Universidade Federal do Estado do Rio de Janeiro
Centro de Ciências Exatas e Tecnológicas
Escola de Informática Aplicada

# Emissão de Certificados Acadêmicos Utilizando a Tecnologia *Blockchain*

Lionel da Rocha Alves de Oliveira

Rio de Janeiro, RJ – Brasil
Fevereiro, 2025

Catalogação informatizada pelo(a) autor(a)

O48 Oliveira, Lionel da Rocha Alves
Emissão de Certificados Acadêmicos Utilizando a Tecnologia Blockchain / Lionel da Rocha Alves Oliveira. -- Rio de Janeiro : UNIRIO, 2025.
53

Orientador: Pedro Nuno de Souza Moura.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2025.

1. blockchain. 2. certificados digitais. 3. processo acadêmico digital. I. Moura, Pedro Nuno de Souza, orient. II. Título.

Emissão de certificados acadêmicos utilizando a tecnologia blockchain

Lionel da Rocha Alves de Oliveira

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada por:

---

Prof. Pedro Nuno de Souza Moura

---

Prof. Jefferson Elbert Simões

---

Prof. Leonardo Luiz Alencastro Rocha

Rio de Janeiro, RJ – Brasil

Fevereiro de 2025

# Agradecimentos

À minha mãe e à dra. Sarah, por não desistirem de mim, e à Medicina.

# Epígrafe

*"A essência da ciência é abrir caminho em direção ao futuro, um pequeno passo de cada vez" - Dr. Stone*

**Resumo**

A crescente demanda por educação on-line e digital no Brasil evidencia a necessidade de soluções seguras e eficientes para a certificação acadêmica. Apesar dos avanços na emissão de certificados digitais, ainda há desafios como perda de documentos e dificuldades de acesso. Neste trabalho, é proposta uma plataforma baseada em *blockchain* para emissão e gestão de certificados educacionais, em que são a modelagem e a arquitetura são detalhadamente descritas. Utilizando as características inerentes à *blockchain*, como imutabilidade e transparência, a solução desenvolvida mitiga a ocorrência de fraudes e simplifica a verificação de autenticidade, contribuindo para maior eficiência e segurança nos processos acadêmicos digitais que concernem a certificados.

**Palavras-chave:** *blockchain*, certificados digitais, processo acadêmico digital.

**Abstract**

The growing demand for online and digital education in Brazil highlights the need for secure and efficient solutions for academic certification. Despite advancements in the issuance of digital certificates, challenges such as document loss and access difficulties persist. This graduation final project proposes a blockchain-based platform for the issuance and management of educational certificates, in which the modeling and architecture are described in detail. By leveraging the inherent characteristics of blockchain, such as immutability and transparency, the developed solution mitigates fraud occurrences and simplifies authenticity verification, contributing to greater efficiency and security in digital academic processes related to certificates.

**Keywords:** blockchain, digital certificates, digital academic process.

# Sumário


**1 Introdução** **1**

1.1 Motivação . . . . . 1

1.2 Objetivos . . . . . 2

1.3 Organização do texto . . . . . 3

**2 Conceitos preliminares** **4**

2.1 Introdução à *blockchain* . . . . . 4

2.2 Características de uma rede *blockchain* . . . . . 4

2.3 Hash SHA-256 . . . . . 5

2.4 Blocos . . . . . 5

2.5 Consenso . . . . . 6

2.6 Protocolos de consenso . . . . . 7

2.7 Ethereum . . . . . 8

2.8 Contratos inteligentes . . . . . 9

2.9 ERC-721 . . . . . 10

2.10 Carteiras . . . . . 11

**3 Revisão da literatura** **12**

3.1 Uso de *blockchain* para emissão de documentos . . . . . 12

3.2 Uso de *blockchain* para emissão de certificados acadêmicos . . . . . 13

**4 Especificação do sistema** **17**

4.1 Visão Geral . . . . . 17

4.2 Regras de negócio . . . . . 17

4.3 Requisitos funcionais . . . . . 18

4.4 Requisitos não funcionais . . . . . 19

4.5 Casos de uso . . . . . 19

4.5.1 **Caso de Uso:** Adicionar endereço emissor de certificado . . . . . 20

4.5.2 **Caso de Uso:** Revogar permissão para emissão de certificado . . . 21

4.5.3 **Caso de Uso:** Emitir certificado . . . . . 23
4.5.4 **Caso de Uso:** Visualizar certificado . . . . . 24
4.5.5 **Caso de uso:** Listar certificados de um endereço . . . . . 25
4.5.6 **Caso de uso:** Obter endereço pelos dados de um certificado . . . . 25

**5 Arquitetura do sistema** **27**
5.1 Arquitetura . . . . . 27
5.1.1 Camada de apresentação . . . . . 27
5.1.2 Camada lógica - programa facilitador . . . . . 28
5.1.3 Camada lógica - contrato inteligente . . . . . 28
5.1.4 Camada de dados . . . . . 28
5.2 Modelagem de dados . . . . . 29
5.3 Tecnologias . . . . . 30

**6 Interações do Sistema** **31**

**7 Conclusão** **35**
7.1 Considerações finais . . . . . 35
7.2 Limitações do estudo . . . . . 35
7.3 Trabalhos futuros . . . . . 36

# Lista de Figuras


1 Estrutura de uma *blockchain* (retirada de [29]). . . . . 6
2 Diagrama de casos de uso para o sistema de emissão de certificados. . . . . 20
3 Diagrama da arquitetura do sistema. . . . . 27
4 Diagrama entidade-relacionamento. . . . . 30
5 Apresentação do sistema. . . . . 31
6 Interação: fornecendo permissão de emissão de certificados a um endereço. 32
7 Interação: removendo permissão de emissão de certificados de um endereço. 32
8 Interação: emitindo certificado a um estudante. . . . . 33
9 Interação: obtendo os dados do certificado de um usuário. . . . . 33
10 Interação: obtendo os IDs dos certificados associados a um estudante. . . . 33
11 Interação: obtendo o endereço de um estudante a partir das informações associadas a um certificado. . . . . 34
12 Interação: ocorrência de falha em uma transação devido a endereço de contrato incorreto. . . . . 34
13 Interação: ocorrência de falha em uma transação e mensagem apresentada pelo sistema. . . . . 34

# 1 Introdução

## 1.1 Motivação

A demanda crescente por educação on-line e digital tem intensificado a busca por soluções que assegurem eficiência, segurança e transparência nos processos de certificação. Segundo dados do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep), nos últimos cinco anos, o número de vagas oferecidas em cursos de graduação no formato de Educação a Distância (EaD) aumentou $167{,}5\%$, refletindo o avanço da educação digital no Brasil [9]. Além dos cursos de graduação, há também uma expansão significativa de cursos de extensão, cursos livres e outras modalidades de ensino on-line, que, juntos, destacam a necessidade de soluções tecnológicas robustas para garantir a integridade e confiabilidade das certificações emitidas.

No Brasil, a Infraestrutura de Chaves Públicas Brasileira (ICP-Brasil)[1] foi estabelecida para assegurar a autenticidade, integridade e validade jurídica de documentos eletrônicos [6]. Atualmente, o país conta com mais de 20 autoridades certificadoras, responsáveis pela identificação precisa de indivíduos e pela emissão de certificados digitais. Qualquer ocorrência de fraude na emissão de um certificado digital pode permitir autenticação fraudulenta em sistemas, acesso a documentos confidenciais e até mesmo a prática de falsidade ideológica [30].

De acordo com Awaji e Solaiman (2022) [3], a fragilidade dos sistemas tradicionais de registro pode resultar na vulnerabilidade dos dados, dificultando a autenticação segura de diplomas e certificados acadêmicos, além de aumentar os riscos associados a fraudes. Mesmo com a existência do ICP-Brasil e a emissão de certificados digitais sem *blockchain*, ainda há fraudes na emissão de diplomas para conseguir cargos, aumentos, entre outras vantagens. Além do prejuízo financeiro, diplomas fraudulentos apresentam riscos à população [25, 32, 47].

Além disso, os alunos, mesmo após a emissão de certificados em formato digital, podem perder os documentos ou terem dificuldades em acessá-los. Isso tende a gerar uma situação de insegurança e ineficiência, uma vez que o processo para recuperar tais documentos costuma exigir tempo, esforço e, frequentemente, custos adicionais tanto para as instituições

[1] https://www.gov.br/iti/pt-br/assuntos/icp-brasil

quanto para os próprios alunos.

Por sua vez, a *blockchain*, por conta de suas características, apresenta-se como uma solução natural para essa problemática, a saber, de perda de certificados emitidos e existência de fraudes. Tal tecnologia foi inicialmente introduzida como a arquitetura subjacente à criptomoeda *Bitcoin* [57]. No documento intitulado "*Bitcoin: A Peer-to-Peer Electronic Cash System*" [44], Nakamoto apresentou o *Bitcoin* como uma alternativa às instituições financeiras tradicionais, introduzindo um sistema monetário digital baseado em *blockchain*, que permite transações diretas entre pares sem a necessidade de intermediários. A tecnologia *blockchain* garante a imutabilidade (não modificação) dos registros e utiliza um mecanismo de consenso conhecido como *Proof of Work* (prova de trabalho), o que confere à rede segurança contra adulterações e ataques externos. O objetivo de Nakamoto foi criar um "sistema de pagamento eletrônico baseado em prova criptográfica, em vez de confiança", permitindo transações seguras e anônimas sem o envolvimento de entidades centralizadas.

Desde então, o conceito de *blockchain* se expandiu além do âmbito das moedas digitais, encontrando aplicações em várias indústrias, como na transparência para cadeia de suprimentos marítima, verificação de identidade digital e na saúde [43, 51, 56].

Um sistema baseado em *blockchain* para emissão, verificação e gerenciamento de certificados de cursos pode oferecer uma solução robusta, segura e eficiente. Utilizando a natureza descentralizada, distribuída e imutável da *blockchain*, este sistema pode agilizar o processo de verificação de certificados, garantir transparência e facilitar a manutenção de registros de forma segura e resistente a adulterações. Além disso, como o foco da aplicação é a construção de confiança, robustez e proveniência dos dados, a blockchain, conforme apontado por Chowdhury, é a melhor solução. [13]

## 1.2 Objetivos

O objetivo geral deste trabalho é principalmente o desenvolvimento de um sistema capaz de lidar com emissão, verificação e gerenciamento de certificados acadêmicos, que uma vez inseridos na blockchain, se tornarão imutáveis. Tal solução apresenta benefícios não apenas para o público inserido no contexto de *web3* e *blockchain*, mas também para o público em geral.

## 1.3 Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, contará com os seguintes conteúdos:

**Capítulo 2:** conceitos preliminares - apresenta e explica os conceitos necessários ao entendimento deste trabalho;

**Capítulo 3:** revisão bibliográfica - descreve os trabalhos relacionados ao tema levantados na literatura;

**Capítulo 4:** especificação do sistema - aborda a especificação do sistema desenvolvido, incluindo os requisitos funcionais e casos de uso;

**Capítulo 5:** arquitetura do sistema - inclui modelagem dos dados e explicação detalhada da estrutura do sistema;

**Capítulo 6:** interações do sistema - trata da interação do usuário com o programa;

**Capítulo 7:** conclusão - reúne as considerações finais, assinala as limitações do estudo e sugere possibilidades de aprofundamento posterior.

# 2 Conceitos preliminares

Este capítulo visa a apresentar os conceitos utilizados no desenvolvimento deste trabalho, a saber, *blockchain* e contratos inteligentes. Aqueles que se sentirem confortáveis com estes conceitos podem pular este capítulo sem prejuízo para o entendimento do trabalho.

## 2.1 Introdução à *blockchain*

Desde a sua criação, a tecnologia *blockchain* demonstrou uma notável versatilidade, expandindo-se muito além de sua aplicação inicial como infraestrutura subjacente a criptomoedas. Como destacado por Nakamoto na obra seminal [44], a *blockchain* surgiu como uma solução descentralizada para eliminar a necessidade de intermediários em transações financeiras. No entanto, o impacto da tecnologia evoluiu significativamente desde então [16, 58].

Além das criptomoedas, a *blockchain* encontrou aplicações em áreas diversas, como contratos inteligentes e finanças descentralizadas. Mais especificamente, contratos inteligentes automatizam a execução de acordos entre as partes [10, 54], enquanto as finanças descentralizadas eliminam entidades intermediárias em serviços financeiros [31]. Esses novos casos de uso, como argumenta Swan [53], têm o potencial de transformar indústrias inteiras, incluindo os setores bancário, jurídico e de seguros.

## 2.2 Características de uma rede *blockchain*

Conforme exposto por Swan [53], a *Blockchain* 1.0 refere-se ao uso de criptomoedas e suas funções primárias, como transferências, pagamentos e outras operações associadas ao manejo de dinheiro. Já a *Blockchain* 2.0 abrange os contratos inteligentes e suas diversas aplicações, incluindo ações, mercados futuros, empréstimos, entre outras possibilidades. Nesse contexto, a iniciativa DREX[2], a moeda digital do Banco Central do Brasil, destaca-se como exemplo. No entanto, ao contrário de outras soluções baseadas em *blockchain*, o acesso aos serviços do DREX será mediado por instituições bancárias, que funcionarão como intermediários. Ainda assim, contratos inteligentes serão empregados para assegurar o cumprimento das condições estabelecidas, permitindo a liberação dos ativos monetários quando as exigên-

[2]https://www.bcb.gov.br/estabilidadefinanceira/drex

cias forem atendidas.

No nível seguinte, denominado *Blockchain* 3.0, Swan [53] prevê a expansão do uso da tecnologia para além do setor financeiro, abrangendo áreas governamentais, culturais, artísticas, educacionais e científicas, o que implica um potencial de transformação significativo em diversos setores da sociedade. Um exemplo de aplicação da *blockchain* nesse terceiro nível corresponde à garantia da integridade e da transparência na totalização de votos em urnas eletrônicas e seu emprego na promoção de uma gestão pública mais transparente [17, 21].

## 2.3 Hash SHA-256

O *SHA-256* [45] é uma função de *hash* criptográfica amplamente utilizada que converte dados de entrada de qualquer comprimento em um valor de *hash* fixo de 256 *bits* (32 *bytes*). O *SHA-256* faz parte da família de funções de *hash SHA-2*, que também inclui o *SHA-384* e o *SHA-512*. O algoritmo foi projetado para ser resistente a colisões, o que significa que é altamente improvável que duas entradas diferentes produzam o mesmo valor de *hash*. Essa propriedade torna o *SHA-256* uma ferramenta crucial para garantir a integridade e a autenticidade de informações digitais, como arquivos, documentos ou transações.

## 2.4 Blocos

De maneira geral, um bloco na *blockchain* é composto por duas partes principais: o cabeçalho do bloco e o corpo do bloco [57].

O cabeçalho contém metadados fundamentais para a validação do bloco e sua conexão com o bloco anterior, em que um dos elementos mais importantes é o *hash* do bloco anterior. A existência desse *hash* do bloco anterior no bloco atual estabelece uma sequência única entre os blocos da cadeia. Dessa forma, qualquer pequena alteração em um bloco $b_1$ invalida todos os blocos posteriores devido à mudança no *hash* de $b_1$, provocando uma inconsistência na rede facilmente detectável [19].

Além disso, o cabeçalho inclui o *Merkle Root*, um resumo criptográfico de todas as transações contidas no bloco, criado a partir de uma estrutura denominada de árvore de *Merkle* [42], que permite a verificação eficiente da integridade das transações sem a necessidade de armazenar todas elas diretamente. Outros componentes do cabeçalho incluem o *timestamp*, que registra a data e a hora da mineração do bloco, e o *nonce*, um número aleatório

ajustado durante o processo de mineração para atender ao critério de dificuldade imposto pelo mecanismo de consenso *Proof of Work* [4].

Já o corpo do bloco contém a lista de transações validadas pela rede antes da inclusão do bloco na *blockchain*. Cada transação inclui informações como o endereço do remetente e do destinatário, valor transferido, taxa de transação e assinatura digital, garantindo a autenticidade e a rastreabilidade das operações [57]. Em *blockchains* como o *Ethereum*, o corpo do bloco também registra dados relacionados ao consumo de energia (refletindo o custo computacional para executar transações e contratos inteligentes) e às taxas pagas pelos usuários para sua execução [10]. A Figura 1 ilustra a estrutura dos blocos em uma *blockchain*.

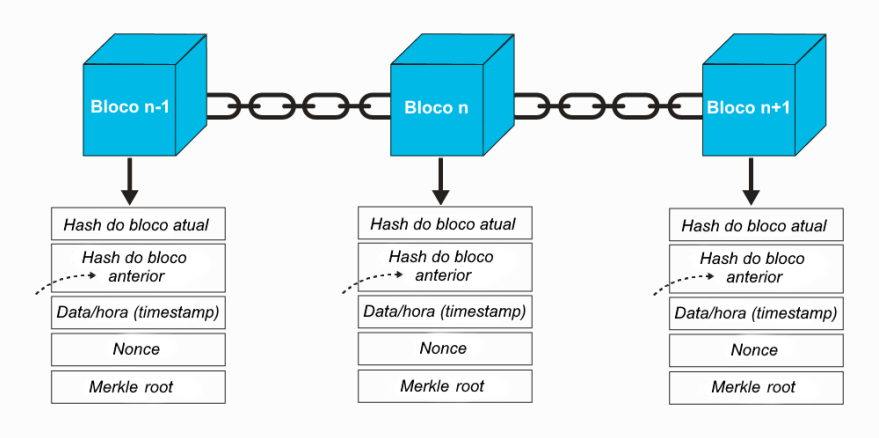


**Figura 1:** Estrutura de uma *blockchain* (retirada de [29]).

É importante destacar que, de maneira geral, as blockchains armazenam referências às transações diretamente na própria blockchain, enquanto os dados dessas transações ficam armazenados em bancos de dados distribuídos. Dessa forma, cada nó faz uma cópia do banco de dados até que ocorra a sincronização com os dados mais recentes. [27] [39] [18] O Hyperledger Besu, por exemplo, utiliza o RocksDB [5].

## 2.5 Consenso

Tipicamente, um protocolo de consenso especifica como fazer com que múltiplos nós concordem sobre algo. No contexto de *blockchain*, o protocolo define como os nós fazem para determinar se um valor deve ser adicionado à cadeia [4]. O consenso em *blockchain*, dessa forma, é o processo pelo qual os nós de uma rede distribuída concordam sobre a validade e a ordem das transações sem a necessidade de uma autoridade central. Esse mecanismo garante que todos os participantes compartilhem a mesma versão do livro-razão, também chamado

de *ledger*, evitando inconsistências e prevenindo ataques maliciosos, como o gasto duplo, um problema que ocorre quando o usuário tenta gastar a mesma criptomoeda duas vezes para ganhar benefícios monetários [44] [35]. Se um usuário conseguir realizar com sucesso um ataque de gasto duplo, efetivamente nenhum dinheiro é gasto, pois ele estaria tentando usar a mesma quantia em duas transações diferentes, levando a um "gasto"inválido. [35] Diferentes algoritmos de consenso, como *Proof of Work* e *Proof of Stake*, a serem detalhados na próxima seção, foram desenvolvidos para equilibrar segurança, descentralização e eficiência computacional [28].

## 2.6 Protocolos de consenso

O protocolo *Proof of Work* (Prova de Trabalho - PoW), que, de acordo com Yaga *et al*. [57], desempenhou um papel fundamental no sucesso de criptomoedas como o *Bitcoin*, é um mecanismo de consenso que exige que os participantes, conhecidos como mineradores, compitam entre si para resolver um problema matemático complexo a fim de validar transações e adicionar novos blocos à *blockchain*. Esse processo, frequentemente chamado de mineração (*mining*), foi projetado para garantir a segurança da rede e impedir que um único participante manipule o livro-razão (*ledger*) [57].

Contudo, a mineração exige um consumo considerável de eletricidade para operar computadores especializados que empregam o algoritmo de consenso *Proof of Work* na validação de transações registradas no livro-razão. A quantidade exata de energia consumida pelas criptomoedas permanece incerta, uma vez que, por padrão, elas são difíceis de monitorar. Mesmo assim, o mecanismo de consenso adotado pelo *Bitcoin* para mineração e transações é notoriamente intensivo em termos de energia [23].

Com a preocupação do consumo de energia em mente, a rede *Ethereum* mudou, em 2022, do protocolo *Proof of Work* para *Proof of Stake* (PoS). Segundo estimativas da CCRI, o processo conhecido como "*The Merge*" reduziu o consumo anual de eletricidade da Ethereum em mais de $99.988\%$. Da mesma forma, a pegada de carbono da rede diminuiu aproximadamente $99.992\%$ – passando de $11.016.000$ para $870$ toneladas de $CO_2$ [**?**].

Diferente do *Proof of Work*, o protocolo de consenso *Proof of Stake* (PoS) seleciona participantes para a criação de novos blocos com base na quantidade de criptomoeda que possuem e alocam como garantia. Essa abordagem não apenas reduz significativamente o consumo de energia, mas também reforça a segurança da rede, uma vez que qualquer

comportamento malicioso pode resultar na perda parcial ou total dos fundos comprometidos pelo participante [26].

Por seu turno, no protocolo *Proof of Authority* (Prova de Autoridade - PoA), um conjunto de "autoridades" são predeterminadas e cada uma tem um tempo fixo no qual pode gerar blocos [2]. Os blocos são criados em rodízio, e a sua criação não requer cálculos complexos ou competição entre os participantes. O tempo de confirmação é reduzido porque os participantes da rede são predeterminados e têm autoridade para gerar blocos imediatamente. De forma sintética, o protocolo *Proof of Authority* requer menos computação e, por isso, tem melhor performance, embora gere maior grau de centralização na rede [20] [48].

## 2.7 Ethereum

*Ethereum*, concebido por Vitalik Buterin em 2013 e lançado formalmente em 2015, foi pensado para ser mais do que uma criptomoeda, e sim uma plataforma de computação distribuída baseada em *blockchain*, de código aberto e pública, que oferece um ambiente versátil para a execução de contratos inteligentes, assunto que será tratado na sessão seguinte. Inicialmente, o protocolo *Ethereum* foi desenvolvido como uma evolução das criptomoedas, incorporando funcionalidades avançadas por meio de uma linguagem de programação altamente genérica e Turing-completa. [10]

No *whitepaper*, Vitalik Buterin [10] declara que "o que o *Ethereum* pretende oferecer é uma blockchain com uma linguagem de programação Turing-completa embutida, capaz de criar 'contratos' que codificam funções arbitrárias de transição de estado." Essa arquitetura suporta a criação e o lançamento de aplicações descentralizadas (dApps), que operam em redes distribuídas, em contraste com os servidores centralizados. A criptomoeda nativa, o Ether, serve tanto como taxa cobrada para a execução de contratos inteligentes quanto como mecanismo para remunerar os mineradores.

A *Ethereum Virtual Machine* (EVM) [24] é o ambiente de execução que opera como o núcleo da rede Ethereum. Ela funciona como um computador descentralizado que processa o código dos contratos inteligentes, garantindo que todos os nós da rede executem transações de maneira consistente e determinística. Além disso, a EVM permite que contratos escritos em linguagens de alto nível, como *Solidity* e *Vyper*, sejam compilados em *bytecode*, possibilitando sua execução autônoma e a interação entre aplicações descentralizadas em um ambiente sem confiança.

## 2.8 Contratos inteligentes

Os contratos inteligentes (*smart contracts*) são acordos digitais que executam automaticamente os termos de um contrato quando condições predefinidas são atendidas. Ao incorporar termos contratuais na *blockchain*, os contratos inteligentes podem otimizar processos empresariais, reduzir custos administrativos e minimizar o risco de erros humanos ou fraudes [33]. Como descrito por Szabo [54], os contratos inteligentes podem atuar controlando a liberação de ativos de acordo com critérios personalizáveis. Dessa forma, um contrato inteligente funciona como um caixa eletrônico em uma operação de saque: caso o usuário tenha fundos, a operação poderá ser concluída com sucesso. Do contrário, o caixa eletrônico não dispensará as notas. Um contrato inteligente pode envolver a compra de ativos digitais e tokens, que só serão fornecidos ao comprador caso ele tenha criptomoeda suficiente para comprá-los e pagar pela taxa da transação.

Os contratos inteligentes possuem diversas funcionalidades de segurança embutidas, como a imutabilidade após serem implantados para garantir a consistência dos dados e a rastreabilidade das transações. Tal característica reduz os riscos de segurança e aumenta a confiança dos usuários em relação a esse tipo de contrato. Entretanto, devido à forte correlação entre contratos inteligentes e o acesso a recursos financeiros, muitos usuários maliciosos têm explorado vulnerabilidades nesses contratos para obter lucros. Mesmo que tais vulnerabilidades sejam detectadas, elas não podem ser corrigidas por meio de atualizações de versão, exceto pela autodestruição do contrato, haja vista a natureza resistente a adulterações dessa abordagem [14].

Essa flexibilidade limitada permitiu, por exemplo, o ataque à Organização Autônoma Descentralizada (*Decentralized Autonomous Organization* - DAO), que ocorreu em junho de 2016, um dos eventos mais notáveis na história das criptomoedas e do *Ethereum*. A DAO foi criada para funcionar como um fundo de capital de risco, permitindo que os investidores participassem em projetos de *blockchain*. No entanto, uma situação não prevista pelos programadores do contrato inteligente, conhecida posteriormente como "ataque de reentrância", permitiu que um *hacker* retirasse cerca de $3{,}6$ milhões de *ETH*, a criptomoeda associada à plataforma *Ethereum*, o que representava aproximadamente $50$ milhões de dólares na época [41].

Nesse sentido, a organização OpenZepellin[3] fornece uma biblioteca de código aberto que

[3]https://www.openzeppelin.com

permite que usuários utilizem na escrita de seus contratos inteligentes outros contratos já auditados, ou seja, que passaram por um processo de verificação de falhas de segurança. Essa biblioteca é utilizada no desenvolvimento de contratos inteligentes para *blockchains* compatíveis com *Ethereum Virtual Machine* e oferece uma coleção robusta e auditada de padrões e implementações de contratos inteligentes, como *ERC-20* e *ERC-721* para *tokens* fungíveis e não fungíveis (*non-fungible token* - NFTs), respectivamente. Além disso, a biblioteca inclui módulos para controle de acesso, governança e proteção contra ataques comuns, como o próprio ataque de reentrância.

Em contratos inteligentes, a operação *revert* aciona uma exceção para sinalizar um erro e reverter a chamada atual. A *Ethereum Virtual Machine* retorna para o cliente uma mensagem opcional em forma de texto, contendo informações sobre o erro. O *revert* assegura que nenhuma alteração no estado da *blockchain* seja feita, mantendo a integridade e a consistência dos dados. Dessa forma, a transação não é registrada no bloco, como se nunca tivesse ocorrido, embora as taxas de computação sejam cobradas, pois a transação foi processada. O prazo para o *revert* de uma transação é o tempo de criação do bloco, ou seja, o tempo definido no arquivo *genesis* da *blockchain* para produção de bloco. Uma transação irá sempre para o próximo bloco a ser criado. [34]

Uma transação que sofreu *revert* é aquela em que, durante sua execução em um contrato inteligente, ocorreu um erro ou violação de uma condição definida, acionando a operação para *revert*. Isso faz com que todas as alterações de estado realizadas pela transação sejam desfeitas, retornando o sistema ao estado anterior à execução da transação. [34]

## 2.9 ERC-721

Conforme proposto em 2018 por William Entriken, Dieter Shirley, Jacob Evans e Nastassia Sachs, o ERC-721 é um padrão de token do Ethereum projetado para criar tokens não fungíveis (NFTs), ativos digitais únicos e indivisíveis. Ao contrário dos tokens fungíveis, como o ERC-20, onde cada token é intercambiável, os tokens ERC-721 são distintos e não podem ser replicados, o que os torna ideais para representar a posse de itens exclusivos. [22]

Ao definir funções essenciais, como rastreamento de propriedade, transferências e aprovações, o ERC-721 simplificou o desenvolvimento de NFTs e fomentou o crescimento de um ecossistema de propriedade digital. Os contratos inteligentes ERC-721 incluem funções principais, como: identificar o proprietário de um token específico, permitir a transfe-

rência de um token único entre endereços e conceder permissão a terceiros para gerenciar um token. Cada token possui um identificador único (por exemplo, um número de série) armazenado na blockchain, garantindo escassez e proveniência. [1]

## 2.10 Carteiras

As carteiras (*wallets*) são ferramentas para a interação com redes de *blockchain*, proporcionando a criação, o armazenamento e o gerenciamento de chaves criptográficas, que são fundamentais para a realização de transações e para a autenticação de identidades digitais. Elas armazenam as chaves públicas e privadas necessárias para acessar e gerenciar ativos digitais baseados em *blockchain*, como criptomoedas e NFTs, permitindo que os usuários enviem, recebam e rastreiem seus ativos digitais de forma segura na *blockchain* [57].

As carteiras *blockchain* funcionam gerando um endereço único, uma sequência de caracteres alfanuméricos que representa a chave pública do usuário. Essa chave pública pode ser usada para receber ativos digitais, enquanto a chave privada correspondente é utilizada para autorizar a transferência de ativos para fora da carteira. A chave privada funciona como uma senha, e os usuários devem mantê-la segura, pois qualquer um com acesso à chave privada pode controlar e manipular os ativos da carteira. Em suma, se um usuário perder uma chave privada de uma dada carteira, qualquer ativo digital que ele tenha nessa carteira pode ser considerado perdido, pois é computacionalmente infactível gerar novamente a mesma chave privada [57].

# 3 Revisão da literatura

Este capítulo apresenta os trabalhos relacionados ao problema abordado, isto é, que tenham lidado com a emissão de documentos e de certificados acadêmicos por meio de *blockchain*.

## 3.1 Uso de *blockchain* para emissão de documentos

A iniciativa da cidade de Zug, Suíça, por meio do lançamento do uPort – uma identidade digital governamental – representa um exemplo pioneiro de como a tecnologia pode transformar a administração pública. O registro da identidade digital baseada em *blockchain* na *Ethereum*, certificada pela cidade de Zug, teve início em 15 de novembro de 2017. Ao oferecer aos cidadãos uma identidade autossoberana e segura, o projeto possibilita que informações pessoais sejam compartilhadas de forma seletiva com instituições governamentais e empresas privadas, fortalecendo o controle individual sobre os próprios dados [37].

Em 2018, cerca de 300 pessoas já usavam a aplicação uPort, que poderia suportar 30 mil usuários. Uma solução de identidade autossoberana que reduz a necessidade de manter repositórios centralizados de informações de identificação. Uma vez que a propriedade e a autenticação das identidades são transferidas para os cidadãos, não há mais necessidade de hospedar servidores e bancos de dados com dados pessoais. Além disso, na arquitetura distribuída, o risco de um grande vazamento de dados pessoais é eliminado, e a nova forma de autenticação gera economia de tempo para os cidadãos no acesso a serviços [37].

O sistema de vouchers inteligentes Stadjerspas, implementado pelo Município de Groningen, foi uma plataforma baseada em *blockchain* projetada para fornecer subsídios direcionados a residentes de baixa renda. Desenvolvido em parceria com a DutchChain e introduzido em 2016, ele substituiu um sistema de *vouchers* em papel que vinha sendo utilizado desde 1994. A solução utilizava contratos inteligentes na *blockchain* Zcash para aplicar critérios de elegibilidade, limites de uso e condições de gasto, garantindo que os fundos públicos fossem utilizados exclusivamente para os propósitos originais. Os cidadãos recebiam códigos QR personalizados vinculados a identidades anônimas na *blockchain*, permitindo o acesso a serviços subsidiados, como clubes esportivos, cinemas ou incentivos para painéis solares. O sistema se destacava pela transparência, programabilidade dos fundos e integração com registros municipais, mantendo os dados pessoais off-chain para proteger a

privacidade [38, 36].

A solução funcionava da seguinte maneira: um cidadão solicitava o Stadjerspas, fornecendo seus dados pessoais, que eram verificados pelo município. Indivíduos elegíveis recebiam identidades anonimizadas na *blockchain* e um passe vinculado a um código QR. Os prestadores de serviço escaneavam esse código QR para ativar os *vouchers*, acionando contratos inteligentes que validavam a elegibilidade e monitoravam os limites de uso. A arquitetura técnica do sistema incluía aplicativos móveis para os usuários, bancos de dados municipais para verificação de elegibilidade, uma camada de API para administração e a infraestrutura *blockchain* gerenciada pela DutchChain [38, 36].

Operacional de 2016 a 2020, o sistema atendeu mais de $20.000$ usuários e processou aproximadamente $4.000$ transações de *vouchers* inteligentes por mês. Ele funcionava em paralelo com os processos municipais existentes, sem substituí-los. Os *vouchers* baseados em *blockchain* ofereciam uma maneira eficiente de programar e monitorar o uso de serviços subsidiados, e os dados de uso registrados serviam para fins de auditoria, aumentando a transparência e a responsabilidade sobre os gastos públicos. Além disso, a automação por meio de contratos inteligentes eliminou processos baseados em papel e reduziu a necessidade de trabalho manual por parte do município [38, 36].

Todavia, até 2019, os custos de desenvolvimento, operação e transações na *blockchain* não eram divulgados publicamente, levantando questões sobre a sustentabilidade financeira a longo prazo. Embora a escalabilidade fosse uma das promessas do sistema, não foram realizados testes práticos sob volumes mais altos de transações para validar sua eficiência em cenários de maior demanda [38].

## 3.2 Uso de *blockchain* para emissão de certificados acadêmicos

A tecnologia *Blockcerts* foi desenvolvida pelo MIT em parceria com a empresa *Learning Machine*, para a emissão, armazenamento e verificação de certificados educacionais na *blockchain*. [40] A tecnologia permite que instituições educacionais registrem certificados de forma imutável, garantindo autenticidade e eliminando a necessidade de intermediários no processo de validação. O sistema utiliza um *hash* criptográfico do certificado, que é armazenado na *blockchain*, permitindo que qualquer pessoa possa verificar sua legitimidade apenas comparando o *hash* registrado com o hash do documento apresentado pelo titular.

Além da segurança e da transparência proporcionadas pela *blockchain*, o *Blockcerts* tam-

bém oferece autonomia aos estudantes, permitindo que armazenem seus certificados digitalmente e compartilhem-nos diretamente com empregadores ou universidades. Além disso, a verificação de um certificado pela plataforma é simples e amigável.

Outra universidade a utilizar as propriedades da tecnologia *blockchain* é a Universidade de Nicosia (UNIC), localizada no Chipre, uma das pioneiras na adoção de tecnologias descentralizadas no ensino superior [55]. Em 2017, passou a emitir todos os seus diplomas na rede do *Bitcoin*. A iniciativa foi implementada como parte de sua estratégia para promover a transparência e segurança na verificação de credenciais acadêmicas.

A solução proposta por Cheng *et al*. [11] concentra-se no uso da *blockchain Ethereum* e de contratos inteligentes para armazenar *hashes* de certificados, que por sua vez, são vinculados a *QR codes*. O objetivo principal é viabilizar uma validação rápida e segura, reduzindo a dependência de intermediários. O foco na validação por *QR codes* e números seriais demonstra uma ênfase maior na acessibilidade da ferramenta ao público em geral e na integração com sistemas existentes.

Por outro lado, Rahardja *et al*. [49] avançam no conceito de um sistema colaborativo entre instituições educacionais, utilizando a *blockchain Vexanium*[4] como infraestrutura. A solução enfatiza a geração de *hashes* criptográficos (mais especificamente, SHA-256) para representar cada certificado e a incorporação de *QR codes* para facilitar a validação por terceiros. A utilização de uma *blockchain* pública promove transparência e permite que qualquer usuário verifique a autenticidade de um certificado, seja escaneando o *QR code* ou consultando diretamente a *blockchain*. Além disso, o artigo introduz a ideia de um consórcio de universidades compartilhando uma infraestrutura *blockchain* para gerenciar certificados, o que representa um avanço em relação à solução de Cheng et al. [11].

Os trabalhos [11] e [49] apresentados anteriormente propõem soluções distintas, mas complementares, para o problema de emissão e validação de certificados digitais por meio de *blockchain*. Eles abordam os desafios de falsificação, perda e verificação de certificados, mas divergem em termos de abordagem tecnológica, foco funcional e implementação, principalmente no que se refere à infraestrutura utilizada - o primeiro, utilizando a *blockchain Ethereum* como base, e o segundo, pela *blockchain Vexanium*, cujos contratos inteligentes são escritos em C++.

O trabalho de Souza, Carneiro e Coutinho [52] propõe uma solução para a geração e validação de diplomas digitais, utilizando a *blockchain* pública *Ethereum* junto ao protocolo

[4]https://www.vexanium.com/

de armazenamento descentralizado IPFS (InterPlanetary File System). Nessa abordagem, cada diploma é registrado como um NFT na *blockchain*, o que assegura a imutabilidade e autenticidade do certificado. No entanto, ao invés de armazenar os dados completos do diploma diretamente na *blockchain* — o que seria ineficiente devido aos custos e limitações do armazenamento da rede —, o diploma contém apenas um hash criptográfico gerado a partir dos dados do diploma, que estão armazenados no IPFS. O IPFS, então, é utilizado para armazenar os dados do diploma de maneira descentralizada. Os dados do diploma são armazenados fora da *blockchain*, mas o hash gerado a partir desses dados é registrado na rede *Ethereum*. Esse hash funciona como um ponteiro para o conteúdo armazenado no IPFS.

Para facilitar a interação com o sistema, a solução inclui uma aplicação *web* que elimina a necessidade de ferramentas especializadas, como o gerenciador de carteiras *Metamask*[5]. Essa interface permite que os usuários, sejam estudantes ou empregadores, consultem e validem diplomas sem conhecimento técnico avançado. O processo de validação envolve a inserção do *hash* da transação ou do identificador do diploma no portal *web*. O sistema consulta a *blockchain* para verificar a autenticidade do NFT associado e, em seguida, utiliza o *hash* registrado para recuperar o conteúdo armazenado no IPFS. Dessa forma, o diploma pode ser baixado e validado por qualquer parte interessada.

Já o trabalho de Chowdhary, Agrawal e Rudra [12] propõe outro sistema para emissão de certificados em *blockchain*, também usando a plataforma *Ethereum* para armazenar os certificados educacionais. De forma análoga ao sistema proposto neste trabalho, apenas organizações autorizadas têm a permissão de emitir certificados, porém utiliza a ferramenta *Metamask* para conectar com a carteira do usuário e o IPFS não é utilizado para armazenar a imagem do certificado, e sim para guardar um *hash* de identificação de emissor ou de destinatário. Há uma inovação na verificação da validade do certificado, que é realizada em três etapas principais: verificação da identidade do estudante, verificação do certificado e verificação de vínculo entre estudante e certificado.

A solução proposta por Reno *et al*.[50] utiliza a tecnologia *Hyperledger* para criar uma rede *blockchain* privada dedicada ao armazenamento de certificados. Todavia, essa abordagem apresenta desvantagens significativas em termos de segurança, transparência e descentralização. Como o principal objetivo da certificação baseada em *blockchain*, no geral, é eliminar intermediários e permitir verificações independentes, o uso de uma *blockchain* privada compromete esse benefício, uma vez que terceiros ainda precisam confiar na entidade

[5]https://metamask.io/

responsável pela gestão da rede. Os autores reconhecem essa limitação e optam por priorizar um maior volume de transações e maior velocidade de processamento em detrimento de um dos princípios fundamentais da *blockchain*, a descentralização. Apesar de criticarem as taxas da rede *Ethereum*, os autores não deixaram claro no artigo se a rede *Hyperledger* adotada possui ou não taxa de transação.

Costa *et al*. [15] apresentam mais uma abordagem para autenticação e preservação de documentos digitais utilizando a tecnologia *blockchain*. A solução propõe a criação de uma plataforma que combina o registro descentralizado na rede *blockchain* com a preservação digital em um repositório, oferecendo um ambiente para o gerenciamento de documentos. Seu diferencial é que as instituições educacionais desempenham um papel ativo no processo, registrando documentos oficiais por meio de uma interface interativa. O sistema registra simultaneamente os dados na *blockchain* e em um repositório digital dedicado à preservação de longo prazo.

Os certificados dos alunos, juntamente com outros documentos acadêmicos oficiais, são armazenados nesse repositório de preservação digital de longo prazo. O componente de blockchain do sistema armazena um registro do registro do documento e o carimbo de data e hora, mas não o próprio documento. Esse registro na blockchain serve como prova de autenticidade e permite a verificação, enquanto o repositório garante a disponibilidade e integridade do documento a longo prazo.

Para permitir uma integração mais ampla, o protótipo dessa proposta inclui uma interface programável de aplicação (*application programming interface* - API). Além dela, a plataforma também oferece um portal interativo destinado a usuários finais, como estudantes, empregadores e outras partes interessadas. Nesse, qualquer pessoa pode verificar a autenticidade de um documento fornecendo seu número de registro. O sistema então consulta a *blockchain* para validar o registro correspondente, confirmando sua integridade e autenticidade.

# 4 Especificação do sistema

Este capítulo tem como objetivo apresentar a especificação do sistema desenvolvido, abordando desde as regras de negócio até o modelo de dados obtido.

## 4.1 Visão Geral

O propósito do sistema é oferecer uma solução segura, transparente e imutável para a emissão e verificação de certificados acadêmicos digitais, utilizando a tecnologia blockchain. O objetivo principal é combater fraudes, como certificados falsificados que afirmam ser emitidos por instituições renomadas, ou certificados adulterados. Dessa forma, busca-se garantir a integridade e a autenticidade das credenciais acadêmicas, tornando mais eficiente o processo de validação em ambientes acadêmicos.

A solução proposta consiste em um contrato inteligente chamado "Certificate", baseado no padrão *ERC-721*, o mesmo dos *tokens* não fungíveis (NFTs), para emitir e gerenciar certificados acadêmicos armazenados em *blockchain*, junto a um programa facilitador para o seu uso. Esse contrato é compatível com qualquer rede *blockchain* baseada em *Ethereum Virtual Machine*, ou seja, além de poder ser utilizado na própria rede *Ethereum*, outras redes públicas como *Polygon*, *Binance Smart Chain* e *Cardano*, bem como redes privadas que tenham compatibilidade, podem usufruir de tal contrato.

## 4.2 Regras de negócio

As regras de negócio estabelecidas para o sistema foram definidas com base em uma análise comparativa de soluções existentes que abordam problemas semelhantes. Foi realizada uma pesquisa sobre sistemas de proposta similar, identificando práticas e estratégias que têm se mostrado eficazes na implementação de funcionalidades similares. Além disso, o processo incluiu a aplicação de princípios de design thinking, visando adaptar as soluções encontradas às necessidades específicas do contexto.

1. Apenas endereços autorizados (emissores) podem emitir certificados;
2. Cada certificado é um *token* não fungível (ERC-721) único, representando um diploma ou certificação;

3. O certificado deve armazenar informações essenciais, como instituição, curso e número de matrícula do estudante;
4. Certificados são emitidos diretamente para a carteira do estudante, garantindo que ele seja o único proprietário;
5. Estudantes não podem transferir seus certificados entre si;
6. O endereço do estudante pode ser recuperado usando a instituição, curso e número de matrícula;
7. Qualquer pessoa pode verificar um certificado pelo seu ID na *blockchain*;
8. Um estudante pode listar todos os certificados que possui;

## 4.3 Requisitos funcionais

A partir das regras de negócio, foram obtidos os requisitos funcionais do sistema, que seguem abaixo:

1. O sistema deve permitir que o implantador do contrato adicione endereços com permissão para emitir certificados;
2. O sistema deve permitir que o implantador do contrato revogue o papel de emissor de certificados de um endereço específico;
3. O sistema deve permitir que endereços com o papel de emissor realizem a emissão de certificados;
4. O sistema deve possibilitar a visualização dos dados de um certificado por meio de seu número de ID;
5. O sistema deve permitir a consulta de todos os certificados associados a um endereço específico;
6. O sistema deve possibilitar a obtenção do endereço de um titular de certificado, utilizando os dados da instituição, curso e matrícula associados ao certificado;
7. O sistema deve impossibilitar a transferência de certificados entre estudantes.

## 4.4 Requisitos não funcionais

A partir das regras de negócio, foram obtidos os requisitos não funcionais do sistema:

- **Usabilidade:** a interface do sistema pela linha de comando deve ser intuitiva e amigável, facilitando a interação dos usuários, desde a emissão até a verificação dos certificados, sem exigir conhecimentos técnicos avançados.
- **Compatibilidade:** o sistema deve ser compatível com diversas redes blockchain baseadas em EVM.
- **Armazenamento descentralizado:** o sistema deve aceitar uso de armazenamento descentralizado, como IPFS, para armazenar dados volumosos de forma confiável e resistente a quedas.

## 4.5 Casos de uso

Com base nos requisitos funcionais descritos anteriormente, o diagrama de casos de uso exposto na Figura 2 foi elaborado. Como é possível visualizar, existem três atores e seis casos de uso no total.

Os três atores foram definidos de maneira que o implantador do contrato seja um gerente responsável pela adição e remoção de emissores, conforme necessário. Os emissores, por sua vez, são as universidades ou organizações incumbidas da emissão dos certificados. Assim, o mesmo contrato pode ser utilizado por diversas partes interessadas, como universidades, plataformas de cursos, entre outras. Os certificados de um aluno serão mantidos em sua carteira, independentemente da organização que os tenha emitido.

Nas próximas subseções, cada um dos casos de uso da Figura 2 serão detalhados, tanto os reais quanto os essenciais. Os casos de uso essenciais são: “emitir certificado” e “visualizar certificado”, enquanto os reais são “adicionar endereço emissor de certificado”, “revogar permissão para emissão de certificado”, “listar certificados de um endereço” e “obter endereço pelos dados de um certificado”.

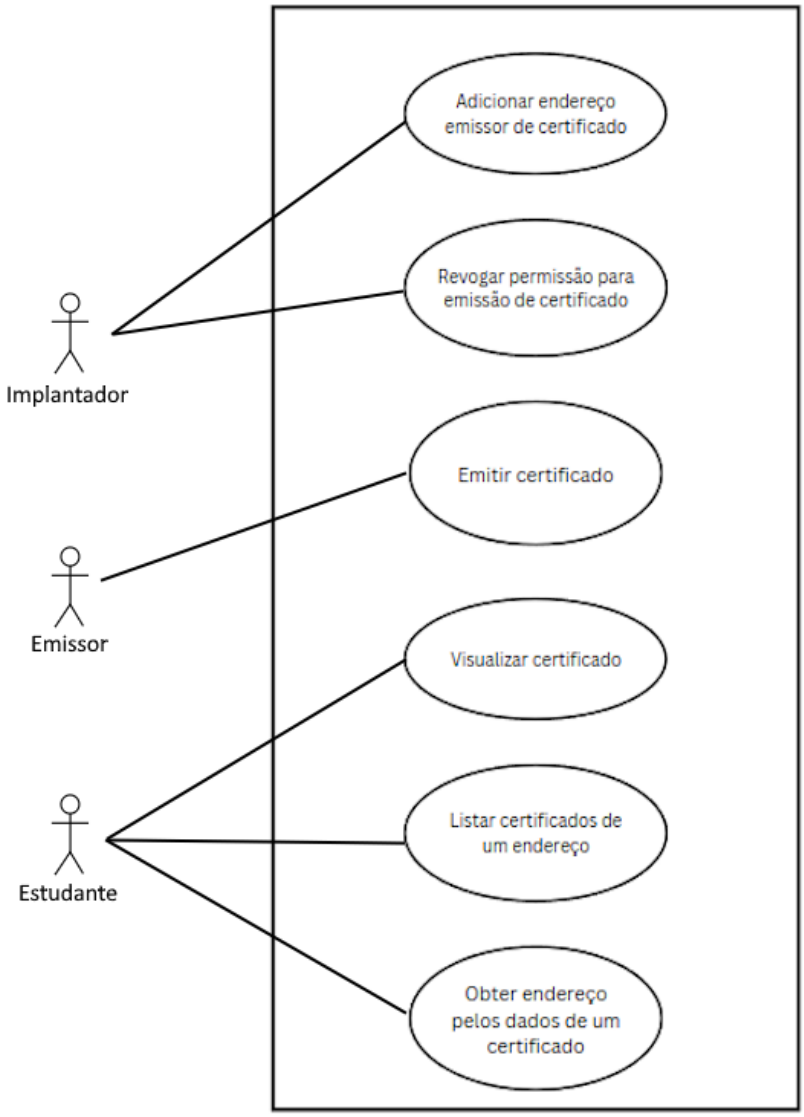


**Figura 2:** Diagrama de casos de uso para o sistema de emissão de certificados.

### 4.5.1 **Caso de Uso:** Adicionar endereço emissor de certificado

- **Objetivo:** permitir que o implantador do contrato adicione um endereço com permissão para emitir certificados.
- **Tipo:** real.
- **Ator:** implantador do contrato.
- **Pré-condição:** o contrato deve estar implantado em rede *blockchain*.
- **Fluxo principal:**

1. O implantador do contrato executa o comando "*grantMinterRole*" do sistema para adicionar um emissor, fornecendo o endereço desejado.
2. O sistema chama a função apropriada do contrato inteligente com as informações enviadas pelo usuário.
3. O sistema envia a transação para a *blockchain*.
4. A *blockchain* processa a transação e registra o endereço como um emissor autorizado.
5. O sistema exibe a confirmação da transação, incluindo seu *hash*, o número do bloco e o *status* da operação.

- **Fluxo alternativo:**

1. Alternativa ao passo 1 - um endereço que não é o do implantador tenta executar o comando no sistema para adicionar um emissor:
   - 1.1: O contrato verifica o papel do endereço que realizou a chamada, utilizando o controle de acesso.
   - 1.2: O contrato detecta que o endereço não possui o papel de implantador do contrato.
   - 1.3: A transação é revertida, emitindo um erro que informa a ausência da permissão necessária.
   - 1.4: O sistema exibe uma mensagem indicando falha na operação.

- **Pós-condições:**
  - O endereço está adicionado no contrato como emissor de certificados.

#### 4.5.2 **Caso de Uso:** Revogar permissão para emissão de certificado

- **Objetivo:** permitir ao implantador do contrato a remoção do papel de um endereço para emitir certificados.
- **Tipo:** real.
- **Ator:** implantador do contrato.
- **Pré-condições:**

- O contrato deve estar implantado em rede *blockchain*;
- O endereço que terá a permissão revogada deve estar registrado no contrato como um emissor.

- **Fluxo principal:**

1. O implantador do contrato executa o comando "*revokeMinterRole*" no sistema para revogar a permissão de um emissor, fornecendo o endereço a ser revogado.
2. O sistema chama a função apropriada do contrato inteligente com as informações enviados pelo implantador.
3. O sistema envia a transação para a *blockchain*.
4. A *blockchain* processa a transação e remove a permissão do endereço.
5. O sistema exibe a confirmação da transação, incluindo seu *hash*, o número do bloco e o *status* da operação.

- **Fluxo alternativo:**

1. Alternativa ao passo 4 - o endereço fornecido não é emissor
   - 4.1: O contrato verifica o papel do endereço cuja permissão de emissor será revogada.
   - 4.2: O contrato verifica que o endereço fornecido não tem permissão para emitir certificados.
   - 4.3: A operação é revertida, emitindo um erro.
   - 4.4: O sistema exibe uma mensagem indicando falha na operação.
2. Alternativa ao passo 1 - um endereço que não é o do implantador tenta executar o comando no sistema para remover a permissão de um emissor:
   - 1.1: O contrato verifica o papel do endereço que realizou a chamada, utilizando o controle de acesso.
   - 1.2: O contrato detecta que o endereço não possui o papel de implantador do contrato.
   - 1.3: A transação é revertida, emitindo um erro que informa a ausência da permissão necessária.
   - 1.4: O sistema exibe uma mensagem indicando falha na operação.

- **Pós-condições:**
  - O endereço não está mais registrado como emissor no contrato inteligente.

#### 4.5.3 **Caso de Uso:** Emitir certificado

- **Objetivo:** registrar um certificado na *blockchain*.
- **Tipo:** essencial.
- **Ator:** usuário cujo endereço possui papel de emissor de certificado.
- **Pré-condições:**
  - O contrato deve estar implantado em rede *blockchain*;
  - Ao menos um endereço com papel de emissor deve existir registrado.
- **Fluxo principal:**
  1. O usuário dono de endereço com papel de emissor executa o comando "*awardCertificate*" no sistema para emitir um certificado, fornecendo o endereço da carteira do estudante, o endereço da imagem no IPFS ou outro metadado apropriado, o nome da instituição, o nome do curso e a matrícula do estudante.
  2. O sistema chama a função apropriada do contrato inteligente com as informações enviados pelo usuário.
  3. O sistema envia a transação para a *blockchain*.
  4. A *blockchain* processa a transação e associa o certificado ao endereço do estudante.
  5. O sistema exibe a confirmação da transação, incluindo seu *hash*, o número do bloco, o *status* da operação e o ID do certificado.
- **Fluxo alternativo:**
  1. Alternativa ao passo 2 - um endereço com formato inválido foi passado como parâmetro:
     - O sistema exibe uma mensagem informando que o endereço é inválido.

2. Alternativa ao passo 1 - um endereço que não tem papel de emissor tenta executar o comando no sistema para emitir um certificado:
   - 1.1: O contrato verifica o papel do endereço que realizou a chamada, utilizando o controle de acesso.
   - 1.2: O contrato detecta que o endereço não possui o papel de emissor de certificado.
   - 1.3: A transação é revertida, emitindo um erro que informa a ausência da permissão necessária.
   - 1.4: O sistema exibe uma mensagem indicando falha na operação.

- **Pós-condições:**
  - O usuário com carteira no endereço fornecido possui um novo certificado.

#### 4.5.4 **Caso de Uso:** Visualizar certificado

- **Objetivo:** permitir que qualquer usuário do sistema consulte os dados de um certificado utilizando o seu número de ID.
- **Tipo:** essencial.
- **Ator:** qualquer usuário.
- **Pré-condição:** o contrato deve estar implantado em rede *blockchain*.
- **Fluxo principal:**
  1. O usuário executa no sistema o comando "*getCertificateData*" e fornece o número de ID do certificado.
  2. O sistema consulta a *blockchain*.
  3. O sistema exibe os dados do certificado.
- **Fluxo alternativo:**
  1. Alternativa ao passo 2 - um certificado com o ID fornecido não existe na *blockchain*:
     - 2.1: O sistema exibe uma mensagem informando que não existe um certificado com o ID informado.

#### 4.5.5 **Caso de uso:** Listar certificados de um endereço

- **Objetivo:** permitir que qualquer usuário visualize todos os certificados associados a um endereço de carteira de estudante.
- **Tipo:** real.
- **Ator:** qualquer usuário.
- **Pré-condição:** o contrato deve estar implantado em rede *blockchain*.
- **Fluxo principal:**
  1. O usuário executa no sistema o comando "*getCertificatesOfStudent*" e fornece o endereço da carteira do estudante.
  2. O sistema consulta a *blockchain*.
  3. O sistema exibe os IDs dos certificados associados ao estudante.
- **Fluxo alternativo:**
  1. Alternativa ao passo 2 - um endereço com formato inválido foi passado como parâmetro:
     - O sistema exibe uma mensagem informando que o endereço é inválido.
  2. Alternativa ao passo 2 - o estudante não possui certificados na *blockchain*:
     - O sistema exibe uma mensagem informando que o estudante não possui certificados associados.

#### 4.5.6 **Caso de uso:** Obter endereço pelos dados de um certificado

- **Objetivo:** permitir a busca por um endereço de carteira de estudante com base nos dados da instituição, curso e número de matrícula.
- **Tipo:** real.
- **Ator:** qualquer usuário.
- **Pré-condição:** o contrato deve estar implantado em rede *blockchain*.
- **Fluxo principal:**

1. O usuário executa no sistema o comando "*getStudentAddressByDetails*" e fornece o nome da instituição, o nome do curso e a matrícula.
2. O sistema consulta a *blockchain*.
3. O sistema exibe o endereço da carteira do estudante solicitado.

- **Fluxo alternativo:**

  1. Alternativa ao passo 2 - o certificado cujos dados foram fornecidos não existe na *blockchain*:
     - O sistema exibe uma mensagem informando que não existe um certificado com os dados fornecidos.

# 5 Arquitetura do sistema

A arquitetura da solução proposta consiste em dois componentes principais: um contrato inteligente implantado em uma rede *blockchain* compatível com a *Ethereum Virtual Machine* (EVM) e um programa facilitador com funcionalidades de visualização e escrita nesse contrato. Considerando que a interação com *blockchain* não é uma tarefa trivial, o programa para simplificar o gerenciamento dos responsáveis pela emissão dos certificados, bem como o fornecimento e a visualização dos certificados, tem vital importância no uso do contrato inteligente.

## 5.1 Arquitetura

A arquitetura da solução proposta, com os seus componentes, é descrita na Figura 3. As próximas seções apresentam em detalhes as camadas componentes da arquitetura do sistema e suas interações.

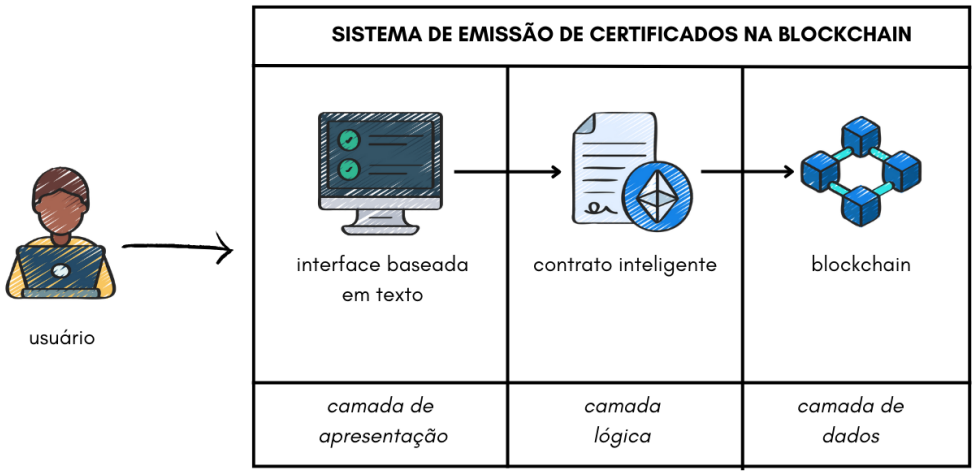


**Figura 3:** Diagrama da arquitetura do sistema.

### 5.1.1 Camada de apresentação

A camada de apresentação é constituída por uma interface baseada em texto, um mecanismo usado para interagir com o programa usando o teclado e desempenha um papel vital

na interação com a *blockchain*, pois faz a intermediação entre os diferentes tipos de usuário e as funções do contrato inteligente. Essa solução foi projetada para mitigar os desafios associados à interação direta com a *blockchain*. Dessa forma, a interface baseada em texto fornece comandos que abstraem as operações subjacentes, permitindo que os usuários executem ações como emissão de certificados, consulta de dados e gerenciamento de permissões de forma mais acessível.

### 5.1.2 Camada lógica - programa facilitador

Caso o usuário tenha permissões de implantador ou de emissão de certificados, antes de usar a interface baseada em texto, é necessário configurar o programa com um arquivo de ambiente que gerencie a chave privada do usuário. Os usuários que não possuem tais permissões não precisam fazer a configuração do programa, pois suas interações são limitadas a somente operações de leitura na *blockchain*.

### 5.1.3 Camada lógica - contrato inteligente

Consiste no contrato inteligente, cujas funções são acessadas e executadas por meio da camada de apresentação. Esse contrato é responsável por realizar as operações fundamentais do sistema, como o gerenciamento de permissões de usuários e a emissão de certificados digitais. Ele também assegura o cumprimento das restrições de acesso e a impossibilidade de transferência de um certificado.

A comunicação entre o contrato implantado na *blockchain* e a camada de apresentação é realizada utilizando a biblioteca *Ethers*[6], que fornece métodos para interagir com redes *blockchain* compatíveis com *Ethereum Virtual Machine*. Tal abordagem abstrai a complexidade técnica da interação direta com a *blockchain*, simplifica a implementação e garante a compatibilidade com diferentes redes públicas e privadas.

### 5.1.4 Camada de dados

A camada de dados é representada pela *blockchain*, responsável por armazenar os certificados digitais emitidos pelo sistema, permitindo que os dados sejam acessados e verificados sempre que necessário.

Dado que a inclusão da imagem digitalizada do certificado diretamente na *block-*

[6]https://docs.ethers.org/v6/

*chain* pode ser ineficiente e dispendiosa devido ao alto custo de armazenamento e às limitações de espaço nas transações de *blockchain*, a abordagem ideal corresponde a armazenar a imagem em outro serviço e gravar apenas uma referência a ela dentro do contrato inteligente. Assim sendo, empregou-se um sistema de armazenamento descentralizado para tal objetivo, a saber, o IPFS.

## 5.2 Modelagem de dados

Neste sistema, cada estudante é representado por uma estrutura que contém informações essenciais para identificar o indivíduo de maneira única, porém sem expor seus dados sensíveis. Essas informações incluem o seu endereço da carteira na *blockchain*, o nome da instituição que emitiu o certificado, o curso realizado pelo estudante e seu número de matrícula. O uso desses dados permite associar um certificado digital de forma intransferível e ainda verificável, sem necessidade de armazenar informações confidenciais, como o CPF, o que garante a privacidade do estudante e a conformidade com a Lei Geral de Proteção de Dados [7].

É possível que um estudante tenha em sua carteira mais de um certificado de diferentes instituições e cursos, uma vez que o contrato inteligente permite que um endereço de carteira seja vinculado a vários certificados, também de instituições diferentes. Isso é possível graças a um mapeamento que armazena uma lista de IDs de certificados para cada endereço de carteira, permitindo a associação de múltiplos certificados a um único estudante. Além disso, de forma a facilitar a verificação de dados, o contrato permite que o endereço de um estudante seja obtido a partir dos dados utilizados para emitir o certificado: instituição, curso e número de matrícula. A partir dessa informação, obtêm-se todos os certificados obtidos pelo estudante responsável pela carteira localizada naquele endereço.

A obtenção dos dados de ambos endereço e certificado é facilitada por quatro mapeamentos: um para vincular o ID do certificado ao estudante, outro para o endereço da carteira do estudante aos IDs de seus certificados, o terceiro para a combinação de instituição, curso e matrícula ao ID do certificado e, por fim, um que faz a ligação entre IDs de certificados e suas informações.

A Figura 4 ilustra o Diagrama de Entidade-Relacionamento para o domínio modelado.

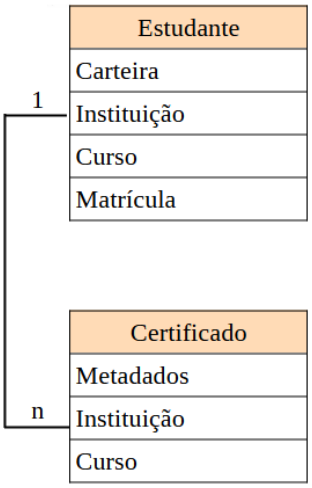


**Figura 4:** Diagrama entidade-relacionamento.

## 5.3 Tecnologias

A escolha pela linguagem JavaScript se justifica pela disponibilidade de uma biblioteca de código aberto para interação com blockchains compatíveis com Ethereum denominada *Ethers* [7]. Além disso, o *Hardhat*[8], um *kit* de código aberto de desenvolvimento voltado para redes compatíveis com *Ethereum* e utilizado para a criação de nós de *blockchain* locais, testes e implantação de contratos, entre outras funcionalidades, é desenvolvido em JavaScript/TypeScript e requer o uso do Node.js[9] para seu funcionamento. Node.js é um software de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web. Ele permite que desenvolvedores criem servidores, aplicações web, ferramentas de linha de comando e scripts. [46]

Adicionalmente, a flexibilidade do JavaScript permite sua utilização tanto no desenvolvimento de back-end quanto de front-end, viabilizando o emprego da mesma biblioteca *Ethers* para a criação de aplicações descentralizadas, seja por meio de interfaces de linha de comando ou de interfaces gráficas, utilizando HTML ou qualquer framework voltado para a criação de páginas. Na implementação realizada, o *token URI* aponta para uma imagem digitalizada do documento, armazenada de forma descentralizada utilizando o IPFS e com o apoio da plataforma *Filebase*[10].

[7] https://github.com/ethers-io/ethers.js
[8] https://github.com/NomicFoundation/hardhat
[9] https://nodejs.org/en
[10] https://filebase.com/

# 6 Interações do Sistema

Neste capítulo, serão apresentadas as telas da interface interativa utilizadas para acessar as funcionalidades do sistema. Ao invés de exigir que os usuários digitem comandos com parâmetros, a interface exibe opções e menus na tela, permitindo que as operações sejam realizadas por meio de seleções. Essa abordagem foi adotada para oferecer uma experiência mais intuitiva e amigável, especialmente para usuários com menor familiaridade com a linha de comando, além de reduzir a necessidade de consulta à documentação durante o uso do programa.

Antes de utilizar o sistema, é necessário configurar um arquivo de variáveis de ambiente. Nesse arquivo, o usuário deve incluir a URL da *Remote Procedure Call* (RPC), que serve como interface para realizar chamadas de funções ou executar comandos em um nó da *blockchain*. Também é necessário adicionar a chave privada do endereço que irá fazer as chamadas de função — um parâmetro opcional, que é utilizado apenas para endereços com permissão para emitir certificados e para o implantador do contrato, além do endereço do contrato na *blockchain*. Na sua inicialização, o sistema exibe um lembrete a fim der orientar o preenchimento correto dessas variáveis de ambiente.

Após essa configuração, o sistema pode ser iniciado. A interface exibe então as opções disponíveis para o usuário escolher, tal como mostrado na Figura 5, o que reduz significativamente a chance de seleção de opções inválidas e melhora a navegação.

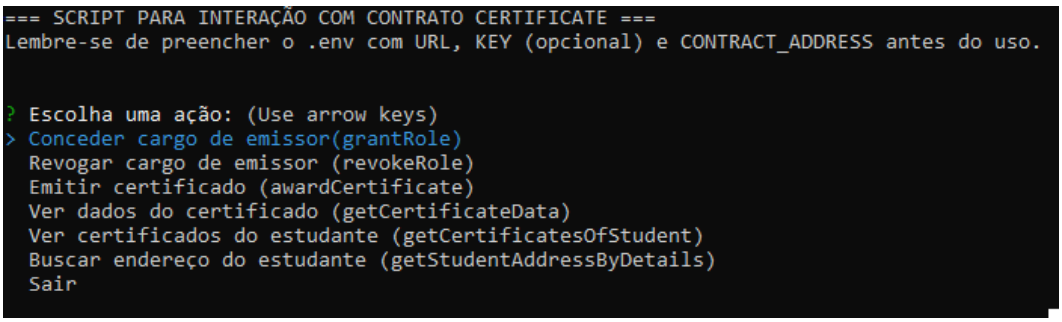


**Figura 5:** Apresentação do sistema.

É importante notar que, nas telas de demonstração do sistema a serem apresentadas a seguir, o número do bloco sempre aparecerá como "*null*". Isso se dá pelo fato de que a demonstração foi feita em um nó local de testes fornecido pelo *Hardhat*, que gera blocos novos conforme transações são requisitadas ao invés de fazer isso em um intervalo de tempo

especificado por um arquivo gênesis.

A primeira opção exibida na Figura 5 permite conceder a permissão de emissão de certificados a um endereço. Ao escolher essa opção, o sistema solicitará o endereço desejado. Se a transação for bem-sucedida, serão exibidos o número do bloco em que a transação foi registrada, o *hash* da transação e uma mensagem de confirmação, conforme ilustrado na Figura 6.

```
? Escolha uma ação: Conceder cargo de emissor(grantRole)
? Digite o endereço: 0x90F79bf6EB2c4f870365E785982E1f101E93b906
Sucesso!
Block number: null
Hash da transação: 0xe112f7b7e3982f8fcd0eb8aed618b826eed8b24ac0b030a3c361f338236e4653
Cargo concedido com sucesso a 0x90F79bf6EB2c4f870365E785982E1f101E93b906
```

**Figura 6:** Interação: fornecendo permissão de emissão de certificados a um endereço.

O fluxo de interação da funcionalidade de revogar permissão de emissão de um endereço é semelhante ao de conceder permissão de emissão, porém com a diferença de que, neste caso, a permissão atribuída ao endereço será removida. O sistema solicita a mesma informação, isto é, o endereço do usuário para a revogação, e, caso a transação seja realizada com sucesso, o sistema retornará informações semelhantes às fornecidas na concessão da permissão: o número do bloco em que a transação foi registrada, o *hash* da transação e uma mensagem de confirmação, indicando que a permissão foi revogada com êxito.

```
? Escolha uma ação: Revogar cargo de emissor (revokeRole)
? Digite o endereço: 0x90F79bf6EB2c4f870365E785982E1f101E93b906
Sucesso!
Block number: null
Hash da transação: 0xc208e5811f76d9c412e14507c2e0575df336622514dd56b0893185bd6ada3cf8
Cargo revogado com sucesso de 0x90F79bf6EB2c4f870365E785982E1f101E93b906
```

**Figura 7:** Interação: removendo permissão de emissão de certificados de um endereço.

Para emitir um certificado, confome mostrado na Figura 8, o sistema solicita o endereço do estudante, o *token* URI — uma URL que aponta para os metadados e recursos associados ao NFT, como imagens, descrições e outras informações que definem o ativo digital —, além do nome da instituição, do curso e do número de matrícula do estudante. Essas informações são necessárias para permitir a obtenção do certificado de um estudante por meio de seu endereço de carteira, ou para localizar o endereço de carteira a partir dos dados do certificado, funcionalidades que serão detalhadas ainda neste capítulo.

A Figura 9 ilustra a execução da opção “Ver dados do certificado” que permite que um usuário acesse informações detalhadas sobre o certificado, como o nome da instituição que emitiu o documento, o curso relacionado e o *token* URI do certificado.


**Figura 8:** Interação: emitindo certificado a um estudante.

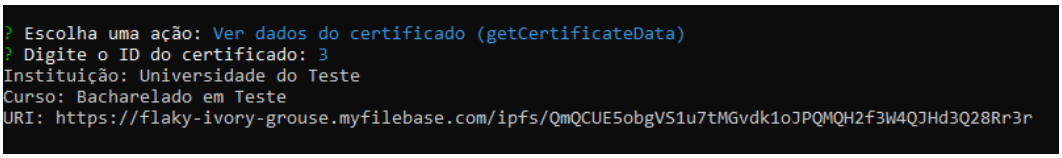


**Figura 9:** Interação: obtendo os dados do certificado de um usuário.

Já a opção "Ver certificados do estudante" exibe os IDs de todos os certificados associados a um determinado estudante. No exemplo apresentado na Figura 10, o estudante com o endereço indicado possui somente o certificado de ID 3, o qual corresponde ao certificado emitido anteriormente e ilustrado na Figura 8.

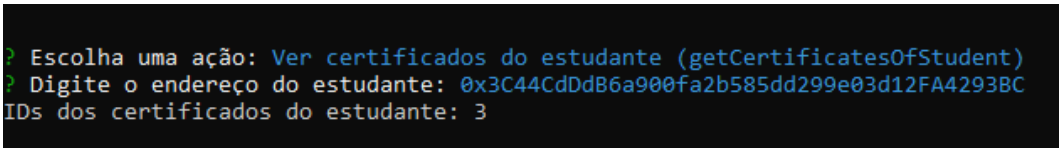


**Figura 10:** Interação: obtendo os IDs dos certificados associados a um estudante.

Por fim, a opção "Buscar endereço do estudante" permite obter o endereço da carteira de um estudante que possui determinado certificado, a partir das seguintes informações associadas: instituição, nome do curso e número de matrícula. A Figura 11 exibe um exemplo de utilização dessa opção.

Dessa forma, os indivíduos podem recuperar seus certificados e validá-los mesmo sem ter acesso ao endereço de suas carteiras, sendo suficiente apenas informar a instituição, o curso e o número de matrícula. Considerando que o endereço de carteira também pode ser obtido a partir da chave privada, essas informações constituem uma forma mais fácil de permitir o acesso de um estudante a seus certificados. Por outro lado, para que outras partes interessadas possam validar os certificados, é necessário que tenham conhecimento dos IDs desses certificados e/ou de outras informações relacionadas, já que a divulgação da chave privada é um risco significativo.

Caso alguma falha ocorra durante uma transação, o sistema exibirá uma mensagem de aviso, tal como mostrado na Figura 13, e retornará ao menu principal. A mensagem de aviso

é a mesma para todos os tipos de falha, exceto no caso de um endereço de contrato inválido, que tem uma mensagem específica alertando sobre essa situação.

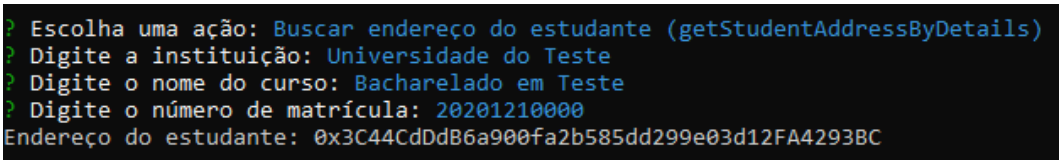


**Figura 11:** Interação: obtendo o endereço de um estudante a partir das informações associadas a um certificado.

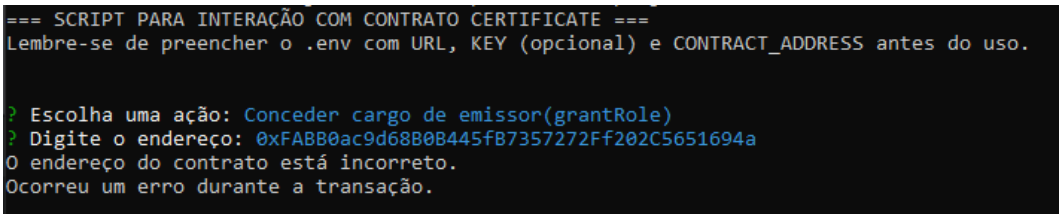


**Figura 12:** Interação: ocorrência de falha em uma transação devido a endereço de contrato incorreto.

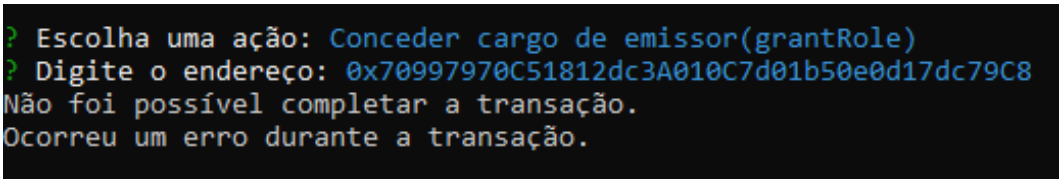


**Figura 13:** Interação: ocorrência de falha em uma transação e mensagem apresentada pelo sistema.

# 7 Conclusão

## 7.1 Considerações finais

O objetivo deste trabalho foi o desenvolvimento de uma ferramenta para a emissão e verificação da autenticidade de certificados acadêmicos, aproveitando duas das principais características dos sistemas baseados em *blockchain*: imutabilidade e transparência. O sistema proposto depende exclusivamente da confiança na instituição emissora do certificado, e a integridade dos dados é garantida pela natureza descentralizada e irreversível da tecnologia. Além disso, ao registrar na *blockchain* uma referência à imagem digitalizada do certificado, é possível que o titular do certificado ou qualquer outra pessoa interessada acesse ou recupere essa informação de forma segura, mitigando as preocupações relacionadas à perda de certificados físicos e ao risco de fraudes.

Todo o código desenvolvido neste trabalho está disponibilizado em um repositório na plataforma *GitHub*[11], acompanhado de instruções com linguagem acessível para instalação e utilização. O código desenvolvido, associado a este documento produzido, podem servir de introdução aos temas de *blockchain* e contratos inteligentes a alunos de graduação.

Caso a plataforma proposta seja implantada de fato, para que ela funcione de maneira plena, é desejável que as universidades de todo o país se conectem a uma única rede *blockchain*, operando nós sob o mecanismo de consenso *Proof of Authority*. Essa abordagem garantiria um ambiente seguro, eficiente e transparente. Nesse contexto, a existência da Rede Blockchain Brasil [8] surge como uma solução facilitadora, fornecendo a infraestrutura necessária para a implantação da tecnologia. Com isso, as universidades precisariam apenas aderir à rede já estabelecida, sem a necessidade de desenvolver ou manter uma infraestrutura própria, simplificando o processo de integração e reduzindo custos operacionais.

## 7.2 Limitações do estudo

Embora a segurança tenha importância vital para o pleno funcionamento do sistema proposto, este trabalho não abordou questões relacionadas de forma detalhada, uma vez que não foi a intenção realizar um estudo de caso. Além disso, a análise de aspectos como esca-

[11] https://github.com/Lionel-Rocha/certificados-blockchain

labilidade, disponibilidade ideal do serviço e a capacidade da rede para suportar um grande número de usuários simultâneos foi intencionalmente omitida, uma vez que esses são problemas de infraestrutura que transcendem o escopo deste trabalho. A infraestrutura necessária para garantir a robustez e o desempenho do sistema, em termos de recursos computacionais, capacidade de processamento e capacidade de rede, deve ser tratada separadamente, focando em aspectos técnicos e operacionais da rede *blockchain* a ser utilizada. Dessa forma, o foco deste trabalho esteve apenas na implementação do contrato inteligente e nos aspectos relacionados às suas funcionalidades, deixando as questões de infraestrutura para futuros estudos de caso práticos.

## 7.3 Trabalhos futuros

Qualquer pessoa interessada poderá criar uma interface gráfica para a adaptação das funcionalidades do sistema, com base na sua flexibilidade e na documentação fornecida. Como trabalho futuro, vislumbra-se o desenvolvimento de outras funcionalidades vistas na revisão da literatura, como validação de certificado por código *QR* e adição de um campo para notas. Além disso, pretende-se disponibilizar uma versão do contrato em Rust para redes que não sejam compatíveis com a *Ethereum Virtual Machine*, que adotam *WebAssembly* (WASM) como ambiente de execução. É interessante destacar que essa abordagem visa ampliar o alcance da aplicação, permitindo sua integração com uma gama mais ampla de plataformas *blockchain*.

# Referências


[1] Erc-721 non-fungible token standard, oct 2020. Atualizado pela última vez em November 19, 2023.

[2] Stefano De Angelis, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, and Vladimiro Sassone. Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain. In Elena Ferrari, Marco Baldi, and Roberto Baldoni, editors, *ITASEC*, volume 2058 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.

[3] Bakri Awaji and Ellis Solaiman. Design, implementation, and evaluation of blockchain-based trusted achievement record system for students in higher education, 2022.

[4] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Consensus in the age of blockchains, 2017.

[5] Hyperledger Besu. Cli options - hyperledger besu documentation, 2023. Acessado em: 3 mar. 2025.

[6] BRASIL. Medida Provisória nº 2.200-2, de 24 de agosto de 2001. Institui a Infraestrutura de Chaves Públicas Brasileira - ICP-Brasil e dá outras providências, 2001. Acesso em: 6 nov. 2024.

[7] BRASIL. Lei nº 13.709, de 14 de agosto de 2018, August 2018. Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965, de 23 de abril de 2014 (Marco Civil da Internet). Diário Oficial da União: seção 1, Brasília, DF, 15 ago. 2018.

[8] BRASIL. Extrato do Acordo de Cooperação nº D-121.2.0014.22, April 2022. Diário Oficial da União: seção 1, Brasília, DF, 18 abr. 2022.

[9] BRASIL. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). Censo da Educação Superior 2023, 2023.

[10] V. Buterin. Ethereum white paper: A next generation smart contract & decentralized application platform. `https://ethereum.org/en/whitepaper/`, 2014. Acesso em: 20 out. 2024.

[11] Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen. Blockchain and smart contract for digital certificate. In *2018 IEEE International Conference on Applied System Invention (ICASI)*, pages 1046–1051, 2018.

[12] Aastha Chowdhary, Shubham Agrawal, and Bhawana Rudra. Blockchain based framework for student identity and educational certificate verification. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 916–921, 2021.

[13] Mohammad Jabed Morshed Chowdhury, Alan Colman, Muhammad Ashad Kabir, Jun Han, and Paul Sarda. Blockchain versus database: A critical analysis. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1348–1353, 2018.

[14] Hanting Chu, Pengcheng Zhang, Hai Dong, Yan Xiao, Shunhui Ji, and Wenrui Li. A survey on smart contract vulnerabilities: Data sources, detection and repair. *Information and Software Technology*, 159:107221, 2023.

[15] Rostand Costa, Daniel Faustino, Guido Lemos, Ademir Queiroga, Cláudio Djohnnatha, Felipe Alves, Jordan Lira, and Mateus Pires. Uso não financeiro de blockchain: Um estudo de caso sobre o registro, autenticação e preservação de documentos digitais acadêmicos. In *Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações*, Porto Alegre, RS, Brasil, 2018. SBC.

[16] M. Crosby, P. Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman. Blockchain technology: Beyond bitcoin. *Applied Innovation Review*, (2):6–19, 2016.

[17] Gabriel Ferreira Gomes da Silva. Totalização de votos das urnas eletrônicas em uma rede blockchain, 2023.

[18] Diptendu Das. All about blockchain databases: Immutabledb, ledgerdb, nodedb, rocksdb, nudb, couchdb. `https://diptendud.medium.com/all-about-blockchain-databases-immutabledb-ledgerdb-nodedb-rocksdb-` 2023. Accessed: 2025-03-10.

[19] Tribunal de Contas da União. Levantamento da tecnologia blockchain. `https://portal.tcu.gov.br/data/files/59/02/40/6E/`

C4854710A7AE4547E18818A8/Blockchain_sumario_executivo.pdf, 2020. Acesso em: 21 nov. 2024.

[20] Tien Tuan Anh Dinh, Ji Wang, Gang Chen, Rui Liu, Beng Chin Ooi, and Kian-Lee Tan. Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*, pages 1085–1100, New York, 2017. Association for Computing Machinery.

[21] H. H. N. dos Santos and M. P. Bueno. Blockchain: Tecnologia sustentável na administração pública municipal / blockchain: Sustainable technology in municipal public administration. *Brazilian Applied Science Review*, 5(1):498–521, 2021.

[22] William Entriken, Dieter Shirley, Jacob Evans, and Nastassia Sachs. Eip-721: Erc-721 non-fungible token standard, jan 2018. [Online serial].

[23] S. Erdogan, M.Y. Ahmed, and S.A. Sarkodie. Analyzing asymmetric effects of cryptocurrency demand on environmental sustainability. *Environmental Science and Pollution Research*, 29:31723–31733, 2022.

[24] Ethereum Foundation. *Ethereum Virtual Machine (EVM)*, January 2025. Última edição em 20 de janeiro de 2025. Criado em 22 de setembro de 2020.

[25] Fantástico. Quatro pessoas são presas pela venda de 50 mil diplomas falsos e milhares carteirinhas de estudante. https://g1.globo.com/fantastico/noticia/2025/02/02/quatro-pessoas-sao-presas-pela-venda-de-50-mil-diplomas-falsos-e-mighthtml, 2025.

[26] Ethereum Foundation. Ethereum energy consumption. Ethereum Foundation Website, 2020. Page last updated: September 3, 2023.

[27] Siddharth Gangwar. Leveldb: Invented by google and used by cryptocurrencies like bitcoin. https://medium.com/coinmonks/leveldb-invented-by-google-and-used-by-cryptocurrencies-like-bitcoi 2022. Accessed: 2025-03-10.

[28] Juan Garay and Aggelos Kiayias. Sok: A consensus taxonomy in the blockchain era. In Stanislaw Jarecki, editor, *Topics in Cryptology – CT-RSA 2020*, pages 284–318, Cham, 2020. Springer International Publishing.

[29] GeeksforGeeks. Blockchain Structure, 2024. Última atualização em 29 de agosto de 2024. Acesso em: 17 fev. 2025.

[30] Fernanda O. Gomes, Bruno M. Agostinho, Julia Baldisera, Raphael S. Silveira, and Jean E. Martina. Detecção de fraudes na emissão de certificados digitais dentro da Infraestrutura de Chaves Públicas Brasileira, 2020.

[31] Laura Grassi, Davide Lanfranchi, Alessandro Faes, and Filippo Maria Renga. Do we still need financial intermediation? the case of decentralized finance – defi. *ArXiv*, 2021.

[32] Henrique Coelho, G1 Rio. Polícia do rio faz operação de combate à emissão de diplomas escolares falsos. `https://g1.globo.com/rj/rio-de-janeiro/noticia/2018/09/24/policia-do-rio-faz-operacao-de-combate-a-emissao-de-diplomas-escolaghtml`, 2018.

[33] Yining Hu, Madhusanka Liyanage, Ahsan Mansoor, Kanchana Thilakarathna, Guillaume Jourjon, and Aruna Seneviratne. Blockchain-based smart contracts - applications and challenges, 2019.

[34] Hyperledger Besu. Revert reason. `https://besu.hyperledger.org/private-networks/how-to/send-transactions/revert-reason`, 2024. Last updated on Sep 17, 2024.

[35] Mubashar Iqbal and Raimundas Matulevičius. Exploring sybil and double-spending risks in blockchain systems. *IEEE Access*, 9:76153–76177, 2021.

[36] Juho Lindman, Jamie Berryhill, Benjamin Welby, and Mariane Piccinin Barbieri. The uncertain promise of blockchain for government, 2020. OECD Working Papers on Public Governance.

[37] D. Llessie, M. Sobolewski, L. Vaccari, and F. Pignatelli. *Blockchain for Digital Government*, pages 31–34. EUR 29677 EN. Publications Office of the European Union, Luxembourg, 2019. JRC115049.

[38] D. Llessie, M. Sobolewski, L. Vaccari, and F. Pignatelli. *Blockchain for Digital Government*, pages 42–45. EUR 29677 EN. Publications Office of the European Union, Luxembourg, 2019. JRC115049.

[39] Frank Mangone. Blockchain 101: Storage. `https://medium.com/@francomangone18/blockchain-101-storage-12d84d0e2c8e`, 2024. Accessed: 2025-03-10.

[40] Massachusetts Institute of Technology (MIT). Digital diplomas, 2022. Acesso em: 16 fev. 2025.

[41] Muhammad Mehar, Charlie Shier, Alana Giambattista, Elgar Gong, Gabrielle Fletcher, Ryan Sanayhie, Henry M. Kim, and Marek Laskowski. Understanding a revolutionary and flawed grand experiment in blockchain: The dao attack. *Journal of Cases on Information Technology*, 21(1):19–32, 2017. Disponível em SSRN: `https://ssrn.com/abstract=3014782` ou `http://dx.doi.org/10.2139/ssrn.3014782`.

[42] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *Advances in Cryptology — CRYPTO '87*, pages 369–378, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

[43] Ziaul Haque Munim, Okan Duru, and Eiji Hirata. Rise, Fall, and Recovery of Blockchains in the Maritime Technology Space. *Journal of Marine Science and Engineering*, 9(3):266, 2021.

[44] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System, 2008. Disponível em: https://bitcoin.org/bitcoin.pdf. Acesso em: 6 nov. 2024.

[45] National Institute of Standards and Technology (NIST). Federal information processing standards publication 180-2: Announcing the secure hash standard. Publicação oficial, 2002. Disponível em: `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-2.pdf`.

[46] OpenJS Foundation. Nodejs, 2009. Acesso em 02 de março de 2025.

[47] Polícia Federal. Pf deflagra operação contra emissão de diplomas falsos. https://www.gov.br/pf/pt-br/assuntos/noticias/2024/02/pf-deflagra-operacao-contra-emissao-de-diplomas-falsos, 2024.

[48] Xiaofang Qiu, Zhi Qin, Wunan Wan, Jinquan Zhang, Jinliang Guo, Shibin Zhang, and Jinyue Xia. A dynamic reputation–based consensus mechanism for blockchain. *Computers, Materials & Continua*, 73(2):2577–2589, 2022.

[49] Untung Rahardja, Achmad Nizar Hidayanto, Panca Oktavia Hadi Putra, and Marviola Hardini. Immutable ubiquitous digital certificate authentication using blockchain protocol. *Journal of Applied Research and Technology*, 19(4):308–321, 2021. Epub 14-Feb-2022. ISSN 2448-6736.

[50] Saha Reno, Mamun Ahmed, Saima Ahmed Jui, and Shamma Dilshad. Securing certificate management system using hyperledger based private blockchain. In *2022 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, pages 46–51, 2022.

[51] Lucas de Souza Ribeiro. Descentralized Student National Identification: A Blockchain Approach, 2019. Orientador: Pedro Nuno de Souza Moura.

[52] Emerson Souza, Elisângela Carneiro, and Antonio Coutinho. Geração e validação de diplomas e certificados utilizando blockchain pública. In *Anais do IV Workshop em Blockchain: Teoria, Tecnologias e Aplicações*, pages 54–59, Porto Alegre, RS, Brasil, 2021. SBC.

[53] M. Swan. *Blockchain: Blueprint for a New Economy*. O'Reilly Media, 2015.

[54] Nick Szabo. Smart contracts: building blocks for digital markets. https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html, 1996. Acesso em: 30 out. 2024.

[55] University of Nicosia (UNIC). Blockchain-based certifications, 2024. Acesso em: 16 fev. 2025.

[56] Noeli Vaz, Matheus Martins, Gislainy Velasco, and Sergio Carvalho. Uma arquitetura para aplicações mhealth descentralizadas baseadas em blockchain. In *Anais do VII Workshop em Blockchain: Teoria, Tecnologias e Aplicações*, pages 134–146, Porto Alegre, RS, Brasil, 2024. SBC.

[57] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. Blockchain Technology Overview, 2018. Acesso em: 10 jan. 2025.

[58] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *2017 IEEE International Congress on Big Data (BigData Congress)*, 2017.