENTAÇÃO TEÓRICA 18

2.1 Autoridade Certificadora 19

2.2 Ethereum 20

2.3 O FRAMEWORK HYPERLEDGER FABRIC 22

2.3.1 Mecanismos de Consenso aplicados ao Blockchain 22

2.3.2 Processo de Desenvolvimento 23

3 METODOLOGIA DE IMPLEMENTAÇÃO 25

3.1 Ferramentas 25

3.2 Dependências 25

3.3 Levantamento de requisitos 26

3.4 Implementação da Autoridade Certificadora 28

3.4.1 Assinatura Digital 29

3.4.2 Revogação dos certificados 29

3.4.3 Telas do Sistema 31

4 CONSIDERAÇÕES FINAIS 33

4.1 Resultados 33

4.2 Conclusão e trabalhos futuros 33

REFERÊNCIAS 35

APÊNDICE A – EXEMPLO DE CÓDIGOS 37

1 INTRODUÇÃO

Na frente de uma revolução tecnológica surgem formas de como pode-se manter a proteção e a privacidade dos nossos dados, pois, com o emergir de uma vasta quantidade de desafios no mundo atual fica cada vez mais difícil lidar com tanta informação. Em simultâneo a isso, os avanços científicos da tão comentada quarta revolução industrial, surge uma tecnologia resultante, o *Blockchain*, que dispõe a ser revolucionária se for usada de forma correta, seja empresarial, ou até mesmo para educação, já que uma das suas principais características é a capacidade de garantir a integridade dos registros e a imutabilidade dos dados. (Rocha et at.,2018).

Logo, o Blockchain é um livro-razão de transação imutável, mantido dentro de uma rede distribuída de nós pares, ou seja, é um sistema de registro de dados onde são demonstrados em blocos encadeados sequencialmente, possuindo a segurança para as informações registradas que não podem ser alteradas ou excluídas sem o consenso da rede. Dessa forma, trata-se de uma tecnologia de armazenamento em nuvem conhecida como *peer-to-peer*. Antes de 2008, a tecnologia Blockchain teria despertado interesse da indústria global, não apenas no setor financeiro, mas também em outras áreas, como: informática, inteligência artificial, fabricantes de semicondutores, telecomunicações, e-commerce e segurança de identidades pessoais (Lessak et al., 2018).

Diante disso, em 2015 surgiu uma das várias plataformas que permite uso de Blockchain, chamada de Hyperledger Fabric[1]. Um do pontos chaves da tecnologia é estar sob a égide da *Linux Foundation* , que em si tem uma longa e muito bem sucedida história de nutrir projetos de código aberto sob governança aberta que cresçam entre comunidades fortes. Assim, como tem uma biblioteca de código-fonte aberta que fica livre para comunidades fazerem implementações relevantes, já que a Fabric é a primeira plataforma de contabilidade distribuída a suportar contratos inteligentes autorais em linguagens de programação de propósito geral, tal como Java, Go e Node.js.

Dessa forma, o presente trabalho tem como propósito demonstrar as diversas aplicações do *blockchain* por meio da plataforma Hyperledger, oferecendo uma solução para a autenticação de certificados digitais. Esses certificados representam documentos eletrônicos contendo informações de indivíduos ou entidades jurídicas. Conforme destacado por Lebre (2022), a certificação digital equivale a uma assinatura virtual e constitui um documento eletrônico que engloba dados do requerente, como nome,

[1] https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html

e-mail, CPF, além dos números referentes à chave pública e privada. Adicionalmente, inclui o nome e a assinatura da Autoridade Certificadora (AC) responsável por sua emissão.

Enfim, o ponto central será um software de fácil entendimento já que é assunto bastante complicado e pouco discutido. Este software usará ferramentas simples da área da programação, envolvendo toda a rede necessária do *blockchain*, a linguagem para interface da plataforma, além de bibliotecas para compilação dos códigos para no fim ser acessível.

1.1 JUSTIFICATIVA

No contexto que a Blockchain se insere popularmente é um tema debatido como o uso em criptmoedas, tal como Bitcoin, Ethereum, Cardano, etc.. No entanto, a blockchain possibilita o registro correto de transações utilizando técnicas de criptografia avançada tais como secure hash algorithm (SHA)-256 e Rivest, Shamir, and Adleman (RSA), entre outras, ultrapassando apenas o mundo das moedas virtuais.

A crescente digitalização de processos e a necessidade de garantir a autenticidade e segurança de informações exigem soluções inovadoras para a autenticação e autorização de dados. No contexto educacional, instituições enfrentam desafios relacionados à confiabilidade e à integridade de certificados acadêmicos, frequentemente sujeitos a falsificações, dificuldades de verificação e centralização em poucas entidades. A adoção da tecnologia *Blockchain*, por meio do *Hyperledger Fabric*, propõe uma abordagem descentralizada e segura para enfrentar essas questões, integrando uma Autoridade Certificadora (AC) capaz de emitir certificados digitais imutáveis e verificáveis. Essa implementação permite a separação clara dos dados, garantindo que cada certificado digital represente de forma única e confiável a ligação entre informações acadêmicas e seus respectivos proprietários. Dessa forma, além de mitigar fraudes e aprimorar a transparência, essa solução proporciona um ambiente escalável e confiável para instituições educacionais, modernizando o processo de certificação e autenticidade de credenciais acadêmicas.

Neste projeto, a justificativa para a criação de um site que utiliza uma AC baseada no HyperLedger Fabric se fundamenta na necessidade iminente de modernizar e fortalecer a infraestrutura educacional, por meio de uma tecnologia em crescimento, estabelecendo um novo paradigma para a gestão de dados no setor educacional.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Criar uma plataforma intuitiva de infraestrutura de chave pública (ICP), a fim de demonstrar a criação e verificação de certificados digitais, apresentando uma forma de validação, que confirme a veracidade do documento adicionado na plataforma, através do framework Hyperleder Fabric.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Desenvolver uma ICP baseada em Blockchain;
- Implementar uma AC baseada no framework HyperLedger Fabric;
- Desenvolver as assinaturas digitais usando Hyperledger Fabric.
- Implementar a criação/revogação de certificados digitais de maneira fácil e intuitiva;
- Publicação da ferramenta e dos resultados obtidos em artigos científicos.

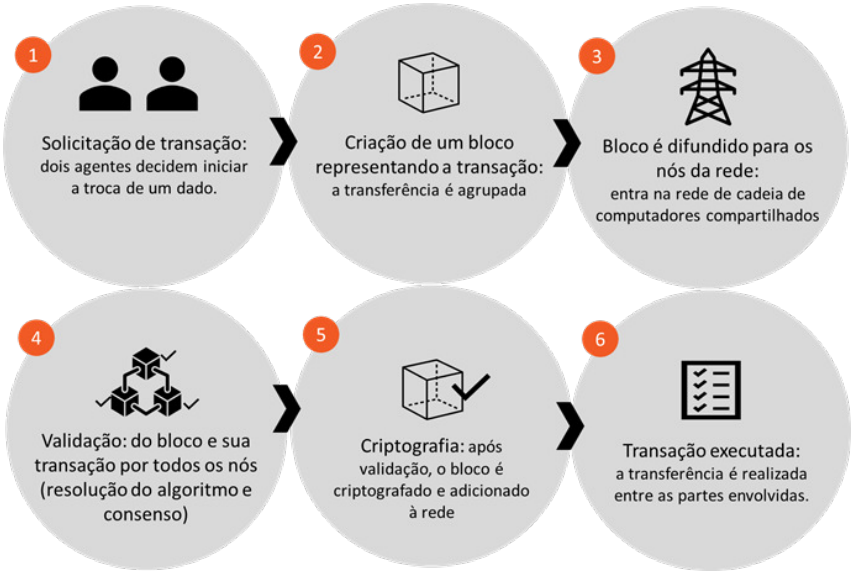
1.3 METODOLOGIA

O projeto propõe o uso da tecnologia Blockchain, especificamente por meio do framework Hyperledger Fabric, para aprimorar o processo de certificação digital em um ambiente educacional. Primeiramente, foi realizado um levantamento teórico sobre as capacidades do Blockchain e as vantagens da descentralização, buscando compreender a transformação que essa tecnologia pode trazer para sistemas de autenticação e certificação. A partir dessa base, desenvolveu-se uma Autoridade Certificadora (AC) utilizando o Hyperledger Fabric, que visa emitir e gerenciar certificados digitais de maneira segura e transparente. A solução foi projetada para superar desafios tradicionais das ACs, como centralização e vulnerabilidades a ataques, ao incorporar as funcionalidades de consenso e controle de acesso do Hyperledger Fabric. Dessa forma, o uso de Blockchain garante que os certificados sejam imutáveis, verificáveis e que o processo de revogação seja realizado de forma eficiente e confiável.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os principais conceitos envolvidos na forma como a rede *blockchain* é projetada e seus benefícios, permitindo que os dados armazenados possam ser avaliados por todos que têm acesso à rede, tornando os eventos executados e registrados totalmente transparentes. Com fundamento nisso, existem várias abordagens para construir uma rede *blockchain* atualmente. Neste trabalho será utilizada a abordagem *Hyperledger Fabric* (Androulaki et al., 2018), proposta e mantida pela Linux Foundation e pela International Business Machines (IBM). Na Figura 1, explica como Blockchain funciona.

Figura 1 – Figura adaptada da 101blockchain e da PwC



Fonte: Retirada de[1,2]

A partir disso, vale evidenciar que o trabalho de Dall'agnol (2022), que elaborou uma plataforma que chama-se Easy Smart Contracts [3], produzida com o intuito de desenvolver uma ferramenta para utilizar pouco código na criação e implantação de contratos inteligentes na rede Ethereum, através de uma interface intuitiva, sem a necessidade de conhecimento prévio em programação. Além disso, o trabalho também teve como objetivo avaliar a usabilidade da plataforma Ethereum. Em seu trabalho, o autor identificou que o Ethereum cria contratos inteligentes com base no *block-*

[1] https://101blockchains.com/pt/tecnologia-blockchain-guia/
[2] https://www.pwc.com.br/pt/temas-atuais/blockchain.html
[3] https://github.com/gabrieldonadel/easy-smart-contracts

chain, e analisou a falta de ferramentas low-code para o desenvolvimento de contratos inteligentes e decidiu criar uma plataforma que pudesse preencher essa lacuna.

Outra contribuição foi apresentada por Santos (2017), que introduziu uma metodologia de avaliação fundamentada nas necessidades tecnológicas para a implementação do *blockchain* no Hyperledger Fabric em ambiente Linux, utilizando contêineres Docker. Essa abordagem leva em consideração o sistema operacional e o sistema de virtualização, juntamente com os serviços necessários para o funcionamento do *blockchain*. Durante os testes realizados em capacidades inferiores de hardware foram obtidos avaliações valiosas para a eficácia da implementação.

Nas seções seguintes, serão demonstradas uma breve definição do que é Blockchain, como funciona o uso da rede, e o raciocínio dos nós, também será apresentado sobre contratos inteligentes (*do inglês: smart contracts*), e as particularidades do framework Hyperledger Fabric com certificados digitais e como funciona a partir do Blockchain, e os mecanismos de Consenso aplicados na rede. Além de explicar sobre Autoridade Certificadora (AC) e suas principais funções que é emitir e revogar certificados digitais e um pouco sobre Ethereum.

2.1 AUTORIDADE CERTIFICADORA

Uma Autoridade Certificadora é uma entidade confiável, em que se pode emitir e gerenciar certificados digitais em um sistema de chaves públicas *(do inglês: Public Key Infrastructure (PKI))*, possuindo várias funções para um ambiente seguro e criptografado. No estudo de Defrémont et al. (2023) é apresentado uma forma de redução da dependência na Autoridade de Certificadora (AC) da rede, destacando a integração de chaves únicas para o envio de transações por terceiros não confiáveis, além do que, destaca a necessidade de garantir a integridade e a rejeição das transações, mesmo quando inseridas por outras identidades, além de sugerir uma solução para superar esses desafios.

No Hyperledger Fabric, a AC é um componente essencial que gerencia identidades e certifica entidades participantes da rede *blockchain*. Sua principal função é emitir, renovar e revogar certificados digitais X.509, garantindo que apenas entidades autorizadas possam interagir com a rede de forma segura. A AC também auxilia na geração do MSP *(do inglês:Membership Service Provider)*, que é o sistema de identidade usado pela rede para autenticar e autorizar transações. Um MSP é um componente que define as regras que governam as identidades válidas para esta organização.

Segundo Araújo e Vieira et al. (2015), as ACs podem ser classificadas em:

AC raiz, que é a autoridade máxima na hierarquia, responsável por emitir certificados para outras, e **ACs intermediárias**, que atuam como um nível abaixo da AC raiz e são responsáveis por emitir certificados para peers, orderers e usuários da rede. Garantem a autenticação das entidades e permitem uma gestão mais distribuída das identidades.

Na prática, a AC raiz emite certificados para as ACs intermediárias, que, por sua vez, emitem certificados para os participantes da rede, ou seja, quando um usuário ou nó tenta interagir com a *blockchain*, sua identidade é validada pela AC intermediária correspondente. Se um certificado precisar ser revogado, é adicionado a uma Lista de Revogação de Certificados (CRL) para impedir seu uso indevido.

2.2 ETHEREUM

O Ethereum é outra plataforma Blockchain versátil, que oferece um ambiente para a criação de contratos inteligentes e aplicativos descentralizados (DApps). Oliveira et al. (2022) conduziram uma avaliação de desempenho de transações em Blockchain, com foco no protocolo Ethereum onde o estudo buscou avaliar o comportamento da rede Ethereum em diferentes cenários, buscando entender como a escalabilidade e o desempenho da rede são afetados em ambientes distribuídos. Além disso o estudo utilizou três tipos de contratos para realizar os testes de desempenho.

Em síntese a isso, o fundamento na escolha do framework Hyperledger Fabric é motivado por ser mais flexível e modular, permitindo a personalização conforme as necessidades. A seguir no Quadro 1 são apresentadas características, tanto do Hyperledger, quanto do Ethereum[4], apresentando implementações no Blockchain sendo possível fazer comparação entre elas.

[4] https://ethereum.org/pt-br/developers/docs/

Quadro 1 – Comparação entre *Hyperledger e Ethereum*

Característica	**Hyperledger**	**Ethereum**
Criadores	Linux Foundation	Desenvolvedores Ethereum
Modelo de Blockchain	Privado	Público e Privado
Descrição	Plataforma blockchain modular	Plataforma menos modular, mas robusta para contratos inteligentes
Geração de Contratos Inteligentes	Suporta contratos inteligentes (chaincode)	Fundamentado em contratos inteligentes nativos
Moeda	Tokens por Smart Contracts	Ether e Tokens por Smart Contracts
Linguagem de Programação	Golang, Node.js (JavaScript), Python, Java	Solidity, Vyper,Go (go-ethereum),Ruby (Ethereum.rb),Rust (Ethcontract-rs),PHP (Ethereum-PHP)
Métodos de Consenso	PBFT, Raft	Proof of Work (PoW), Proof of Stake (PoS)
Desempenho e Escalabilidade	Alta velocidade e escalabilidade para empresas	Menor escalabilidade devido ao processamento global
Casos de Uso Comuns	Empresas,identidade digital, certificação	Finanças descentralizadas (DeFi), NFTs, aplicações descentralizadas (DApps)

Fonte: Elaborado pela autora (2025)

A comparação entre Hyperledger e Ethereum destaca suas diferenças essenciais. O Hyperledger, é uma *blockchain* modular voltada para ambientes empresariais e privados. Já o Ethereum, criado por uma comunidade descentralizada, pode operar tanto em redes públicas quanto privadas e é amplamente utilizado para contratos inteligentes. Uma diferença importante está no modelo de permissão: o Hyperledger é permissionado, garantindo que apenas participantes autorizados acessem a rede, enquanto o Ethereum é não-permissionado, permitindo participação pública.

Além disso, o mecanismo de consenso difere entre as plataformas: o Hyperledger utiliza PBFT, Raft , que garantem alta eficiência e velocidade, enquanto o Ethereum inicialmente usava *Proof of Work (PoW)* e atualmente está migrando para *Proof of Stake (PoS)*, o que melhora a escalabilidade, mas ainda mantém um modelo mais descentralizado. No quesito desempenho, o Hyperledger se destaca por sua escalabilidade

e rapidez em ambientes empresariais, enquanto o Ethereum tem um processamento global mais lento devido à validação distribuída. Por fim, os casos de uso mostram que o Hyperledger é mais adequado para gestão de identidades e certificação digital, enquanto o Ethereum domina finanças descentralizadas (DeFi), NFTs e aplicativos descentralizados (DApps).

2.3 O FRAMEWORK HYPERLEDGER FABRIC

O principal objetivo do Hyperledger Fabric é fornecer uma plataforma modular e extensível que permita a criação e a operação de blockchains permissionadas de forma eficiente e adaptável a diversos casos de uso. Está sendo apresentado como um sistema pioneiro que redefine a forma como as blockchains permissionadas são projetadas e implementadas, promovendo uma solução flexível e eficiente para as necessidades das empresas na era digital.

Prover uma plataforma para o gerenciamento de identidades descentralizadas, segura, capaz de oferecer serviços de autenticação de forma genérica e abstraindo a complexidade técnica de padrões e protocolos de comunicação é um desafio. A plataforma precisa garantir um repositório de identidades descentralizadas confiável, permitir a manipulação de credenciais (emissão e revogação), e prover autenticação (Queiroz et al.,2021).

2.3.1 Mecanismos de Consenso aplicados ao Blockchain

De acordo com Miers et al.(2019), análise de mecanismos de consenso distribuído aplicados ao blockchain é fundamental para garantir a segurança, a descentralização e a confiabilidade da rede. O estudo aborda diferentes algoritmos, como *Proof of Work (PoW), Proof of Stake (PoS), Practical Byzantine Fault Tolerance (PBFT) e Proof of Authority (PoA)*, onde cada um deles representa diferentes níveis de segurança, escalabilidade e uso de energia. A seguir vamos esmiuçar cada uma das formas de consenso, tais como:

- Proof of Work (PoW): utilizado em redes bitcoin, que requer a solução de problemas matemáticos complexos, e com mais alto consumo de energia;
- Proof of Stake (PoS): presente em Ethereum 2.0, seleciona validadores com base na quantidade de criptomoedas que possuem, sendo mais eficiente em termos energéticos;

- Practical Byzantine Fault Tolerance (PBFT): adotado em redes como Hyperledger Fabric, que permite consenso mesmo com a presença de nós maliciosos, priorizando a confiança das redes privadas;
- Proof of Authority (PoA): utilizado em redes privadas, onde validadores confiáveis são responsáveis pela validação das transações, garantindo rapidez e eficiência.

Conforme está sendo abordado sobre as formas de consenso em redes blockchain, o consenso que está sendo utilizado neste projeto é o consenso *Raft*[5] , já que é uma rede privada e permissionada, ele é um algoritmo padrão do Hyperledger para garantir a ordem das transações e a imutabilidade do Ledger, sendo baseado em líder e seguidores, em que um nó líder é eleito (por canal) e suas decisões são replicadas pelos seguidores. Diferente do *PBFT (Practical Byzantine Fault Tolerance)*, que é projetado para redes com participantes potencialmente maliciosos e comunicação não confiável, o Raft é mais adequado para redes permissionadas, onde todas as entidades participantes são conhecidas e confiáveis. No contexto do Hyperledger Fabric, o Raft é preferido em redes privadas porque é mais leve, eficiente e adequado para cenários onde não há desconfiança entre os participantes. Como todos os nós na rede já possuem permissões e identidades verificadas, a necessidade de um consenso mais robusto contra ataques maliciosos, como o PBFT, se torna desnecessária. Assim, o Raft equilibra segurança, desempenho e simplicidade, tornando-se a escolha ideal para a ordenação das transações nesta rede permissionada.

2.3.2 Processo de Desenvolvimento

Segundo a documentação do Hyperledger Fabric, diversos componentes desempenham papéis fundamentais na operação da rede *Blockchain*. Os Peers são nós responsáveis por manter, validar e operar o livro-razão (*ledger*), garantindo a sincronia da rede. Os Orders determinam a ordem das transações antes de serem registradas no *ledger,* assegurando a consistência e integridade dos dados. O Channel permite a comunicação isolada e controlada entre diferentes partes da rede, possibilitando transações seguras e privadas. O Chaincode, equivalente a contratos inteligentes, automatiza e implementa a lógica de negócios dentro da *Blockchain*, permitindo interações diretas com o ledger. Por fim, o ledger funciona como um livro-razão distribuído, armazenando de forma imutável todas as transações validadas em blocos encadeados e protegidos criptograficamente. Dentro dessa estrutura, os principais elementos que compõem a rede Hyperledger Fabric incluem:

[5] https://hyperledgerfabric.readthedocs.io/pt/latest/orderer/ordering_service.html

- **Peers** = Referem-se a nós na rede que desempenham um papel essencial na manutenção, validação e operação do livro-razão (do inglês: ledger), mantendo em sincronia com a rede.
- **Orders** = São componentes que determinam a ordem das transações em uma rede até que elas sejam adicionadas no livro-razão.
- **Channel** = É uma estrutura que permite a ligação de uma parte da rede a outra rede, possibilitando a comunicação e a visibilidade entre as duas, para que possam haver transações, sendo compartilhadas de forma isolada e controlada dentro da rede.
- **Chaincode** = Responsável pela automação e a implementação necessária de informações da lógica de negócios na blockchain, referindo-se a contratos inteligentes. Além de que ao interagir com o chaincode, você interage com o ledger.
- **Ledger** = É um livro-razão que serve para armazenar todas as transações na rede, sendo estruturado como armazenamento em blocos, onde contém um conjunto de transações que foram validadas e agrupadas para serem adicionadas em cadeias. Uma vez bloco adicionado no ledger , ele não pode ser alterado ou removido , ou seja, o registro das transações é criptografadas e distribuído entre todos da rede.

3 METODOLOGIA DE IMPLEMENTAÇÃO

3.1 FERRAMENTAS

- Node.js[1] = é um ambiente de execução JavaScript,ele permite a criação de aplicativos escaláveis e assíncronos no lado do servidor. No contexto do Hyperledger Fabric, Node.js é frequentemente utilizado para desenvolver aplicações cliente que interagem com a *blockchain*
- Docker / Docker Compose [2] = Docker é uma plataforma que permite criar, distribuir e rodar aplicativos em containers, garantindo consistência entre diferentes ambientes. O Docker Compose é uma ferramenta que permite definir e gerenciar aplicações multi-container usando um arquivo YAML. No Hyperledger Fabric, Docker é essencial para executar os diferentes componentes da rede, como peers, orderers e CAs.
- Hyperledger Fabric Samples[3] = O Hyperledger Fabric Samples é um repositório que contém exemplos práticos para aprender e testar o Hyperledger Fabric. Esses exemplos incluem redes de teste, contratos inteligentes (chaincodes) e aplicações cliente.
- Hyperledger Fabric CA[4] = O Fabric CA é o serviço de gerenciamento de identidades do Hyperledger Fabric. Ele é responsável pela emissão, renovação e revogação de certificados digitais X.509 para os participantes da rede blockchain.

3.2 DEPENDÊNCIAS

- fabric-network[5] = O pacote `fabric-network` fornece APIs para interagir com o Hyperledger Fabric. Ele permite que aplicações cliente acessem a rede blockchain, enviem transações e consultem o ledger.
- fabric-ca-client[6] = O `fabric-ca-client` é uma biblioteca Node.js usada para interagir com o Hyperledger Fabric CA. Ele permite registrar e autenticar identidades na rede.

[1] https://nodejs.org/pt
[2] https://www.docker.com
[3] https://github.com/hyperledger/fabric-samples
[4] https://hyperledger-fabric-ca.readthedocs.io/en/latest/
[5] https://www.npmjs.com/package/fabric-network
[6] https://www.npmjs.com/package/fabric-ca-client

- express[7] = Express é um framework web para Node.js que facilita a criação de APIs e aplicações web. No Hyperledger Fabric, ele é frequentemente usado para desenvolver serviços REST que interagem com a blockchain.
- body parser[8] = `body-parser` é um middleware para Express que analisa requisições HTTP e converte o corpo das requisições para JSON, facilitando o processamento de dados enviados pelo cliente.

3.3 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos desempenha um papel crucial no desenvolvimento de sistemas de informação, servindo como a base para todo o processo de criação de software. É nessa fase que os analistas se reúnem com os clientes e usuários para compreender as funcionalidades essenciais do sistema a ser desenvolvido.(Mendonça et al.,2014).

Os requisitos funcionais da aplicação estão apresentados no Quadro 2. Eles descrevem as funcionalidades que o sistema deve realizar para atender às necessidades do usuário, como a realização de operações específicas e a interação com o sistema. Esses requisitos são essenciais para garantir que o sistema cumpra seu propósito e forneça as funcionalidades esperadas.

Já os requisitos não funcionais, descritos no Quadro 3, abordam aspectos qualitativos do software, como desempenho, segurança, usabilidade e confiabilidade. Esses requisitos não estão diretamente ligados a funcionalidades específicas, mas são fundamentais para garantir que o sistema seja eficiente, seguro e agradável de usar, atendendo às expectativas do usuário em relação à qualidade e à performance.

[7] https://expressjs.com
[8] https://www.npmjs.com/package/body-parser

Quadro 2 – Requisitos Funcionais

ID	Requisito Funcional	Descrição
RF1	Tela de Login	O usuário deve conseguir acessar o sistema informando e-mail e senha.
RF2	Tela de Cadastro	O usuário deve poder criar uma conta com nome, e-mail e senha.
RF3	Gerenciamento de Certificados	O usuário pode assinar, verificar, revogar e definir a data de expiração de certificados.
RF4	Assinatura de Certificados	O usuário pode assinar digitalmente um certificado.
RF5	Verificação de Certificados	O usuário pode verificar a autenticidade de um certificado.
RF6	Revogação de Certificados	O usuário pode revogar um certificado tornando-o inválido.
RF7	Definição de Data de Expiração	O usuário pode definir uma data de expiração para um certificado.
RF8	Menu Hamburguer	No canto da tela, um botão de menu deve exibir os dados do usuário e a opção de sair.
RF9	Logout	O usuário pode sair do sistema e retornar à tela de login.
RF10	Login de Administrador	O administrador pode acessar o sistema com credenciais específicas.
RF11	Listagem de Certificados Válidos	O administrador pode visualizar todos os certificados válidos.
RF12	Listagem de Certificados Revogados	O administrador pode visualizar todos os certificados revogados

Fonte: Elaborado pela autora (2025)

Quadro 3 – Requisitos Não Funcionais

ID	Requisito Não Funcional	Descrição
RF1	Desempenho	O sistema deve processar operações de certificação (assinatura, verificação, revogação) em menos de 2 segundos.
RF2	Segurança	As senhas dos usuários devem ser armazenadas de forma segura usando hashing (ex: bcrypt).
RF3	Criptografia	A assinatura dos certificados deve utilizar criptografia forte, como ECDSA.
RF4	Autenticação	O sistema deve implementar autenticação baseada em tokens JWT para maior segurança.
RF5	Controle de Acesso	Apenas administradores podem visualizar a lista de certificados válidos e revogados.
RF6	Usabilidade	O sistema deve ser intuitivo, com menus claros e fluxos de navegação simples.

Fonte: Elaborado pela autora (2025)

3.4 IMPLEMENTAÇÃO DA AUTORIDADE CERTIFICADORA

A AC é a responsável por emitir e gerenciar os certificados digitais . No contexto do Hyperledger Fabric, a AC é geralmente um Fabric CA, que gera os certificados usados para autenticação e assinatura digital dentro da rede blockchain. No Algoritmo 1, Apêndice A, o código importa o módulo FabricCAServices do Hyperledger Fabric, que é responsável por criar a conexão com a Certificate Authority (CA) , depois a função *getCAClient()* retorna um cliente da autoridade certificadora, onde esse cliente é utilizado para registrar/inscrever o usuário que interage diretamente com a AC.

Na inscrição do usuário o método *ca.register()* é utilizado para registrar um novo usuário na CA. Durante o processo de registro, o administrador da rede (adminUser) autoriza o registro de um novo usuário com o ID fornecido (userId), e a AC gera um *enrollment secret* (segredo de inscrição). Após o registro, o usuário é inscrito usando o método *ca.enroll()*. O enrolamento é o processo onde o usuário se autentica com o segredo de inscrição gerado pelo registro, e a AC emite um certificado digital e uma chave privada, que são usados para autenticar a identidade do usuário na rede.

Quando um usuário se registra em um sistema baseado em blockchain, a Autoridade Certificadora (AC) gera um par de chaves criptográficas: uma chave pública

e uma chave privada. A AC, então, assina o certificado digital, que contém a chave pública do usuário, seus dados de identificação e agora, um campo adicional que especifica a data de expiração do certificado. Esse certificado assinado é armazenado no blockchain, garantindo a transparência, integridade e imutabilidade dos dados.

O certificado digital, juntamente com a chave privada correspondente, é armazenado na carteira do usuário (do inglês: *wallet*), que serve como um repositório seguro para credenciais, incluindo certificados e chaves privadas. A adição de uma data de expiração ao certificado é uma medida importante para aumentar a segurança do sistema, pois limita o tempo de validade do certificado e previne o uso indevido em caso de comprometimento da chave privada.

3.4.1 Assinatura Digital

A assinatura digital é um mecanismo criptográfico que garante a autenticidade e integridade do documento eletrônico. Neste caso, tem-se a função *sign* (Algoritmo 4/Apêndice A) que permite, que um usuário assine um documento digitalmente, o código examina a wallet buscando a identidade do usuário e sua chave privada, em seguida o documento é assinado criando um hash criptografico (SHA-256), que salva em um arquivo de texto na pasta *signatures*. Além disso, agora a assinatura digital inclui uma data de expiração, que determina até quando a assinatura é considerada válida, aumentando a segurança e a conformidade com regulamentações de validade temporal.

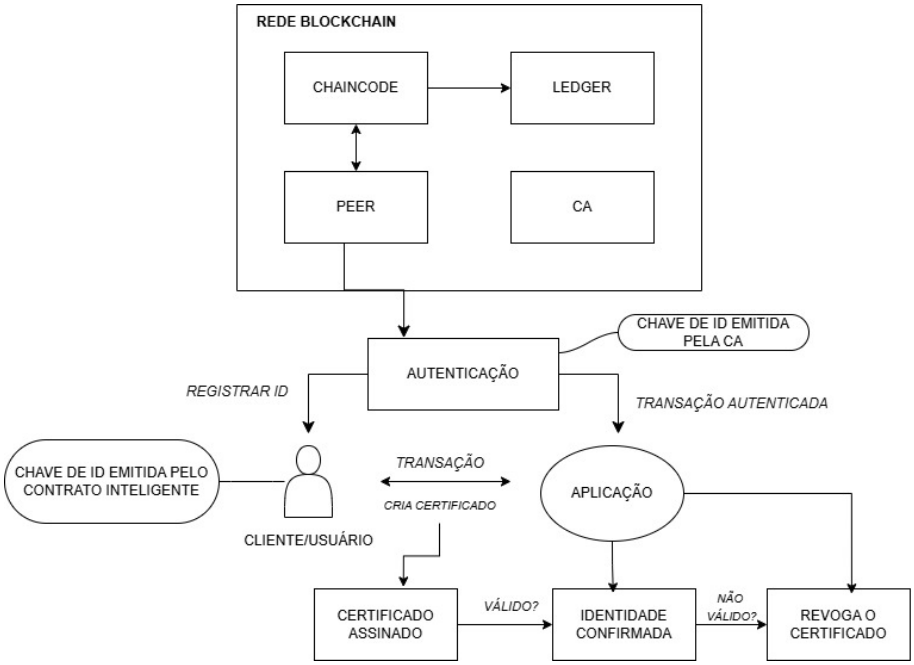
3.4.2 Revogação dos certificados

A revogação de certificados é o processo de invalidar um certificado digital antes do seu prazo de expiração, pois impede o uso indevido de credenciais comprometidas. No Algoritmo 5, Apêndice A, a revogação é registrada de forma que conecta a *network* e chama o contrato inteligente, garantindo que todos os nós saibam que o certificado não é mais válido e depois desconecta após a transação. No Hyperledger Fabric , não é possível excluir ou modificar o bloco já registrado, garantindo assim a integridade, então cria um novo bloco que revoga o certificado anterior ou corrige o erro, mas o historico da transação anterior continua na rede.

Para exemplificar a forma como a arquitetura é dividida, a Figura 2 apresenta sua estrutura e organização, destacando os principais componentes e suas interconexões, como: a rede Blockchain composta por Peers, CAs, Ledger e o Chaincode. Além disso, tem a Autenticação, responsável por repassar os tokens para o blockchain. O Cliente/Usuário gera o tokens necessários para seu registro e para realizar as transações autenticadas, enquanto a Aplicação fornece os serviços da rede ao

usuário, garantindo recebimento e validação dos tokens para as transações. Para que um contrato inteligente funcione corretamente na rede, são necessárias configurações adicionais que garantem a conformidade com os protocolos. Além disso, o serviço autenticado deve ser desenvolvido e integrado à rede. É importante acrescentar que os contratos somente executam funções se forem chamados por uma transação. Os contratos nunca podem chamar a si próprios ou atuarem em background, mas podem chamar outros contratos em cadeia. As transações são atômicas, e caso a execução ocorra sem erros até o final, toda a transação é registrada (Abijaude et al., 2021).

Figura 2 – Arquitetura de proposta



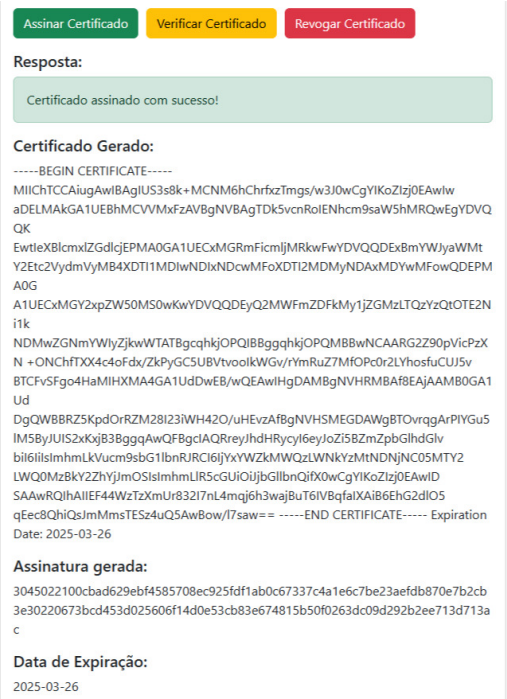
Fonte: Elaborado pela autora (2025)

O fluxo inicia quando um cliente/usuário recebe uma chave de ID emitida por um contrato inteligente e tenta registrar sua identidade no sistema. Esse processo passa pela **Autenticação**, onde a CA emite uma chave de ID para o usuário e autentica a transação. Após isso, a **Aplicação** recebe a transação autenticada e gera um **Certificado Assinado**. Quando um usuário tenta acessar o sistema, a aplicação verifica a validade do certificado. Se for válido, a identidade do usuário é confirmada e ele pode interagir com a rede blockchain. Caso contrário, o sistema revoga o certificado, impedindo o acesso do usuário. Esse fluxo garante segurança e controle sobre identidades dentro da rede Hyperledger Fabric.

3.4.3 Telas do Sistema

Esta seção apresenta uma visão geral das principais interfaces do sistema, destacando suas funcionalidades e o fluxo de navegação. Aqui, o usuário pode visualizar as telas de Gerenciamento de certificados, a tela de Revogação, e a tela do Admin, onde faz uma listagem de certificados válidos e revogados, e outras seções essenciais, compreendendo como cada uma contribui para a experiência do sistema. A Figura 3 ilustra a tela do certificado exibida após o cadastro do usuário, contendo a assinatura gerada e a data de expiração já definida.

Figura 3 – Imagem da interface dos Certificados



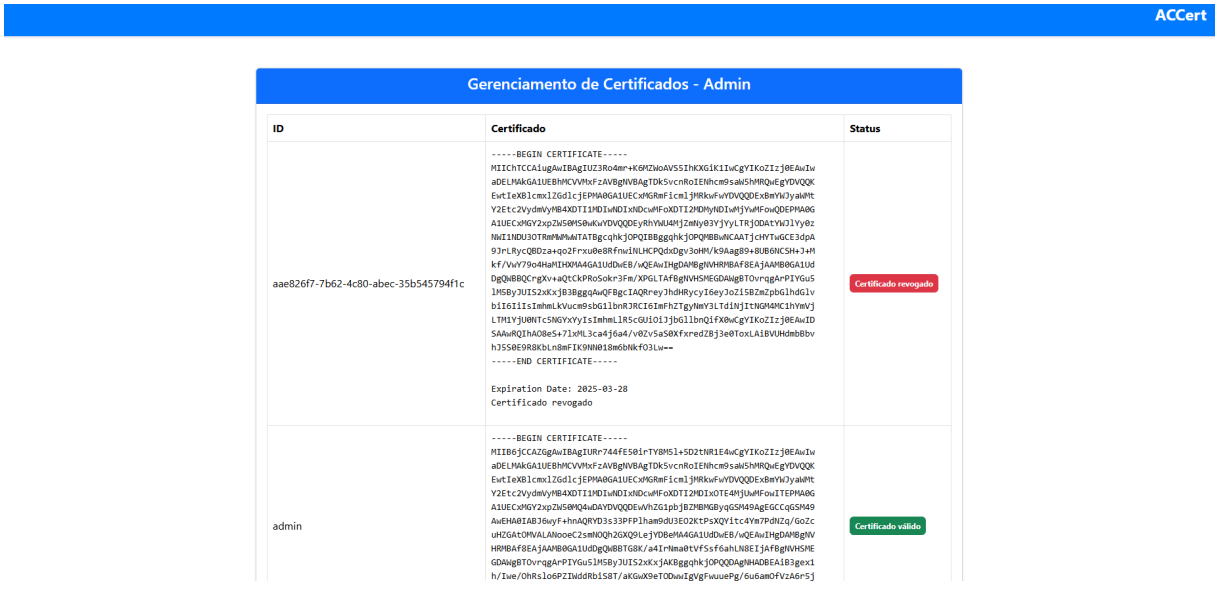
Fonte: Elaborado pela autora (2025)

Em seguida, a Figura 4 apresenta a tela de revogação. Após a revogação, todos os dados do certificado são removidos, restando apenas uma mensagem informativa para o usuário.

Figura 4 – Tela de Revogação de Certificação

Por fim, a Figura 5 exibe a tela do administrador, acessível apenas por meio de login exclusivo. Nessa tela, o administrador pode visualizar a listagem de certificados, incluindo aqueles válidos e revogados, juntamente com os dados dos respectivos usuários.

Figura 5 – Tela do Administrador



4 CONSIDERAÇÕES FINAIS

4.1 RESULTADOS

O framework Hyperledger Fabric foi identificado como altamente versátil, capaz de se adaptar a diversas condições devido à sua arquitetura descentralizada e distribuída. Essa flexibilidade também contribui para sua escalabilidade em ambientes favoráveis, uma vez que opera sobre a tecnologia de contêineres do Docker. Através do estudo da implementação, foi possível estabelecer uma estrutura funcional para o Blockchain, abordando as necessidades de criação das regras de negócio e das estruturas de dados essenciais, utilizando Smart Contracts (Chaincode).

Na camada de interconexão, foram definidas as exigências para o desenvolvimento de uma API em Node.js, destinada à troca de informações com sistemas externos, com a estrutura implementada, é possível persistir e consultar os dados, considerando que sistemas baseados em blockchain não permitem alterações nas informações após a criação de um bloco.

A partir das respostas fornecidas pela API, foi criada uma interface que permite ao usuário interagir com os dados retornados de forma compreensível e eficiente. A interface exibe as informações de maneira clara, permitindo uma visualização simplificada dos resultados. Essa abordagem possibilita a construção de uma aplicação mais interativa e funcional, facilitando a experiência do usuário ao acessar e manipular as informações fornecidas pela rede, mostrando a parte de assinatura, verificação e revogação dos certificados.

4.2 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma arquitetura segura e eficiente para o envio de transações por meio de identidades e aplicações externas a uma rede blockchain, abordando os desafios de garantir segurança robusta e controle de acesso à rede. A proposta inclui a implementação de um serviço de autenticação em contratos inteligentes, bem como métodos para geração de tokens únicos e seguros, que fortalecem a integridade e autenticidade das transações. Além disso, foi desenvolvida uma autoridade certificadora, com emissão e revogação de certificados assinados digitalmente, assegurando um processo confiável e auditável.

Embora o trabalho apresente algumas limitações e desafios na implementação prática, a proposta oferece avanços significativos no uso da rede Hyperledger Fabric,

destacando-se na combinação de segurança, escalabilidade e controle. Durante o processo de desenvolvimento, há uma oportunidade de otimizar a arquitetura para ser implementada em ambientes mais performáticos, visando reduzir o tempo de execução das respostas e proporcionar uma interface mais intuitiva para os usuários, aumentando assim a eficiência geral do sistema.

Em trabalhos futuros podem explorar melhorias na comunicação do sistema com outras plataformas blockchain, permitindo um ecossistema mais flexível e adaptável a diferentes aplicações. Além disso, a implementação de algoritmos mais eficientes para verificação de identidade e controle de acesso pode fortalecer ainda mais a segurança e reduzir a latência nas transações. Outra possível evolução seria o desenvolvimento de uma interface gráfica aprimorada para facilitar a gestão de certificados e transações pelos administradores e usuários. Esses aprimoramentos não apenas consolidariam os avanços apresentados, mas também abririam caminho para novas pesquisas e aplicações práticas na área de segurança e autenticação em blockchain.

REFERÊNCIAS

LESSAK, A. L.; DIAS, R. A.; FREY, I. A. Blockchain: Prospecção tecnológica em Bases de Patentes. Cadernos de Prospecção, [S. l.], v.11, n. 3, p. 876, 2018. Disponível em: https://periodicos.ufba.br/index.php/nit/article/view/27006. Acesso em: 21 nov. 2023. DOI: 10.9771/cp.v11i3.27006.

COSTA, Giovana. Tecemos,2021. Blockchain: desmistificando essa tecnologia e suas aplicações. Disponível em:https://www.tecemos.com/blog-Blockchain.html. Acesso em: 23 março 2023.

GOMES, Alan Nascimento; COUTINHO, Emanuel Ferreira. Um Estudo Inicial sobre a Importância de Simular Contratos Inteligentes em Blockchain. In: WORKSHOP EM MODELAGEM E SIMULAÇÃO DE SISTEMAS INTENSIVOS EM SOFTWARE (MSSIS), 4. , 2022, Uberlândia/MG. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022 . p. 1-10. DOI: https://doi.org/10.5753/mssis.2022.225407.

ANDROULAKI, E.; BARGER, A.; BORTNIKOV, V.; CACHIN, C.; CHRISTIDIS, K.; CARO, A. D.; ENYEART, D.; FERRIS, C.; LAVENTMAN, G.; MANEVICH, Y.; MURALIDHARAN, S.; MURTHY, C.; NGUYEN, B.; SETHI, M.; SINGH, G.; SMITH, K.; SORNIOTTI, A.; STATHAKOPOULOU, C.; VUKOLIć, M.; COCCO, S. W.; YELLICK, J. Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference. New York, NY, USA: ACM, 2018. (EuroSys '18), p. 30:1–30:15. ISBN 978-1-4503-5584-1. Disponível em: <http://doi.acm.org/10.114 /3190508.3190538>.

LEBRE, O. C. das N.; ANDRADE, E. R.; MIRANDA, M. R. Prospecção Tecnológica em Bases de Patentes Sobre Certificação Digital e Blockchain. Cadernos de Prospecção, *[S. l.]*, v. 15, n. 2, p. 588–603, 2022. DOI: 10.9771/cp.v15i2.42509. Disponível em: DOI: https://doi.org/10.9771/cp.v15i2.42509

DALL'AGNOL, Gabriel Donadel. Plataforma de Desenvolvimento de Smart Contracts Baseada em Interface Gráfica. 2022. 75 f. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) - Universidade Federal de Santa Catarina, Florianópolis. Disponível em: https://repositorio.ufsc.br/handle/123456789/237939

DEFRÉMONT, André; SOUSA, Jeffson Celeiro; PINHEIRO, Billy Anderson; ARAÚJO, Roberto Samarone; ABELÉM, Antônio Jorge G.. Transações Autenticadas para Aplicativos Não Confiáveis em Blockchains Permissionadas. In: WORKSHOP EM BLOCKCHAIN: TEORIA, TECNOLOGIAS E APLICAÇÕES (WBLOCKCHAIN), 6. , 2023, Brasília/DF. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2023 . p. 57-70. DOI: https://doi.org/10.5753/wblockchain.2023.770.

OLIVEIRA, Danilo Bizarria de; AMARAL, Geraldo Lucas Fernandes do; MACEDO, Markel Pedrosa Duarte de; MESQUITA, Matheus Ferreira; BIANCHINI, Calebe de Paula. Avaliação de desempenho de transações em protocolos blockchain: um estudo de caso com Ethereum. In: WORKSHOP DE INICIAÇÃO CIENTÍFICA - SIMPÓSIO EM SISTEMAS COMPUTACIONAIS DE ALTO DESEMPENHO (WSCAD), 23. , 2022, Florianópolis. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022 . p. 49-56. DOI: <https://doi.org/10.5753/wscad_estendido.2022.226339.>

SANTOS, Júlio Sérgio Quadros dos. Implementação do Blockchain Hyperledger

Fabric em ambiente Linux. 2017. 18 f. Trabalho de Conclusão de Curso (Tecnologia em Sistemas para Internet). Instituto Federal Sul-rio-grandense, Campus Passo Fundo, Passo Fundo, Rio Grande do Sul. <https://painel.passofundo.ifsul.edu.br/uploads/arq/20180412172403161465538 7.pdf>

QUEIROZ, Silvio; GREVE, Fabíola; SAMPAIO, Leobino N.; MARQUES, Eduardo. Plataforma para Gestão de Identidades Descentralizadas Baseada em Blockchain. *In*: SIMPÓSIO BRASILEIRO DE SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS (SBSEG), 21. , 2021, Belém. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021 . p. 29-42. DOI: https://doi.org/10.5753/sbseg.2021.17304.

MIERS, Charles Christian; KOSLOVSKI, Guilherme Piêgas; PILLON, Maurício Aronne; MARQUES, Marco Antonio. Blockchains com Hyperledger: conceitos, instalação, configuração e uso. In: SOARES, Marcelo; et al. (Org.). *Minicursos da XVIII Escola Regional de Redes de Computadores (ERRC 2020)*. Porto Alegre: Sociedade Brasileira de Computação (SBC), 2020. p. 15-40. Disponível em: https://books-sol.sbc.org.br/index.php/sbc/catalog/book/58. Acesso em: 17 ago. 2025.

ABIJAUDE, Jauberth; SERRA, Henrique; SANTIAGO, Levy; SOBREIRA, Péricles; GREVE, Fabíola. Blockchain, contratos inteligentes, sistemas web: teoria e prática. In: DORÇA, Fabiano Leandro (org.). *Tópicos Especiais em Sistemas de Informação: Minicursos SBSI 2021*. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 1-30. DOI: https://doi.org/10.5753/sbc.6453.5.1

MIEIRS, Charles C.; KOSLOVSKI, Guilherme P.; PILLON, Maurício A.; SIMPLÍCIO JR., Marcos A.; CARVALHO, Tereza C. M. B.; RODRIGUES, Bruno B.; BATTISTI, João H. F. Análise de mecanismos para consenso distribuído aplicados a Blockchain. In: SIMPÓSIO BRASILEIRO DE SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS, 19., 2019, Anais... [S.l.]: SBC, 2019. Disponível em: DOI: https://doi.org/10.5753/sbc.8589.4.3

ROCHA, Luiz Augusto Pinto. *Quarta Revolução Industrial: Tecnologia Blockchain e as Cadeias Globais de Valor*. 2018. Trabalho de Conclusão de Curso (Especialização em Relações Internacionais) — Universidade de Brasília, Brasília, 2018. Disponível em: https://bdm.unb.br/bitstream/10483/24340/1/2018_LuizAugustoPintoRocha_tcc.pdf

MENDONÇA, R. A. R. de. Levantamento de requisitos no desenvolvimento ágil de software. Semana da Ciência e Tecnologia da PUC Goiás, p. 12, 2014.

ARAÚJO, Wagner Junqueira de; VIEIRA, Yasmin Brito de Lemos. Comparação entre produtos e serviços oferecidos pelas Autoridades Certificadoras. RDBCI: Revista Digital de Biblioteconomia e Ciência da Informação, Campinas, SP, v. 13, n. 2, p. 366–385, 2015.Disponível em: https://doi.org/10.20396/rdbci.v13i2.2120 .

APÊNDICE A – EXEMPLO DE CÓDIGOS

Este apêndice apresenta uma seleção de códigos relevantes para a compreensão dos tópicos técnicos abordados neste trabalho. Os códigos a seguir estão diretamente relacionados à **implementação da autoridade certificadora**, abrangendo processos como **verificação de certificados, criação de assinaturas digitais e revogação de certificados**. Estes trechos de código são fornecidos com o intuito de complementar o entendimento e detalhar a implementação desses aspectos críticos no contexto do trabalho.

1. **Importação do módulo FabricCAServices, onde é o responsável por criar a conexão com a Autoridade Certificadora (AC).**

Algoritmo 1 – Função do código de *RegisterUser*

```
const FabricCAServices = require('fabric-ca-client');
const { Wallets } = require('fabric-network');
const path = require('path');

const getCAClient = require('./getCAClient');

const registerUser = async (userId, username, password, name, email) => {
    try {
        const walletPath = path.join(process.cwd(), 'wallet');
        const wallet = await Wallets.newFileSystemWallet(walletPath);

        const userExists = await wallet.get(username);
        if (userExists) {
            throw new Error(`Usuário com username "${username}" já está registrado.`);
        }

        const adminIdentity = await wallet.get('admin');
        if (!adminIdentity) {
            throw new Error('Admin não encontrado na wallet. Execute primeiro o registro do administrador.');
        }

        const ca = getCAClient(); // Obtemos a CA client
        const provider = wallet.getProviderRegistry().getProvider(adminIdentity.type);
        const adminUser = await provider.getUserContext(adminIdentity, 'admin');

        // Verifica se o admin tem permissão para registrar
        const enrollmentSecret = await ca.register(
            { enrollmentID: userId, role: 'client' },
            adminUser
        );

        const enrollment = await ca.enroll({
            enrollmentID: userId,
            enrollmentSecret,
        });

        const userIdentity = {
            credentials: {
                certificate: enrollment.certificate,
                privateKey: enrollment.key.toBytes(),
            },
            mspId: 'Org1MSP',
            type: 'X.509',
            metadata: { userId, username, email, name, password },
        };

        // Salvando a identidade do usuário na wallet
        await wallet.put(userId, userIdentity);
        await wallet.put(email, userIdentity);

        return userIdentity;

    } catch (error) {
        console.error(`Erro ao registrar usuário: ${error.message}`);
        throw error;
    }
};

module.exports = { registerUser };
```

Fonte: Elaborado pela autora (2025)

No trecho de código a seguir, são utilizadas as funções de verificação de assinatura do documento *verifySignature* ou *verifyCertificate*, onde são usadas a chave pública do certificado do usuário. Este certificado foi emitido por uma AC e pode ser validado na cadeia de certificação da rede.

2. **Configuração de Verificação da assinatura**

Algoritmo 2 – Função do código *verifySignature*

```
const verifySignature = (data, signature, publicKey) => {
    const verifier = crypto.createVerify('SHA256');
    verifier.update(data);
    verifier.end();

    const isValid = verifier.verify(publicKey, signature, 'hex');
    return isValid;
};
```

Fonte: Elaborado pela autora (2024

Além disso, a função representada no Algoritmo 3 é responsável por verificar a autenticidade da assinatura em qualquer certificado. Durante o processo de verificação, essa função não apenas valida a assinatura da AC, mas também verifica se o certificado ainda está dentro do período de validade definido pela data de expiração. Caso o certificado tenha expirado, ele é considerado inválido, reforçando a segurança do sistema e prevenindo possíveis ataques ou usos indevidos.

3. **Verificação do Certificado , junto com assinatura e data de expiração**

Algoritmo 3 – Função do código *verifyCertificate*

```
const verifyCertificate = async (req, res) => {
    try {
        const { userId, certificateData, signature } = req.body;
        const walletPath = path.join(process.cwd(), 'wallet');
        const wallet = await Wallets.newFileSystemWallet(walletPath);

        const userIdentity = await wallet.get(userId);
        if (!userIdentity) {
            throw new Error(`Usuário com ID ${userId} não encontrado.`);
        }

        const publicKey = userIdentity.credentials.certificate;
        const isValid = verifySignature(certificateData, signature, publicKey);

        if (!isValid) {
            return res.status(400).json({ message: "Certificado não é autêntico." })
                ;
        }

        const expirationMatch = certificateData.match(/Expiration Date: (.+)/);
        if (!expirationMatch) {
            return res.status(400).json({ message: "Certificado não contém data de
                expiração." });
        }

        const expirationDate = new Date(expirationMatch[1]);
        const currentDate = new Date();

        if (currentDate > expirationDate) {
            return res.status(400).json({ message: "Certificado expirado." });
        }

        if (certificateData.includes("Certificado revogado")) {
            return res.status(400).json({ message: "Certificado foi revogado e não é
                mais válido." });
        }

        res.status(200).json({ message: "Certificado é válido e autêntico." });
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
};
```

Fonte: Elaborado pela autora (2024

4. **Função para a criação da Assinatura digital**

Algoritmo 4 – Função do código *sign*

```
1   const sign = async (req, res) => {
2    try {
3        const { userId, certificateData, expirationDate } = req.body;
4        const walletPath = path.join(process.cwd(), 'wallet');
5        const wallet = await Wallets.newFileSystemWallet(walletPath);
6
7        const userIdentity = await wallet.get(userId);
8        if (!userIdentity) {
9            throw new Error(`Usuário com ID ${userId} não encontrado.`);
10       }
11
12       const certificateWithExpiration = `${certificateData}\nExpiration Date: ${
             expirationDate}`;
13
14       const privateKey = userIdentity.credentials.privateKey;
15       const signature = await signDocument(certificateWithExpiration, privateKey);
16
17       saveSignature(userId, signature);
18
19       res.status(200).json({ certificate: certificateWithExpiration, signature });
20   } catch (error) {
21       res.status(400).json({ error: error.message });
22   }
23 };
```

Fonte: Elaborado pela autora (2024)

5. **Função de Revogação do Certificado**

Algoritmo 5 – Função do código *revokeCertificate*

```
const revokeCertificate = async (req, res) => {
    try {
        const { userId, email, certificateData } = req.body;
        const walletPath = path.join(process.cwd(), 'wallet');
        const wallet = await Wallets.newFileSystemWallet(walletPath);

        const userIdentityById = await wallet.get(userId);
        if (!userIdentityById) {
            throw new Error(`Usuário com ID ${userId} não encontrado.`);
        }

        const userIdentityByEmail = await wallet.get(email);
        if (!userIdentityByEmail) {
            throw new Error(`Usuário com email ${email} não encontrado.`);
        }

        if (userIdentityById && userIdentityByEmail) {
            const privateKey = userIdentityById.credentials.privateKey;
            const revocationMessage = `Certificado revogado`;

            const revokedCertificateData = `${certificateData}\n${revocationMessage
                }`;

            const revokedSignature = await signDocument(revokedCertificateData,
                privateKey);

            userIdentityById.credentials.certificate = revokedCertificateData;
            userIdentityByEmail.credentials.certificate = revokedCertificateData;

            await wallet.put(userId, userIdentityById);
            await wallet.put(email, userIdentityByEmail);

            return res.status(200).json({ revokedCertificate: revokedSignature,
                message: revocationMessage });
        }

        throw new Error("Erro ao revogar o certificado: não foi encontrado o usuário
             correspondente ao ID ou email fornecido.");
    } catch (error) {
        res.status(400).json({ error: error.message });
    }
};
```

Fonte: Elaborado pela autora (2025

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Cajazeiras - Código INEP: 25008978
Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

Documento do TCC aprovado e com ficha catalográfica

Assunto:	Documento do TCC aprovado e com ficha catalográfica
Assinado por:	Alana Cristina
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Alana Cristina Felix da Silva**, **DISCENTE (202112010028) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 28/03/2025 11:20:54.

Este documento foi armazenado no SUAP em 28/03/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1438555
Código de Autenticação: 00d6fd060c