UNIVERSIDADE DO SUL DE SANTA CATARINA

LUCAS PASIN

PAULO CORREA MARIA

DESENVOLVIMENTO DE UM SOFTWARE EM BLOCKCHAIN: INTEGRAÇÃO E APLICAÇÕES PRÁTICAS

Florianópolis

2023

Lucas Pasin

Paulo Correa Maria

DESENVOLVIMENTO DE UM SOFTWARE EM BLOCKCHAIN: INTEGRAÇÃO E APLICAÇÕES PRÁTICAS

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade do Sul de Santa Catarina como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Flávio Ceci.

Florianópolis

2023

Lucas Pasin

Paulo Correa Maria

DESENVOLVIMENTO DE UM SOFTWARE EM BLOCKCHAIN: INTEGRAÇÃO E APLICAÇÕES PRÁTICAS

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Sistemas de Informação e aprovado em sua forma final pelo Curso de Sistemas de Informação da Universidade do Sul de Santa Catarina.

Florianópolis, 12 de Junho de 2023.

_____

Professor e orientador Flávio Ceci, Dr.
Universidade do Sul de Santa Catarina

_____

Prof. Nome do Professor, Dr./Ms./Bel./Lic
Universidade...

_____

Prof. Nome do Professor, Dr./Ms./Bel./Lic
Universidade do Sul de Santa Catarina

“A tecnologia blockchain é como uma pedra fundamental para a construção de um futuro confiável, transparente e descentralizado” (Autor desconhecido).

RESUMO

Com o exponencial avanço tecnológico decorrido das últimas tecnologias, a sociedade e as empresas estão sendo forçadas a rever as formas como as interações financeiras e sociais ocorrem no mundo digital. Os Sistemas descentralizados vêm se destacando e se tornando fundamentais no desenvolvimento dessa nova etapa. E nesse sentido, a busca por reformular os processos que antes nos davam as garantias necessárias para a convivência em um ambiente tecnológico tem se intensificado nos últimos tempos, sendo necessário a utilização de ferramentas de auxílio a fim de se obter maior segurança e independência nos acordos mútuos entre duas ou mais partes. O presente trabalho tem como objetivo desenvolver um protótipo funcional de um sistema em blockchain para a criação de contratos inteligentes. O conhecimento teórico foi obtido por meio de pesquisa científica caracterizada como bibliográfica, com abordagem qualitativa, tendo como objetivo o método exploratório. Após o desenvolvimento do protótipo funcional, o mesmo foi avaliado por usuários, tendo uma avaliação positiva e, apesar de possuir funcionalidades simples, o sistema se mostrou eficiente em demonstrar o potencial do uso dessas tecnologias na sociedade.

Palavras-chave: Blockchain. Contratos inteligentes. Web 3.0.

ABSTRACT

With the exponential technological advancements in recent years, society and companies are being forced to reconsider how financial and social interactions take place in the digital world. Decentralized systems have been gaining prominence and have become essential in this new phase. In this regard, there has been an intensified effort to reformulate the processes that previously provided the necessary guarantees for coexistence in a technological environment. The use of auxiliary tools has become necessary to achieve greater security and independence in mutual agreements between two or more parties. The aim of this work is to develop a functional prototype of a blockchain-based system for smart contract creation. The theoretical knowledge was obtained through scientific research characterized as bibliographic, using a qualitative approach with an exploratory method. After the development of the functional prototype, it was evaluated by users, receiving positive feedback. Despite having simple functionalities, the system proved to be efficient in demonstrating the potential of using these technologies in society.

Keywords: Blockchain. Smart contracts Web 3.0.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO:

Em 2008 um usuário com o pseudônimo de Satoshi Nakamoto publicou um artigo acadêmico chamado **Bitcoin: um sistema financeiro eletrônico peer-to-peer**, essa foi a primeira aparição da blockchain, que se tornaria anos depois de sua publicação a tecnologia mais promissora das próximas décadas. A blockchain é uma base de dados distribuída composta por registro de transações ou eventos digitais que foram executadas ou compartilhadas entre os participantes, cada uma dessas transações é validada e aprovada pela maioria dos participantes (HUAIQING et al, 2016).

> E se não tivesse apenas uma internet de informação, tivéssemos também uma internet de valor? Um tipo de livro razão vasto, global e distribuído, rodando em milhões de computadores e disponível para todo mundo, onde um todos os tipos de ativos, desde dinheiro até música pudesse ser armazenado, negociado, trocado ou gerido, tudo sem intermediários poderosos (TAPSCOTT, 2016).

Como descrito por Don Tapscott as possibilidades da utilização da blockchain são colossais, podendo influenciar desde transações bancárias até direitos de músicas e artes. Vários projetos vêm sendo implementados desde 2008 com a intenção de utilizar a tecnologia do blockchain para eliminar a necessidade de intermediários e aumentar a confiabilidade do sistema, deixando o sistema mais otimizado e simples, ao mesmo tempo em que cria camadas de segurança mais robustas.

Com a chegada desta nova tecnologia foram também desenvolvidas novas tecnologias com base na blockchain e sua teoria, um exemplo de tecnologia baseado nesses fundamentos são as criptomoedas, que são uma moeda digital conectada a uma rede peer-to-peer que pode ser usado como pagamento de forma totalmente descentralizada usando para isso uma rede blockchain onde são registradas as transações (ULRICH, 2014).

Outra Aplicação são os Tokens não fungíveis, mais conhecidos pela sua abreviação NFT. Segundo Sam Dean, uma NFT é um certificado de autenticidade sobre um objeto virtual ou real, esse certificado é armazenado em uma rede blockchain garantindo a confiabilidade da veracidade do mesmo e permitindo ser usado como uma forma de cartório descentralizado.

Como descrito anteriormente esta monografia tem como propósito disseminar o conhecimento sobre a tecnologia da blockchain e suas aplicações, demonstrando os conceitos técnicos e históricos de usos da blockchain demonstrando suas funcionalidades e capacidades perante a sociedade atual.

1.1 PROBLEMÁTICA

O termo blockchain está se tornando cada vez mais comum, e muitas empresas estão buscando aproveitar esse hype. No entanto, ainda existem várias dúvidas que não estão claras para o público em geral. Por exemplo: como e onde essa tecnologia pode ser utilizada, quais são os benefícios e malefícios, como funciona a mineração, o que garante a segurança e qual é o nível de conhecimento necessário para compreender esses conceitos. Diante disso, surge a pergunta:

É possível contribuir para a divulgação do conhecimento sobre blockchain, apresentando de forma acessível as possibilidades de utilização para contratos inteligentes, assim como os desafios e avanços nas aplicações baseadas nessa tecnologia?

1.2 OBJETIVOS

Tendo em vista o tema proposto, foram definidos o objetivo geral e os específicos a seguir.

1.2.1 Objetivo Geral

Desenvolver uma aplicação baseada em blockchain, com enfoque na área de contratos inteligentes, a fim de demonstrar o funcionamento da ferramenta estudada ao longo do desenvolvimento desta monografia.

1.2.2 Objetivos específicos

- Apresentar o funcionamento da tecnologia.
- Descrever os pontos positivos e negativos da tecnologia.

- Demonstrar aplicações da tecnologia na área de criptomoedas.
- Apresentar aplicações da tecnologia para impacto social.
- Apresentar aplicações culturais e de entretenimento da tecnologia.
- Expor as inovações e possibilidades futuras da blockchain e seus derivados.
- Demonstrar a criação e funcionamento na prática.

1.3 JUSTIFICATIVA

Ao aprofundarmos nosso conhecimento nas diversas aplicações baseadas em blockchain, conseguimos ampliar a visibilidade e compreensão do uso dessa tecnologia. Esta monografia tem como objetivo proporcionar uma reflexão a respeito dos possíveis benefícios e desvantagens que a aplicação dessa tecnologia pode trazer.

Embora muitos cientistas e especialistas nessa área sejam altamente especializados, é importante discutir os desafios que a blockchain apresenta de uma forma mais ampla e aplicada, abrangendo diversos setores da sociedade. Ao divulgar os conhecimentos e avanços nessa área para um público leigo no assunto, promovemos a inclusão desses participantes nos debates e discussões necessários para o avanço contínuo do campo.

Por meio dessa inclusão, permitimos que indivíduos de diferentes áreas de conhecimento compreendam o potencial da tecnologia blockchain, suas aplicações práticas e as transformações que podem ser alcançadas em diferentes setores, como finanças, cadeia de suprimentos, saúde e muito mais. Além disso, essa disseminação de conhecimento possibilita uma participação mais ativa desses indivíduos nos processos de tomada de decisão e contribui para a criação de soluções inovadoras e inclusivas.

Portanto, ao compartilhar os conhecimentos e avanços da área de blockchain com um público mais amplo e diversificado, estamos impulsionando o desenvolvimento e a aplicação dessa tecnologia revolucionária, incentivando a colaboração entre especialistas e leigos, e promovendo uma sociedade mais informada e participativa no que diz respeito às possibilidades e desafios da blockchain.

1.4 ESTRUTURA DA MONOGRAFIA

Esta monografia está dividida em capítulos, conforme especificado:

- Capítulo 1: Neste capítulo é apresentada a introdução ao tema abordado, problemática, objetivos e justificativa do trabalho.
- Capítulo 2: Apresenta os resultados da pesquisa bibliográfica referente a definição, histórico e tecnologias da blockchain e suas tecnologias derivadas.
- Capítulo 3: O terceiro capítulo apresenta o método de pesquisa utilizado para o desenvolvimento do presente trabalho.
- Capítulo 4: Nesta etapa estudamos alguns casos já existentes da aplicação da blockchain.
- Capítulo 5: Desenvolvemos e demonstramos a funcionalidade da blockchain usando como base os cases citados no capítulo anterior.
- Capítulo 6: O último capítulo do trabalho apresenta a conclusão.

2 REVISÃO DA LITERATURA

Este capítulo aborda de forma abrangente os conceitos fundamentais da tecnologia blockchain e suas diversas aplicações derivadas. O objetivo principal deste trabalho é fornecer uma descrição detalhada do funcionamento dessas ferramentas baseadas em descentralização, assim como explorar os conceitos que permeiam seu funcionamento.

2.1 BLOCKCHAIN

É amplamente aceito que a primeira implementação da tecnologia blockchain nos dias atuais veio de Satoshi Nakamoto.(Lewis Popovski and George Soussou, 2018)

Em meados de 2008 uma pessoa ou grupo identificado como Satoshi Nakamoto publicou um artigo nomeado: “Bitcoin: A Peer-to-Peer Electronic Cash System” que hipotetiza criação de um pagamento online direto entre pessoas sem a necessidade de intermediários para prover confiança entre as partes. O artigo propõe um sistema baseado em prova criptográfica em vez de confiança.

2.1.1 O que é a blockchain?

Para responder esta pergunta, observa-se os elementos que a compõe:

2.1.1.1 O que é um bloco?

O bloco é formado por quatro partes principais, a primeira é o conteúdo, alguma informação, que pode ser uma palavra, texto ou documento digital, cada bloco deve conter também uma chave (também chamado de hash), formado por um processo chamado Hash/Digest, que é um identificador único que forma uma impressão digital do bloco em relação à rede, permitindo que qualquer um com o identificador possa encontrá-lo. As outras duas partes do bloco são: o hash ou chave do bloco anterior (permitindo a inserção do bloco na rede e realizando o encadeamento dos registros) e o horário de criação do bloco (essa informação é

guardada no formato timestamp com data e hora em que o bloco foi inserido na Rede) (AKITA, 2018).

Tendo o bloco construído, precisamos então entender como o hash é formado e como ele garante que o conteúdo não foi alterado. O hash é uma codificação dos dados imputados no bloco utilizando, no caso da rede bitcoin, o formato SHA256. Este formato visa gerar um código único para aquele conjunto de dados. Importante ressaltar que não se trata de uma encriptação pois o hash não pode ser resolvido de volta no conteúdo do bloco, é uma representação codificada do conteúdo, sendo que caso seja alterado qualquer bite dos dados do bloco, o hash será alterado (MIRANDA e ZUCHI, 2018).

Uma vez desenhado o bloco, é possível avançar no entendimento da Blockchain. O primeiro bloco de qualquer blockchain será o bloco chamado de Gênesis, não terá um bloco anterior e deve ser criado pelo criador da blockchain, e terá também um hash para ser identificado. O segundo bloco, terá o hash do bloco anterior(Gênesis) concatenado com o conteúdo do bloco atual e assim é calculado o seu próprio hash. Nasce assim a estrutura base de uma blockchain, uma corrente de blocos interligados que não podem ser alterados e podem ser rastreados até a origem (MIRANDA e ZUCHI, 2018).

Figura 1 - Representação da estrutura de blocos da blockchain

Fonte: Elaborada pelos autores (2023)

Por uma questão de simplificação, neste exemplo é utilizado o hash puro, em que os dados são concatenados com o hash anterior e passam pelo seguinte código em C# para converter a string recebida em um hash SHA256:

Figura 2 - Exemplo de código de conversão de uma String em hash

```
2 referências
public static String sha256_hash(String value)
{
    StringBuilder Sb = new StringBuilder();

    using (SHA256 hash = SHA256Managed.Create())
    {
        Encoding enc = Encoding.UTF8;
        Byte[] result = hash.ComputeHash(enc.GetBytes(value));

        foreach (Byte b in result)
            Sb.Append(b.ToString("x2"));
    }

    return Sb.ToString();
}
```

Fonte: Elaborada pelos autores (2023)

Após essa conversão, é criada uma cadeia de blocos que estão interligados pelo seu hash. Caso ocorra qualquer alteração no bloco 1, além de recalculado o hash do bloco 1, é necessário recalcular também o hash do bloco 2. Dessa forma, quanto maior for o número de blocos na cadeia, maior será a dificuldade exponencial para retroceder na história da cadeia (AKITA, 2018).

2.1.1.2 Como é que acontece a mineração?

A mineração em um sistema de blockchain é necessário que um dispositivo com capacidade computacional concorra dentro da rede pelo direito à escrita de um bloco, existem dois principais sistemas de concorrência: o baseado em capacidade computacional e o baseado em capacidade monetária (SENA e DIAN, 2020).

2.1.1.3 Proof of work (PoW):

Tem-se a mineração utilizando PoW(proof of work), que consiste em uma corrida entre todas as pessoas tentando minerar o próximo bloco. Todos estão

tentando encontrar um número que quando concatenado com o hash do bloco retorna um hash com um determinado número de caracteres no início (SAPRA et al, 2023).

Isso se resume em um loop como o da figura 3:

Figura 3 - Exemplo de código de mineração baseada em PoW

```
referências
public static string minerar_PoW(string dificuldade, string hash)
{
    int nonce = 0; //numero procurado
    while(true)
    {
        string meu_hash = sha256_hash(hash + nonce);
        if (meu_hash.StartsWith(dificuldade))
        {
            Console.WriteLine("nonce = " + nonce);
            return meu_hash;
        }
        nonce += 1;
    }
}
```

Fonte: Elaborada pelos autores (2023)

Na figura 3 pode-se ver um loop feito em c# que recebe o hash do bloco anterior e procura um número(nonce), que vá retornar um hash com a dificuldade solicitada no início. Esta dificuldade costuma ser um conjunto de 0s.

Ao executar esse código em uma máquina com capacidade de processamento mediana, como as comumente encontradas entre os brasileiros, pode-se perceber a escala da complexidade:

Figura 4 - Log de resposta de um algoritmo de mineração PoW com nível 4

```
Hash do bloco: 3a69fa94f454b44094781269fa00aa60628920e82d7194067fb3422a77c20360
Dificuldade solicitada:0000
nonce = 58883
Hash final: 00006b8a1f1456cc8c4696650f8a27d636a17031346518656c45428dd1152e86
Tempo decorrido: -00:00:00.2427639
```

Fonte: Elaborada pelos autores (2023)

Pode-se comprovar pela figura 4 que ao requisitar uma dificuldade de 4 0s, o mesmo levou apenas 0,24 segundos para encontrar a resposta desejada.

Figura 5 - Log de resposta de um algoritmo de mineração PoW com nível 5

```
Hash do bloco: 3a69fa94f454b44094781269fa00aa60628920e82d7194067fb3422a77c20360
Dificuldade solicitada:00000
nonce = 1203262
Hash final: 00000fb8d586274d3e15f47ed86c73b45cfa26467d8b58adccc295d7d377f613
Tempo decorrido: -00:00:03.9149406
```

Fonte: Elaborada pelos autores (2023)

Ao elevarmos a dificuldade para 5 0s, observamos na figura 5 que o tempo de pesquisa aumenta significativamente, chegando a quase 4 segundos para encontrar a resposta desejada.

Figura 6 - Log de resposta de um algoritmo de mineração PoW com nível 6

```
Hash do bloco: 3a69fa94f454b44094781269fa00aa60628920e82d7194067fb3422a77c20360
Dificuldade solicitada:000000
nonce = 35761267
Hash final: 000000af17b9fff67f276844448abd6fe165bbaf1391fab6278c0163457c98fc
Tempo decorrido: -00:01:55.4794346
```

Fonte: Elaborada pelos autores (2023)

Ao elevarmos ainda mais a dificuldade para 6 0s, notamos pela figura 6 um aumento considerável no tempo de pesquisa, alcançando quase 2 minutos para encontrar a resposta desejada. No caso do Bitcoin, em 2021, é empregada uma dificuldade aproximada de 20 0s (GREVE, 2018).

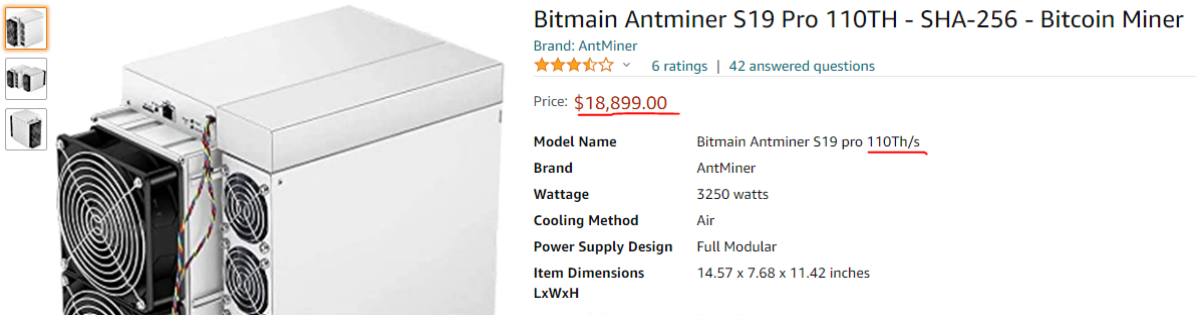
Conforme o próprio nome sugere, é crucial apresentar uma prova desse trabalho para que os demais participantes da rede possam validar e aceitar um novo bloco antes de ser adicionado à cadeia principal. Essa prova é representada pelo número denominado nonce, o qual pode ser transmitido junto com o hash descoberto. Qualquer participante pode utilizá-lo para chegar ao mesmo resultado do minerador bem-sucedido (GREVE, 2018).

Por ser uma competição, o participante que conseguir minerar primeiro com sucesso será recompensado pela criação do bloco. No entanto, para que alguém possa inserir informações alteradas, seria necessário ter controle sobre pelo menos 51% do poder de processamento de toda a rede, a fim de garantir que o bloco alterado seja aceito e adicionado à cadeia (GREVE, 2018).

Para os mineradores, o fator crucial é o hash rate. Atualmente, dispomos de máquinas especializadas projetadas para minerar algoritmos SHA256. Essas máquinas são otimizadas para realizar cálculos intensivos e gerar hashes de forma

eficiente, maximizando assim as chances de encontrar soluções válidas para os blocos de transações. O hash rate é um indicador fundamental da capacidade de processamento dessas máquinas e desempenha um papel crucial na competição pela mineração bem-sucedida.

Figura 7 - Hardware para mineração de hash



Fonte: Amazon 2023

Máquinas como a demonstrada na figura 7 são capazes de realizar incríveis 110 trilhões de hashes por segundo. Em comparação, um código consegue executar em média 340 mil hashes por segundo em um computador normal não focado em mineração. Essa discrepância no poder de processamento demonstra a imensa vantagem que as máquinas especializadas possuem em termos de eficiência e velocidade na mineração de criptomoedas.

2.1.1.4 Proof of stake (PoS):

Diante da demanda por uma capacidade computacional significativa exigida pelo Proof of Work (PoW), surgiram alternativas que buscam manter a segurança da rede por meio de métodos de criptografia avançados.

Após o Proof of Work (PoW), a alternativa mais reconhecida é o Proof of Stake (PoS). No PoS, o processo de "mineração" assemelha-se mais a um sorteio, no qual os participantes adquirem entradas com a moeda da blockchain em questão, havendo um limite máximo de compra por usuário. Dentro desse grupo de pessoas que adquiriram suas entradas para o sorteio, uma delas será selecionada para gerar o bloco e receber a recompensa correspondente (MAUNG et al, 2018).

Nesse modelo de mineração, não é utilizado poder computacional para resolver inúmeros hashes e encontrar a chave. Os mineradores do PoS precisam investir seus recursos financeiros para terem a chance de serem sorteados. Isso nos leva à situação em que é necessário que um desses mineradores adquira mais de 51% das entradas para tentar realizar alguma possível fraude. No entanto, tal ação colocaria em risco todo o investimento desse minerador fraudulento. Como todo o valor investido deve ser na mesma moeda da blockchain em questão, caso essa blockchain perca seu valor, o minerador fraudulento perderia todo o seu lucro, criando um desincentivo para ações fraudulentas (MAUNG et al, 2018).

2.1.1.5 Como funciona a distribuição?

Os dados na rede são distribuídos em todas as máquinas, permitindo que todos tenham acesso a eles a qualquer momento. É possível fazer uma analogia com um repositório público no GIT, onde qualquer pessoa pode criar uma ramificação (branch) e fazer um commit de volta para a versão principal (master). No contexto da mineração, são os mineradores que criam as ramificações, buscando gerar o próximo bloco. Aquele que conseguir primeiro gera um tipo de commit, que é validado pelo restante da rede. Se aprovado, esse commit passa a fazer parte da versão principal do blockchain (branch principal) (AKITA, 2018).

2.1.2 Pontos positivos e negativos da tecnologia.

A blockchain ficou muito conhecida acompanhando o bitcoin e conforme comentado por Rose Jacobs em 2018, no início havia uma visão de fazer todas as transações na mesma rede, sendo essa rede a blockchain que acompanha o Bitcoin, no entanto logo as empresas que foram utilizar a tecnologia iniciaram a criar suas próprias concepções da tecnologia (SAKAMOTO SARAH GOMES, 2020).

A privacidade da blockchain é uma das principais questões da rede pois apesar de ser uma rede extremamente segura e robusta, é completamente pública e está disponível para todos os nós da rede, usando esse fato para reforçar sua segurança (SAKAMOTO SARAH GOMES, 2020).

O conceito inicial por traz da rede é ter todas as transações sendo feitas no mesmo local para garantir a segurança e assim remover todas as taxas de intermediários no entanto com cada empresa criando a sua própria blockchain corremos o risco de termos o mesmo cenário que o anterior onde para realizar uma transação é necessário passar por diversos intermediários.

Com as empresas criando suas próprias blockchain, temos uma expansão nas possibilidades e uso da tecnologia, pode-se citar o exemplo do Walmart que buscou o uso da blockchain como uma forma de garantir que o suprimento de tilápia de Shangai para Dakota do Sul entre outros cuidados, seja descongelado pela primeira vez pelo consumidor final garantindo a melhor qualidade (CANT, 2019).

Para o funcionamento da rede, todos os envolvidos no processo devem em algum momento registrar e fazer o upload das informações na rede, o que requer uma conexão com a internet e pode ser um problema em algumas regiões (CANT, 2019).

2.2 CRIPTOMOEDAS

Diferente do que muitos pensam a ideia da criptomoeda foi descrita primeiramente em um artigo de 1998, publicado por Wei Dai no grupo de internet chamado Cypherpunks. Os Cypherpunks eram um grupo com grande preocupação com a privacidade e o anonimato no ambiente da internet, para isso utilizavam de criptografia para transmitir informações pela rede de forma que apenas o destinatário tivesse a fórmula para ler a informação (REIS, 2017).

O termo criptografia vem do grego Kryptós Graphién que significa escrita escondida, em outras palavras, criptografia é uma forma de enviar e receber mensagens de forma segura de forma que apenas o destinatário e o remetente consigam acessar o conteúdo da mesma (ASSANGE, 2012).

Um grande fator das criptomoedas é que elas não são sequer impressas, os parâmetros da moeda definidas pelo seu criador no início de seu tempo de vida, sendo assim quando o protocolo dessa nova moeda começa a circular todos sabem completamente como vai ser o funcionamento da moeda, qual o limite que pode ser

gerado, como que novas moeda são geradas ou como que é definido quem recebe essas novas moedas geradas, e todos essas definições se tornam imutáveis por conta do funcionamento da blockchain (REIS, 2017).

2.3 NFTs

Token não fungível, mais conhecido como NFT, é um ativo digital que representa algo único, diferente das criptomoedas e de outros tokens, as NFTs não são mutuamente intercambiáveis (SCHROEDER, 2018). Um ativo fungível como uma criptomoeda pode ser trocado por outra do mesmo tipo diretamente, um bitcoin pode ser trocado por outro sem que tenha alteração nas posses das entidades envolvidas, já um objeto não fungível representa algo único no ambiente então não pode ser trocado por outro igual, pois cada token tem um valor diferente e representa algo diferente na rede.

2.3.1 Utilidade das NFTs

As NFTs são usadas para simular a escassez de objetos no ambiente digital de forma verificável e segura, possibilitando que qualquer um possa ter os direitos sobre um ativo digital e que essa propriedade possa ser provada definitivamente (BUSSINESS WIRE, 2019). As principais utilidades mais conhecidas para o NFT atualmente ficam no mercado de arte/música e jogos eletrônicos.

No mercado de arte e música a NFT é utilizada como forma de garantir os direitos autorais sobre as obras, nesse sentido qualquer pessoa pode criar uma música ou obra de arte e atrelar sua criação com um registro de um token não fungível em uma rede blockchain, dessa forma sabemos que é o dono original daquele token logo sabemos também quem é o dono da obra vinculada ao token. Caso o criador queira vender sua obra ele também pode transferir o NFT relacionado garantindo assim que a obra tem um novo detentor dos direitos sobre a mesma (DEAN, 2021).

No caso dos jogos eletrônicos, a NFT é utilizada de forma que alguns dos recursos ou objetos dentro do jogo são gerados processualmente e cada um desses

objetos se vincula a um token, assim cada objeto se torna único, raro e escasso. A escassez desses objetos dentro do jogo acabam criando uma cada financeira crescente que permite que jogadores ganhem dinheiro fazendo a venda desses token para outros jogadores (DEAN, 2021).

2.4 CONTRATOS INTELIGENTES

Contratos inteligentes são programas de computador que executam automaticamente termos e condições acordados entre as partes envolvidas em uma transação. Esses contratos são projetados para serem transparentes, seguros, confiáveis e auto executáveis, eliminando a necessidade de intermediários e reduzindo a possibilidade de fraudes ou interferências (SZABO, 2006).

Os contratos inteligentes são executados em uma blockchain, que é um registro digital descentralizado e imutável. A tecnologia blockchain permite que as transações sejam verificadas e registradas de forma transparente, tornando os contratos inteligentes altamente seguros (MOHANTA, 2018).

Os contratos inteligentes podem ser utilizados em uma ampla gama de setores e aplicações, incluindo finanças, imóveis, cadeia de suprimentos, seguros e muito mais. Por exemplo, em um contrato de compra e venda de imóveis, um contrato inteligente pode ser usado para automatizar o processo de transferência de propriedade, verificando se todas as condições, como pagamento e documentos, foram cumpridas antes de efetivar a transação (MOHANTA, 2018).

As vantagens dos contratos inteligentes incluem:

- **Automatização:** Os contratos inteligentes são executados automaticamente sem a necessidade de intervenção humana, reduzindo erros e melhorando a eficiência.
- **Transparência:** Todas as transações e execuções de contrato são registradas na blockchain, tornando o processo transparente e audível.
- **Segurança:** Os contratos inteligentes são imutáveis e criptograficamente seguros, tornando-os altamente resistentes a fraudes e ataques cibernéticos.

- **Redução de intermediários:** Como os contratos inteligentes são autoexecutáveis, eles eliminam a necessidade de intermediários, reduzindo custos e complexidade.
- **Rapidez:** Os contratos inteligentes podem ser executados de forma instantânea, eliminando a necessidade de esperar por processos manuais e burocráticos.

Os contratos inteligentes funcionam por meio da execução de código em uma blockchain. Tomando como exemplo a plataforma Ethereum, que é uma das mais conhecidas para a criação e execução de contratos inteligentes.

- **Linguagem de programação:** Para escrever um contrato inteligente no Ethereum, é utilizada a linguagem de programação Solidity. Essa linguagem permite que os desenvolvedores descrevam as regras e lógica do contrato em forma de código.
- **Implantação na blockchain:** Após o desenvolvimento do contrato inteligente, ele é implantado na blockchain Ethereum. Isso envolve o envio do código para a rede e a gravação do contrato em um bloco da blockchain. Esse processo requer o pagamento de uma taxa de transação, conhecida como "gas".
- **Execução automática:** Uma vez implantado na blockchain, o contrato inteligente está disponível para execução. Quando as condições pré-definidas no contrato são atendidas, ele é automaticamente acionado. Essas condições podem ser determinadas por eventos específicos, como um pagamento recebido, um prazo atingido ou uma ação realizada.
- **Verificação e consenso:** Antes de ser executado, o contrato inteligente passa por um processo de verificação e consenso na rede blockchain. Isso envolve a validação do contrato por nós (nodes) na rede, que verificam se o código foi escrito corretamente e se as transações associadas ao contrato são válidas.
- **Transparência e imutabilidade:** Uma vez que o contrato inteligente é executado, todas as transações e ações realizadas são registradas na blockchain de forma transparente e imutável. Isso significa que qualquer

pessoa pode visualizar o histórico completo do contrato e sua execução, garantindo transparência e rastreabilidade.

- **Auto Execução e autenticidade:** Os contratos inteligentes são autoexecutáveis, o que significa que eles são executados sem a necessidade de intermediários ou intervenção humana. Além disso, a autenticidade do contrato é garantida pela criptografia e pelas regras definidas em seu código.
- **Pagamentos e tokens:** Os contratos inteligentes também podem lidar com pagamentos e ativos digitais. Por exemplo, eles podem facilitar o envio e recebimento de criptomoedas ou tokens específicos, permitindo a transferência de valor de forma direta e segura.

É importante destacar que a execução de contratos inteligentes está sujeita a limitações técnicas, como a capacidade de processamento da rede blockchain e as restrições de escalabilidade. Além disso, questões legais e regulatórias podem surgir em relação à validade jurídica dos contratos inteligentes em diferentes jurisdições (KHAN et al, 2021).

2.5 CASES DE APLICAÇÃO DO BLOCKCHAIN

Neste capítulo, são descritos os conceitos e aplicações das tecnologias derivadas da blockchain, sua utilidade e potencial futuro, bem como alguns casos existentes de aplicações reais dessas tecnologias.

2.5.1 CRIPTOMOEDAS

Quando falamos de criptomoedas certamente o maior expoente deste ramo é o bitcoin, tanto por ser o primeiro quanto por ser o mais conhecido no mundo. Se perguntarmos sobre bitcoin para qualquer pessoa que tenha acessado a internet no último ano e ela já tenha ouvido sobre, esse projeto recebeu uma grande notoriedade por causa do grande aumento do seu valor unitário em comparação com o dólar, especialmente nos últimos dois anos (BINANCE, 2021).

O bitcoin é uma criptomoeda, ou seja, um ativo digital que facilita transações em uma sociedade baseada em propriedades privadas, servindo como um substituto

descentralizado das moedas de governos como o Dólar dos Estados Unidos da América ou o Iene do Japão (BINANCE, 2021).

2.5.1.1 Por que o Bitcoin tem valor?

O Bitcoin é um ativo digital sem lastro em nenhum ativo físico, isso significa que não existe nada físico no mundo que seja usado como referência para pautar o valor do bitcoin, com isso o ativo tende a ser mais volátil e especulativo, fazendo seu valor variar mais que moedas comuns. Outro grande fator que incentiva o aumento do seu valor é o fato que ser uma moeda deflacionária, ou seja, o bitcoin desde sua criação tem um valor limitado e por questões de perdas desses ativos ela tende a diminuir a quantidade em circulação com o tempo, aumentando proporcionalmente seu valor com relação a raridade (BINANCE, 2021).

Segundo a Binance em um artigo postado em janeiro de 2021, os principais fatores para o valor do bitcoin são a descentralização, a imutabilidade e a divisibilidade. A descentralização dificulta que pessoas ou instituições manipulem a moeda para benefício próprio, o fato do ativo ser gerenciado em cima de um protocolo da blockchain impede que sua base monetária seja expandida e garante que todas as transações do ativo são válidas, esse fator também é o motivo principal pela qual a imutabilidade é um dos pilares da criptomoeda.

O bitcoin é uma ativo divisível, isso facilita sua transação pois dessa forma um bitcoin pode ser dividido em dois 0.5 bitcoins, e assim sucessivamente e o mesmo possa ser transacionado na blockchain sem problemas, cada fração do bitcoin é chamada de satoshi e cada satoshi pode ter uma transação e um dono único no sistema. Esse fator é importante pois garante que valores pequenos possam circular no sistema aumentando não só sua utilidade como também sua liquidez (BINANCE, 2021).

2.5.1.2 Quais as principais diferenças em comparação com moedas governamentais?

O primeiro fator que diferencia o bitcoin de moedas governamentais é o fato do bitcoin ser um ativo totalmente virtual, enquanto moedas como o Real ou o Dólar tem um relativo físico que precisa ser cunhado no caso das moedas e impresso no caso das cédulas. Isso favorece a moeda corrente dos países pois permite transações sem o uso da tecnologia, enquanto o bitcoin precisa estar dentro da rede blockchain para poder ser transacionado, por outro lado o fato da criptomoeda ser completamente digital dificulta fraudes e golpes pois a blockchain garante a segurança de todos os ativos na rede (DE ARAUJO, 2021)

O segundo principal fator que diferencia o bitcoin das moedas físicas é o fato do ativo digital ser completamente descentralizado, isso garante ao bitcoin muito mais segurança e previsibilidade quanto ao seu futuro pois diferente do Real não existe ninguém que possa expandir a base monetária para manipular o preço ou alterar algo no protocolo de transações do bitcoin sem que todos da rede concordem (NAKAMOTO, 2008).

2.5.1.3 Como é mantido?

O bitcoin é baseado em um protocolo criado por Satoshi Nakamoto lançado juntamente com seu artigo, com isso o bitcoin não tem uma base de código que precise ser usada para o funcionamento, dessa forma qualquer um pode criar um programa que acessa e contribui na mesma rede do bitcoin desde que o sistema siga o protocolo definido no começo do funcionamento da moeda (NAKAMOTO, 2008)

2.5.2 Impacto social

Com o passar dos anos desde a popularização da tecnologia, muitas pessoas estão tentando utilizar a blockchain das mais variadas maneiras. Uma forma nobre de contribuir que foi encontrada por algumas dessas pessoas é fazer o uso da

tecnologia para uma finalidade nobre e facilitar a vida dos usuários além do financeiro, buscando trazer segurança e confiança para a vida das pessoas como um todo não apenas o financeiro. Abaixo alguns exemplos que não são ligados diretamente a dinheiro sendo armazenado na blockchain. (WELIGHT, 2016).

Hoje quando é feita uma doação, só é possível esperar que seja entregue no local que dizem que será entregue, porém com a blockchain é possível mudar isso. Supondo um cenário onde pode ser escolhido entre cashBack ou a doação para uma instituição de combate ao câncer, caso seja escolhido o cashBack, se pode facilmente observar o valor correto entrando em nossa conta mas, quando se está fazendo uma doação não tem como rastrear este valor até chegar em seu destino, neste ponto entra a blockchain, podendo ser registrado todos os passos até a transferência para a instituição desejada e todos os registros estariam disponíveis publicamente e com a possibilidade de serem validados por qualquer interessado. (WELIGHT, 2016).

As cooperativas do nosso país têm uma grande dificuldade para fazer microfinanciamento já que para tal é necessário passar por alguma instituição financeira e pagar os juros de acordo. Com o uso da blockchain e a iniciativa de instituições como a unicafes, é possível criar tokens(baseados em Ethereum no caso da unicafes) e utilizá-los para que pessoas do mundo todo invista nessas pequenas cooperativas e possam em primeiro lugar apoiar o projeto que precisa ser feito e uma vez que esteja concluído e gerando receita, o valor pode ser devolvido com uma taxa de sucesso. (UNICAFES e moedasSeeds).

2.5.3 Entretenimento e cultural

Como comentado por William Ribeiro de Paula em seu TCC, as mudanças que blockchain causou no mundo vão muito além de criar uma moeda digital, ele também abriu as portas para que diferentes mídias pudessem ser armazenadas e trabalhadas. Com a popularização dos NFTs temos cada vez mais jogos que utilizam a blockchain como um livro razão para armazenar seus recursos e também, garantir

que as transações entre players sejam feitas sem problemas e que seus recursos estão sempre salvos de forma imutável.

Além de jogos, temos a utilização de NFT para música, vídeos, textos ou qualquer propriedade intelectual que alguém deseja armazenar com segurança, sendo possível negociar de forma livre sobre o recurso armazenado.

Um dos principais e mais antigos dos jogos é chamado Axie Infinity, desenvolvido pelo estúdio Vietnamista Sky Mavis, que utiliza moedas baseadas em Ethereum(AXS e SLP) para que seus jogadores possam negociar seus animais e itens uns com os outros. Infelizmente para jogar esse tipo de jogo temos uma barreira de entrada, para iniciar no jogo é necessário comprar de antemão 3 animais para que seja possível batalhar e seguir o jogo com eles.

2.5.4 Inovação

Proposta recente feita por Araujo, Guilherme Patricio de (2021), sugere a blockchain como modelo para inovar na sistematização das informações na exportação de café, um mercado que corresponde a um terço da receita total do país e que tem muitas irregularidades devido a diferentes impostos em diferentes estados possibilitando fraudes como a exposta pela operação “Expresso” em que estima-se mais de 1 bilhão de reais foram sonegados no setor. Com a blockchain seria possível implementar um sistema que tenha os produtores e consumidores finais como usuários e seja detalhado cada tributo ao longo do caminho para que seja transparente de forma clara e imutável o quanto está sendo gasto em cada etapa do processo gerando mais confiança no produto e evitando fraudes.

3 MÉTODO

Neste capítulo é abordado o tipo de pesquisa sobre a blockchain e NFTs, as etapas metodológicas, o planejamento das atividades.

3.1 METODOLOGIA DE PESQUISA

Nesta monografia foi desenvolvido em duas pesquisas principais, a primeira de pesquisa onde será realizado um estudo sobre o funcionamento de uma aplicação blockchain e sua utilização com contratos inteligentes e seu funcionamento no mercado brasileiro, esse estudo será básico onde apenas será descrito e investigado um case existente. A abordagem dessa pesquisa será qualitativa descrevendo todos os aspectos qualitativos do sistema.

A segunda parte da pesquisa foi o desenvolvimento de uma aplicação de contratos inteligentes que possa criar e gerenciar os contratos, essa parte é uma pesquisa básica e exploratória, no final sendo possível compreender os desafios e facilidades desse ambiente.

3.2 METODOLOGIA DO TRABALHO

O desenvolvimento da pesquisa se dá pela realização das etapas a seguir:

- Para entender os aspectos técnicos relacionados aos assuntos do tema foi realizada uma pesquisa bibliográfica;
- Escolha da ferramenta para desenvolvimento do modelo de aplicação que permite a criação de uma plataforma blockchain(pesquisa aplicada);
- Estudo e aprendizado da ferramenta selecionada para desenvolvimento da aplicação;
- Desenvolvimento de um protótipo blockchain que crie e verifique contratos inteligentes;
- Validação dos resultados;
- Conclusões da pesquisa.

3.3 DELIMITAÇÕES

As aplicações da blockchain são incalculáveis, já que se trata de uma tecnologia disruptiva e devido ao tamanho e a complexidade que essa pesquisa pode chegar, seguem abaixo as limitações deste trabalho:

- Foi criado um estudo em cima de uma aplicação já existente na área de blockchain.
- Foi criado um protótipo para validação de contratos inteligentes em blockchain.

4 PROPOSTA DE SOLUÇÃO

Neste capítulo são apresentadas as definições e metodologias utilizadas para o desenvolvimento do sistema. São discorridas as definições de UML, requisitos, e casos de uso.

As propostas apresentadas nas seções seguintes foram definidas por meio do conhecimento adquirido no decorrer do desenvolvimento deste trabalho, porém é sabido que existem módulos mais completos e complexos que deixariam o protótipo mais robusto, mas por conta do tempo do trabalho, optou-se por fazer uma versão simples, com as funcionalidades essenciais, como uma prova de conceito.

4.1 UML

UML (Unified Modeling Language) é uma linguagem de modelagem padronizada que é amplamente utilizada na engenharia de software para representar visualmente sistemas complexos. Ela fornece um conjunto de notações gráficas e semânticas para descrever estruturas e comportamentos de sistemas, permitindo que equipes de desenvolvimento comuniquem e documentem de forma eficaz as diferentes perspectivas de um sistema.

A UML foi desenvolvida originalmente pela Rational Software Corporation na década de 1990 e, desde então, tornou-se um padrão de fato na indústria de desenvolvimento de software. Ela é suportada por várias ferramentas de modelagem e é usada em diferentes fases do ciclo de vida do desenvolvimento de software, desde a concepção até a implementação e manutenção.

4.2 REQUISITOS

Nesta seção, serão apresentados os requisitos funcionais, requisitos não funcionais e regras de negócios para a solução do protótipo funcional proposto. Com base na análise realizada anteriormente e na experiência adquirida durante o desenvolvimento deste trabalho, foram identificados os requisitos que serão apresentados nas próximas seções.

4.2.1 Requisitos funcionais

Requisitos funcionais são declarações específicas que descrevem as funcionalidades e as ações que um sistema, software ou produto deve ser capaz de realizar. Esses requisitos descrevem as principais características e comportamentos do sistema, fornecendo uma visão clara e detalhada do que ele deve fazer.

Os requisitos funcionais geralmente descrevem as interações do sistema com os usuários, outros sistemas ou componentes, bem como as respostas esperadas do sistema a entradas específicas. Eles definem as tarefas que o sistema deve ser capaz de executar, os dados que devem ser processados e os resultados que devem ser produzidos.

Os requisitos funcionais são geralmente expressos em forma de declarações ou casos de uso, que descrevem a ação que o sistema deve executar e as condições sob as quais essa ação deve ocorrer. Essas declarações podem ser escritas em linguagem natural ou podem ser representadas de forma mais formal usando técnicas de modelagem, como diagramas de casos de uso, fluxogramas ou especificações de linguagem de programação.

Para que um contrato inteligente seja desenvolvido e possa ser utilizado de forma funcional e confiável, são necessárias diversas funcionalidades e características. Os requisitos funcionais para o desenvolvimento e implementação de uma aplicação baseada em contratos inteligentes são apresentados no quadro a seguir:

Tabela 1 - Requisitos funcionais

Identificação	Requisito
RF001	O sistema deve exigir Autenticação para ser utilizado.
RF002	O sistema deve possibilitar a escolha e configuração de contratos inteligentes.
RF003	O sistema deve viabilizar a publicação dos contratos criados na blockchain.
RF004	O sistema deve efetuar a publicação de um anúncio em seu marketplace.

Identificação	Requisito
RF005	O sistema deve possibilitar e registrar o/os interessados na compra do/s item/ns ofertado/s no marketplace
RF006	O sistema deve acompanhar o andamento do contrato bem como notificar as partes envolvidas quando houver alterações.

Fonte: Elaborada pelos autores (2023)

4.2.2 Requisitos não funcionais

Requisitos não funcionais são critérios e restrições que definem os atributos de qualidade de um sistema, software ou produto, em vez de suas funcionalidades específicas. Esses requisitos descrevem características importantes que não estão relacionadas diretamente com as ações executadas pelo sistema, mas sim com aspectos como desempenho, segurança, usabilidade, confiabilidade, manutenibilidade e escalabilidade.

Ao contrário dos requisitos funcionais, que se concentram no "o que" o sistema deve fazer, os requisitos não funcionais se concentram no "como" o sistema deve realizar suas funcionalidades. Eles definem critérios para avaliar o desempenho geral do sistema e para garantir que ele atenda aos padrões e regulamentações aplicáveis.

Os requisitos não funcionais para o desenvolvimento e implementação de uma aplicação baseada em contratos inteligentes são apresentados no quadro a seguir:

Tabela 2 - Requisitos não funcionais

Identificação	Requisito
RNF001	O sistema utilizará a blockchain da Ethereum
RNF002	O sistema deve ser responsivo para ser acessado de qualquer dispositivo.
RNF003	O sistema deve garantir que apenas usuários cadastrados tenham acesso.

Identificação	Requisito
RNF004	O sistema deve possuir uma interface simples e de fácil utilização.

Fonte: Elaborada pelos autores (2023)

4.2.3 Regras de negócio

As regras de negócio são diretrizes ou restrições que definem como um sistema de software deve funcionar em termos de lógica de negócio. Elas representam as políticas, as regras e os processos específicos que regem as operações e as funcionalidades do software, garantindo que ele atenda aos requisitos e às necessidades do negócio ou da organização.

As regras de negócio podem abordar diferentes aspectos, como validações de dados, cálculos, fluxos de trabalho, autorizações, regras de acesso, regras de precificação, entre outros. Elas são formuladas com base nas necessidades do negócio e podem variar de acordo com o domínio de aplicação e as particularidades do software em questão.

As regras de negócio estabelecidas para este desenvolvimento são as apresentados no quadro a seguir:

Tabela 3 - Regras de negócio

Identificação	Regra
RN001	O sistema deve exigir uma senha forte.
RN002	O sistema não deve permitir criar conta com email já cadastrado.
RN003	O sistema deve exigir a escolha de um modelo de contrato durante a criação.
RN004	O sistema deve exigir o preenchimento de configurações mínimas para a criação do contrato.
RN005	O sistema deve possibilitar ordenação e pesquisa no marketplace
RN006	Apenas o dono do contrato pode alterar suas configurações e apenas enquanto o contrato estiver aberto.
RN007	O contrato só pode ser criado se os itens nele descritos estiverem disponíveis.

Identificação	Regra
RN008	Caso o comprador não interaja com o contrato em um prazo estipulado durante sua criação, o contrato deve retornar para status disponível e os fundos depositados pelo comprador serão devolvidos.

Fonte: Elaborada pelos autores (2023)

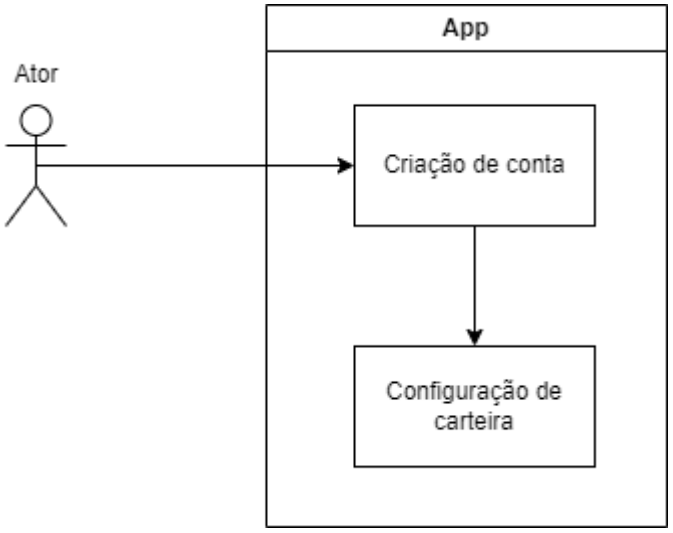
4.2.4 Casos de uso:

Um caso de uso é uma técnica de modelagem usada no desenvolvimento de software para descrever as interações entre os usuários e um sistema. Ele representa uma situação específica em que o sistema é utilizado para alcançar um objetivo, descrevendo as ações executadas pelos usuários e as respostas esperadas do sistema. Os casos de uso fornecem uma visão clara e detalhada das funcionalidades do sistema, ajudando na compreensão dos requisitos e no alinhamento entre os stakeholders. Eles também servem como base para o desenvolvimento, o teste e a validação do sistema, auxiliando na identificação de cenários de uso e na detecção de possíveis erros ou lacunas nos requisitos.

4.2.4.1 Caso de uso 1:

Na figura 8, é apresentado um ilustrativo caso de uso que exemplifica as iterações disponíveis para o usuário.

Figura 8 - Ilustrativo do caso de uso 1



Fonte: Elaborada pelos autores (2023)

Tabela 4 - UC001

UC001 - Criar conta
Descrição: O ator cria sua conta e conecta sua carteira.
Pré-condições:
Pós-condições: Conta criada e carteira conectada a conta.
Requisitos Funcionais: RF001
- Fluxo Principal Passo 1: O ator preenche o campo email. Passo 2: O ator preenche o campo senha, obedecendo a RN001. Passo 3: O ator preenche o campo confirma a senha. Passo 4: O ator clica no botão criar. Passo 5: O ator é direcionado para a página de login. Passo 6: O ator efetua o login com a conta criada. Passo 7: O ator é direcionado a tela inicial onde lhe é solicitado a conexão com sua carteira.
- Fluxo alternativo A Passo 1: O ator escolhe logar com o Google. Passo 2: O ator informa o usuário e senha do google. Passo 3: O ator é redirecionado para a tela de login. Passo 4: O ator efetua o login com a conta criada. Passo 5: O ator é direcionado a tela inicial onde lhe é solicitado a conexão com sua carteira.
- Fluxo alternativo B Passo 1: O ator preenche o campo e-mail com um e-mail já registrado, não obedecendo a RN002.

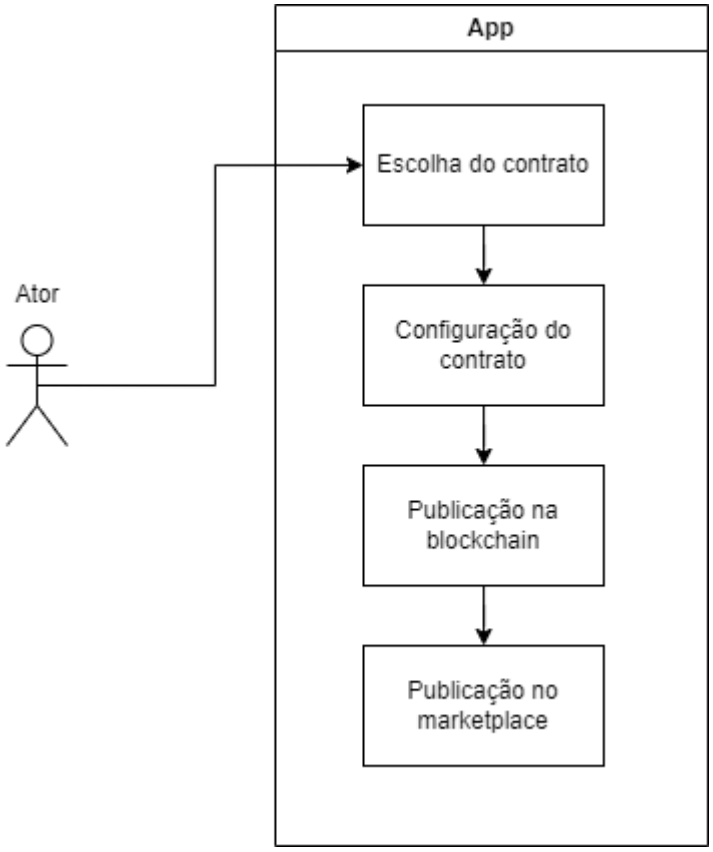
UC001 - Criar conta
Passo 2: O ator preenche o campo senha, obedecendo a RN001. Passo 3: O ator preenche o campo confirma a senha. Passo 4: O ator clica no botão criar. Passo 5: O ator é informado pelo sistema que já existe uma conta criada com o email informado.

Fonte: Elaborada pelos autores (2023)

4.2.4.2 Caso de uso 2:

A seguir, é apresentado um caso de uso que demonstra a publicação de um novo bloco e o subsequente anúncio no marketplace:

Figura 9 - Ilustrativo do caso de uso 2



Fonte: Elaborada pelos autores (2023)

Tabela 5 - UC002

UC002 - Criação, configuração e publicação do contrato
Descrição: O ator cria o seu contrato-o e publica na blockchain que por sua vez, resulta em uma publicação no marketplace.

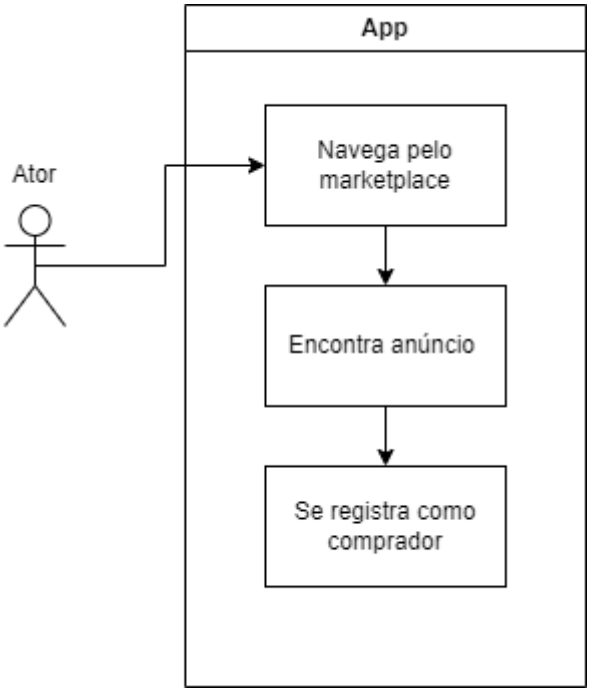
UC002 - Criação, configuração e publicação do contrato
Pré-condições: UC001
Pós-condições: Contrato registrado na blockchain e incluído no marketplace
Requisitos Funcionais: RF002, RF003 e RF004
- Fluxo Principal Passo 1: O ator escolhe um dos modelos de contrato que deseja criar obedecendo a RN003. Passo 2: O ator configura o contrato de acordo com sua necessidade obedecendo a RN004. Passo 3: O ator confirma a criação do contrato com as configurações selecionadas. Passo 4: O ator clica no botão criar contrato e efetua o pagamento utilizando a carteira cadastrada. Passo 5: O sistema publica o contrato na blockchain e cria o anúncio no marketplace. Passo 6: O ator é direcionado para a tela contendo seus anúncios. Passo 7: O ator faz o acompanhamento de seu anúncio pela plataforma.
- Fluxo alternativo A Passo 1: O ator escolhe um dos modelos de contrato que deseja criar. Passo 2: O ator configura o contrato de acordo com sua necessidade Passo 3: O ator tenta configurar um item para venda que já tem um contrato aberto. Passo 4: O sistema notifica o ator que o item já está preso em outro contrato, obedecendo a RN007.

Fonte: Elaborada pelos autores (2023)

4.2.4.3 Caso de uso 3:

A seguir é demonstrado um caso de uso com o registro de um comprador:

Figura 10 - Ilustrativo do caso de uso 3



Fonte: Elaborada pelos autores (2023)

Tabela 6 - UC003

UC003 - Registro de comprador no contrato
Descrição: O ator navega pelo marketplace, encontra o contrato que deseja comprar e se registra como comprador.
Pré-condições: UC001 e UC002
Pós-condições: Contrato com ator registrado como comprador.
Requisitos Funcionais: RF005
- Fluxo Principal Passo 1: O ator navega pelo marketplace. Passo 2: O ator encontra um anúncio que faz sentido para ele. Passo 3: O ator lê os detalhes do contrato referentes aos bens oferecidos e as condições da compra. Passo 4: O ator decide se registrar como comprador do contrato. Passo 5: O ator confirma a transação em sua carteira. Passo 6:. O ator é redirecionado para a página de acompanhamento do pedido.
- Fluxo alternativo A Passo 1: O ator navega pelo marketplace. Passo 2: O ator encontra um anúncio que faz sentido para ele. Passo 3: O ator lê os detalhes do contrato referentes aos bens oferecidos e as condições da compra. Passo 4: O ator decide que quer negociar e entra em contato com o dono

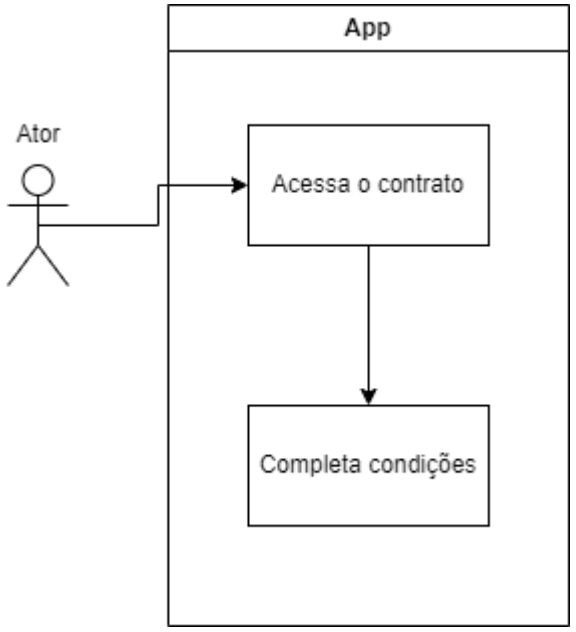
UC003 - Registro de comprador no contrato
do contrato com sua oferta. Passo 5: O ator é redirecionado para a página de chat para acompanhar a negociação.

Fonte: Elaborada pelos autores (2023)

4.2.4.4 Caso de uso 4:

A seguir se tem um caso de uso demonstrando o ator comprador buscando completar as condições do contrato:

Figura 11 - Ilustrativo do caso de uso 4



Fonte: Elaborada pelos autores (2023)

Tabela 7 - UC004

UC004 - Conclusão das condições do contrato
Descrição: O ator acessa o contrato e completa as condições
Pré-condições: UC001, UC002 e UC003
Pós-condições: Contrato com ator registrado como comprador.
Requisitos Funcionais: RF005
- Fluxo Principal

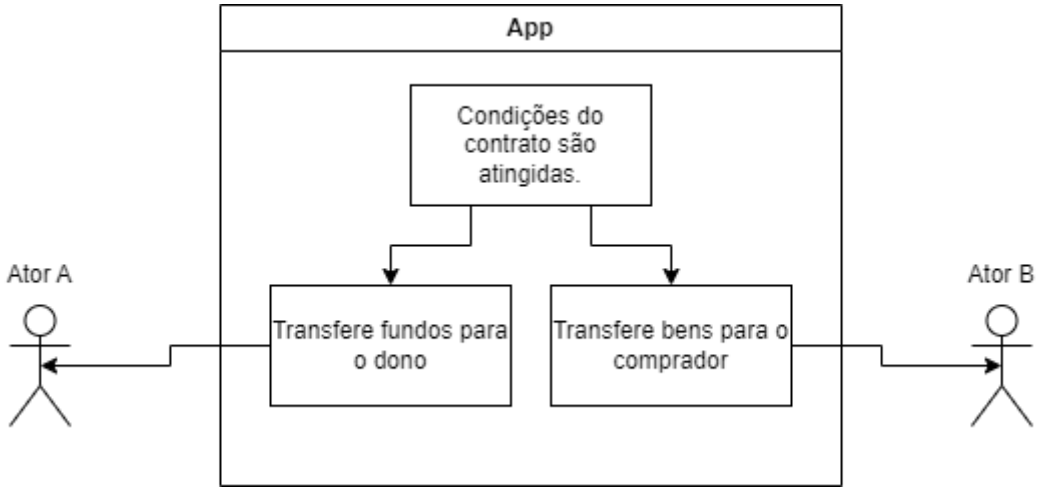
UC004 - Conclusão das condições do contrato
Passo 1: O ator acessa o contrato. Passo 2: O ator define quais condições deseja cumprir. Passo 3: O ator cumpre o necessário para atender as condições. Passo 4: O sistema é atualizado exibindo o novo status do contrato.
- Fluxo alternativo A Passo 1: O ator acessa o contrato. Passo 2: O ator define qual condição deseja cumprir. Passo 3: O ator cumpre o possível da condição. Passo 4:.O sistema atualiza e exibe o status atual do contrato.

Fonte: Elaborada pelos autores (2023)

4.2.4.5 Caso de uso 5:

A seguir pode se visualizar um caso de uso demonstrando a execução de um contrato:

Figura 12 - Ilustrativo do caso de uso 5



Fonte: Elaborada pelos autores (2023)

Tabela 8 - UC005

UC005 - Registro de comprador no contrato
Descrição: Quando as condições do contrato são atingidas a transferência de bens é automaticamente executada.
Pré-condições: UC001, UC002, UC003 e UC004
Pós-condições: Contrato executado e bens transferidos.

Requisitos Funcionais: RF006
- Fluxo Principal Passo 1: O contrato é executado e os bens são transferidos para os respectivos donos. Passo 2: O sistema recebe a atualização do contrato. Passo 3: O sistema notifica os atores. Passo 4: Os atores validam suas páginas de acompanhamento e podem observar os detalhes da transação.

Fonte: Elaborada pelos autores (2023)

5 DESENVOLVIMENTO

Neste capítulo, são apresentadas as ferramentas e tecnologias utilizadas para o desenvolvimento de um dos contratos inteligentes propostos.

5.1 FRAMEWORK E FERRAMENTAS APLICADAS:

Com base nas necessidades descritas, pode ser considerado o uso do framework Alchemy para desenvolver a plataforma blockchain baseada em Ethereum. O Alchemy é uma poderosa infraestrutura de desenvolvimento blockchain que simplifica a interação com a rede Ethereum e oferece recursos avançados.

5.1.1 Alchemy:

O Alchemy é um framework de desenvolvimento blockchain que fornece uma API fácil de usar e escalável para interagir com a rede Ethereum. Ele oferece recursos poderosos, como nós de infraestrutura otimizados, monitoramento de transações, suporte a eventos em tempo real e análises detalhadas. O Alchemy permite que você se concentre no desenvolvimento da lógica de negócios da plataforma, enquanto lida com a complexidade da infraestrutura blockchain.

5.1.2 Linguagem de Programação: Solidity

O Solidity é uma linguagem de programação amplamente utilizada e versátil, desenvolvida especificamente para escrever contratos inteligentes na blockchain Ethereum. Com o Solidity, você pode criar contratos inteligentes que definem a lógica e o comportamento dos aplicativos descentralizados (DApps) na plataforma Ethereum. Ele oferece recursos para definir estruturas de dados, funções, eventos e permite a interação com outras contas e contratos na rede Ethereum.

5.1.3 Biblioteca Web3.js:

A biblioteca web3.js é uma escolha popular para interagir com a blockchain Ethereum usando JavaScript. Ela fornece uma API rica para envio de transações, interação com contratos inteligentes, leitura e escrita de dados na blockchain, gerenciamento de contas e muito mais. A biblioteca web3.js é compatível com o framework Alchemy e oferece recursos adicionais para desenvolver sua plataforma de gerenciamento de cadeia de suprimentos e autenticação de produtos.

5.1.4 Remix IDE:

A Remix IDE é um ambiente de desenvolvimento integrado (IDE) projetado especificamente para contratos inteligentes na plataforma Ethereum. Ele oferece uma série de recursos e ferramentas que tornam o processo de desenvolvimento e teste de contratos inteligentes mais fácil e eficiente.

No geral, a Remix IDE é uma opção popular para o desenvolvimento de contratos inteligentes devido à sua interface amigável, recursos de depuração, facilidade de compilação e implantação, testes automatizados e integração com o ecossistema Ethereum.

5.1.5 Ether Scan:

O Etherscan é um serviço online e um explorador de blocos para a rede Ethereum. Ele fornece uma interface amigável para visualizar, explorar e analisar informações sobre transações, contratos inteligentes, endereços e blocos na blockchain do Ethereum. O Etherscan é uma escolha comum para o desenvolvimento de contratos inteligentes devido à sua facilidade de uso, recursos de monitoramento, verificação de contratos, exploração de informações e integração com outras ferramentas.

5.1.6 Metamask:

Metamask é uma extensão de navegador que permite aos usuários interagir com aplicativos descentralizados (dApps) baseados na tecnologia blockchain. Ele funciona como uma carteira digital e uma ponte de comunicação entre o navegador web e a rede Ethereum, permitindo que os usuários gerenciem suas criptomoedas e interajam com contratos inteligentes.

Sua escolha foi feita devido à sua interface amigável, segurança, compatibilidade com a rede Ethereum e suporte aos desenvolvedores. Ele desempenha um papel fundamental na simplificação da interação dos usuários com a tecnologia blockchain, abrindo caminho para uma adoção mais ampla e facilitando a criação de aplicativos descentralizados inovadores.

5.1.7 Ethereum:

Ethereum é uma plataforma descentralizada baseada na tecnologia blockchain que permite a criação e execução de contratos inteligentes. Foi proposto por Vitalik Buterin em 2013 e lançado em 2015. O Ethereum se diferencia do Bitcoin por sua capacidade de suportar além da criptomoeda nativa (ETH), a execução de programas computacionais complexos e descentralizados.

A principal razão para escolher o Ethereum para o desenvolvimento de aplicativos é sua funcionalidade de contratos inteligentes.E a linguagem de programação Solidity é usada para escrever contratos inteligentes no Ethereum e possui suporte robusto por parte da comunidade de desenvolvedores.

5.1.8 Sepolia:

A Sepolia é uma rede interna da Ethereum destinada a testes, ela foi lançada em 2021 com o intuito de facilitar o desenvolvimento de aplicações dentro da rede principal sem ter o risco de perda ou conflitos. Dentro dessa rede é possível gerar e transferir valores sem ter a necessidade de despender os valores relativos às

moedas da transação, isso permite que o desenvolvedor realize operações pagando as taxas sem ter custo real.

A rede Sepolia foi escolhida por conta da segurança no desenvolvimento, permitindo uma maior variedade de testes e custo, e por conta deste projeto ser uma prova de conceito não tendo a intenção de ser publicado em uma rede real. Vale ressaltar que a rede é um parâmetro referenciado dentro do código, durante a execução do código, podendo ser facilmente trocado por uma rede real da Ethereum.

5.2 CENÁRIO DE APLICAÇÃO

Em um mundo cada vez mais globalizado, a gestão eficiente da cadeia de suprimentos se tornou essencial para o sucesso das empresas. Nesse contexto, uma aplicação blockchain de contratos inteligentes pode revolucionar a forma como os processos logísticos são conduzidos, proporcionando maior transparência, segurança e eficiência.

Imagine um grande produtor de grãos que hoje tem que recorrer a uma empresa para fazer a venda de seus produtos, outra empresa para gerenciar o rastreamento do envio e ainda corre o risco de sofrer com algum tipo de fraude na transação. Pensando nesse caso, é desenvolvida a plataforma projetada nessa monografia, permitindo que o produtor possa substituir todas essas empresas criadoras de confiabilidade da transação e possa substituir por apenas um contrato que pode ser gerenciado de qualquer local e está registrado na rede de forma a garantir a execução e confiabilidade do mesmo.

A plataforma consiste em um sistema que consegue criar e gerenciar contratos inteligentes e permitir a execução do mesmo de acordo com as regras pré-estabelecidas no contrato. Também é possível substituir qualquer grande software de rastreamento pelo contrato no qual consegue manter a localização atual do objeto escolhido por meio de um dispositivo IOT e tudo isso sem que tenha que passar por nenhuma empresa de serviço apenas enviando informações para dentro da rede blockchain.

5.3 ARQUITETURA DE IMPLANTAÇÃO E SOLUÇÃO

Neste capítulo é apresentada a estrutura escolhida e as integrações que serão feitas com aplicações e sistemas de terceiros, assim como o motivo para cada uma das escolhas.

5.3.1 Design da arquitetura:

A blockchain é a maior aposta da web.3.0 onde as conexões e sistemas são descentralizados, por conta disso a nossa abordagem para esse desenvolvimento foi utilizar o maior número possível de integrações de forma que a nossa aplicação só implemente as lógica do sistema e as regras de negócio. Com isso o nosso sistema será composto por um core desenvolvido que se integra com outras aplicações, sendo essas uma camada de abstração para a execução de tarefas em uma rede blockchain.

5.3.1.1 Blockchain:

A blockchain é uma ferramenta que ganha confiabilidade à medida que o número de usuários ativos aumenta. Portanto, uma rede com muitos usuários é significativamente mais segura e difícil de ser fraudada do que uma rede nova com poucos adeptos. Devido a essa consideração, foi optado por utilizar a rede Ethereum para o desenvolvimento dos nossos contratos inteligentes. Essa rede é amplamente conhecida e se destaca em comparação com outras redes devido ao seu constante crescimento e evolução.

No entanto, levando em conta custos e segurança, foi feito uso da rede Ethereum chamada Sepolia. Trata-se de uma rede de testes disponibilizada para desenvolvedores que desejam testar sua integração com a rede Ethereum principal, sem expor-se aos riscos de gerar transações reais. A grande vantagem de se conectar à rede Ropsten é o fato de que os valores transacionados são falsos, o que significa que não é necessário realizar nenhum investimento real para gerar e executar os contratos. Isso é especialmente benéfico, já que toda transação na rede

blockchain implica em uma taxa que deve ser paga no momento da geração do contrato.

5.3.1.2 Carteira:

Para interagir com a blockchain, cada usuário do sistema deve possuir sua própria carteira única. Essas carteiras não devem ser geradas dentro do nosso sistema, uma vez que são bens pessoais dos próprios usuários. Nosso sistema apenas gerencia os contratos utilizando o acesso fornecido pelas carteiras. Por essa razão, foi escolhido o Metamask como o gerenciador de carteiras. Essa ferramenta é a mais renomada do mercado e permite gerenciar não apenas valores em criptomoedas, mas também artigos NFT.

O Metamask será utilizado em nosso fluxo de usuário de duas formas. Primeiramente, ele será usado para a geração das credenciais e da carteira, garantindo que nosso sistema não tenha vínculo direto com as carteiras dos usuários. Assim, é possível aceitar carteiras externas para a execução das tarefas. A segunda função do Metamask em nosso sistema é possibilitar o envio de valores para dentro dos contratos. Isso nos permite terceirizar o envio de moedas e nos concentrar apenas no escopo responsável do nosso sistema, que é criar e gerenciar contratos dentro da blockchain.

5.3.1.3 Api de acesso à rede:

Para simplificar a comunicação com a blockchain Ethereum, foi feito uso do Alchemy, uma ferramenta de abstração que facilita a interação entre nossa aplicação e o servidor Ethereum. Especificamente, utilizando o SDK do Alchemy, uma biblioteca na qual pode ser utilizada apenas indicando as credenciais necessárias. Através do SDK, é possível invocar os métodos estabelecidos para executar as rotinas desejadas.

O SDK do Alchemy é uma biblioteca de software fornecida pelo Alchemy, uma plataforma de infraestrutura blockchain. O SDK, abreviação de Software Development Kit (Kit de Desenvolvimento de Software), é um conjunto de

ferramentas, bibliotecas e documentação que facilita o desenvolvimento e a integração de aplicativos com a rede Ethereum.

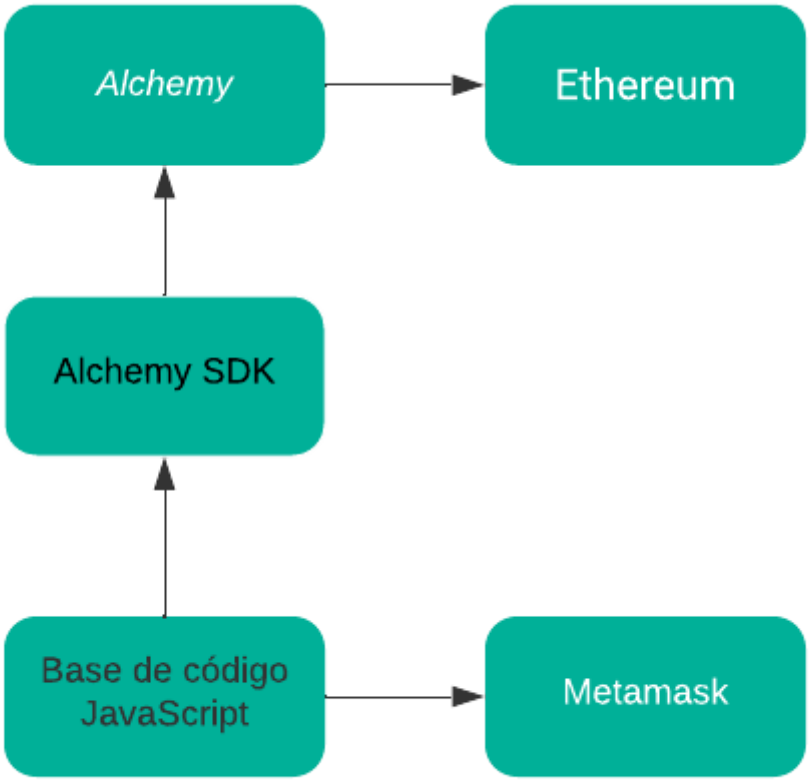
O SDK do Alchemy oferece uma série de recursos e funcionalidades para simplificar a interação com a blockchain Ethereum. Ele fornece uma abstração sobre as complexidades da comunicação direta com os nós da rede Ethereum, permitindo que os desenvolvedores se concentrem no desenvolvimento de seus aplicativos, em vez de se preocuparem com detalhes técnicos.

Com o SDK do Alchemy, os desenvolvedores podem se conectar à rede Ethereum, enviar transações, interagir com contratos inteligentes, recuperar dados da blockchain e muito mais. Ele fornece uma interface de programação de aplicativo (API) simples e intuitiva, que pode ser integrada a aplicativos desenvolvidos em várias linguagens de programação, como JavaScript, Python, Java e outros.

Além disso, o SDK do Alchemy oferece recursos adicionais, como gerenciamento de chaves e assinaturas criptográficas, cache de dados em tempo real, monitoramento de transações e análise de desempenho. Esses recursos são projetados para ajudar os desenvolvedores a criar aplicativos mais eficientes, seguros e escaláveis na blockchain Ethereum.

5.3.1.4 Desenho da arquitetura:

Figura 13 - Representação da arquitetura proposta



Fonte: Elaborada pelos autores (2023)

Na arquitetura do projeto tem como base de código em JavaScript que utiliza o SDK do Alchemy para se comunicar com a rede Ethereum, passando pelo servidor fornecido pelo Alchemy. O objetivo principal é desenvolver e gerenciar contratos inteligentes na rede Ethereum.

No código desenvolvido ocorre o armazenamento seguro das credenciais obtidas do Metamask. Essas credenciais são utilizadas para estabelecer a conexão com a rede Ethereum por meio do SDK do Alchemy. Esse código atua para gerenciar as operações do aplicativo blockchain, validando solicitações, executando a lógica de negócios e facilitando a interação com a rede Ethereum.

Os contratos inteligentes contêm a lógica de negócios e as regras específicas que serão executadas na rede Ethereum. O SDK do Alchemy permite que a aplicação interaja com esses contratos inteligentes, realizando operações como implantação, leitura e escrita de dados.

O SDK do Alchemy desempenha um papel crucial na arquitetura, fornecendo uma interface simplificada para a comunicação com a rede Ethereum. Ele permite que a aplicação envie transações, consulte informações da blockchain e interaja diretamente com os contratos inteligentes implantados.

Ao utilizar o SDK do Alchemy, a aplicação se beneficia da infraestrutura robusta fornecida pelo Alchemy, que inclui servidores otimizados e escaláveis para processar as transações e operações na rede Ethereum de forma eficiente.

Dessa forma, a arquitetura do projeto em JavaScript combina o Metamask, o SDK do Alchemy e o código desenvolvimento para criar uma aplicação blockchain completa e segura, com foco na criação, implantação e gerenciamento de contratos inteligentes na rede Ethereum.

5.3.2 Desenvolvimento do contrato:

Foi escolhido por utilizar a Ethereum como base para nossa solução de blockchain, devido à sua ampla adoção pela comunidade ao longo dos últimos anos. Além disso, a Ethereum nos proporciona um ambiente completo para trabalhar com Smart Contracts.

Para o desenvolvimento do nosso contrato, foi utilizada a linguagem Solidity, uma linguagem específica para contratos inteligentes na plataforma Ethereum. Essa linguagem nos oferece a flexibilidade e as funcionalidades necessárias para criar contratos robustos e seguros.

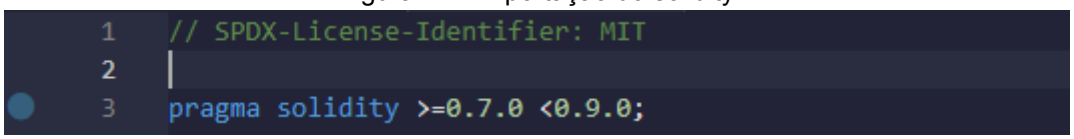
Para facilitar o processo de desenvolvimento, utilizou-se o Remix IDE, uma plataforma que nos permite escrever, testar e implantar nossos contratos de forma eficiente. O Remix IDE oferece recursos avançados, como depuração de código e simulação de transações, tornando o processo de desenvolvimento mais intuitivo e produtivo.

Essas escolhas estratégicas na utilização da Ethereum, Solidity e Remix IDE destacam nosso compromisso em construir uma solução sólida e confiável, aproveitando as melhores ferramentas e tecnologias disponíveis no ecossistema da blockchain.

Para garantir o funcionamento adequado do nosso arquivo Solidity (.sol), é necessário fornecer duas informações básicas:

- Licença: É preciso especificar a licença que será aplicada a este código. A lista completa de licenças pode ser encontrada no site spdx.org/licenses. No exemplo, foi utilizada a licença disponível em spdx.org/licenses/MIT.html.
- Versão do Solidity: É necessário indicar a versão do Solidity que é exigida para compilar a aplicação. No código de exemplo, foi definido que a versão mínima do Solidity deve ser 0.7.0 e que ela deve ser inferior a 0.9.0. Mais informações sobre a diretiva pragma podem ser encontradas em: docs.soliditylang.org/en/v0.8.20/layout-of-source-files.html#version-pragma.

Figura 14 - Importação do solidity

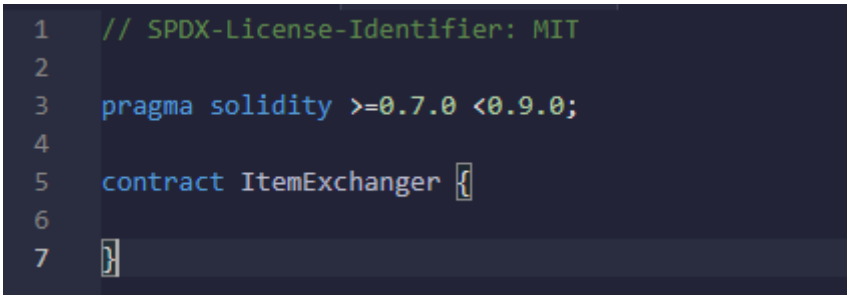


Fonte: Elaborada pelos autores (2023)

Essas informações são cruciais para garantir a compatibilidade e o correto funcionamento do arquivo Solidity. Ao fornecer a licença e a versão do Solidity, é assegurado que o código esteja em conformidade com as normas e requisitos específicos estabelecidos.

Uma vez que o arquivo .sol esteja configurado com essas informações, os erros relacionados devem ser eliminados. Após isso é possível prosseguir com a criação do contrato. A estrutura do contrato é semelhante à de uma classe em Java ou C#, e requer o uso da palavra-chave "contract", seguida pelo nome que será atribuído ao contrato.

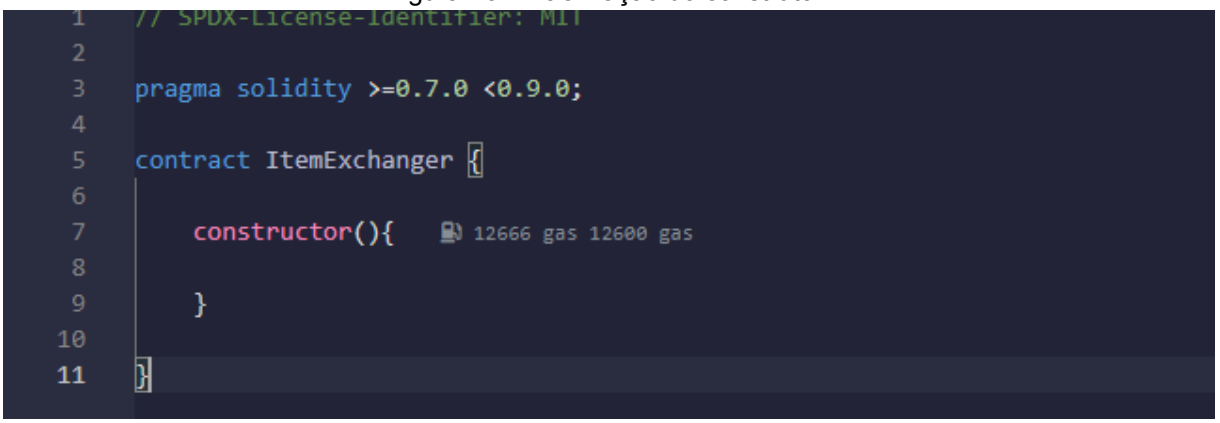
Figura 15 - Estrutura básica do contrato



Fonte: Elaborada pelos autores (2023)

Com o contrato devidamente definido, a etapa seguinte é preenchê-lo com propriedades, funções, construtores e outros elementos. Começando pelo construtor:

Figura 16 -Inicialização do construtor



Fonte: Elaborada pelos autores (2023)

No exemplo acima, é demonstrada a estrutura básica de um contrato com a inclusão do construtor. O construtor é um bloco especial de código que é executado uma única vez quando o contrato é implantado na blockchain. Ele é utilizado para inicializar variáveis, definir valores padrão e realizar outras tarefas necessárias para o funcionamento do contrato.

Com a adição da propriedade "owner" (dono) e a utilização do construtor para preencher essa informação com o endereço da carteira que está criando o contrato, pode-se melhorar ainda mais o contrato. Conforme o exemplo da figura 17:

Figura 17 - Configuração do dono do contrato no construtor

Fonte: Elaborada pelos autores (2023)

Nesse exemplo, foi adicionada a propriedade "owner" do tipo "address" (endereço) como uma variável privada. Isso permite que apenas o contrato e os métodos internos possam acessar e alterar o endereço do dono do contrato.

No construtor, é utilizada a variável global "msg.sender" para obter o endereço do remetente da transação que está implantando o contrato na blockchain. Dessa forma, o endereço do dono é atribuído à propriedade "owner" durante a criação do contrato.

Agora, com a inclusão da propriedade "owner", é possível identificar facilmente quem é o dono do contrato, o que pode ser útil para implementar lógicas específicas e atribuir permissões especiais dentro do contrato.

Agora com o proprietário do contrato definido, é possível avançar para o destinatário do item que será transferido. Para isso, é necessário criar uma nova propriedade denominada "buyer" (comprador) e implementar a função para atribuir seu valor:

Figura 18 - Criação da primeira função do contrato SetBuyer

```
// SPDX-License-Identifier: MIT

pragma solidity >=0.7.0 <0.9.0;

contract ItemExchanger {

    address private owner;
    address private buyer;

    constructor(){    212098 gas 187600 gas
        owner = msg.sender;
    }

    function SetBuyer(address buyerAddress) public {    29038 gas
        require(buyer == address(0), "Comprador ja preenchido, nao e possivel alterar.");
        require(owner == msg.sender, "Apenas o dono do contrato pode preencher.");

        buyer = buyerAddress;
    }

}
```

Fonte: Elaborada pelos autores (2023)

Aprimorando o código, é definida a propriedade "buyer" como privada e do tipo "address", seguindo o mesmo padrão da propriedade "owner". Além disso, é criada a função "setBuyer", que espera um endereço da carteira do comprador e tem o modificador de acesso público. Veja o exemplo abaixo:

Dentro da função, são realizadas duas validações para garantir a segurança do nosso contrato. Primeiro, se verifica que o comprador deste contrato ainda não foi preenchido, pois uma vez preenchido, não pode ser alterado. Em seguida, verfica se a carteira que está tentando preencher é a carteira do dono do contrato, pois somente ele deve ter permissão para preencher o comprador.

Finalmente, após todas as validações, posso atribuir o endereço recebido como sendo o comprador do contrato, o que resulta no travamento do contrato entre o comprador e o vendedor, já que não será possível alterar essas informações posteriormente.

Para o próximo passo, será necessário estabelecer um método para armazenar as informações do item que está sendo negociado e seu respectivo valor. Para isso, é criada duas novas propriedades: "item" (referente ao item) e "itemValue" (referente ao valor do item). O valor do item será medido em Wei, que é uma unidade de medida utilizada para facilitar transações operacionais na Ethereum, onde 10^18 ou 1,000,000,000,000,000,000 wei é equivalente a 1 ether. É possível

encontrar um conversor de Wei para Gwei para Ether no site eth-converter.com. Internamente este contrato mantém o balanço e o valor do item na unidade de medida Wei, no entanto, ao utilizar a carteira para transferência de valores, a unidade é em Ether. Devido ao alto valor da moeda é sempre necessário tomar precauções com a unidade de medida utilizada a todo momento.

Figura 19 - Criação das variáveis do contrato e incremento do construtor

```
address payable private owner;
address payable private buyer;
string public item;
uint256 public itemValue;
uint256 public balance;
bool public isFinalized;

constructor(string memory itemToSell, uint256 itemValueToSell){    infinite gas 488000 gas
    owner = payable(msg.sender);
    item = itemToSell;
    itemValue = itemValueToSell;
}
```

Fonte: Elaborada pelos autores (2023)

Ambas as informações são necessárias durante a criação do contrato. Portanto, no construtor do contrato, são solicitadas essas informações como parâmetros que preenchem as respectivas propriedades.

Para concluir, é necessário criar um método que seja capaz de receber Ether e armazená-lo no contrato. É criado o método e a propriedade correspondente para armazenar o Ether (que também será contabilizado em Wei). A palavra-chave escolhida para esta função é "receive", e indica que este é o método responsável por gerenciar o recebimento de Ether pelo contrato. Além disso, o método é marcado como "payable" para que ele possa enviar ou receber Ether e como "external" para informar que ele só pode ser chamado de fora do contrato, reduzindo assim o custo da taxa de mineração do bloco.

Figura 20 - Criação da segunda função do contrato receive

```
uint256 public balance;

receive() payable external{
    require(msg.sender == buyer, "Apenas o comprador pode depositar");
    balance += msg.value;
}
```

Fonte: Elaborada pelos autores (2023)

É inserida também uma validação para garantir que apenas o comprador possa depositar fundos no contrato.

Agora, é preciso adicionar uma regra para que, quando o valor inserido no contrato for maior ou igual ao valor do item, o item seja transferido para o comprador e o valor armazenado no contrato seja enviado ao dono do contrato.

Para permitir a transferência de fundos para o dono do contrato, é necessário marcar a variável "owner" como "payable". Além disso, se houver um valor excedente transferido, deve-se tornar a propriedade "buyer" também "payable" para que o valor excedente possa ser devolvido ao comprador.

Aqui está o código completo com todas as alterações mencionadas:

Figura 21 - Visão completa do código do contrato

```
// SPDX-License-Identifier: MIT

pragma solidity >=0.7.0 <0.9.0;

contract ItemExchanger {

    address payable private owner;
    address payable private buyer;
    string public item;
    uint256 public itemValue;
    uint256 public balance;
    bool public isFinalized;

    constructor(string memory itemToSell, uint256 itemValueToSell){    infinite gas 489600 gas
        owner = payable(msg.sender);
        item = itemToSell;
        itemValue = itemValueToSell;
    }

    function SetBuyer(address buyerAddress) public {    29082 gas
        require(buyer == address(0), "Comprador ja preenchido, nao e possivel alterar.");
        require(owner == msg.sender, "Apenas o dono do contrato pode preencher.");

        buyer = payable(buyerAddress);
    }

    receive() payable external{    undefined gas
        require(msg.sender == buyer, "Apenas o comprador pode depositar");
        balance += msg.value;
        if(balance >= itemValue){
            owner.transfer(itemValue);
            balance = balance - itemValue;
            buyer.transfer(balance);
            balance = 0;
            isFinalized = true;
        }
    }
}
```

Fonte: Elaborada pelos autores (2023)

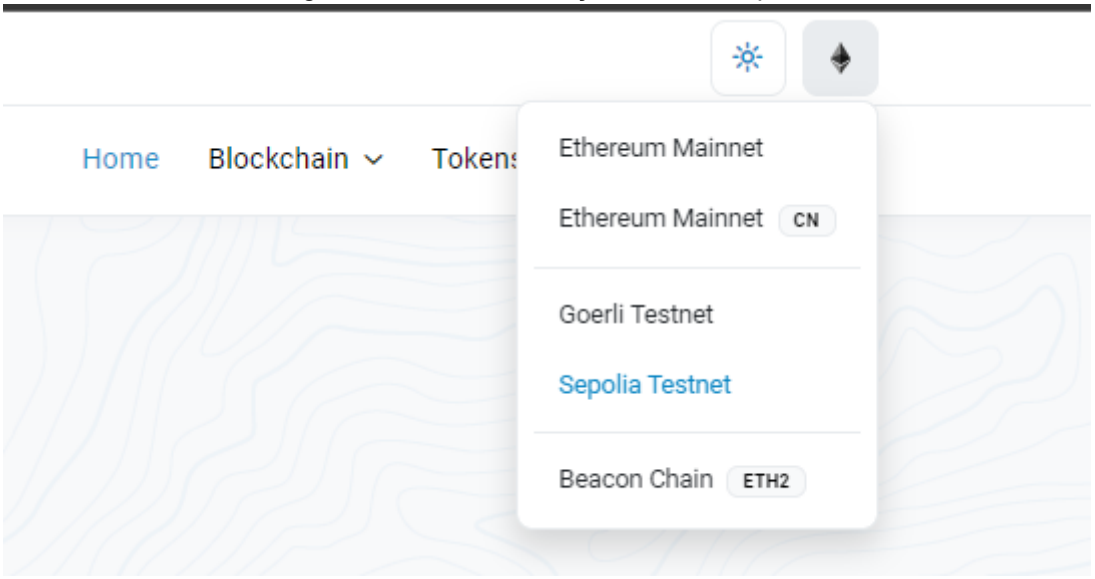
Foi adicionada a variável "isFinalized" para indicar que o contrato foi finalizado e o pagamento foi efetuado com sucesso. Com este código, é possível criar um contrato, definir o proprietário, transferir fundos para o contrato e efetuar saques quando as condições forem atendidas.

5.3.3 Implementação dos recursos de rastreabilidade:

A rastreabilidade é garantida devido à utilização da blockchain. Ela assegura que, uma vez que um bloco seja registrado na rede, não será alterado. No caso específico da Ethereum, pode-se utilizar o site etherscan.io para pesquisar qualquer

contrato na rede, desde que se possua o identificador correspondente. Neste exemplo, é utilizada a rede de testes chamada Sepolia, a qual está disponível no site mencionado anteriormente, no seguinte local:

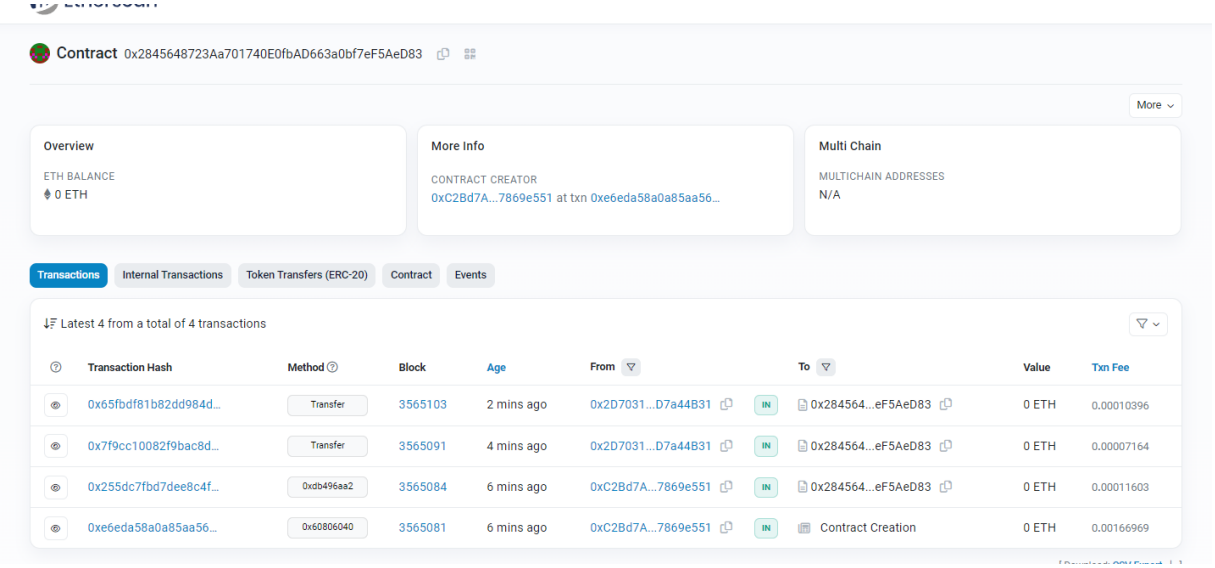
Figura 22 - Visão da seleção da rede Sepolia



Fonte: Ethercan (2023)

Ao pesquisar um endereço, é possível ter acesso a todos os dados disponíveis. Por exemplo:

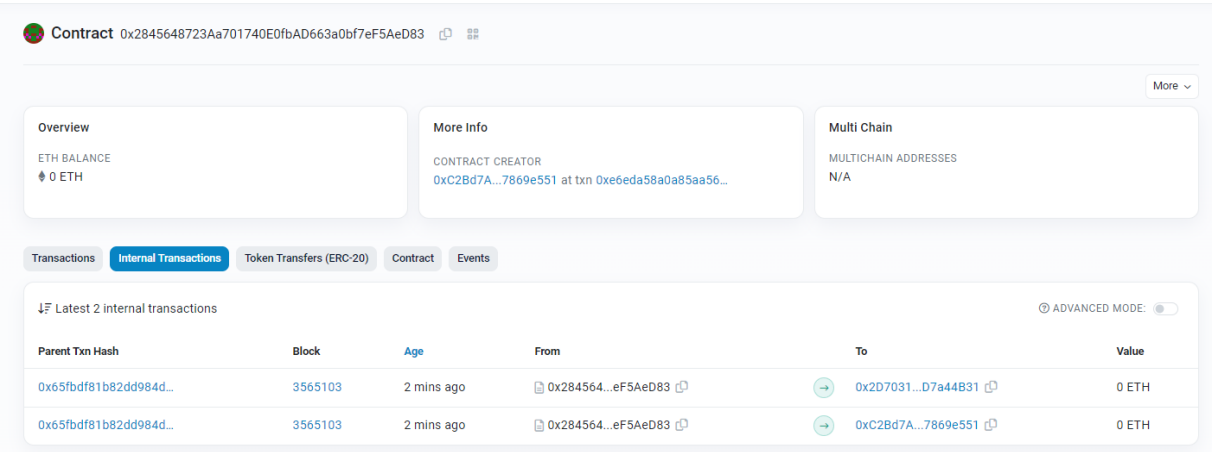
Figura 23 - Exemplo de dados de um contrato



Fonte: Ethercan (2023)

Neste exemplo do código acima em execução, pode-se verificar o seguinte fluxo de eventos: o contrato foi criado, em seguida o comprador foi informado. Posteriormente, ocorreram duas transferências: uma de 50000 Wei e outra de 100000 Wei, para um item que custava 100000 Wei. Isso resultou no contrato entregando o valor do item ao dono e devolvendo o restante ao comprador. Na imagem acima, pode-se observar as transferências recebidas. No entanto, as transferências feitas pelo contrato são exibidas em uma aba separada, conforme mostrado na imagem abaixo:

Figura 24 - Visualização de um exemplo da aba de transferências



Fonte: Ethercan (2023)

Todos os blocos da blockchain seguem esse comportamento, no qual todos

os dados do bloco são acessíveis. No entanto, é necessário ter o endereço do bloco para acessá-los. Assim como em outros aspectos da blockchain, se houver perda de acesso à carteira ou ao contrato de qualquer forma, não há meios de recuperá-lo.

5.3.4 Autenticação e verificação:

No projeto, a segurança e a autenticação são fundamentais para garantir a integridade das transações na rede Ethereum. Para isso, é utilizado o Metamask, uma carteira digital e extensão do navegador que permite aos usuários gerenciar suas chaves privadas e interagir com aplicativos descentralizados.

O Metamask atua como uma camada de segurança adicional, pois armazena as chaves privadas do usuário localmente, criptografadas e protegidas por uma senha. Essas chaves privadas são utilizadas para assinar digitalmente as transações antes de serem enviadas para a rede Ethereum, garantindo a autenticidade e a integridade dos dados.

Quando um usuário interage com a aplicação blockchain, ele é solicitado a conectar sua carteira Metamask, que permite a autenticação segura e confiável. Ao conectar a carteira, o Metamask realiza um processo de autenticação, solicitando ao usuário que forneça sua senha para desbloquear as chaves privadas e permitir a assinatura de transações.

Dessa forma, a segurança e autenticação do projeto são asseguradas pelo Metamask, que protege as chaves privadas dos usuários e garante que apenas transações assinadas digitalmente por essas chaves privadas sejam enviadas para a rede Ethereum. Isso oferece uma camada adicional de segurança, impedindo o acesso não autorizado às carteiras e transações na blockchain.

O Metamask não é a única ferramenta que faz o gerenciamento e criação de carteiras, mas dentre as opções é a mais simples, segura e confiável dentre as mais utilizadas. A autenticação na blockchain se dá pelo endereço das carteiras envolvidas na transação. Utilizando um aplicativo qualquer de criação e manipulação de carteira é possível criar e gerenciá-las.

Um fator importante a se levar em consideração é que as credenciais da carteira são gerenciadas pelas aplicações portanto a aplicação desenvolvida neste

projeto não se responsabiliza por guardar nem armazenar credenciais, as mesmas devem ser acessadas quando necessárias e removidas após o uso da aplicação, deixando a responsabilidade do armazenamento dessas chaves para as aplicações de terceiros.

5.4 AVALIAÇÃO

A avaliação foi realizada por meio de um questionário aplicado tanto a pessoas com conhecimento técnico no assunto quanto àquelas sem esse conhecimento, com o objetivo de demonstrar se o desenvolvimento atende aos critérios estabelecidos. A seguir, apresentam-se as etapas da avaliação do sistema proposto.

5.4.1 Cenário da avaliação

O sistema foi criado e desenvolvido para auxiliar usuários na criação e acompanhamento de contratos inteligentes criados na blockchain.

O protótipo do sistema foi apresentado por meio de um vídeo explicativo mostrando o funcionamento planejado para o sistema bem como um contrato inteligente sendo criado e utilizado.

O número de avaliadores foi de 12 possíveis usuários da plataforma. As pessoas envolvidas nesta avaliação possuem diferentes profissões e faixas etárias, representando uma ampla parcela de perfis para usuários do sistema.

5.4.2 Elaboração do questionário

Para Silva e Menezes (2005, p. 33) o questionário “deve ser objetivo, limitado em extensão e estar acompanhado de instruções As instruções devem esclarecer o propósito de sua aplicação, ressaltar a importância da colaboração do informante e facilitar o preenchimento”.

A elaboração das perguntas do questionário foi feita a partir da análise dos seguintes tópicos:

- Objetivos do trabalho.
- Problemática do trabalho.
- Justificativa do trabalho.
- Requisitos da modelagem do protótipo.
- Melhorias ao protótipo.

Após feita essa análise, foram elaboradas perguntas de múltipla escolha (questões fechadas) e um pergunta dissertativa (questão aberta) para o usuário expor o que poderia ser melhorado no sistema.

Para a formulação das perguntas, foram levados em consideração alguns critérios estabelecidos por Gil (2008), sendo eles:

- As perguntas devem ser formuladas de maneira clara, concreta e precisa.
- A pergunta deve possibilitar uma única interpretação;
- A pergunta não deve sugerir respostas;
- As perguntas devem referir-se a uma única ideia de cada vez.

Nas perguntas fechadas, foi utilizado a escala Likert, pois utiliza uma elaboração simples e de caráter ordinal, não medindo o quanto uma atitude é mais ou menos favorável Gil(2008). Como opções de resposta, foi utilizado as seguintes alternativas: Concordo totalmente, Concordo parcialmente, Indiferente, Discordo parcialmente e Discordo totalmente.

Tabela 9 - Perguntas do questionário de avaliação

Requisito	Perfil do usuário	Pergunta	Tipo de pergunta
	Usuário	O contrato apresentado atende as necessidades mínimas de um contrato inteligente?	Fechada
	Usuário	Por ser um contrato inteligente automaticamente executado, não tem a intervenção de terceiros na negociação, isso é um ponto positivo?	Fechada

Requisito	Perfil do usuário	Pergunta	Tipo de pergunta
RF006	Usuário	Poder acompanhar o andamento do contrato sem necessidade de acesso ou conhecimento da blockchain é um diferencial?	Fechada
RF002	Usuário	Ter um catálogo de contratos pré-configurados para o usuário escolher é um ponto positivo?	Fechada
RF004	Usuário	Ter um marketplace para pesquisa e negociação de contratos é um diferencial para a plataforma?	Fechada
RF001	Usuário	O fato de o sistema permitir acesso apenas aos usuários cadastrados, torna-o mais seguro?	Fechada
	Usuário	O sistema pode facilitar a transação de qualquer bem entre usuários?	Fechada
	Usuário	Quais melhorias, sugestões e funcionalidades podem ser aplicadas no sistema?	Aberta

Fonte: Elaborada pelos autores (2023)

A tabela 9 apresenta as perguntas que compõem o questionário, nela é possível perceber a quais requisitos cada pergunta está ligada e para qual perfil de usuário a mesma é destinada. Algumas perguntas não estão ligadas a requisitos do sistema, porém foram julgadas relevantes para a avaliação deste trabalho.

5.4.3 Aplicação do questionário

A aplicação do questionário resultou em dados que ao passarem por uma análise determinou se o sistema desenvolvido atingiu os objetivos iniciais e as expectativas. Devido a isso para que os avaliadores pudessem ter uma melhor visão do sistema a aplicação do questionário seguiu alguns passos:

- Apresentação dos propósitos do sistema.
- Apresentação do protótipo e suas funcionalidades.
- Aplicação do questionário.

O questionário foi desenvolvido utilizando a ferramenta Google Forms, a escolha foi feita baseada nas vantagens oferecidas pela mesma:

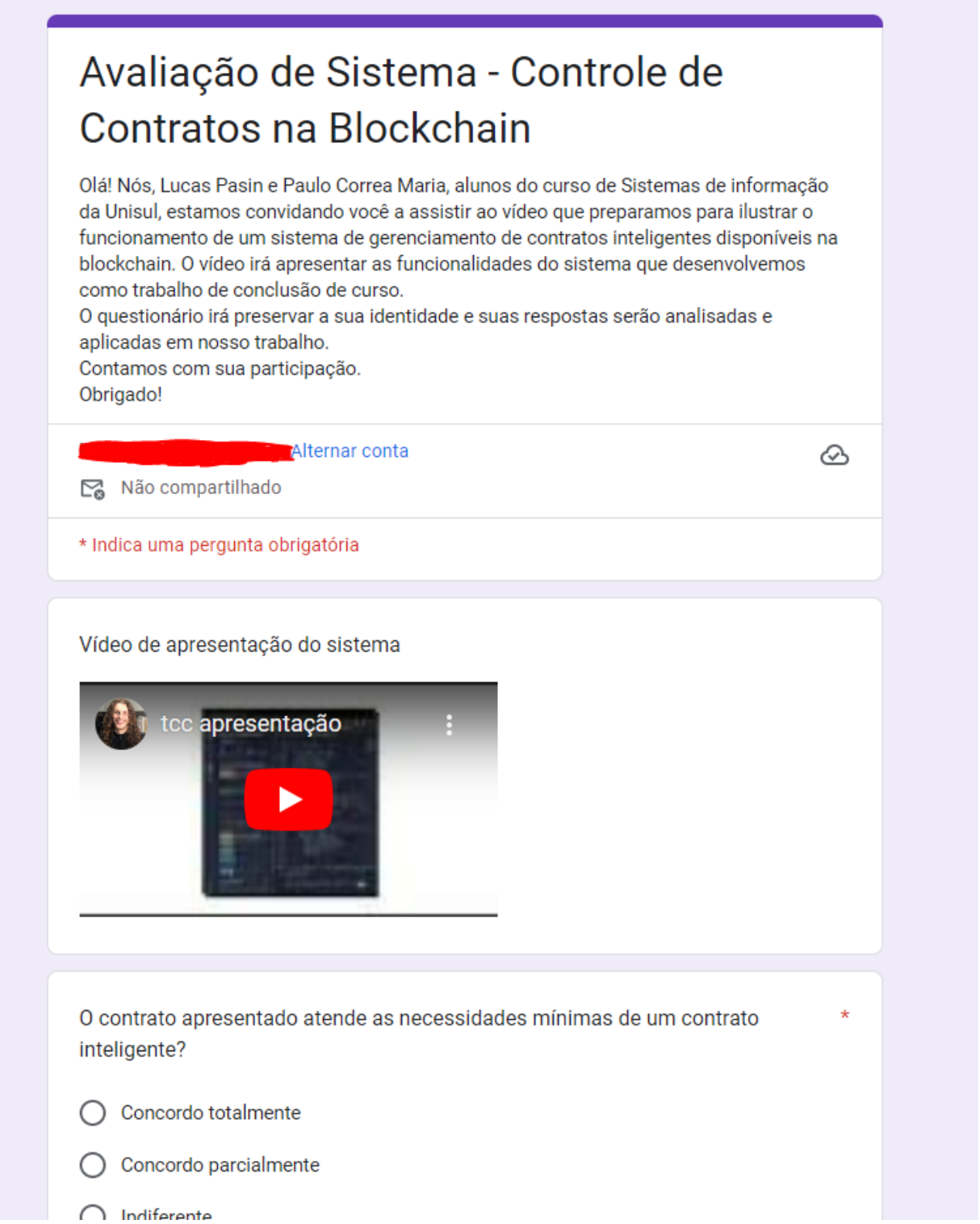
- Ferramenta gratuita.
- Fácil acesso.
- Interface simples e amigável.
- Fácil manuseio do questionário.

5.4.4 Análise dos resultados

Nesta seção, é apresentado o resultado do questionário realizado para validação do sistema desenvolvido.

De acordo com os objetivos estabelecidos, este questionário pretende identificar se o sistema proposto alcançou os mesmos.

Figura 25 - Formulário de avaliação



Fonte: Google Form (2023)

Na figura 25 se demonstra o questionário enviado para que os participantes pudessem avaliar o sistema proposto. No início existe um enunciado para introdução e explicação, em seguida um vídeo com a apresentação do protótipo e por fim as

perguntas.Ao final dos questionários, uma pergunta aberta onde foi utilizada para sugestão de melhorias, as respostas dessa questão foram utilizadas para elaboração da seção de trabalhos futuros.

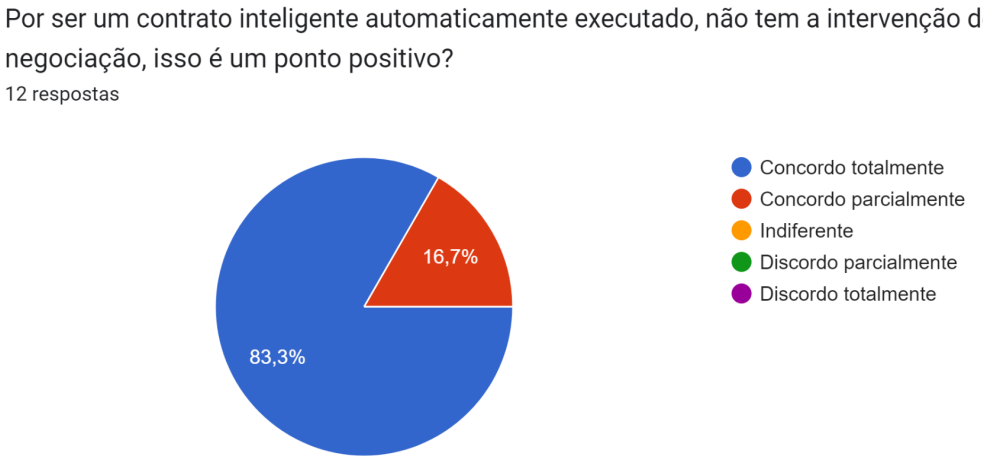
A seguir são apresentadas as respostas do questionário aplicado aos possíveis usuários do sistema:

Figura 26 - Resposta questão 1

Fonte: Google Form (2023)

Na figura 26 é apresentado o gráfico com as respostas do questionário destinado aos possíveis usuários do sistema sobre os requisitos para um contrato inteligente onde é possível visualizar que 75% das pessoas entrevistadas estão confortáveis com o contrato apresentado e 25% esperava algo mais completo, trazendo assim pontos para melhoria.

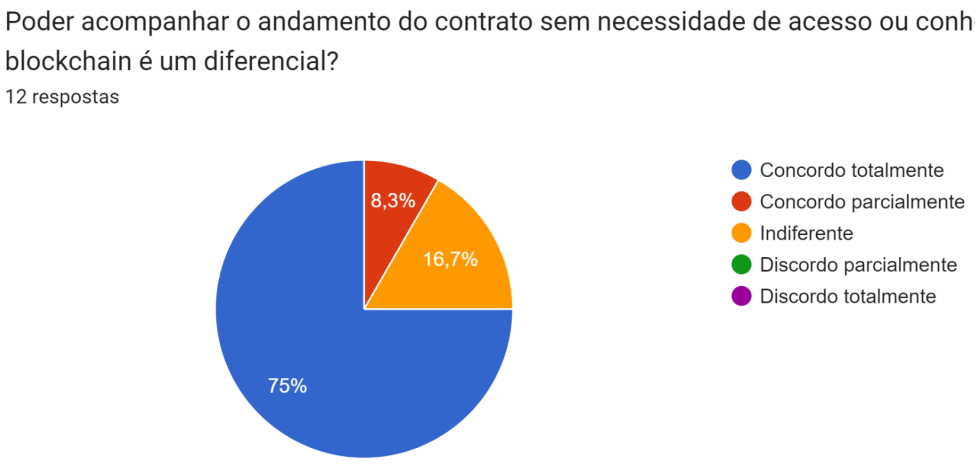
Figura 27 - Resposta questão 2



Fonte: Google Form (2023)

Na figura 27 é apresentado o gráfico com as respostas do questionário destinado aos possíveis usuários do sistema sobre não haver intervenção de terceiros na execução do contrato e é possível perceber que a maioria dos entrevistados observou como totalmente positivo porém tivemos algumas objeções apontando a falta da segurança que um terceiro traz para a negociação.

Figura 28 - Resposta questão 3

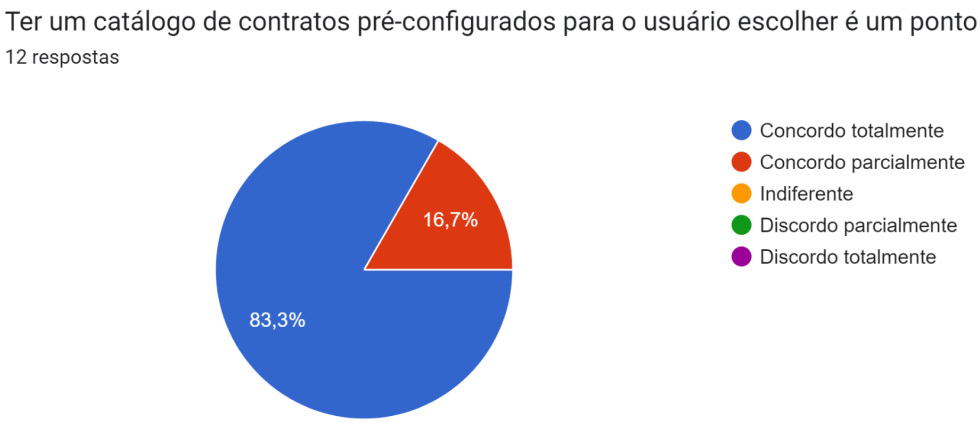


Fonte: Google Form (2023)

Na figura 28 é apresentado o gráfico com as respostas do questionário destinado aos possíveis usuários do sistema sobre a possibilidade de usar contratos

sem conhecimento técnico da blockchain e podemos observar que a maior parcela dos entrevistados foi favorável à rastreabilidade dos contratos pela plataforma e alguns sentiram que não necessariamente é um diferencial.

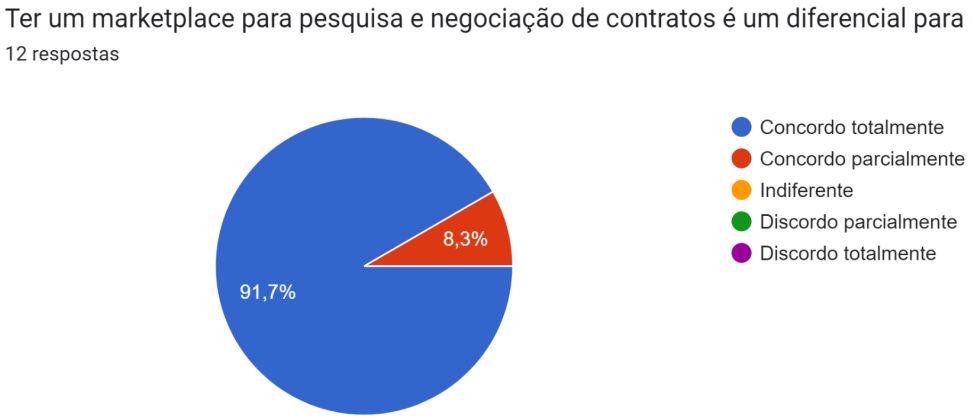
Figura 29 - Resposta questão 4



Fonte: Google Form (2023)

Na figura 29 é apresentado o gráfico com as respostas do questionário destinado aos possíveis usuários do sistema sobre o catálogo de contratos disponível e a maioria dos entrevistados sentiu que poder escolher de um modelo de contrato é um ponto positivo para a plataforma.

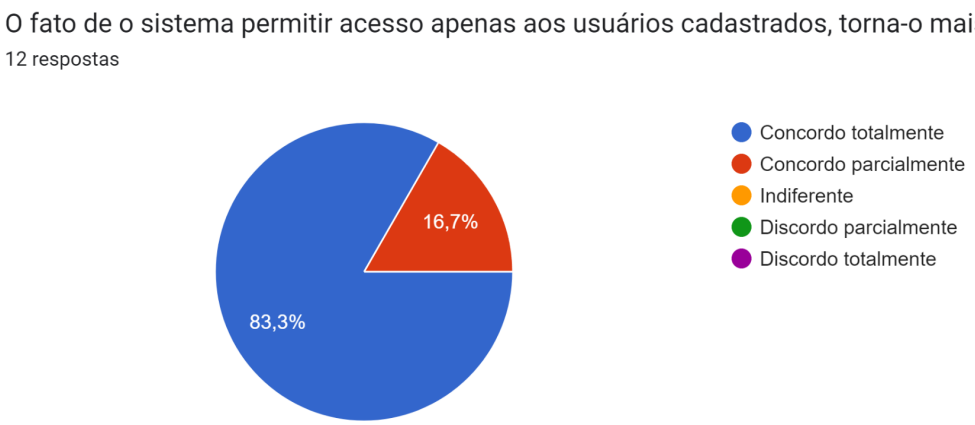
Figura 30 - Resposta questão 5



Fonte: Google Form (2023)

Na figura 30 é apresentado o gráfico com as respostas do questionário destinado aos possíveis usuários do sistema sobre o marketplace e observamos que a maior parcela dos entrevistados concorda com o marketplace presente na plataforma ser um diferencial para seu funcionamento.

Figura 31 - Resposta questão 6



Fonte: Google Form (2023)

Na figura 31 é apresentado o gráfico com as respostas do questionário destinado aos possíveis usuários do sistema sobre a permissão de acesso apenas a usuários registrados é visível a grande aceitação do login como forma de garantir a segurança da aplicação porém uma parcela dos entrevistados sentiram a necessidade de alguma forma de segurança adicional.

5.4.5 Conclusão da avaliação

A partir da análise das respostas dos questionários, verifica-se que, de acordo com a primeira pergunta, todos os participantes concordaram que o sistema proposto atende aos requisitos esperados para um contrato inteligente. Dentre eles, 75% concordaram completamente, enquanto 25% concordaram parcialmente com essa afirmação.

Isso demonstra que, no mínimo, 25% dos participantes conseguiram compreender o funcionamento do sistema e o conceito da tecnologia, identificando pontos de melhoria e lacunas no sistema. Esses resultados evidenciam que o

sistema, juntamente com o conhecimento associado a ele, conseguiu envolver pessoas de diferentes setores da sociedade no debate sobre essa tecnologia.

Quanto às demais questões, seu objetivo era avaliar a capacidade de análise e questionamento dos participantes em relação aos temas relacionados aos contratos inteligentes. É importante ressaltar que mais de 50% dos participantes não possuíam conhecimento prévio sobre o assunto.

Ao analisarmos as respostas, observamos que o percentual de concordância parcial com as questões variou entre 8% e 16%. Isso indica que uma discussão entre os participantes sobre o assunto não resultaria em um consenso, mas sim em um debate com opiniões divergentes.

Em relação ao protótipo desenvolvido para este trabalho, o sistema proposto recebeu aprovação dos avaliadores. Embora seja um protótipo com limitações, focado apenas na avaliação do treinamento e com funcionalidades básicas, é possível concluir que esse tipo de sistema pode ser útil na facilitação da troca de bens entre usuários e trazer maior confiabilidade para o mercado de blockchain como um todo.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões envolvendo o sistema proposto. Além disso, são apresentadas sugestões de novas funcionalidades e ideias que poderão ser implementadas no sistema como parte de trabalhos futuros.

6.1 CONCLUSÕES

O sistema foi desenvolvido com o intuito de demonstrar uma prova de conceito do uso de blockchain e contratos inteligentes em um ambiente de compra e venda de bens entre usuários. Com seu desenvolvimento foi possível provar o grande benefício que uma ferramenta de criação de contratos inteligentes em blockchain podem trazer para o dia a dia, economizando tempo, dinheiro e dando mais confiabilidade para as transações feitas entre pessoas.

Com o desenvolvimento do capítulo da revisão bibliográfica, conhecimentos foram adquiridos, favorecendo a amplitude de visão sobre o tema, agregando mais insumos para a elaboração, confecção da solução, ajudando a aproximar de uma solução mais adequada, robusta e eficiente para a proposta do trabalho.

No processo de desenvolvimento do software, fizemos primeiramente o levantamento de requisitos funcionais e não funcionais, regras de negócio, casos de uso, modelo da estrutura e integração com sistemas terceiros. O propósito da modelagem do sistema foi desmistificar as características, estrutura e o objetivo do sistema, para assim melhor produzir uma solução e alcançar o objetivo proposto.

Após a etapa de definições e modelagem, iniciou-se o desenvolvimento da prova de conceito, sendo usada inúmeras ferramentas e integrações tanto para auxiliar o desenvolvimento como para minimizar a complexidade da ferramenta, e com muito aprendizado no final conseguimos provar o que pretendemos conforme os objetivos propostos.

Após a finalização do desenvolvimento do sistema o mesmo foi avalizado por pessoas de dentro e fora da área de tecnologia, provando que o sistema pode ser abrangente e útil não só para pessoas com expertise na área de tecnologia como

também para pessoas que são leigas no assunto. E por mais que o sistema tenha muitos pontos de melhoria ele ainda sim consegue fazer o que foi proposto e facilitar as transações entre duas entidades.

Chegando ao final, temos que o objetivo foi alcançado em demonstrar as possibilidades da tecnologia, garantindo mais velocidade, segurança e democratização nos processos em que foi aplicado. Muitas dificuldades apareceram pelo caminho, desenvolver ao mesmo tempo que aprende sobre a tecnologia, ferramentas e os sistemas que foram utilizados para o funcionamento e desenvolvimento da solução.

Com esse sistema abrimos precedente para o desenvolvimento de ferramentas que possibilitam a otimização de inúmeros processos do nosso dia a dia, assim atingindo a resolução para a problemática proposta neste trabalho.

6.2 TRABALHOS FUTUROS

O trabalho inicial propõe as funcionalidades necessárias apenas para a demonstração da ferramenta, porém no decorrer do desenvolvimento do trabalho e com as informações e opiniões dos usuários, percebeu-se que novas funções poderiam ser implementadas no mesmo. Devido ao curto espaço de tempo para a modelagem e desenvolvimento do sistema, essas funcionalidades serão efetuadas com parte de trabalhos futuros como mostra a tabela 10.

Tabela 10 - Trabalhos futuros

Funcionalidade	Descrição
Transferências dentro do sistema	Criar a função de enviar valores diretamente por dentro do sistema
Armazenamento das credenciais da carteira dentro do sistema	Permitir que as credenciais possam ser armazenadas dentro do sistema de forma segura, sem precisar de uma aplicação terceira para o uso.
Criar um login interno da plataforma	Ter um login próprio além de apenas as chaves de carteira
Criar uma interface gráfica para o usuário final	Ter uma interface de interação com o qual o usuário final possa interagir

Funcionalidade	Descrição
Ter o registro das ações que são feitas	Adicionar logs no sistema que possibilitem verificar quais as ações que foram feitas dentro do sistema
Permitir alteração dos dados do contrato	Permitir que algumas informações do contrato sejam alteradas após a criação do mesmo
Criar templates de contratos	Permitir criar contratos a partir de templates pré definidos no sistema

Fonte: Elaborada pelos autores (2023)

A partir das ideias citadas, o sistema pode tornar-se mais completo e eficiente contribuindo ainda mais para quem fizer o seu uso.

REFERÊNCIAS

NAKAMOTO, Satoshi. Bitcoin: “A Peer-to-Peer Electronic Cash System .”. Bitcoin.org, 2008.

HUAIQING Wang, KUN Chen, DONGMING Xu. A maturity model for blockchain adoption. Disponível em: https://link.springer.com/content/pdf/10.1186/s40854-016-0031-z.pdf. Acesso em: 10 de outubro de 2021.

TAPSCOTT Don. How the blockchain is changing money and business. Disponível em: https://www.ted.com/talks/don_tapscott_how_the_blockchain_is_changing_money_and_business. Acesso em: 10 de outubro de 2021.

ULRICH Fernando. Bitcoin - A moeda na era digital. Brasília, 2016. Disponível em: https://books.google.com.br/books?id=s-lDDwAAQBAJ&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false Acesso em: 10 de outubro de 2021.

Dean Sam. $69 million for digital art? The NFT craze explained. Disponível em: https://www.latimes.com/business/technology/story/2021-03-11/nft-explainer-crypto-trading-collectible. Acesso em: 05 de novembro de 2021.

Maung Wai Yan Maung, Dong Naipeng, Griffith Guangdong Bai,Dong Jin Song Formal Analysis of a Proof-of-Stake Blockchain. Disponível em: https://baigd.github.io/files/ICECCS2018b.pdf. Acesso em: 08 de novembro de 2021.

SAKAMOTO SARAH GOMES, SEGURANÇA, PRIVACIDADE E BLOCKCHAIN NO CONTEXTO DE INTERNET DAS COISAS. Curitiba, 2020 . Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/16143/1/CT_CEIOT_II_2019_10.pdf. Acesso em: 12/11/2021

Lewis Popovski and George Soussou , A Brief History of Blockchain, 2018. Disponível em: https://www.pbwt.com/content/uploads/2018/05/010051804-Patterson.pdf. Acesso em: 10 de novembro de 2021.

Rose Jacobs, The Good and Bad of Blockchain, 2018. Disponível em: https://www.chicagobooth.edu/review/good-and-bad-blockchain. Acesso em: 18 de novembro de 2021.

William Ribeiro de Paula, Blockchain e sua Correlação com Teoria dos Jogos, 2019. Disponível em: https://www.ime.usp.br/~map/tcc/2019/WilliamRibeiroV1.pdf. Acesso em: 18 de novembro de 2021.

Welight. Disponível em: https://welight.co/about-us, Acesso em: 21 de novembro de 2021.

UNICAFES. Disponível em: https://www.unicafes.org.br/, Acesso em: 20 de novembro de 2021.

MoedasSeeds. Disponível em: https://moedaseeds.com/, Acesso em: 18 de novembro de 2021.

DE ARAUJO Guilherme Patricio, Blockchain como modelo de inovação na sistematização das informações para exportação de café, 2021. Disponível em: https://repositorio.ufscar.br/bitstream/handle/ufscar/14626/Araujo%2c%20Guilherme%20Patricio%20de.pdf?sequence=1&isAllowed=y , Acesso em: 22 de novembro de 2021.

Binance. Disponível em: https://www.binance.com/pt-BR/blog/all/por-que-o-bitcoin-vale-tanto-421499824684901557, Acesso em: 28 de novembro de 2021.

Bitcoin.org. Disponível em: https://bitcoin.org/pt_BR/faq#como-alguem-consegue-bitcoins, Acesso em: 28 de novembro de 2021.

Reis, Luis Filipe Gold Coelho de Almeida dos. Disponível em: https://www.acervodigital.ufpr.br/handle/1884/55424, Acesso em: 28 de novembro de 2021.

Schroeder Stan. Disponível em: https://mashable.com/article/gods-unchained-trailer, Acesso em: 28 de novembro de 2021.

Assange, J. (2012). Liberdade e o Futuro da internet Cypherpunks . Sao paulo: Boitempo.

Business Wire. Disponível em: https://venturebeat.com/2019/04/16/enjin-is-creating-a-real-life-ready-player-one-and-its-powered-by-blockchain/, Acesso em: 28 de novembro de 2021.

VOTEM. Disponível em: https://www.votem.com/. Acesso em: 01 maio 2023.

Serto. Disponível em: https://www.serto.id/. Acesso em: 02 maio 2023.

VeChain. How It Works. Disponível em: https://www.vechain.org/. Acesso em: 02 maio 2023.

ETHEREUM Whitepaper. [S. l.], 2014. Disponível em: https://ethereum.org/en/whitepaper/. Acesso em: 2 jun. 2023.

SZABO, Nick. Smart Contracts: Building Blocks for Digital Markets. [S. l.], 30 dez. 2006. Disponível em: https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LO

Twinterschool2006/szabo.best.vwh.net/smart_contracts_2.html. Acesso em: 2 jun. 2023.

SOLIDITY. [S. l.], 2016. Disponível em: https://docs.soliditylang.org/en/v0.8.20/. Acesso em: 2 jun. 2023.

ANTONOPOULOS, Andreas. Mastering Ethereum. [S. l.: s. n.], 2018.

MIRANDA, J. C. de; ZUCHI, J. D. TECNOLOGIA BLOCKCHAIN: a disrupção na indústria financeira. Revista Interface Tecnológica, [S. l.], v. 15, n. 2, p. 457–469. Disponível em: https://revista.fatectq.edu.br/interfacetecnologica/article/view/376. Acesso em: 10 jun. 2023.

SENA, L. G. B. de; DIAN, M. de O. A CRIPTOMOEDA: como obtê-la através da mineração. Revista Interface Tecnológica, [S. l.], v. 17, n. 2, p. 364–375, 2020. Disponível em: https://revista.fatectq.edu.br/interfacetecnologica/article/view/1053. Acesso em: 10 jun. 2023.

SAPRA, N.; SHAIKH, I.; DASH, A. Impact of Proof of Work (PoW)-Based Blockchain Applications on the Environment: A Systematic Review and Research Agenda. Journal of Risk and Financial Management, v. 16, n. 4, p. 218, 31 mar. 2023.

GREVE, Fabíola Greve et al. Blockchain e a Revolução do Consenso sob Demanda. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) - Minicursos, [S.l.], may 2018. Disponível em: http://143.54.25.88/index.php/sbrcminicursos/article/view/1770. Acesso em: 14 jun 2023.

CANT, Joeri. Walmart quer usar tecnologia blockchain para rastrear frutos do mar. Cointelegraph, Https://br.cointelegraph.com/news/walmart-uses-blockchain-tech-to-track-shrimp-supply-chains, 4 out. 2019.

B. K. Mohanta, S. S. Panda and D. Jena, "An Overview of Smart Contract and Use Cases in Blockchain Technology," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 2018, pp. 1-4

Khan, S.N., Loukil, F., Ghedira-Guegan, C. et al. Blockchain smart contracts: Applications, challenges, and future trends. Peer-to-Peer Netw. Appl. 14, 2901–2925 (2021). https://doi.org/10.1007/s12083-021-01127-0