UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SUZANE CARVALHO DE SOUSA

USO DA *BLOCKCHAIN* DO ETHEREUM PARA A REALIZAÇÃO DE LEILÕES DE ENERGIA ELÉTRICA ATRAVÉS DA CRIAÇÃO DE FERRAMENTA WEB

FORTALEZA

2019

SUZANE CARVALHO DE SOUSA

USO DA *BLOCKCHAIN* DO ETHEREUM PARA A REALIZAÇÃO DE LEILÕES DE ENERGIA ELÉTRICA ATRAVÉS DA CRIAÇÃO DE FERRAMENTA WEB

Monografia apresentada ao Curso de Engenharia Elétrica do Departamento de Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial para a obtenção do Título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Demercil Oliveira

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S698u Sousa, Suzane Carvalho de.
Uso da blockchain do Ethereum para a realização de leilões de energia elétrica através da criação de ferramenta WEB / Suzane Carvalho de Sousa. – 2019.
79 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2019.
Orientação: Prof. Dr. Demercil de Souza Olivera Júnior.

1. Blockchain. 2. Contratos Inteligentes. 3. Ethereum. I. Título.

CDD 621.3

SUZANE CARVALHO DE SOUSA

USO DA *BLOCKCHAIN* DO ETHEREUM PARA A REALIZAÇÃO DE LEILÕES DE ENERGIA ELÉTRICA ATRAVÉS DA CRIAÇÃO DE FERRAMENTA WEB

Monografia apresentada ao Curso de Engenharia Elétrica do Departamento de Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial para a obtenção do Título de Bacharel em Engenharia Elétrica.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

_____
Prof. Dr. Demercil de Souza Olivera Júnior (Orientador)
Universidade Federal do Ceará (UFC)

_____
Prof. Assist. Msc. Lucas Melo Silveira
Universidade Federal do Ceará (UFC)

_____
Prof. Msc. Jorge Fredericson De Macedo Costa da Silva
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Ao meu avô, Joaquim (*in memoriam*).

AGRADECIMENTOS

À minha mãe, Socorro, pela paciência, amor e encorajamento para chegar até aqui.

À minha família de Jaguaruana, em especial às minhas tias Beta e Loura, por sempre torcerem e comemorarem minhas vitórias.

À minha família de Caucaia, em especial Isabelle, por me oferecerem os meios necessários para enfrentar a mudança e o curso.

À minha melhor amiga, Samile, por sempre estar presente, sendo luz e suporte no fim de todos os dias.

Aos meu amigos e companheiros de curso, Adrielly, Caroline, Danielle, Murilo, Raísa e Samara, por estarem comigo nas maiores adversidades e nas melhores conquistas.

Ao meu melhor amigo, Rodrigo, pelo companheirismo, ajuda, caronas e risadas em todos os momentos que precisei.

Ao melhor grupo, GEP, em especial ao Luís, por me mostrarem que a diversão deve fazer parte da vida e também pelos infinitos conselhos compartilhados.

Às empresas Eletra Energy e Instituto Experimentes, por terem me dado a oportunidade de experimentar.

À toda a equipe da AMBEV, que me ofereceu oportunidade de crescimento e me ensinou valores, em especial meus colegas de estágio Caio, Cammilla, Lucas e Marília pelos desabafos compartilhados e pela amizade criada.

Ao time da engenharia, em especial ao Mário, por toda paciência e parceria.

A todos os professores da Universidade Federal do Ceará, que contribuíram para minha formação como pessoa e profissional, em especial ao Professor Lucas, pela orientação paciente e cautelosa.

“A ciência é, portanto, uma perversão de si mesma, a menos que tenha, como fim último, melhorar a humanidade.” (Nikola Tesla)

RESUMO

O setor de energia, como um todo, vem passando por um movimento de desregulamentação nos últimos anos com o aumento da geração distribuída. A entrada de novos agentes torna as redes de distribuição mais complexas e deve ser acompanhada de mecanismos que facilitem a sua operação, bem como incentivem a inserção de prosumidores. Atualmente, a estrutura do sistema elétrico, além de complexa, é burocrática e dificulta a entrada de prosumidores, além de tratar o consumidor como agente passivo do sistema. Nesse cenário, as concessionárias de energia são figuras obrigatórias em todos os processos, trazendo pouca autonomia aos consumidores de eletricidade. Este trabalho apresenta uma ferramenta que utiliza a tecnologia *Blockchain*, que vem mostrando seu poder disruptivo em vários setores, sendo seu uso mais conhecido de implementação a criptomoeda Bitcoin, para criar uma aplicação WEB responsável por fazer leilões de energia entre prosumidor e consumidor, sem interferência de intermediários. O projeto consiste em uma aplicação WEB desenvolvida em HTML, CSS e JavaScript, com banco de dados MySQL que por meio da biblioteca Web3.js se integra a um contrato inteligente em linguagem Solidity, através da IDE Remix nativa do Ethereum para realizar leilões de energia diários. Nela, existem três casas que simulam a produção de energia solar fotovoltaica e estabelecem preços mínimos para aquisição de sua energia. Desta forma, o consumidor pode acessar a aplicação via internet e verificar os termos de venda de cada casa para realizar ofertas em Ether (Criptomoeda do Ethereum) pela energia oferecida. É realizada uma simulação em que cinco compradores efetuam lances nas três casas e os resultados do leilão são apresentados e analisados.

Palavras-chave: *Blockchain*. Contratos Inteligentes. Ethereum.

ABSTRACT

The energy sector as a whole has been experiencing a deregulation movement in recent years with increasing distributed generation. The entry of new agents makes more complex distribution networks and must be accompanied by mechanisms that facilitate their operation, as well as encourage the insertion of small producers. Currently, the electrical system structure, besides being complex, is bureaucratic and hinders the entry of prosumers, in addition to a consumer's treatment as a passive agent of the system. In this scenario, the energy utilities are mandatory in all processes, bringing little autonomy to the network. This paper presents a tool that uses Blockchain technology, which shows its disruptive power in various sectors, being known best for implementing an encrypted Bitcoin, creating a WEB application responsible for making prosumer-to-consumer energy auctions without interference from intermediaries. The project consists of a WEB application developed in HTML, CSS and JavaScript, with MySQL database, which through Web3.js library, integrates with an intelligent contract in Solidity language, using Ethereum's native IDE Remix to hold daily power auctions. There are three houses that simulate photovoltaic solar production and set minimum prices for energy acquisition. Thus, the consumer can access the application on the internet and check the sale terms for each house to make offers in Ether (Ethereum's cryptocurrency) for the energy offered. A demo is performed in which five bids placed in the three houses and the auction results are presented and analyzed.

Keywords: *Blockchain*. Smart contracts. Ethereum.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ABI	Application Binary Interface
ANEEL	Agência Nacional de Energia Elétrica
CSS	Cascading Style Sheets
ETH	Ether
EVM	Ethereum Virtual Machine
FTP	File Transfer Protocol
GD	Geração Distribuída
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
PHP	Hypertext Preprocessor
RPC	Remote Procedure Call
SQL	Structured Query Language
URL	Uniform Resource Locator
VM	Virtual Machine

SUMÁRIO

1 INTRODUÇÃO

1.1 Contextualização

No mercado de energia, a tecnologia *Blockchain* tem uma capacidade disruptiva tão grande quanto no mercado financeiro, podendo atingir, inclusive, pessoas que as criptomoedas não atingem, fornecendo um sistema seguro para registro e automação (PETER, 2019).

A médio prazo, o setor de energia se voltará para aplicativos cada vez mais automatizados, como nas medições inteligentes e processos de documentação autônomos, firmando assim uma nova base para o mercado (PETER, 2019).

A maioria dos projetos neste campo está atualmente em estágio inicial, mesmo existindo casos já notórios de como a *Blockchain* pode interferir de forma positiva nos sistemas elétricos. Portanto, ainda é muito difícil precisar sobre como a tecnologia se consolidará e se os governos e autoridades investirão nesses novos modelos (PETER, 2019).

Diante deste cenário, é necessário um sistema integrado que traga a automação que o futuro procura e, sobretudo, apoio e incentivo dos governos para que os contratos estabelecidos dentro da tecnologia *Blockchain* sejam regulamentados. Então, se torna cada vez mais importante o desenvolvimento de aplicações que possam trazer desenvolvimento ao sistema.

A natureza inerente da *Blockchain* agrega confiança ao setor elétrico e elimina a necessidade de agentes que sejam os intermediários de negociações. Com isso, traz autonomia ao mercado e convida participantes que antes eram passivos, como compradores e prosumidores, a se tornarem partes ativas das redes de energia.

1.2 Objetivos

Divididos em objetivos gerais e específicos.

1.2.1 Objetivos gerais

O objetivo principal deste trabalho é desenvolver uma ferramenta para realização de transações de compra e venda de energia utilizando contratos inteligentes e mostrar em detalhes a viabilidade desse tipo de aplicação da tecnologia no processo proposto. Trazendo autonomia para prosumidores e consumidores na rede distribuída através da criação de uma interface WEB integrado à tecnologia *Blockchain*, oferecendo uma solução simples.

1.2.2 Objetivos específicos

Neste trabalho, os objetivos específicos são:

i) Justificar a escolha da *Blockchain* do Ethereum para esse tipo de aplicação;
ii) Desenvolver um contrato em Solidity para administrar o leilão;
iii) Desenvolver interface WEB para interface com o usuário que interaja com o contrato.

1.3 Metodologia

O presente trabalho utiliza a tecnologia *Blockchain* do Ethereum para integração de contratos inteligentes a uma interface WEB criada para exposição de três casas que participarão de leilões de energia produzida por placas solares e que, individualmente, têm dados de produção e tabelas de preço diferentes. Esta aplicação foi desenvolvida utilizando a linguagem de front-End HTML e CSS, ou seja, na aplicação cliente do navegador de internet, bem como PHP no back-End, ou seja, na aplicação do servidor. A integração com a Blockchain do Ethereum integração é feita através da biblioteca Web3.js, em JavaScript, com contratos inteligentes desenvolvidos na linguagem Solidity na plataforma Remix do Ethereum.

1.4 Estrutura do trabalho

O Capítulo 1 deste trabalho busca apresentar as principais motivações, método e estrutura que foram utilizados para seu desenvolvimento.

O Capítulo 2 busca esclarecer a tecnologia utilizada e os motivos da sua escolha frente aos desafios que o mercado de energia irá enfrentar nos próximos anos.

O Capítulo 3 apresenta os conceitos gerais da tecnologia *Blockchain* através da

apresentação dos blocos mais relevantes da plataforma Ethereum.

O Capítulo 4 elucida o desenvolvimento da aplicação, os softwares utilizados e como a estrutura da interface WEB foi dividida em back-end e front-end. Além da criação e integração com os contratos inteligentes do Ethereum.

O Capítulo 5 mostra um caso de teste na simulação de um leilão que acontece para as três casas ao longo de um dia, o funcionamento da realização de ofertas e a apresentação dos resultados.

O Capítulo 6 traz as conclusões do trabalho e propostas de trabalhos futuros.

2 O SETOR ELÉTRICO E A TECNOLOGIA *BLOCKCHAIN*

Identificar os mercados e as oportunidades que a tecnologia *Blockchain* proporciona são primordiais. Dessa forma, provar sua viabilidade como facilitadora do mercado de energia e sua escalabilidade são fundamentais para que o seu uso possa se consolidar no setor elétrico. Neste Capítulo serão discutidos os aspectos da tecnologia, bem como as características que facilitam e dificultam sua implantação.

2.1 Cenário atual da tecnologia *Blockchain* e perspectivas no mercado de energia

2.1.1 Geração distribuída no Brasil e no mundo

Os sistemas de energia estão passando por diversas transformações para acomodar e suportar o crescente aumento percentual de energias renováveis à matriz energética. Os incentivos dos governos e das empresas têm justificado e assegurado esse crescimento acentuado.

No Brasil, através da Agência Nacional de Energia Elétrica (ANEEL) entrou em vigor a Resolução Normativa nº482/2012, em 07 de abril de 2012. Por conseguinte, o consumidor brasileiro, que antes era participante passivo do mercado de eletricidade, pôde assumir o papel de mini e micro gerador, através do emprego de energias renováveis ou cogeração qualificada. O que possibilitou o fornecimento da energia excedente para a rede a qual está conectado (ANEEL, 2015).

Decorrente disso, a geração distribuída (GD) vem aumentando sua participação no setor elétrico, proporcionando maior consciência socioambiental e incentivando a sustentabilidade. Sustentabilidade esta que decorre de benefícios inerentes à GD para o meio ambiente como: redução das emissões de gases, baixo impacto ambiental na instalação e diversificação da matriz energética brasileira (ANEEL, 2015).

Esse crescimento da GD no Brasil tem acompanhado as tendências mundiais. Dados da ANEEL mostram que a quantidade de instalações subiu de 22183 em 2017 para 53345 em 2018, um aumento de 140% (ANEEL, [s.d.]). Apenas em janeiro de 2019 esse número aumentou em 1835 instalações, os números podem ser vistos na Figura 2.1.

Figura 2.1 – Geração distribuída no Brasil

Fonte: ANEEL, [s.d.] com modificações.

Para sustentar e consolidar esse crescimento na geração distribuída, é necessário desenvolver tecnologias que possam controlar a operação e administrar o comércio em redes distribuídas.

As fontes energéticas renováveis mais utilizadas por prosumidores são, em sua grande maioria, a geração fotovoltaica e eólica, que têm como características em comum a dificuldade de previsão de suas potências de saída e dependência das condições climáticas (EID et al., 2016).

Para se adaptar às novas condições de fluxo de potência, ou seja, do prosumidor para a rede de distribuição tradicional, é necessário aprimorar os sistemas de medição para que proporcionem maior segurança e flexibilidade no controle e na comercialização de energia (EID et al., 2016; ZHOU; BROWN, 2017).

Com os investimentos em energia que estão sendo realizados, os novos sistemas trarão desafios, pois serão mais ativos, descentralizados, complexos e com múltiplos agentes. Assim, com a transformação da distribuição num sistema pouco previsível, a troca de informações entre partes deve ser ainda mais clara e eficiente (AHSAN; BAIS, 2017).

2.1.2 O surgimento da tecnologia Blockchain

A tecnologia *Blockchain* se insere nesse cenário como uma forma de realizar transações sem a necessidade de uma entidade centralizada que detenha regras e recursos, além de oferecer maneiras de lidar com desafios que podem aparecer na descentralização do comércio (ANDONI et al., 2019).

A Blockchain é uma estrutura de dados compartilhada que pode armazenar transações digitalmente sem a necessidade de uma autoridade central. Sendo assim, múltiplos usuários podem realizar mudanças simultâneas nos livros razão, isto é, a tecnologia é baseada numa cópia do chamado livro razão que é localizado em todos as máquinas ligadas a rede e armazenam todas as transações realizadas. Essas operações resultam em diferentes versões desses livros, cujas mudanças podem ser checadas por cada usuário conectado em suas próprias máquinas (ANDONI et al., 2019).

Todas essas alterações são validadas pelos usuários através dos livros razão e é pouco provável desfazer qualquer tipo de transação da rede. Dessa forma, as novas transações dentro do código são sempre interligadas a transações anteriores, garantindo transparência e confiança aos contratos (ANDONI et al., 2019).

A *Blockchain* foi desenvolvida na sua concepção para dar suporte ao lançamento de criptomoedas, mais especificamente, o Bitcoin. Entretanto, o futuro das moedas digitais é incerto enquanto a tecnologia *Blockchain* como um todo oferece diversas oportunidades de aplicação.

O uso da tecnologia vem sendo discutido e incentivado por governos como o da Alemanha, por exemplo. Essas aplicações têm alterado de maneira disruptiva a forma como funcionam os mercados financeiros, cadeias de fornecedores e as relações com os consumidores (WALPORT, 2015).

Em Tapscott (TAPSCOTT; TAPSCOTT, 2016), diversas aplicações são apresentadas e a comparação do surgimento da internet como o surgimento da *Blockchain* é feita, evidenciando o poder da ferramenta em seus campos de aplicação. Mas, diferentemente da internet, a *Blockchain* otimiza a forma como as negociações são feitas redefinindo a confiança no mundo digital e eliminando a necessidade de intermediários entre transações.

De acordo com (FRONI; MEULEN, 2017), a fase inicial, onde a tecnologia sofre fortes especulações e que existem picos de uso, já foi superada, sendo esperado que a mesma se torne uma tecnologia padrão em cerca de 2 a 5 anos.

2.1.3 Perspectivas de uso da Blockchain no setor elétrico

Uma pesquisa da Agência Alemã de Energia (BURGER et al., 2016a) mostrou que cerda de 20% dos administradores do setor elétrico acreditam que a tecnologia pode trazer grandes mudanças no fornecimento de energia. Essa pesquisa baseou-se na visão de 70 executivos no ramo de energia.

Com a participação ativa dos consumidores, três pontos principais serão mandatórios para a evolução da indústria de energia, são eles: a descarbonização, a descentralização e a digitalização (PACKAGE, 2015). No entanto, atualmente, a estrutura do setor elétrico é inadequada para suportar a visão que a *Blockchain* traz, já que pequenos participantes da rede, tanto prosumidores quanto consumidores, são excluídos e os incentivos para sua participação ainda são escassos (ANDONI et al., 2019).

Entretanto, com a aderência à tecnologia, os pequenos participantes passam a ser parte integrante do sistema como tomadores de decisão, acrescentando maior competitividade e demandando mecanismos que proporcionem maior controle da demanda e dos preços (ANDONI et al., 2019).

Ainda no contexto que engloba as vantagens de se utilizar essa tecnologia, as aplicações que empreguem a *Blockchain* podem oferecer reduções de custos operacionais, maior eficiência, rapidez, processos automatizados, transparência e possibilidade de redução de capital, segundo o relatório comercial da Deloitte (GREWAL-CARR; STEPHEN, 2016).

Além disso, a redução de custos não é restrita às concessionárias e pode ser relevante para consumidores, que estão pagando mais caro pela compra da energia, e prosumidores, que enfrentam reduções de incentivo, como é citado em (ANDONI; ROBU; FLYNN, 2017). Essa economia é gerada através do comércio P2P em mercados locais centrados no consumidor (PINSON et al., 2017).

Essa tecnologia proporciona melhorias na segurança para os consumidores quando oferece um maior controle das transações envolvidas na rede. Com a digitalização do sistema, proporciona maior cibersegurança para o comércio e a medição inteligente (ANDONI et al., 2019).

Por outro lado, também existem barreiras que a tecnologia deve transpor para mostrar sua escalabilidade mantendo as propriedades desejadas de custo e segurança. Um desses obstáculos é a limitação física que a transferência de energia requer, desta forma isso restringe o espaço em que esse mercado pode estar presente.

Consequentemente, de acordo com um relatório da (EURELETRIC, 2017), essa

necessidade de meios de transmissão físicos para a transferência de energia vem retardando a adoção da tecnologia *Blockchain* no setor elétrico, o que não acontece no mercado financeiro, por exemplo (ANDONI et al., 2019).

2.2 A tecnologia *Blockchain*

2.2.1 Conceitos básicos

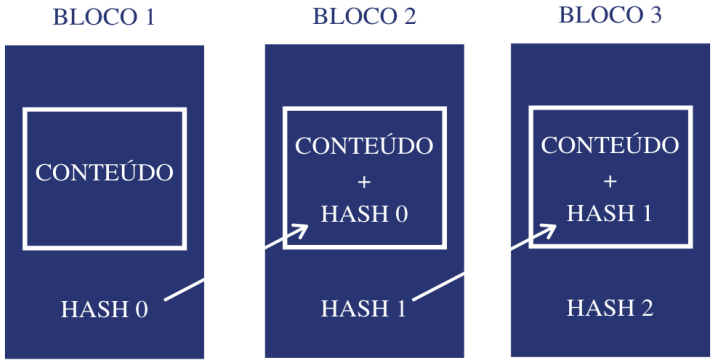
No momento atual, para realizar negociações no meio digital é necessário que algum agente intermediário seja responsável pela confiabilidade da operação. Existem alguns problemas decorrentes disso, como os custos adicionais de acrescentar um intermediário no serviço, a confiabilidade de uma agência centralizada e por existir um ponto único de falha em caso de problemas técnicos e ataques maliciosos (LEE, 2013).

Para isso, a *Blockchain* é um livro de razão pública onde estão todas as transações efetuadas dentro da plataforma, que podem ser arquivos de dados ou executáveis. As informações registradas no livro consistem em quem efetuou o envio de dados, quem recebeu, o que foi enviado, quando foi enviado e em que lugar estes dados estão armazenados (ANDONI et al., 2019).

Este livro razão está presente nas máquinas de todos os usuários da plataforma e a alteração é feita, simultaneamente, em todos eles, garantindo a segurança do sistema e tornando quase impossível cometer fraudes (ANDONI et al., 2019).

O armazenamento dessas informações é feito através de blocos que são interligados aos anteriores, formando uma relação de dependência, marcando cada um com registro de tempo e data. A cada período de tempo é formado um novo bloco de transações, como ilustra a Figura 2.2.

Figura 2.2 – Funcionamento dos blocos na *Blockchain*



Fonte: (PRADO, 2017) com modificações.

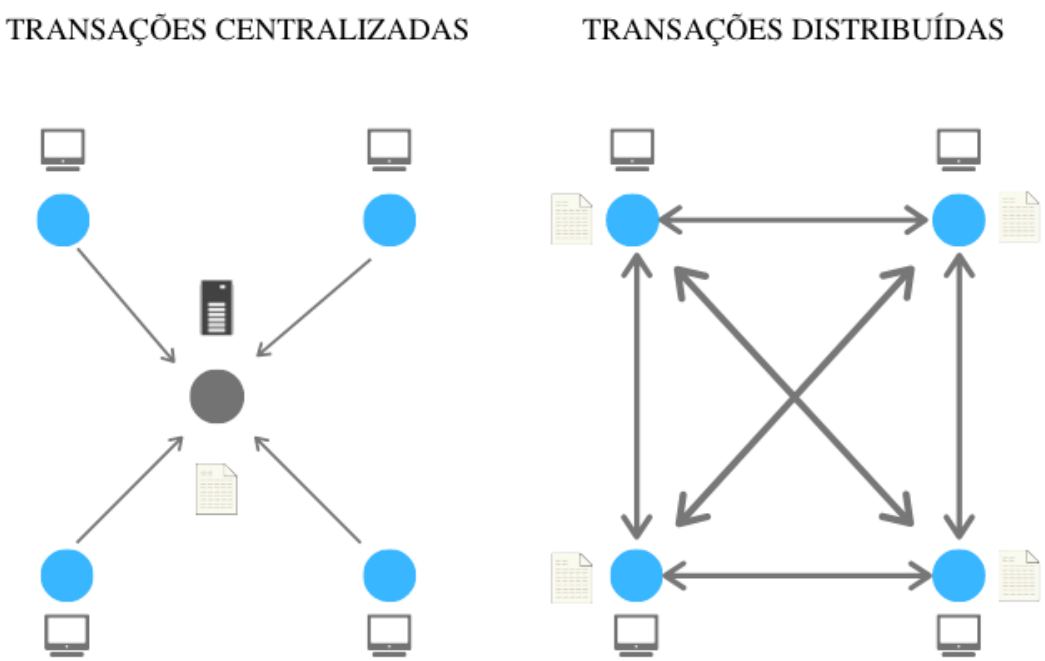
Os *hash*s apresentados na Figura 2.2 são funções matemáticas responsáveis por transformar um determinado dado de entrada, podendo ser uma mensagem ou um arquivo, e gera um código alfanumérico criptografado a partir desse dado.

Essencialmente, essas funções transformam qualquer coisa em um pequeno conjunto de informações, sendo a impressão digital de um arquivo e, neste caso, de um bloco. Logo, com uma pequena alteração das informações de entrada, todo o *hash* é alterado (ANDONI et al., 2019).

Então, quando um *hash* de um novo bloco é gerado, ele também irá possuir a assinatura do *hash* anterior. Dessa forma, uma espécie de selo é criada e se um dos blocos anteriores sofrer alterações, é possível invalidá-lo. Todos os *hash*s são escritos no livro razão e depois de escritos, não podem ser apagados (ANDONI et al., 2019).

Outro dos princípios da tecnologia é substituir as redes centralizadas atuais por uma rede compartilhada de usuários que juntos podem verificar a integridade dos livros e a segurança das transações. Adicionalmente, a diferença entre as transações centralizadas e distribuídas é ilustrada na Figura 2.3 (ANDONI et al., 2019).

Figura 2.3– Tipos de transação



Fonte: (ANDONI et al., 2019) com modificações

Como resultado desse tipo de rede, qualquer usuário pode ter acesso ao histórico de transações do livro e verificar a validade das mesmas. Mas com a remoção da central reguladora, foi necessário encontrar uma maneira adequada para consolidar as informações sincronizadas (ANDONI et al., 2019).

Esse processo de validação varia de acordo com os diferentes tipos de plataforma de *Blockchain*. Assim, como característica comum está a comparação das versões do livro razão que cada usuário tem em sua máquina. A comparação acontece de forma semelhante a uma votação distribuída que busca um consenso entre as partes envolvidas sobre um estado válido (MATTILA, 2016).

Outra forma de garantir a confiabilidade do sistema é utilizando a criptografia de chave-pública, que se trata de um protocolo assimétrico (DIFFIE; HELLMAN, 2016). Cada usuário da rede possui duas chaves criptografadas que consistem em caracteres numéricos e alfanuméricos, uma chave privada e uma chave pública, semelhantes ao *hash*s das transações. Essas chaves são matematicamente relacionadas de forma que informações criptografadas por uma parte, só podem ser verificadas com a outra (ANDONI et al., 2019).

Para que a tecnologia atinja seu maior potencial, de acordo com (WALPORT, 2015), esta precisa ser combinada aos contratos inteligentes, que são criados por usuários da rede e possuem característica imutável. Ademais, são escritos em linguagem própria da *Blockchain* em que estão inseridos, uma vez que são publicados na rede, não podem mais ser apagados e recebem um *hash* para monitoramento, bem como as contas de usuário e transações (SWAN, 2015).

2.2.2 A criação do Bitcoin e a plataforma Blockchain

Em 2009, foi lançado o Bitcoin, a primeira criptomoeda do mundo, que serviria como um sistema de pagamento em moeda virtual entre usuários anônimos e desconhecidos da rede, usando a tecnologia de comunicação P2P. Sua criação é atribuída a Nakatomo, cuja identidade é desconhecida e que publicou um artigo que a antecedeu (NAKAMOTO, 2008).

Essencialmente, essa troca de moedas virtuais entre usuários não era controlada por um banco digital, mas utilizavam criptografia para proteger as transações e os usuários eram responsáveis por sua verificação e controle (MUFTIC, 2016). Apesar da descrença inicial, a moeda cresceu mais de 1700% e fechou cotação valendo em torno de 20000 dólares no final de 2017 (COINMARKETCAP, 2018).

Para armazenar as moedas virtuais, cada usuário possui uma carteira digital que possui duas chaves de segurança, uma pública e uma privada. Essa carteira só pode ser acessada via chave privada e o endereço dessa carteira para os outros usuários da rede é derivado diretamente da chave pública e serve para identificação (ANDONI et al., 2019).

Numa transação de moedas, é necessário conhecer esse endereço para que seja feito

o depósito do valor especificado, mas se a carteira não possuir moedas, a transação não é iniciada. Contudo antes de ser transmitida, a operação então é criptografada com a chave pública da carteira de destino e assinada digitalmente pelo remetente (ANDONI et al., 2019).

Além dos nós comuns que constituem a rede, existem ainda nós especiais e nós validadores. Os primeiros agregam as últimas transações acontecidas na rede em um único bloco com a duração de 10 minutos, além de serem responsáveis por selecionar as informações e agrupá-las para facilitar o processo de validação desse bloco (DWYER, 2015).

Os nós validadores, chamados de mineradores, são responsáveis por resolver um problema matemático a cerca de um novo bloco. Como recompensa, o primeiro nó a solucionar o problema proposto ganha o direito de adicioná-lo ao livro razão da rede, além de recompensas em criptomoedas. A resolução do problema matemático é algo difícil e demanda esforço computacional e energia elétrica, por isso se chama prova de trabalho e é uma das bases do sistema de consenso distribuído da rede (NAKAMOTO, 2008).

Durante o processo de validação das transações, é possível que estas sejam inseridas em dois ou mais blocos, isso resulta numa estrutura ramificada de blocos (WOOD, 2014), que eventualmente, através dos algoritmos de consenso serão reduzidas a apenas uma cadeia válida, geralmente a mais longa ou a que agregou maior esforço computacional (ANDONI et al., 2019).

Desta forma, quando apenas um bloco sofre alteração, toda a cadeia criada deve ser validada novamente pelos outros nós da rede. Se qualquer divergência for encontrada, a transação é rapidamente invalidada pela plataforma que não chegará ao consenso, processo semelhante ao apresentado na Figura 2.2 (ANDONI et al., 2019).

2.3 Conceito de consenso distribuído

O consenso é um problema fundamental em sistemas distribuídos onde se deve ter uma visão idêntica da estrutura. Existem vários algoritmos desenvolvidos para que se chegue a um consenso, todos partindo de pontos distintos, mas com a mesma finalidade. O "*Proof-of-work*", ou prova de trabalho, é o algoritmo de consenso mais amplamente utilizado e será explicado com mais detalhes.

Nas redes *Blockchain*, os algoritmos devem ser escolhidos de acordo com características como escalabilidade, velocidade de transação, finalização da transação, segurança e gasto de recursos como eletricidade, isso garantirá a melhor operação possível para o sistema (ANDONI et al., 2019).

Uma série de transações na rede da *Blockchain* se concentram em um bloco criado

por algum nó da rede, este bloco deve ser gerado e inserido na rede para validação. Então os outros nós da rede utilizarão o algoritmo definido até que cheguem ao consenso, esse bloco será inserido na *Blockchain* e terá seu endereço associado aos novos blocos criados. A depender do algoritmo escolhido, só após um tempo esse bloco se tornará parte permanente da rede, isso evita que ataques maliciosos se concretizem na plataforma (ANDONI et al., 2019).

2.4 Fatores para a escolha da tecnologia *Blockchain* no desenvolvimento de aplicações

A segurança, a resistência à censura e a transparência, características inerentes à tecnologia *Blockchain*, juntamente com os algoritmos de consenso distribuídos, podem ser atraentes e úteis para diversos tipos de sistema (MATTILA, 2016).

Entretanto, as aplicações consolidadas da tecnologia ainda são poucas, pois pela natureza recente da mesma é difícil predizer se uma aplicação que a usa terá sucesso. Todavia, devido aos benefícios já citados, é importante entender quais critérios devem ser levados em conta para a tomada de decisão. Alguns trabalhos, como em (SEPPÄLÄ, 2016), buscam levantar os critérios da sua aplicação e o que deve ser observado para sua escolha.

Posto isto, o primeiro critério a ser considerado é se as partes da negociação podem ser representadas em forma digital, como código ou base de dados, para que a transação possa ser efetiva (SEPPÄLÄ, 2016).

O segundo critério é que esses dados precisam ser compartilhados dentro de toda a rede *Blockchain*, tendo fácil acesso de pessoas desconhecidas, que podem não ser confiáveis. E todas as decisões podem influenciar na aplicação (SEPPÄLÄ, 2016).

O mais importante dos critérios refere-se a entender a necessidade da descentralização para o funcionamento desse mercado específico. Alguns motivos para isso são: reduzir os custos agregados por intermediários, alcançar transações mais rápidas e seguras, clareza dos procedimentos automáticos, resistência a censura, maior resiliência a falhas e a eliminação da necessidade de confiar em intermediários (ANDONI et al., 2019).

2.5 Impactos da tecnologia no desenvolvimento do setor

2.5.1 Potencial de impacto

Existe uma série de aplicações do uso da tecnologia para vários aspectos do mercado de energia, as mesmas, levantadas em (BURGER et al., 2016b) e (CANTO, 2017) são

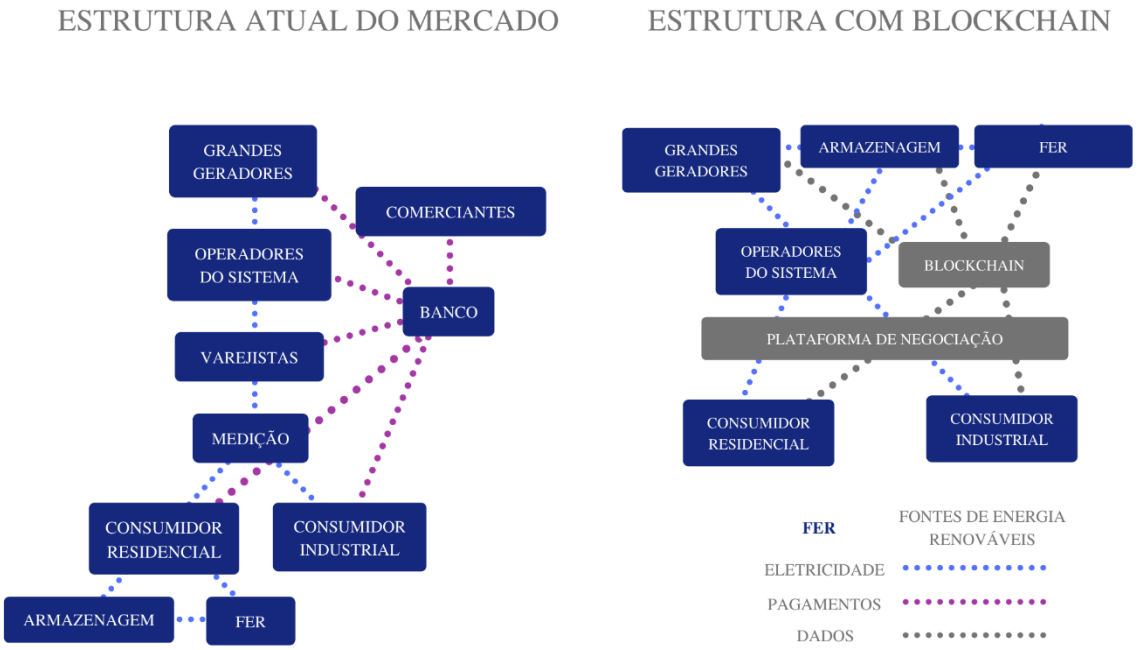
listadas a seguir de acordo com (ANDONI et al., 2019):

- Faturamento: *Blockchain*, contratos inteligentes e medições inteligentes podem realizar cobranças automáticas, que promoveriam soluções de pagamento conforme o uso ou na forma pré-paga.
- Vendas e marketing: a forma como a energia é vendida, por se adaptar a partir do perfil energético do consumidor.
- Negociação e mercados: plataformas de geração distribuída podem se tornar disruptivas para as operações de mercado atuais, como a administração do mercado, as transações e o gerenciamento de riscos.
- Automação: melhoria do controle de sistemas de energia descentralizados ou micro redes.
- Aplicativos de rede inteligente e transferência de dados: uso na comunicação entre dispositivos inteligentes, transmissão de dados e armazenamento.
- Gerenciamento de rede: ajudando no gerenciamento de redes descentralizadas.
- Gerenciamento de segurança: segurança oferecida pela forma anônima que as transações são feitas, assim como a cibersegurança.
- Compartilhamento de recursos: a tecnologia pode proporcionar soluções de cobrança de recursos entre vários usuários da rede, assim como armazenamento comunitário.
- Concorrência: contratos inteligentes podem simplificar e acelerar a entrada de agentes no mercado e a participação ativa de consumidores que afetariam a demanda.
- Transparência: registros imutáveis e processos transparentes podem melhorar, significativamente, a auditoria e conformidade regulamentar.

2.5.2 Comércio e fornecimento de energia por atacado

Os mercados atacadistas de energia possuem procedimentos complexos que exigem diversos intermediários em suas estruturas. Na Figura 2.4 resumem-se as entidades presentes atualmente e, ao lado, uma ideia de como o mercado se resumiria funcionando integrado a *Blockchain* (ANDONI et al., 2019).

Figura 2.4 – Estruturas do mercado de energia



Fonte: (ANDONI et al., 2019) com modificações

As transações do mercado atacadista envolvem pós-processamento manual e maior comunicação para consolidar informações mantidas separadamente por cada parte da estrutura. O resultado disso é um sistema lento e burocrático que torna difícil a inserção de produtores de menor escala (ANDONI et al., 2019).

Com a utilização de contratos inteligentes, os prosumidores negociariam diretamente com os consumidores ou varejistas através de agentes comerciais autônomos, que eliminariam os intermediários (ANDONI et al., 2019).

No entanto, substituir todas as partes da estrutura atual por um sistema completamente descentralizado é uma tarefa demorada e complexa. Visto isso, todo o desenvolvimento do sistema deve ser bem fundamentado, apesar de ainda se correr o risco de não conseguir suprir a velocidade das demandas atuais. Por conta disso, o foco das aplicações criadas até o momento é de substituir apenas parte dessa estrutura (ANDONI et al., 2019).

2.5.3 Internet das coisas e a Blockchain

A internet das coisas pode trazer ainda mais benefícios a tecnologia *Blockchain*. Além dos contratos inteligentes que surgem entre prosumidores e consumidores através da tecnologia P2P, a troca de informações entre máquinas com a tecnologia M2M também pode oferecer um grande avanço tecnológico no consumo de energia (RISTESKA STOJKOSKA; TRIVODALIEV, 2017).

Um número crescente de dispositivos com essa tecnologia deverá ser conectado à internet até 2020 (BURGER et al., 2016b). No setor elétrico, medidores inteligentes estão sendo cada vez mais adotados nos sistemas de potência (JARADAT et al., 2015).

Essa tendência pode trazer, além da segurança e da descentralização trazida da *Blockchain*, a automação dos sistemas, gerando sobretudo economia de energia. Através dos medidores inteligentes, é possível configurar contratos e aplicações que façam as transações de forma otimizada. Isso é bom tanto para o comercializador de energia, que pode acompanhar o mercado e se adequar ao mesmo, quanto para o consumidor que pode comprar a energia com melhores preços e condições, tudo isso de forma automática através de um algoritmo (ANDONI et al., 2019).

Já com a comunicação entre máquinas e aparelhos, os consumos podem ser monitorados e, através dos dados coletados e analisados, decisões relacionadas a demanda da residência podem ser conhecidas e tomadas, permitindo também a interação até mesmo entre unidades de uma mesma comunidade (ANDONI et al., 2019). Uma problemática a respeito do conceito se refere a privacidade dos consumidores que optem por esse tipo de monitoramento, havendo a necessidade de definir os limites e termos para cada uso.

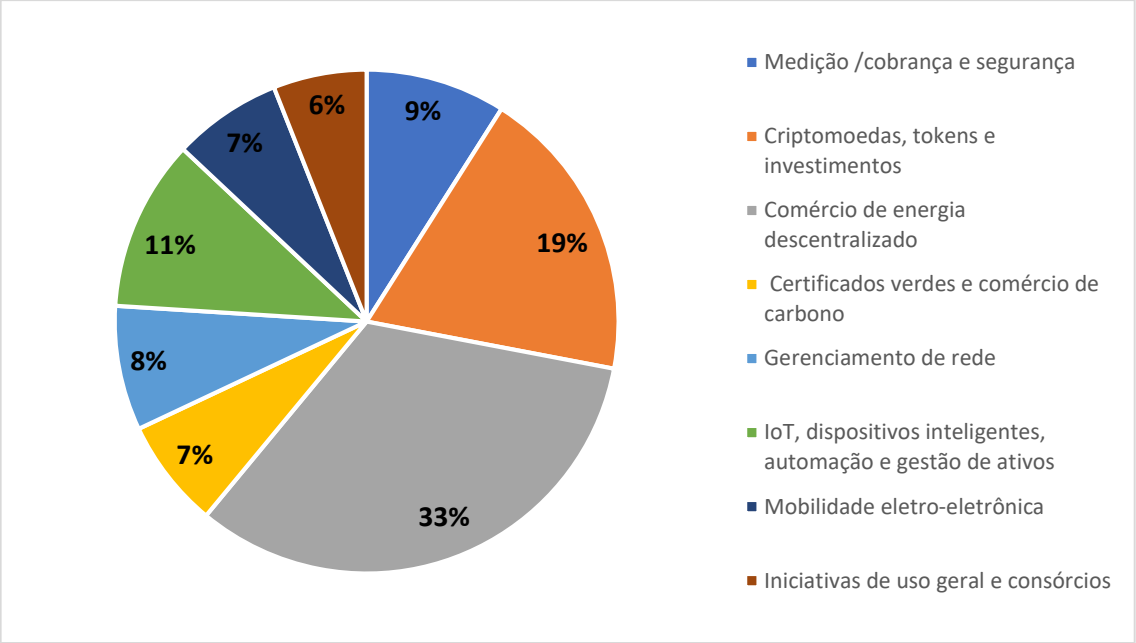
2.6 Dados da *Blockchain* na indústria de energia

Em (ANDONI et al., 2019) uma pesquisa envolvendo mais de 140 artigos relacionados a aplicação da tecnologia *Blockchain* no setor de energia foi realizada, verificando quais deles têm maior interesse de investidores e de administradores do mercado.

Nele, os casos de uso foram separados em oito grandes grupos de acordo com seus propósitos, são eles: 1) medição/cobrança e segurança; 2) criptomoedas, tokens e investimentos; 3) comércio de energia descentralizado; 4) certificados verdes e comércio de carbono; 5) gerenciamento de rede; 6) IoT, dispositivos inteligentes, automação e gestão de ativos; 7) mobilidade eletro-eletrônica; 8) e iniciativas de uso geral e consórcios.

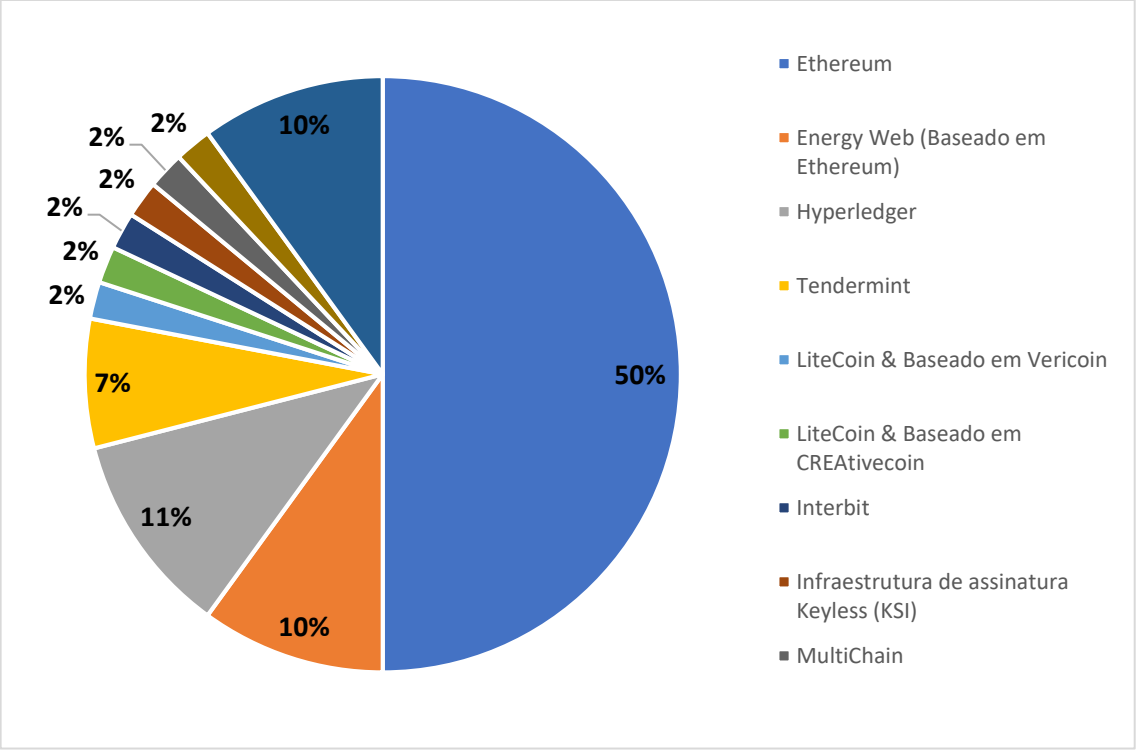
Um a cada três casos de uso relaciona-se ao comércio de energia descentralizado. O segundo mais popular é o de criptomoedas, tokens e investimentos, representando um em cada cinco casos. O resumo dos casos de uso pode ser visto na Figura 2.5. Também foram classificadas atividades de *Blockchain* de acordo com a plataforma utilizada, onde 60% são soluções baseadas no Ethereum e os detalhes podem ser vistos na Figura 2.6.

Figura 2.5 – Casos de uso da *Blockchain* no mercado de energia



Fonte: (ANDONI et al., 2019) com modificações

Figura 2.6 – Plataformas da *Blockchain* mais utilizadas



Fonte: (ANDONI et al., 2019) com modificações

3 ETHEREUM E SUAS FERRAMENTAS

O Ethereum é uma plataforma baseada na tecnologia da *Blockchain* que permite a programação de aplicativos descentralizados, contratos inteligentes e transações de criptomoedas Ether e outros tokens.

A plataforma foi idealizada no final de 2013 por Vitalik Buterin e foi lançada no dia 30 de julho de 2015 com 11.9 milhões de Ethers pré-minerados e, desde então, se tornou a segunda moeda com maior capitalização do mundo, ficando atrás do Bitcoin (FOXBIT, [s.d.]).

A *Blockchain* surgiu juntamente com o Bitcoin, cuja a proposta era de realizar transações monetárias e de valorizar a nova moeda digital. No Ethereum o objetivo vai além de agregar valor ao ether. A ideia é que se utilize a tecnologia para desenvolver aplicações, dos mais variados empregos, que possam se beneficiar da segurança, da confiabilidade e da transparência que a plataforma oferece, objetivando, por fim, o seu desenvolvimento.

O ether, desta forma, é utilizado para realizar operações dentro da Ethereum Virtual Machine (EVM), empregando uma linguagem Turing completa, o que possibilita a compra de poder computacional dos usuários ao redor do mundo para que as movimentações possam ser realizadas (ANDONI et al., 2019).

Nesse capítulo, serão abordados os tópicos mais relevantes em relação ao desenvolvimento de um aplicativo descentralizado, as publicações de contratos inteligentes e as ferramentas necessárias para isso.

3.1 O Ethereum

De acordo com (WOOD, 2014), o Bitcoin demonstrou o poder dos mecanismos de consenso e o respeito aos contratos que torna possível usar a internet para criar um sistema de transferência de valor descentralizado, compartilhado ao redor do mundo e virtualmente livre para uso.

Desta forma, o Ethereum é um projeto que busca construir a mesma generalização da tecnologia vista no Bitcoin, que permitirá que todas as transações baseadas em conceitos de estado de máquina sejam realizadas. Para a construção desse conceito é necessário especificar um sistema de mudança de estado através de uma linguagem simples, mas com todas as funções necessárias. E, ainda, arquitetar um sistema em que todos possam confiar em sua autonomia (WOOD, 2014).

O artigo de (WOOD, 2014), foi publicado depois do artigo de (BUTERIN, 2013),

que apresentou a proposta inicial do Ethereum, e acrescentou conceitos importantes, como cofundador, criador da linguagem Solidity e coordenador de tecnologia da fundação. Além disso, apresentou os conceitos de mudanças de estado e algoritmos relacionados à prova de trabalho e consenso, que são a base de todo o sistema da *Blockchain* do Ethereum.

3.2 Prova de trabalho

Em 1992, no trabalho de (DWORK; NAOR, 1992), o conceito de prova de trabalho é introduzido como uma maneira de transferir valor na internet. Naquele tempo, esse valor foi utilizado como uma maneira de prova de integridade do serviço e demonstrou o potencial futuro do mecanismo para transferências de valor econômico com confiança, mesmo desconhecendo o remetente da operação.

A prova de trabalho é um valor escalar seguro, quanto a sua criptografia, que demonstra a quantidade de trabalho computacional necessária para a mineração do bloco ou token (WOOD, 2014).

Esse algoritmo é utilizado para trazer segurança a *Blockchain*, dando significado e credibilidade ao trabalho computacional. No entanto, como a mineração de novos blocos acarreta em recompensa monetária, a prova de trabalho não serve apenas como garantia de segurança, mas também como rico mecanismo de distribuição (WOOD, 2014).

Uma forma de limitar a mineração, é usar o acesso sequencial de memória, isto é, para determinar o valor é necessário grande largura de banda, de modo que a memória não possa ser usada em paralelo para descobrir vários valores simultaneamente (WOOD, 2014).

Em resumo, a prova de trabalho funciona da seguinte forma: um algoritmo é executado para acrescentar um novo bloco na *Blockchain* resolvendo o problema criptográfico de gerar um *hash* de saída que comece com um valor consecutivo de zeros nas posições significativas. Isto é, a implementação exige que o minerador procure por um valor de nonce, explicado mais a frente, que, adicionado aos dados do bloco e ao mempool (que contém as novas transações), gere um *hash* com uma determinada quantidade de zeros no seu início (MATOS, 2018).

3.3 Componentes da tecnologia *Blockchain*

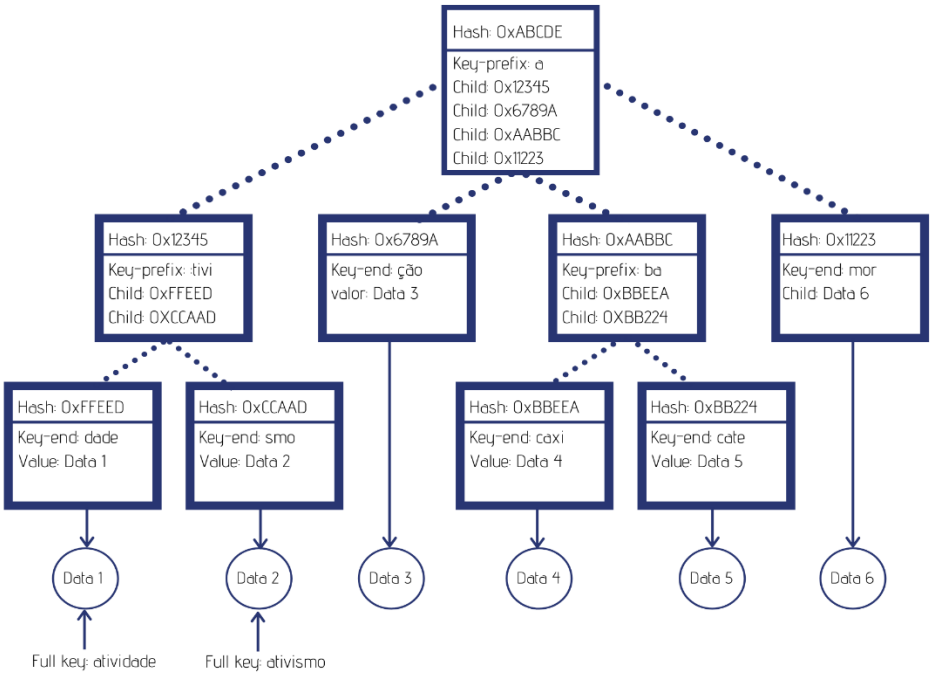
São apresentados a seguir conceitos de blocos, transações e estados e seus funcionamentos dentro no Ethereum, os parâmetros e definições citados são citações do artigo

original de publicação da plataforma e pode ser visualizado em (WOOD, 2014).

3.3.1 Estados

O estado é um mapeamento entre endereços, que possuem identificadores de 160 bits (WOOD, 2014). Embora os estados não sejam armazenados diretamente na *Blockchain*, a sua implementação manterá o mapeamento em uma árvore modificada de Merkle Patricia, isto é, o valor final do bloco será composto pelos valores agrupados desses estados. Esta árvore é apresentada na Figura 3.1.

Figura 3.1 – Árvore de Merkle Patricia



Fonte: (PEYROTT, 2017) com modificações.

A árvore requer uma base de dados simples no back-end que mantenha o mapeamento dos dados. Usualmente, esse banco de dados recebe o nome de banco de dados de estado. Portanto, primeiramente, esse sistema traz benefícios, principalmente porque o nó raiz da estrutura tem sua identidade atrelada a todos os dados internos, logo, uma única identidade de nó guarda as identidades dos demais componentes internos. Em segundo lugar, sendo a estrutura de dados imutável, permite que qualquer estado anterior, cujo *hash* raiz seja conhecido, seja reestabelecido simplesmente alterando o *hash* raiz de acordo com a alteração (WOOD, 2014).

De acordo com (WOOD, 2014), o estado compreende os quatro campos a seguir:

- nonce: um valor escalar igual ao número de transações enviadas de um endereço, ou, no caso de contas com código associado, o número de criações de contrato feita pela conta;
- balance: um valor escalar igual ao número de Wei possuído pelo endereço;
- storageRoot: um *hash* de 256 bits do nó raiz de uma árvore Merkle Patricia que codifica o armazenamento de conteúdo da conta;
- code*Hash*: o *hash* do código da EVM dessa conta, esse é o código que quando executado este endereço deve receber uma chamada de mensagem, é imutável e diferente dos outros não pode ser alterado após a construção.

3.3.2 Transações

Uma transação é uma operação criptografada que ocorre de forma externa a *Blockchain*, ou seja, é acionada por um agente externo que pode ser humano ou software. Sendo responsável pela transmissão de dados de uma conta para outra em forma de mensagem, e também pela criação de contratos (WOOD, 2014).

Esses dois tipos de transação possuem um número comum de componentes, de acordo com (WOOD, 2014) são eles:

- Nonce: um valor escalar igual ao número de transações enviadas pelo remetente;
- gasPrice: valor escalar igual ao número de Wei a ser pago por unidade de gas para os custos computacionais decorrentes da execução da transação;
- gasLimit: um valor escalar igual a máxima quantidade de gás que pode ser usada para executar a transação;
- To: o endereço de 160 bits do destinatário da mensagem no caso de transferência;
- Value: um valor escalar igual ao número de Wei transferido para o destinatário ou, no caso de criação de contrato, como dotação para a conta recém-criada;
- V,r,s: três valores correspondentes a assinatura da transação e usados para determinar o remetente.

Além disso, uma transação de criação de contrato possui:

- Init: uma matriz de bytes de tamanho ilimitado que especifica o código EVM para o processo de inicialização da conta formalmente.

3.3.3 Blocos

O bloco no Ethereum é composto pelo agrupamento das partes mais importantes da informação. Junto das informações das transações, possui os endereços de remente e destinatário, os dados de tempo e demais informações relevantes. Estas são listadas a seguir por (WOOD, 2014):

- parent*Hash*: *hash* de 256 bits do bloco de origem;
- ommers*Hash*: *hash* de 256 bits da lista dos blocos que possuem mesma origem;
- beneficiary: o endereço de 160 bits para qual serão transferidos os honorários da mineração bem-sucedida;
- stateRoot: o *hash* de 256 bits do nó raiz da árvore de estados, depois que todas as transações são executadas e aplicadas as finalizações;
- transactionsRoot: o *hash* de 256 bits do nó raiz da estrutura da árvore preenchida com cada transação na porção da lista de transações do bloco;
- receiptsRoot: *hash* de 256 bits do nó raiz da estrutura da árvore preenchida com os recibos de cada transação na porção da lista de transações do bloco;
- logsBloom: o filtro Bloom composto de informações indexáveis contidas em cada entrada do destinatário nas transações da lista de transações;
- difficulty: valor escalar correspondente ao nível de dificuldade do bloco, pode ser calculado a partir do nível de dificuldade do bloco anterior e pelo timestamp;
- number: um valor escalar igual ao número de blocos passados, por exemplo, o bloco de gênesis tem o número igual a zero;
- gasLimit: um valor escalar igual ao atual limite de gás consumido pelo bloco;
- gasUsed: um valor escalar igual ao total de gás usando em transações neste bloco;
- timestamp: valor escalar igual a saída do tempo do Unix() no início do bloco;
- extraData: uma matriz de bytes arbitrária que contém dados relevantes para o bloco, deve conter 32 byes ou menos;
- mix*Hash*: um *hash* de 256 bits que, combinado com o nonce, prova que esforço computacional suficiente foi realizado neste bloco;
- nonce: um *hash* de 64 bits que, combinado com o mix*Hash*, prova que esforço computacional suficiente foi realizado neste bloco.

3.4 Contratos inteligentes

Em 1994, muitos anos antes do nascimento da *Blockchain*, o termo "Smart Contract" foi introduzido por (SZABO, 1996), criptógrafo e cientista da computação, que idealizou uma representação digital de contratos convencionais com objetivo de reduzir a necessidade de confiança em intermediários, ele cita:

> Novas instituições e novas formas de formalizar as relações que compõem essas instituições agora são possíveis graças à revolução digital. Eu chamo esses novos contratos de "inteligentes", porque eles são muito mais funcionais do que seus ancestrais inanimados baseados em papel. Nenhum uso de inteligência artificial está implícito. Um contrato inteligente é um conjunto de promessas, especificadas em formato digital, incluindo protocolos nos quais as partes cumprem essas promessas.

Os smart contracts, em português "contratos inteligentes", devem garantir o cumprimento dos termos de ambas as partes através de linhas de código imutáveis que podem ser executadas automaticamente, agregando imparcialidade e segurança aos contratos convencionais. Desta forma, ele obtém as informações referentes aos termos contratuais e garante sua execução.

Até o momento, pode-se identificar algumas vantagens e desvantagens para os contratos. Como vantagens, além da autonomia para criação e administração de contratos pelos próprios usuários, da confiabilidade, da segurança e do backup, podem-se destacar: a velocidade como os contratos ocorrem, através de validações computacionais autoaplicáveis, a economia de se fazer transações sem intermediários e a precisão, que retira possíveis falhas de preenchimento por interferência humana.

Em contrapartida, seu estado legal incerto pode ser um problema. Atualmente, os contratos inteligentes não são regulados por nenhum governo e há risco caso as instituições governamentais estabeleçam quadros legislativos desfavoráveis (TAR, 2017).

Adicionado a isso, o fator humano oferece riscos, pois o contrato é criado por um usuário e, por isso, é passível de brechas. Devido a essa característica, podem ser alvo de ataques que são capazes de minar a confiabilidade e todas as vantagens que o modelo oferece. Um exemplo foi o TheDAO, um projeto de organização autônoma descentralizada lançado na plataforma Ethereum, que foi atacado por hackers que se utilizaram das fragilidades do contrato, resultando em um prejuízo de 50 milhões de dólares (DHILLON et al., 2017).

3.5 Moeda e tecnologia do Ethereum

O Bitcoin é uma aplicação construída com base na tecnologia *Blockchain*, que pode

servir de suporte para muitas outras. O Ethereum oferece os recursos para desenvolvimento de aplicações na criação de contratos inteligentes e transações. Para isso, moedas também precisam ser utilizadas e esforços computacionais são exigidos para garantir todos os recursos de segurança e confiabilidade da rede.

O Ether (ETH) é a moeda do Ethereum, serve como pagamento para as transações que ocorrem no ambiente virtual e é paga como gratificação ao esforço dos mineradores da rede. Essa moeda é volátil e sofre alterações de valor constantemente. Pensando nisso, a unidade gás foi criada para medir o esforço que deve ser empregado para a realização de uma transação ou a publicação de um contrato. Portanto, diferentes tipos de transações requerem quantidades diferentes de gás para serem concluídas. Fazendo uma analogia, é comum a sua comparação com a unidade de kW, onde a quantidade de energia consumida não é medida em reais, mas sim em kilowatts hora (HENRIQUE, 2019).

Existe um limite de gás definido pelo usuário, que é a quantidade que o mesmo se dispõe a pagar numa transação. Transferências padrões de Ether demandam um limite de gás de 21000 unidades. Essa quantidade irá depender da complexidade do código a ser executado que, por sua vez, se utiliza das interfaces para calcular automaticamente os limites para um valor padrão (HENRIQUE, 2019).

O preço do gás é quantidade de ETH que o usuário se dispõe a pagar por cada unidade de gás, geralmente medido em “Gwei”. Os valores são apresentados na Tabela 3.1.

Tabela 3.1 – Sub denominações do Ether

Multiplicador	Nome
10^0	Gwei
10^{12}	Szabo
10^{15}	Finney
10^{18}	Ether

Fonte: (WOOD, 2014) com modificações.

3.6 Redes no Ethereum

Para que os contratos sejam lançados e as transações sejam realizadas, é necessário que os mesmos estejam na rede e em operação dentro do sistema de uma plataforma *Blockchain*, como o Ethereum.

O ambiente onde isso acontece são as “Nets”, a “MainNet” e as “TestNets”, como o nome diz, a primeira se refere a *Blockchain* original, onde as transações ocorrem no livro e o

valor econômico é real.

As TestNets, que podem ser privadas ou públicas, utilizam Ether fictícios para efetuar operações. Esses Ethers só funcionam dentro de cada rede, mas são minerados de forma real dentro delas. Nesses ambientes os desenvolvedores podem experimentar funcionalidades antes de publicar o contrato diretamente na rede principal.

As quatro maiores TestNets públicas são Ropsten, Rinkeby, Morden e Kovan, que têm os nomes de estações de metrô ao redor do mundo (AZIZ, [s.d.]).

Um novo nó na rede, pode migrar entre as duas redes sem dificuldades, ora desenvolvendo na rede principal, ora nas redes de teste, dependendo da necessidade do momento (AZIZ, [s.d.]).

3.7 Usuários e carteiras

Realizar publicação de contratos e interagir com eles, requer que o usuário esteja dentro da rede do *Blockchain*. Para comprar ou minerar Ether, o usuário deve possuir uma carteira eletrônica.

Diferentemente de uma carteira utilizada no mundo real, as criptomoedas que o usuário possui não estão dentro da carteira eletrônica, mas sim flutuando na rede da *Blockchain*. A carteira é tão somente um endereço, semelhante ao *hash* associados aos contratos inteligentes, que registra todas as operações que são realizadas dentro da rede e sinalizam a todos quem são seus donos (HENRIQUE, 2019).

Ao criar uma carteira, é criado um par de chaves criptográficas únicas. A parte pública dessas chaves é o endereço de sua carteira, que permite que terceiros enviem Ether para ela. Já a parte privada permite que o usuário transfira as criptomoedas associadas a carteira, esta é secreta e só o dono tem acesso (HENRIQUE, 2019).

3.8 A linguagem dos contratos inteligentes

Na *Blockchain* do Ethereum, é possível criar contratos inteligentes com a linguagem de programação solidity.

A Solidity é uma linguagem de programação de alto nível, orientada a contratos, com a sintaxe semelhante a JavaScript e desenhada para ser executada na Máquina Virtual Ethereum (EVM). É estatisticamente tipada, suporta herança, bibliotecas e tipos complexos definidos pelo usuário, dentre outras características (ETHEREUM, 2018).

A linguagem pode ser executada em alguns tipos de IDE, do inglês *Integrated Development Environment*. A mais difundida e de uso mais simples é o Remix, que é baseada em *browser* e possui compilador integrado. Outros tipos de linguagem já estão sendo desenvolvidas e a Vyper, está disponível também no Remix, sendo semelhante a Python.

4 DESENVOLVIMENTO DA APLICAÇÃO WEB

Tendo em vista as necessidades que surgirão no mercado de energia nos próximos anos, particularmente, devido ao crescimento da geração distribuída e às diversificações do setor, conforme apresentado nos capítulos anteriores, é proposta uma aplicação WEB na forma de página da internet que emprega a tecnologia *Blockchain* por meio da plataforma Ethereum. Esta aplicação tem como objetivo proporcionar um novo modo de efetuar transações de compra e venda de energia elétrica, de forma prática e fácil, eliminando, em parte ou totalmente, a necessidade da concessionária ou de comercializadores de energia elétrica na intermediação de transações. Favorecendo, desta forma, o crescimento da geração autônoma de energia elétrica, tornando o consumidor mais ativo e o convidando a ter mais domínio e consciência sobre seu consumo e sua receita.

4.1 Visão geral do sistema

O modelo proposto do sistema WEB é a realização de leilões diários de energia elétrica, onde são leiloadas as produções diárias de energia de três residências com sistemas de geração fotovoltaica instaladas. Por meio do sistema proposto, o usuário que possui conta no Ethereum efetua um lance em qualquer uma das plantas de geração, sendo este lance determinado pelo preço que ele deseja pagar por kWh e pela quantidade de kWh desejada. No fim do dia, quando o leilão é encerrado, o usuário que deu o maior lance vence e o valor pago, em Ether, é transferido para a residência escolhida.

A estrutura do sistema elétrico é composta por várias partes e propor um modelo utilizando a tecnologia *Blockchain* que substitua todas elas é inviável para a fase atual do sistema. Portanto, o objetivo do sistema WEB é promover melhoria e facilidade na relação de troca entre prosumidores e consumidores residenciais, utilizando a tecnologia segura, automatizada e imutável da *Blockchain*.

4.1.1 Aplicação WEB

A proposta da plataforma é apresentar as informações, de forma clara e coesa, referentes a cada unidade geradora. Dessa forma, a estrutura gráfica da interface WEB foi desenvolvida de forma a exibir os principais dados a serem considerados pelo cliente durante a escolha, logo, é posto lado a lado as informações referentes a cada sistema de geração,

facilitando, portanto, a comparação e a eleição.

Desse modo, as três plantas de geração são apresentadas lado a lado, em forma de coluna, e as informações apresentadas em tela são: a produção acumulada de energia até a hora da consulta, a previsão de estoque no final do dia e o preço mínimo por kWh aceito pela residência. Onde, a produção acumulada refere-se aos dados de produções reais de energia elétrica amostrados ao longo de um dia e armazenados no banco de dados de cada unidade geradora. A previsão do estoque é uma média das produções por intervalo coletado até o respectivo momento de consulta sendo multiplicada pelo dia inteiro de produção, o preço mínimo é um valor mandatório definido pelo proprietário de cada sistema de geração.

De posse dessas informações, cada comprador tem autonomia de escolha quanto a definição de qual residência deseja efetuar o lance, tendo em vista quanto irá pagar por kWh e qual unidade geradora atende melhor a quantidade de energia desejada por ele. Para tal, foi implementado um botão para concretização da oferta que, após ser selecionado, disponibiliza campos para preenchimento de informações pelo usuário, sendo eles: “Valor por kWh” e “Quantidade de kWh”. Ao comprador é retornando o valor em Reais (R$) e em Ether, sendo este último calculado a partir de uma conversão automática a partir do valor em Real (R$). Após a definição de todas as etapas anteriores, então, a proposta poderá ser submetida.

A proposta submetida pelo usuário passa por algumas verificações de aceite e, então, a comunicação com o contrato inteligente do Ethereum é estabelecida. Em seguida, a carteira de confirmação de transação é aberta para aceitação e, assim, a transação é realizada. Por fim, de acordo com os parâmetros do leilão, o comprador pode ser bem-sucedido ou não em sua oferta.

4.1.2 Contrato do Leilão

Um leilão é uma modalidade de negociação, nela o preço de um bem não é fixado e depende da demanda de compradores. Assim, quando maior a procura, mais o bem ganha valor. Usualmente, um valor mínimo é estabelecido para evitar grandes prejuízos na venda sob risco de demanda (7ª REGIÃO, 2010).

Nesta aplicação, o leilão se enquadra bem tanto para o comercializador quanto para o comprador. Portanto, a proposta é que o comercializador estabeleça os requisitos mínimos do leilão, como o preço mínimo, e a partir disso o comprador seja capaz verificar qual o comercializador mais adequa às suas necessidades.

A lógica pode ser dividida em início, meio e fim. De início, é estabelecido quem

será o beneficiado e o tempo total em que a oferta estará ativa, após isso as propostas podem ser submetidas e uma variável com o endereço da maior oferta juntamente com seu valor é armazenada no contrato, se ela for superada, é restituída, bem como ofertas abaixo dela serão revertidas no ato do seu envio.

Por fim, quando o tempo de leilão é atingido, o valor da maior oferta é transferido para o endereço do beneficiado e as propostas que foram superadas são restituídas para seus endereços automaticamente.

4.2 Desenvolvimento da aplicação

Explicadas as lógicas de funcionamento da plataforma WEB e do contrato, a seguir, serão detalhadas as ferramentas utilizadas para o desenvolvimento da aplicação, assim como os seus detalhes de funcionamento.

4.2.1 Aplicação WEB

A interface WEB, intitulado Appa Energy, é o ambiente virtual que possibilita a interface com o usuário e, além disso, é responsável pela integração com o contrato do Ethereum. Para sua criação, foram necessárias ferramentas de hospedagem e de desenvolvimento back-end e front-end, assim como o uso de um pequeno banco de dados para armazenamento dos dados de produção de energia elétrica.

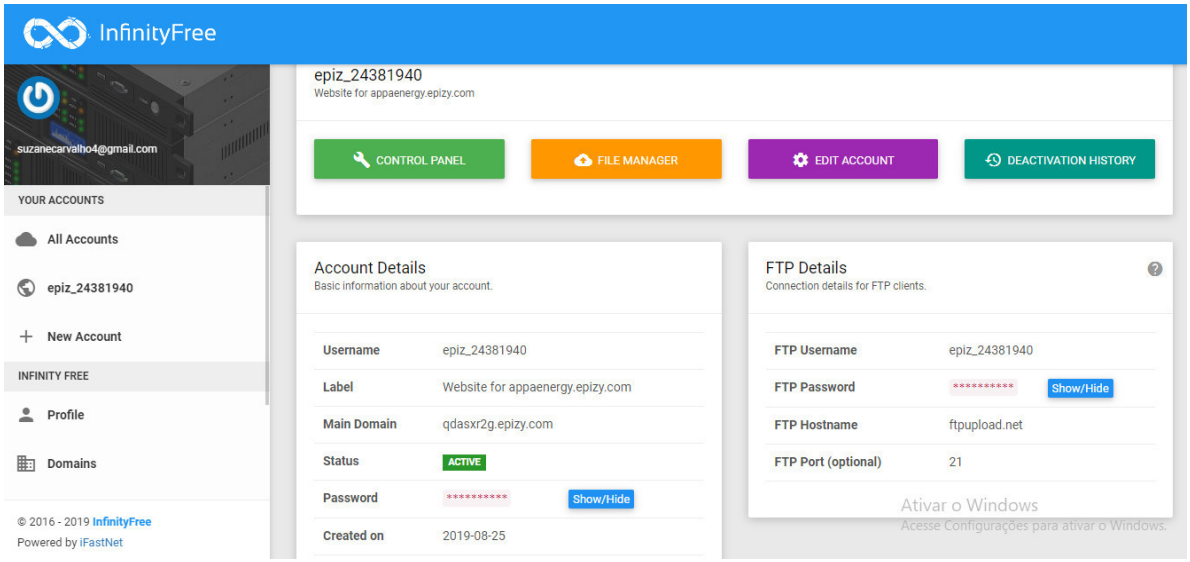
4.2.1.1 Hospedagem e Infinity Free

O processo de hospedagem de uma interface WEB significa dizer que o mesmo estará disponível na internet. Portanto, o ambiente virtual deve estar implementado em um servidor que garanta que ele permaneça no ar 24 horas por dia e protegido de ataques. Existem diversos servidores capazes de hospedar interface WEBs e também diferentes formas de hospedagem.

Para a hospedagem do Appa Energy, o servidor Infinity Free foi escolhido, esse servidor é gratuito por tempo indeterminado, o que garantiu que a interface WEB continuasse no ar durante todo o desenvolvimento, além de oferecer serviços satisfatórios para a aplicação, como banco de dados. Para aplicações mais robustas e que necessitassem de mais ferramentas, existe a opção de pagar para obtê-las.

Gratuitamente é possível criar até três interfaces WEB, e a forma como a conta é exibida é mostrada na Figura 4.1. Como apresentado, as informações iniciais são os detalhes da conta e os detalhes FTP, do inglês *File Transfer Protocol*, este último permite a transferências dos arquivos do computador do desenvolvedor para o servidor em que a interface WEB ficará hospedado.

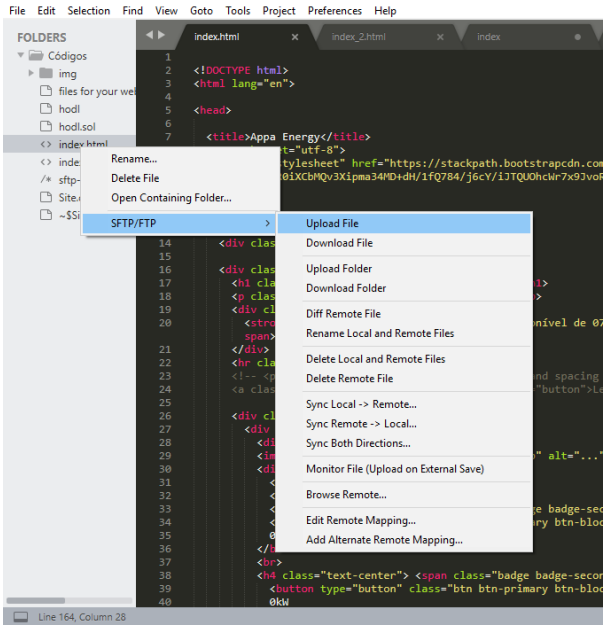
Figura 4.1 – Visão da interface WEB de hospedagem InfinityFree



Fonte: Própria autora.

Durante o desenvolvimento da plataforma, o Sublime Text foi usado para a edição de códigos, este possui IDE leve e simples de usar. Além disso, existem diversos *softwares* para integração de um cliente FTP com o Sublime Text, e para atingir este objetivo, foi utilizado um *plugin* chamado de FTPSync, o que tornou fácil e acessível a transferência de arquivos para o servidor. Na Figura 4.2, é possível observar a opção “SFTP/FTP” disponibilizada pelo FTPSync, que agrega, dentre outras, as funcionalidades de *upload* e *download* do protocolo.

Figura 4.2 – FTPSync na IDE Sublime Text



Fonte: Própria autora.

4.2.1.2 Front-end

O front-end é responsável por toda a interface visual da interface WEB, ou seja, aquilo com o que o cliente pode interagir, e é executado no ambiente de um navegador de internet. As validações de dados de entrada foram desenvolvidas com HTML e CSS, e a ferramenta BootStrap possibilitou a conFiguração da página oferecendo modelos prontos de *design*. Já o framework javascript jQuery assegurou as verificações de dados fornecidos pelos usuários.

4.2.1.2.1 HTML e CSS

As linguagens presentes no front-end possuem padronização feita pelo W3C, *World Wide Web Consortium*, que é responsável pela padronização da *World Wide Web* e foi fundada por Tim Bernes-Lee para levar a web ao seu máximo potencial, orientando seu desenvolvimento.

Já o HTML, do inglês *Hypertext Markup Language*, é uma linguagem de marcação também criada por Tim, ela é conhecida por compor todas as páginas da internet conhecidas e foi criada para ser de fácil entendimento tanto para o homem quanto para a máquina.

Como uma linguagem de marcação, ela é sinalizada por uma *tag*/elemento. Ou seja, para um título é usada a marcação <h1>Texto do Título<h1> e para um parágrafo <p> Texto do parágrafo<p>, por exemplo. Assim, os elementos entre <> são as chamadas *tags* e definem como a linguagem se comporta.

Unicamente, a linguagem HTML se parece com um texto em preto e branco com alguns *links* azuis. Desse modo, para adicionar *designs* a página e tratar a parte gráfica, é necessário acrescentar mais uma linguagem, chamada de CSS, do inglês *Cascada Style Sheets*, que traduzida significa Folha de Estilo em Cascatas e também foi criada pelo W3C.

As *tags* como <font> foram introduzidas na versão 3.2 do HTML e causaram muitos problemas para os desenvolvedores. Como os interface WEBs tinham diferentes fontes, cores e estilos, era um processo longo, doloroso e caro para reescrever o código. Assim, o CSS foi criado pelo W3C para resolver este problema.

A relação entre HTML e CSS é bem sólida. Como o HTML é uma linguagem de marcação (o alicerce de uma interface WEB) e o CSS é focado no estilo (toda a estética), eles andam juntos (GONÇALVES, 2019).

No CSS, existe um seletor que aponta para o elemento HTML que o desenvolvedor deseja estilizar, o bloco de declaração conterá declarações separadas por ponto-e-vírgula. Cada declaração inclui um nome de propriedade e um valor, separados por dois pontos. Uma declaração CSS sempre termina com um ponto-e-vírgula e os blocos de declaração são cercados por chaves (GONÇALVES, 2019). Por exemplo:

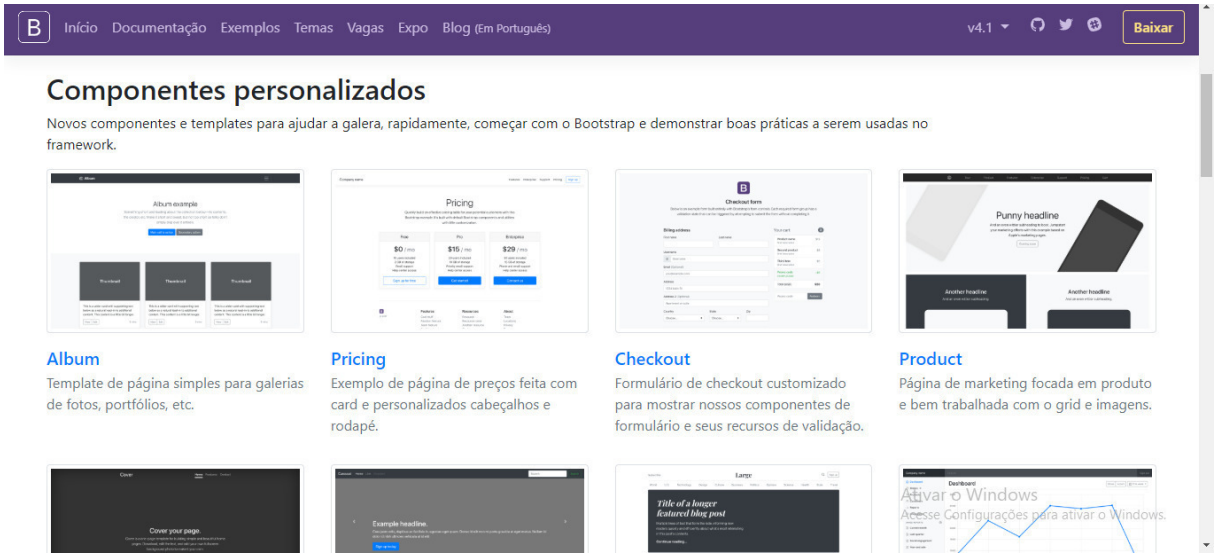
```
<style>
p {
text-align: center;
font-size: 16px;
color: pink;
}
</style>
```

Tendo em vista essas duas linguagens, ainda com o objetivo de facilitar o desenvolvimento da interface WEB e melhorar sua aparência e funcionalidade, o Bootstrap foi utilizado. O Bootstrap é um projeto de código aberto que foi originalmente desenvolvido para o Twitter.

O Bootstrap é um framework web com código-fonte aberto para desenvolvimento de componentes de interface e front-end para interface WEBs e aplicações web usando HTML, CSS e JavaScript, baseado em modelos de design para a tipografia (BOOTSTRAP, 2019). Com

ele, é possível baixar código fonte de exemplos que podem ser modificados da melhor forma possível para sua aplicação, como é mostrado na Figura 4.3.

Figura 4.3 – Interface WEB do Framework Bootstrap



Fonte: Própria autora.

Para usá-lo foi necessário, além do código fonte, copiar e colar o arquivo de estilo <link> dentro de <head> antes de todos os outros arquivos de estilo para carregar seu próprio CSS.

4.2.1.2.2 JavaScript e jQuery

O JavaScript vem para complementar as funcionalidades do HTML e CSS, como se fosse a camada por trás das outras duas, enquanto HTML e CSS fornecem a informação e a aparência da mesma, o JavaScript vem agregar o dinamismo.

Algumas das possibilidades da linguagem é armazenar conteúdo útil em variáveis, operações com *strings*, executar códigos quando eventos são disparados na página web, entre outras coisas.

No Appa Energy, a linguagem é utilizada no momento das verificações dos dados de entrada de usuário que foram explicados anteriormente. Então, o programa identifica a entrada e faz as comparações necessárias, permitindo ou não que o código proceda para a próxima etapa.

Para isso, é utilizada a biblioteca Jquery que interage com HTML, criada em 2006 por John Resig com o propósito de facilitar a vida dos desenvolvedores com o lema "*Escreva menos, faça mais.*". Esta não é uma linguagem separada e funciona em conjunto com o JavaScript e tem a função de compactar várias linhas de código em uma única função.

4.2.1.3 Back-end e PHP

O back-end e o front-end caminham juntos, no front-end a aparência e interface com o usário é implementada. Já no back-end toda a comunicação entre banco de dados e servidor é feita; dessa forma, informações são tratadas e validadas em uma camada que não pode ser alterada pelo cliente.

Na aplicação deste trabalho, foi utilizado PHP (um acrônimo recursivo para PHP: *Hypertext Preprocessor*) é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento WEB e que pode ser embutida dentro do HTML.

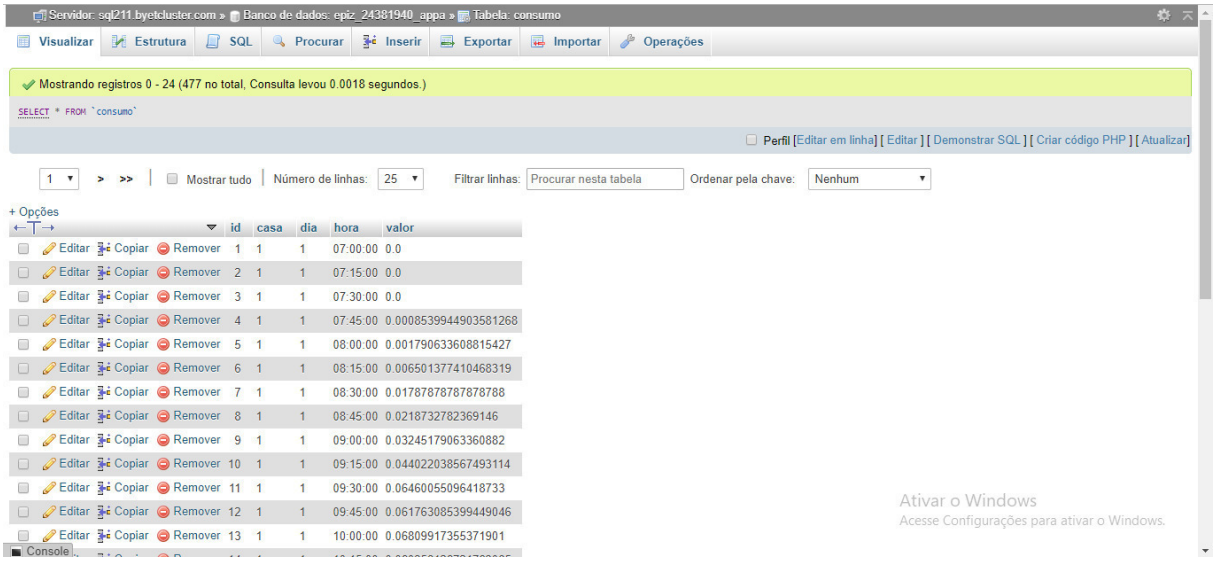
A linguagem foi utilizada para criar um código separado que manipulasse o banco de dados utilizado para armazenar as informações referentes a produção de energia elétrica pelas placas fotovoltaicas das residências, adequando os parâmetros. Depois disso, no código principal, é chamado o código em PHP para que seja apresentado na interface de saída as informações manipuladas.

4.2.1.4 Banco de dados e MySQL

Um pequeno banco de dados foi utilizado armazenar as informações referentes a produção acumulada de cada casa, os valores recolhidos são da interface WEB (PVOUTPUT, 2018) e relacionados à três sistemas diferentes, que serão detalhados posteriormente.

Para que esses dados sejam utilizados no código e apresentados na interface WEB, eles também precisam estar no servidor. Sendo assim, o Infinity Free disponibiliza o banco de dados MySQL para que seja realizado esse armazenamento. A estrutura final do banco de dados utilizado é apresentada na Figura 4.4.

Figura 4.4 – Banco de dados no PHPAdmin



Fonte: Própria autora.

O MySQL é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (*Structure Query Language* – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado em um banco de dados.

4.2.2 Contrato inteligente

Um contrato inteligente precisa ser desenvolvido em linguagem confiável e não ambígua, para isso foi criada a linguagem Solidity, desenvolvida por (WOOD, 2014) para aplicações na plataforma Ethereum. Para a publicação do contrato na rede foi utilizada a carteira MetaMask e para acompanhamento de transações dentro dele utilizou-se a interface WEB Etherscan.

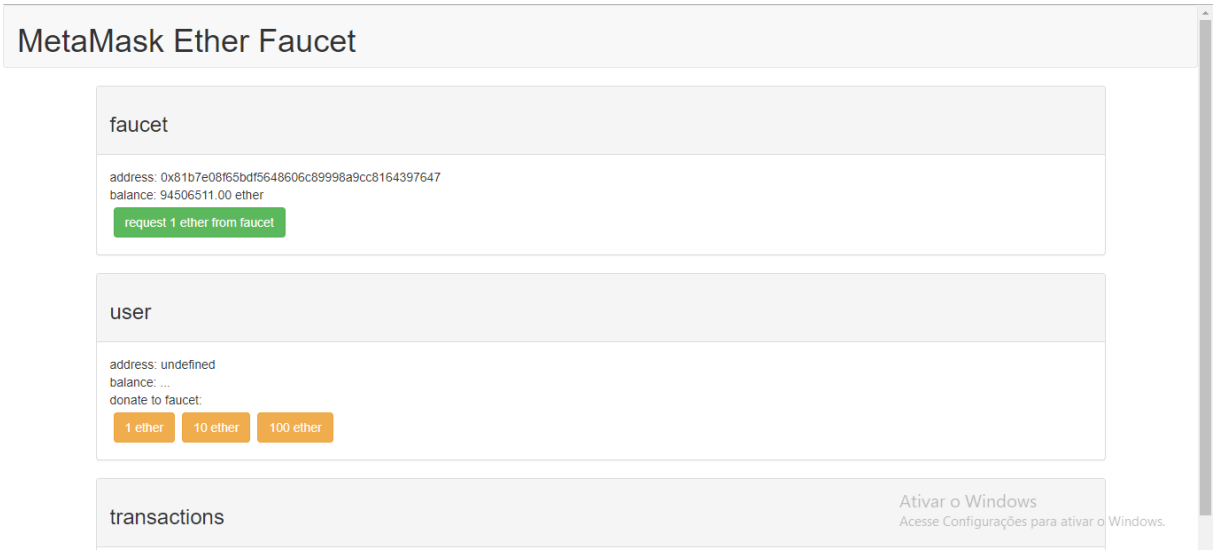
4.2.2.1 MetaMask

MetaMask é uma carteira Ethereum, ou seja, armazena as moedas dos usuários e pode interagir com contratos na Web. Ela funciona como uma extensão dos principais navegadores, como Chrome, Opera, Chromium e Brave, portanto, é uma ferramenta online. Sua instalação acontece como qualquer extensão de navegador convencional, após instalada é possível criar um conta, que inicia sem fundos.

Essa conta possuirá um *hash* único e pode ser usada tanto na main net quanto nas test nets. Quando o MetaMask está com o *login* ativado no navegador, ao acessar interface WEBs com vínculo ao Ethereum, a conta é automaticamente identificada.

Para conseguir REthers, os Ethers da rede de teste Ropsten, que será utilizada neste trabalho, existem interfaces WEB que doam a moeda, mas não é possível pedir muitas de uma vez, pois a quantidade, como os Ethers reais, também é limitada. A interface WEB utilizado é mostrado na Figura 4.5.

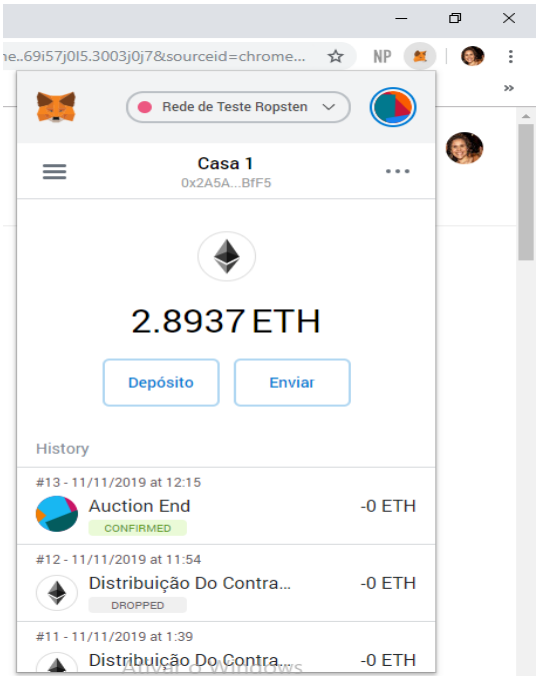
Figura 4.5 – Faucet para obtenção de REthers



Fonte: Própria autora.

No MetaMask, além da quantidade de Ethers que o usuário possui, também é possível enxergar as últimas transações em que esteve envolvido, como mostrado na Figura 4.6. Existe a possibilidade de gerenciar mais de um nó Ethereum, com a mesma conta criada no MetaMask. É importante destacar a rede Ropsten selecionada no canto superior da extensão para a realização de todas as transações deste trabalho.

Figura 4.6 – Extensão Metamask no navegador Chrome

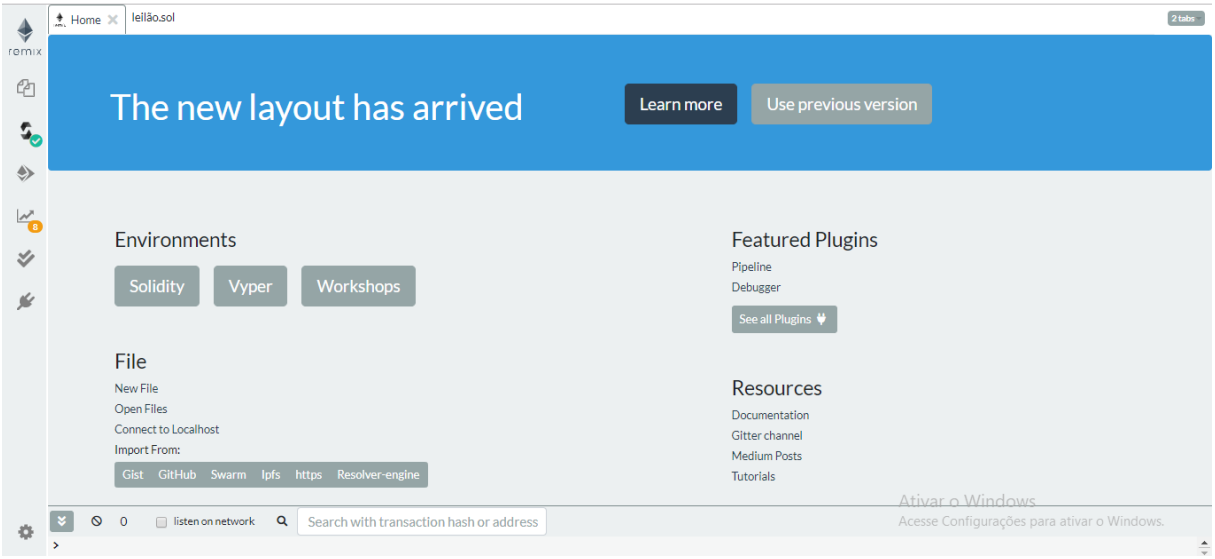


Fonte: Própria autora.

4.2.2.2 Remix

O Remix é uma IDE baseada em *browser* e pode ser acessada por qualquer navegador. Em sua página inicial, apresentada na Figura 4.7, existem guias de uso, recursos e opções de escolha de qual ambiente o desenvolvedor deseja programar, assim como a opção de criação de novos arquivos e abertura de arquivos pré-existentes.

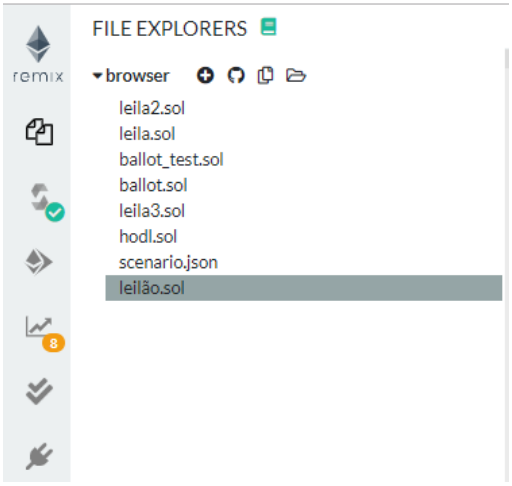
Figura 4.7 – IDE Remix



Fonte: Própria autora.

Existem ícones na barra lateral esquerda que serão úteis para o desenvolvimento, compilação e publicação do contrato. O primeiro ícone, mostrado na Figura 4.8, exibe o explorador de arquivos onde ficam os contratos escritos e alguns contratos padrões que podem ajudar iniciantes na programação. Arquivos em Solidity são salvos com a extensão .sol.

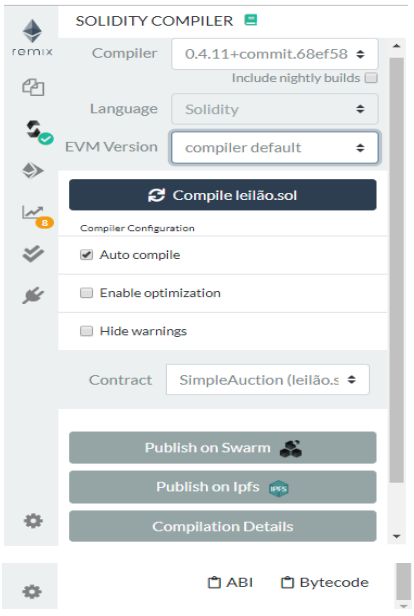
Figura 4.8 – Explorador de arquivos do Remix



Fonte: Própria autora.

O segundo ícone, apresentado na Figura 4.9, se refere ao compilador do contrato, no leilão foi utilizada a linguagem Solidity e o compilador utilizado foi a versão 4.11 como apresentado abaixo. Além disso, existe a possibilidade de selecionar a caixa "Auto compile" para que a compilação aconteça de forma automática. Nessa aba, a ABI, do inglês *Application Binary Interface*, ou seja, Interface Binária de Aplicação, do contrato fica armazenada e disponível para cópia.

Figura 4.9 – Compilador do Remix

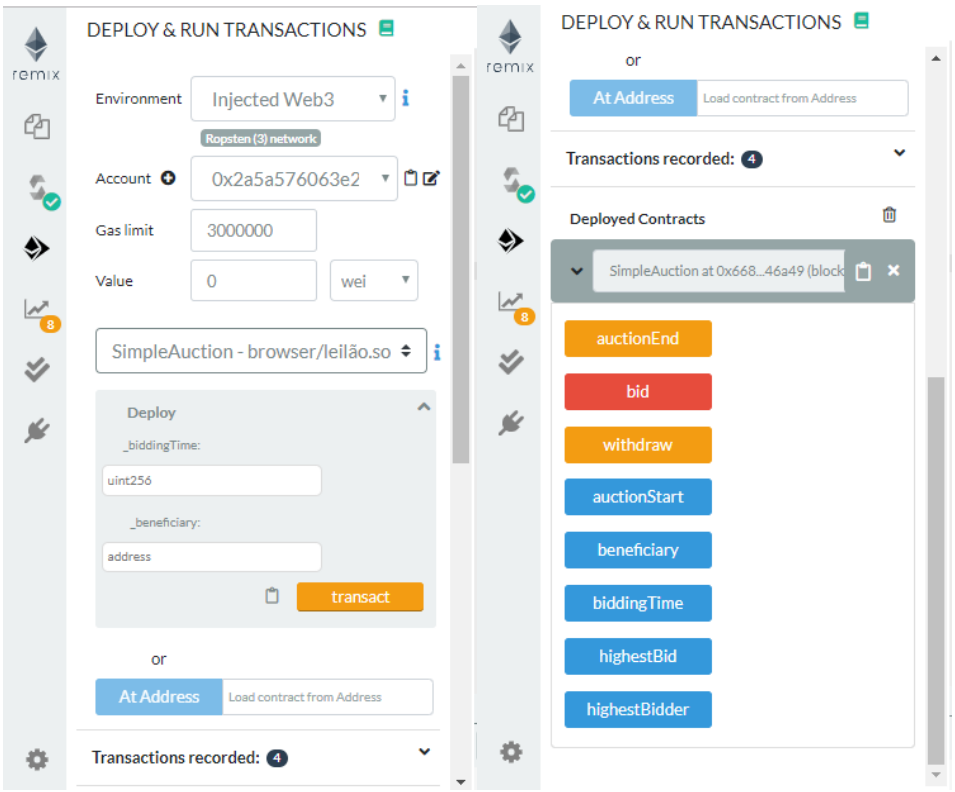


Fonte: Própria autora.

O terceiro ícone, da Figura 4.10, é responsável pela publicação do contrato, chamado de *deploy*, para iniciar é necessário escolher o "Enviroment", que é o ambiente no qual se dará essa publicação, são três opções: JavaScript VM, Injected Web3 e Web3 provider. A primeira diz respeito à quando são realizados apenas testes do contrato, sem se preocupar com a interação externa a IDE. As duas opções seguintes permitem sua publicação para a *Blockchain* do Ethereum, a diferença entre elas está na forma de conexão do nó com a *Blockchain.* Enquanto na opção Injected Web3, o Metamask será responsável pela conexão com o nó da rede, na Web3 provider a configuração do nó é manual.

Neste trabalho optou-se por utilizar a Injected Web3 e o desenvolvedor precisará usar a conta criada no MetaMask. Parte de seus Ethers serão utilizados na publicação do contrato, pagando os custos da publicação na rede, mesmo utilizando a rede de teste Ropsten.

Figura 4.10 – Transações e deploy no Remix



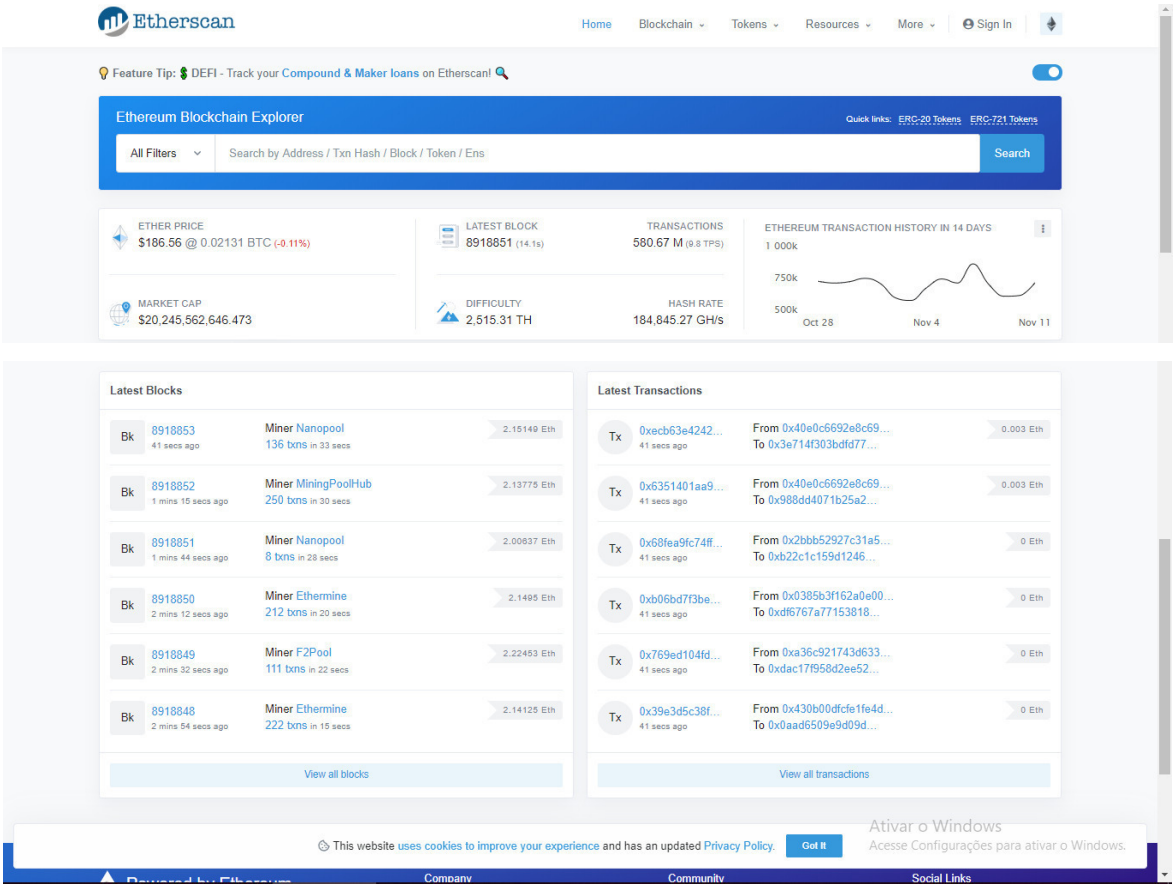
Fonte: Própria autora.

4.2.2.3 Etherscan

O Etherscan é uma interface WEB desenvolvido para acompanhamento da *Blockchain* do Ethereum, ele mantém um histórico de transações e serve para acompanhar as entradas e saídas de contratos, contas e *tokens*. Esta ferramenta foi usada neste trabalho para o monitoramento da criação dos contratos e das transações que exibirão os resultados de leilão.

Na Figura 4.11, é possível ver a página inicial do Etherscan, com informações de preço do Ether em dólares, que pode ser alterado nas preferências para outras moedas, além de informações como os blocos movimentados e os níveis de dificuldade.

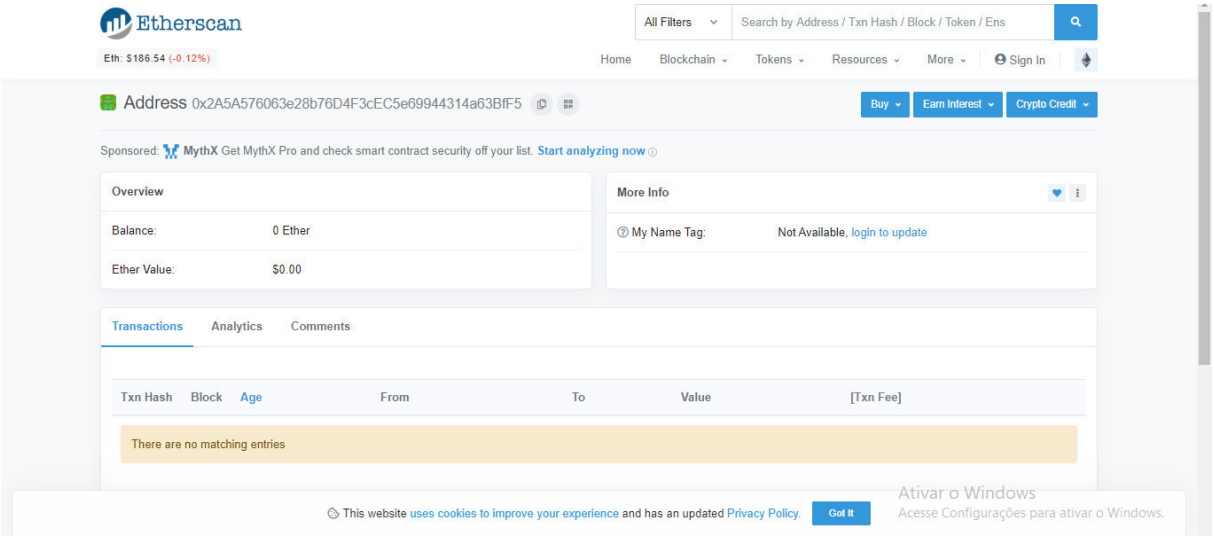
Figura 4.11 – Página Inicial do Etherscan



Fonte: Própria autora.

Para consultar o histórico, basta informar o *hash* do contrato ou da conta a ser monitorada no campo "*Search by Adress/Txn Hash/Block/Token/Ens*". Uma página como a mostrada na Figura 4.12 será apresentada. No *Overview* é possível ver o balanço da conta, o valor do Ether, e as transações realizadas podem ser analisadas no canto inferior da página, além de outras observações.

Figura 4.12 – Detalhes de transações em contas/contratos



Fonte: Própria autora.

4.2.3 Integração das funcionalidades

Após entender as ferramentas utilizadas nos desenvolvimentos da interface WEB e do contrato inteligente, faz-se necessário compreender de que forma as duas irão trocar dados entre si para configurar uma aplicação WEB dentro do contexto descentralizado da *Blockchain* do Ethereum.

O Web3.js permite que interfaces WEB interajam com a *Blockchain*, é uma coleção de bibliotecas que permite executar ações como: enviar Ether de uma conta para outra, ler e gravar dados de contratos inteligentes, criar contratos inteligentes, dentre outras ações relacionadas a *Blockchain*.

Existe uma semelhança nas funções do Web3.js e do Jquery, ou seja, ao invés de usar o Jquery para ler e gravar dados de um servidor da WEB, como foi apresentado anteriormente, o Web3.js lê e grava dados na *Blockchain* do Ethereum.

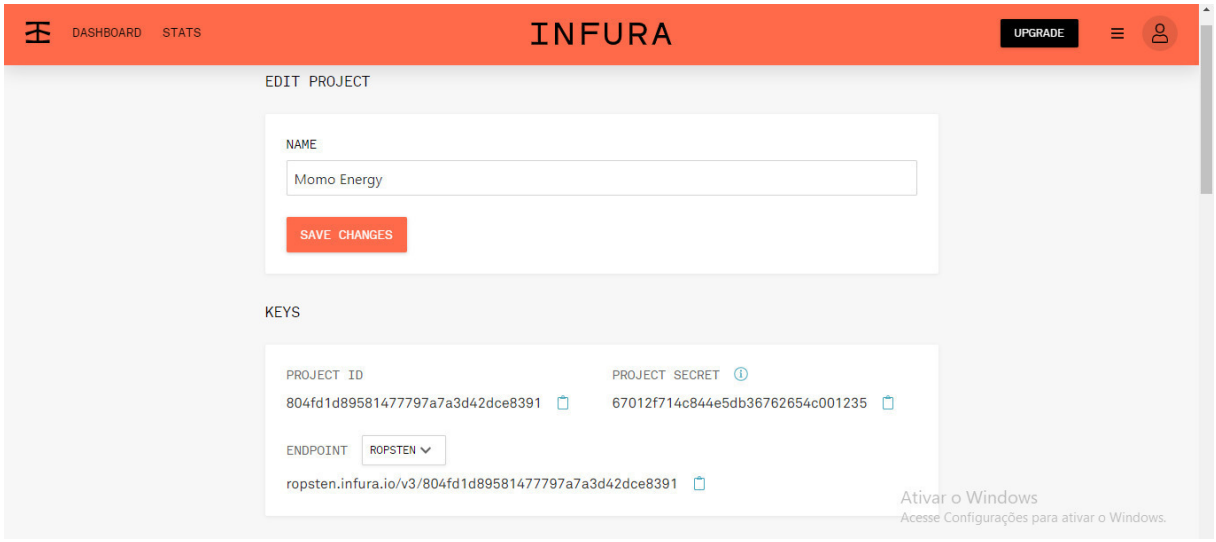
Além disso, o Web3.js interage com a *Blockchain* do Ethereum com o JSON RPC, do inglês *Remote Procedure Call*, ou seja, chamada de procedimento remoto. O Ethereum é uma rede de nós ponto a ponto e o Web3.js permite fazer solicitações para um desses nós individuais com o JSON RPC para ler e gravar dados na rede (GREGORY, 2019).

Para se conectar a um nó Ethereum com JSON RPC na rede, é possível executar um nó próprio, mas isso exige que muitos dados da *Blockchain* sejam baixados e precisem ser sincronizados.

Para facilitar esse processo, o serviço Infura dá acesso a um nó do Ethereum sem

que o desenvolvedor precise executá-lo. O Infura é um serviço que fornece um nó remoto gratuitamente, é necessário apenas se inscrever e obter a URL RPC da rede a qual é desejada a conexão. Neste trabalho, a rede Ropsten foi utilizada e o serviço Infura com essas informações é apresentado na Figura 4.13.

Figura 4.13 – Nó remoto na interface WEB Infura



Fonte: Própria autora.

5 IMPLEMENTAÇÃO E ESTUDO DE CASO

Explicadas as lógicas de funcionamento do sistema WEB e dos contratos inteligentes, além das ferramentas utilizadas para seus desenvolvimentos, são mostradas a seguir as particularidades da programação e os fluxos envolvidos na modelagem do sistema.

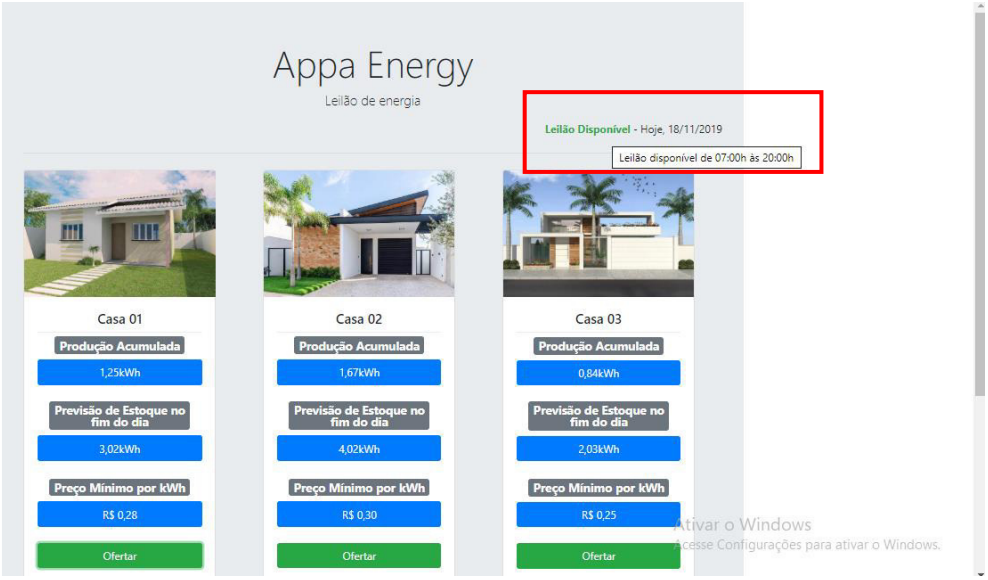
5.1 Definições gerais da implementação

5.1.1 Interface WEB

No design da interface WEB, utilizou-se o framework CSS Bootstrap para o posicionamento das colunas e para a criação de botões e campos para entradas de dados pelos usuários. Além disso, foram realizadas pequenas alterações nos códigos em HTML e CSS para ajustar os elementos da maneira desejada.

Na tela principal é exibida, no canto superior direito, a data em que o usuário está acessando a plataforma, além de informações relativas ao leilão, como “disponível” ou “indisponível”. Os horários definidos para que o leilão fique ativo são entre 07h00 e 20h00, ou seja, o motivo para o leilão ficar indisponível é justificado pelos horários de produção das placas solares. Sendo assim, em horários que não há irradiação solar o leilão permanece indisponível à lances. Por isso, nesses momentos, a interface WEB apresenta a mensagem de “Leilão Indisponível”. A interface WEB é mostrada na Figura 5.1.

Figura 5.1– Leilão disponível



Fonte: Própria autora.

Os dados de saída da interface WEB para o usuário também podem ser vistos na Figura 5.1, são eles: Produção Acumulada (prod_acum), Previsão de Estoque no fim do dia (previsao_dia) e Preço Mínimo por kWh. O valor de produção acumulada é coletado do banco de dados por meio de manipulação em PHP. Esse valor é atualizado a cada 15 minutos e carregado no servidor.

Os valores de produção das casas foram coletados da interface WEB PVOutput (PVOUTPUT, 2018), que oferece um serviço grátis de compartilhamento e comparação de informações de geração fotovoltaica. Os dados recolhidos são referentes à três sistemas diferentes que estão localizados na cidade de Embu das Artes, no estado de São Paulo. A potência instalada dos dois primeiros sistemas é de 6,05 kW cada, e do terceiro é de 12,1 kW.

A taxa de amostragem dos dados era de 5 minutos e em (TAVARES, 2019) foi realizado um tratamento de dados para que se passasse a representar uma taxa de aquisição de 15 minutos. Neste trabalho foram utilizados os dados tratados, ou seja, com taxa de amostragem de 15 minutos.

No entanto, os dados representavam a potência média no intervalo de 15 minutos e estavam normalizados em relação a potência instalada de cada planta. Para esta aplicação, se fazia necessário os valores em energia (kWh), então, foram efetuadas simples operações para essa conversão. Neste trabalho, utilizou-se dois dias de cada sistema e eles alternam-se na interface WEB.

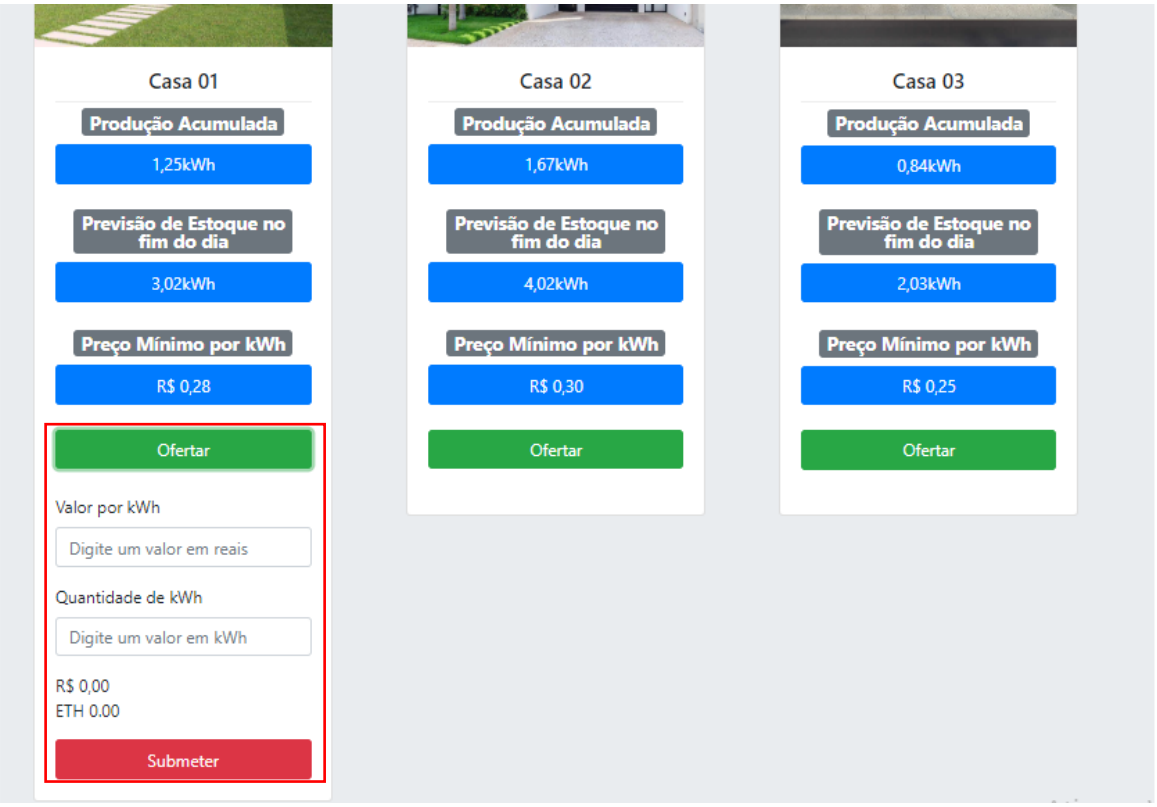
O valor da previsão de estoque foi projetado da seguinte forma: o valor de produção acumulada é dividido pelo número de intervalos capturados até o momento, ou seja, a média continua utilizando os valores anteriores e a cada 15 minutos é incrementada com os novos. Em seguida, essa média é multiplicada pelo número de intervalos que existem entre 07h00 e 20h00 horas, então obtêm-se uma previsão de estoque que varia ao longo do dia de acordo com as produções reais anteriores, de acordo com a Equação 1.

$$previsao_{dia} = \frac{produção_{acumulada}}{intervalos_{capturados}} * 53 \quad (1)$$

Já o preço mínimo é um valor fixado por cada casa, e o critério utilizado baseou-se num valor inferior às tarifas de energia convencionais. De acordo com (ENEL, 2019), no Ceará é cobrado em média 0,45R$ para cada kWh consumido de energia elétrica para a classe residencial atendida em baixa tensão, chamada de grupo B, na faixa entre 31 e 100kWh em bandeira amarela. Então, para manter um preço competitivo, a média dos preços escolhida é 30% inferior a cobrada pela distribuidora de energia elétrica.

Depois de visualizar os valores e ofertas de todas as casas, o usuário é capaz de realizar uma oferta para uma ou mais delas. Ao pressionar o botão verde “Oferta”, uma aba com novas informações é aberta e o usuário estará habilitado a inserir sua proposta. Esta aba está destacada na Figura 5.2.

Figura 5.2 – Aba de oferta



Fonte: Própria autora.

Ademais, existem dois campos que precisam ser preenchidos pelo usuário, são eles o Valor por kWh (valor_kwh), ou seja, o valor que ele pretende pagar por cada kWh e a Quantidade de kWh (qtd_kwh), que é a quantidade de kWh que ele deseja comprar ao fim do dia. Esses dois valores são multiplicados e, por fim, é exibido o valor da oferta no canto inferior aos campos, tanto em reais quanto em ETH. Como é indicado na Figura 5.3.

Figura 5.3 – Cálculo da oferta

Fonte: Própria autora.

Para a conversão da oferta do usuário de reais para ETH, é necessário escolher um valor adequado. Sabendo que o preço das criptomoedas varia diariamente de acordo com sua valorização, semelhante às moedas convencionais, as cotações utilizadas nesta plataforma foram tomadas da interface WEB Etherscan.

Existem alguns sistemas WEB que acompanham, em períodos que variam de minutos a dias, a valorização da criptomoeda ETH. Existe uma API desenvolvida por (MERCADOBITCOIN, [s.d.]) que retorna os principais dados do mercado de criptomoedas no formato JSON. Então, esta API foi integrada a aplicação WEB apresentada nesse trabalho que utiliza o valor do ETH fornecido pelo sistema para a conversão de reais para ether.

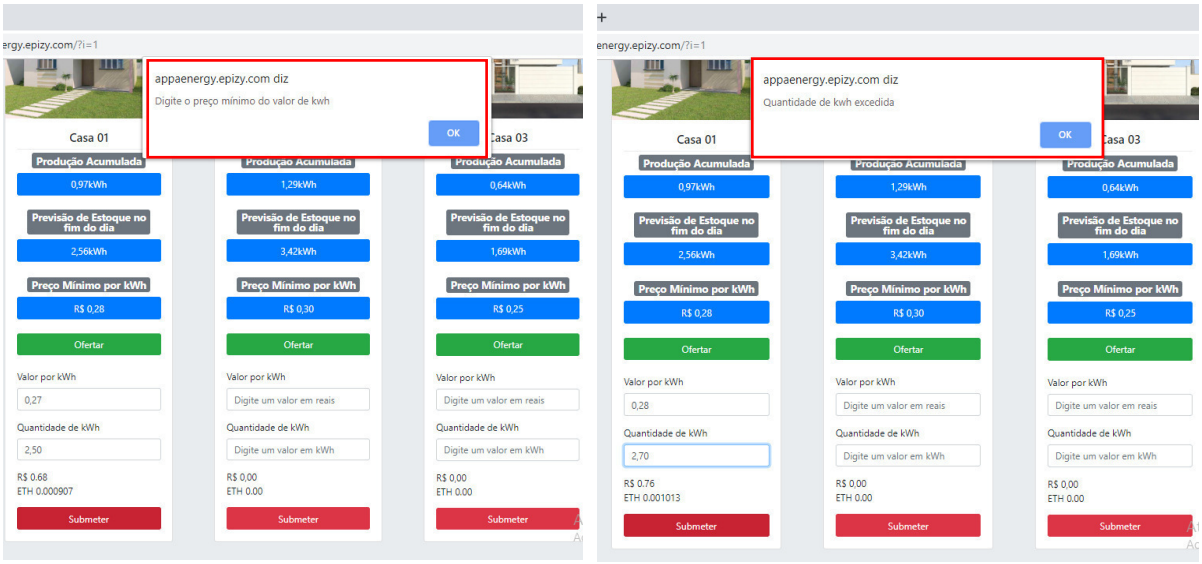
Na Figura 5.4 é possível ver esse comportamento oscilatório no ano de 2019. Os preços mínimo e máximo foram, respectivamente, R$ 388,79 e R$ 1303,55 e o preço médio até metade de novembro deste ano foi de R$ 738,58.

Figura 5.4 – Cotação de ETH ao longo de 2019

Fonte: Própria autora.

Para que o usuário seja capaz de submeter uma oferta, os valores inseridos precisam cumprir alguns requisitos, que são: o valor do kWh inserido pelo usuário não pode ser menor que o preço mínimo estabelecido pela casa; e a quantidade de kWh não pode exceder a previsão de estoque no fim do dia, para que o cliente não faça um lance para um valor que não pode ser entregue pela casa. Se o usuário tentar submeter valores que não cumpram os critérios, dois avisos podem ser exibidos na tela e são apresentados na Figura 5.5.

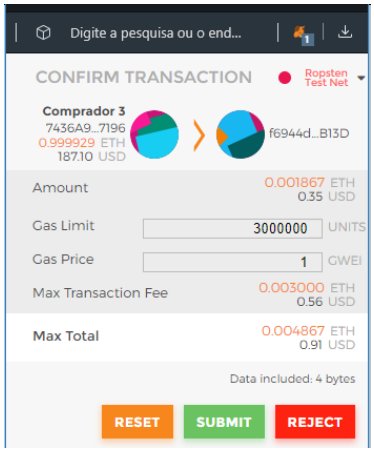
Figura 5.5 – Aviso de preço mínimo e quantidade de kWh excedida



Fonte: Própria autora.

Após a verificação dos dados inseridos, o back-end da aplicação estabelece uma conexão com o nó de rede e, através dos comandos fornecidos pela biblioteca Web3.js, uma transação é gerada. Por conseguinte, o MetaMask, que deve estar previamente logado na conta do usuário, abre uma janela para confirmação da transação com os dados inseridos via programa e o valor em Ether convertido pela oferta do usuário, como é mostrado na Figura 5.6.

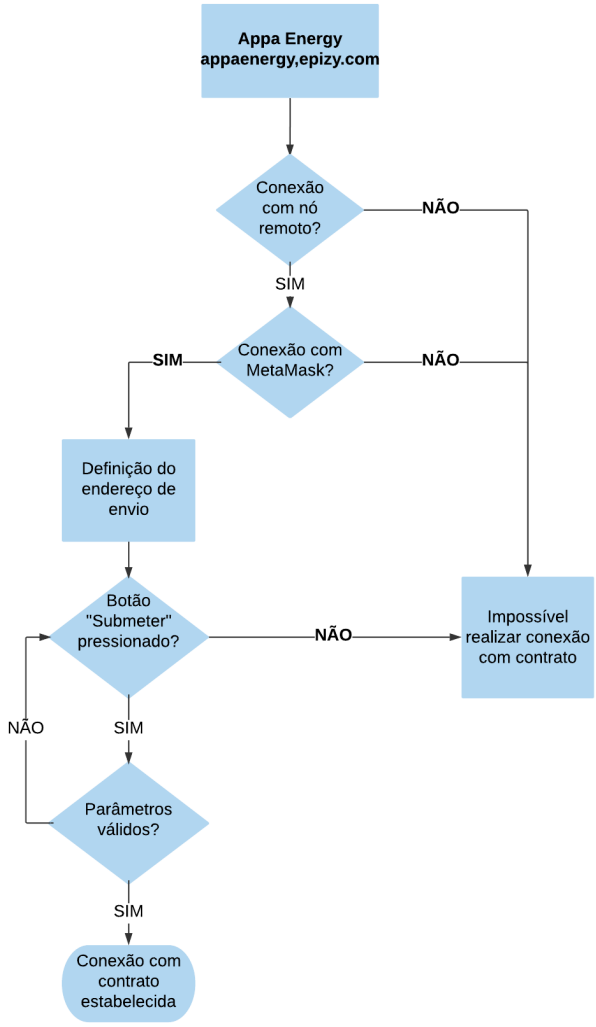
Figura 5.6 – Transação no MetaMask



Fonte: Própria autora.

Nesse momento, é necessário entender como a transação foi criada a partir do programa integrado com o Web3.js. O fluxograma que esboça a integração que ocorre ao acessar a aplicação até que seja estabelecida a conexão com o contrato é apresentado na Figura 5.7.

Figura 5.7 – Fluxograma de integração



Fonte: Própria autora.

Então, ao estabelecer uma conexão com a interface WEB, o usuário se conecta ao nó de rede remoto do Infura. Logo após, é definido se há uma injeção de web3 por parte do *browser*, ou seja, é verificado se o usuário está usando algum programa que possibilita a interação com a *Blockchain* do Ethereum.

Em seguida, define-se que o endereço (*account*) que foi realizado o acesso deve ser o endereço que fará as chamadas das funções da interface WEB, no caso será o endereço que realizará uma transação para dentro do contrato.

Para instruir o *browser* como interagir com o contrato, é necessário especificar a interface binária de aplicação ou ABI, que pode ser copiada da IDE remix. Então, uma variável com o endereço do contrato publicado é definida, que orientará para onde será realizada a transferência em “to”.

Então, é definido o que acontecerá quando o botão “Submeter” for acionado pelo usuário. Após a verificação dos valores, a função do contrato que será executada é definida. Neste trabalho, a função “bid” é a função de depósito.

Em seguida, algumas especificações são feitas:

1) From: “Endereço account”;
2) Gas: “Limite de gás”;
3) To: “Endereço do contrato”;
4) Value: 1000000000000000000*(Valor da conversão para Ethereum realizada anteriormente), multiplicador necessário para converter de Gwei para ETH;

Então, são definidos os termos da transação. E, por fim, uma janela é exibida para que o usuário possa conferir os dados propostos e escolher as opções de “Submit”, para prosseguir com a transação, ou “Reject”, para cancelar.

5.1.2 Lógica do contrato inteligente

Os contratos inteligentes utilizados neste trabalho são baseados no código de leilão simples apresentados na documentação do Solidity (ETHEREUM, [s.d.]). Para efeitos de simplificação e automatização, algumas funções foram alteradas ou removidas. As definições e principais aspectos do código são detalhados a seguir.

A versão do compilador utilizada no contrato foi a 0.4.21 definida no início do código, em seguida o contrato é definido através da função “c*ontract*”. Algumas variáveis são criadas para a inicialização do contrato, são elas: “casa”, “ini_leilao”, “tempo_leilão” e “fim”.

“casa”: recebe o valor de “*msg.sender*”, isso significa que o endereço que cria o contrato será o endereço associado a essa variável. É definida como tipo “*address public*”, “*address*” é um tipo de variável Solidity que corresponde a um *hash* de 160 bits, tamanho referente aos endereços da plataforma. “*public*” significa que essa variável pode ser acessada externamente por outros usuários e contratos;

“ini_leilão”: recebe o valor de “*now*”, valor em “Unix timestamps” que é dado em segundos desde 01/01/1970 até agora. É definida como “*uint public*”, “*uint*” sendo o tipo para números inteiros sem sinal;

“tempo_leilão”: é definido como “*15 hours*”, que se refere ao tempo disponível de leilão entre 07h00 e 20h00 e como “*uint public*”;

"fim": recebe o valor "*false*" e é definida como "*bool"*, tipo booleano que pode ter apenas os valores "*false*" e "*true*".

Em seguida, mais 4 variáveis são definidas para a operação das funções, são elas a "maior_oferta", "antiga_oferta", definidas como "*uint public*" e "maior_ofertante", "antigo_ofertante", definidos como "*addres public*", todas recebem por padrão o valor inicial zero.

Duas funções são definidas no contrato, a função de "proposta" e a função "fim_leilão". A função "proposta" tem como objetivo receber a oferta em Ether realizada através da aplicação WEB, para receber transações em Ether da plataforma, as funções devem ser definidas pelo parâmetro "*payable*".

Depois de sua definição, existem três "*requires*", funções nativas da linguagem Solidity com condições que interrompem o código caso não sejam atendidas. As condições que devem ser cumpridas para a execução da função são: o valor enviado ("*msg.value*") deve ser maior que zero, superior a maior oferta já realizada e o marcador de tempo "*now*", próprio da linguagem, deve ter o valor inferior a soma de "ini_leilao" mais "tempo_leilao".

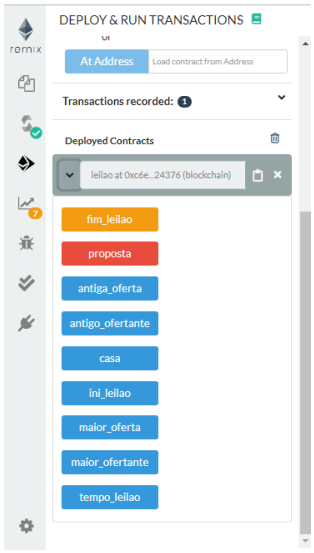
Então, os valores de "maior_oferta" e "maior_ofertante" serão armazenados nas variáveis "antiga_oferta" e "antigo_ofertante", é realizado um teste usando a função "*if*" para saber se esse valor é superior a zero e o valor é devolvido para o comprador que teve sua oferta superada, então um evento é disparado, na linguagem Solidity os eventos servem para sinalizar a execução de uma função específica, nesse caso, o evento é chamado através da função "*emit*" junto de "RestituiçaoOferta".

As variáveis de "maior_oferta" e "maior_ofertante então recebem os valores de "*msg.sender*" e "*msg.value*", que são referentes ao comprador que executou a função e o evento de "NovaMaiorProposta" é chamado.

A função "fim_leilão" é definida sem o parâmetro payable, pois não é necessário que o usuário que a chame efetue qualquer pagamento. Duas funções "*require*" são criadas, uma exige que o usuário ("*msg.sender*") seja igual a "casa", ou seja, o dono do contrato e a outra requer que o tempo "*now*" seja superior ao tempo estimado do leilão, "ini_leilão" mais "tempo_leilao". Então, o valor da "maior_oferta" é enviado para "casa" através da função "*transfer*", própria da linguagem e um evento de "FimDoLeilao" é emitido.

Na Figura 5.8, em amarelo é observada a função que executa o fim do leilão, em vermelho tem-se a função pagável para realização das propostas e em azul estão as variáveis que são manipuladas ao longo do código. A estrutura do código pode ser vista no Apêndice A.

Figura 5.8 – Funções de chamada



Fonte: Própria autora.

5.2 Estudo de caso

Neste item, será proposta uma simulação de cinco compradores diferentes fazendo ofertas em cada uma das três casas, as propostas feitas por cada um deles é conhecida e, portanto, os resultados esperados de vencedores devem condizer com a proposta mais alta.

5.2.1 Criação das partes do leilão

Para a realização dessa simulação, foi necessário criar mais endereços no MetaMask, a mesma conta pode ser utilizada. Desta forma, foram criadas outras duas casas e cinco endereços de compradores, que estão expostos na Tabela 5.1.

Tabela 5.1 – Partes integrantes do leilão

Participante	Endereço
Casa 1	0x2A5A576063e28b76D4F3cEC5e69944314a63BfF5
Casa 2	0x6e2784B0E04eEA4d504D2e11eA33661aAEaa5046
Casa 3	0xA84716227668128aB9Ec7376044f8b40C5b0b838
Comprador 1	0x36255d2048f3Daf941401c0b13f08B0891259179
Comprador 2	0x29457870079878Af5c6C7f0ecbaadD30C9B02fAe
Comprador 3	0x7436A938F4e3Ab34204Cb80692C49F1BfdF67196
Comprador 4	0xCC84156D9cDEe9B683178ed14E7ddF58E9AD406B
Comprador 5	0x17A11C72c5deaB5A8a55bABb619083Bf0E76c707

Fonte: Própria autora.

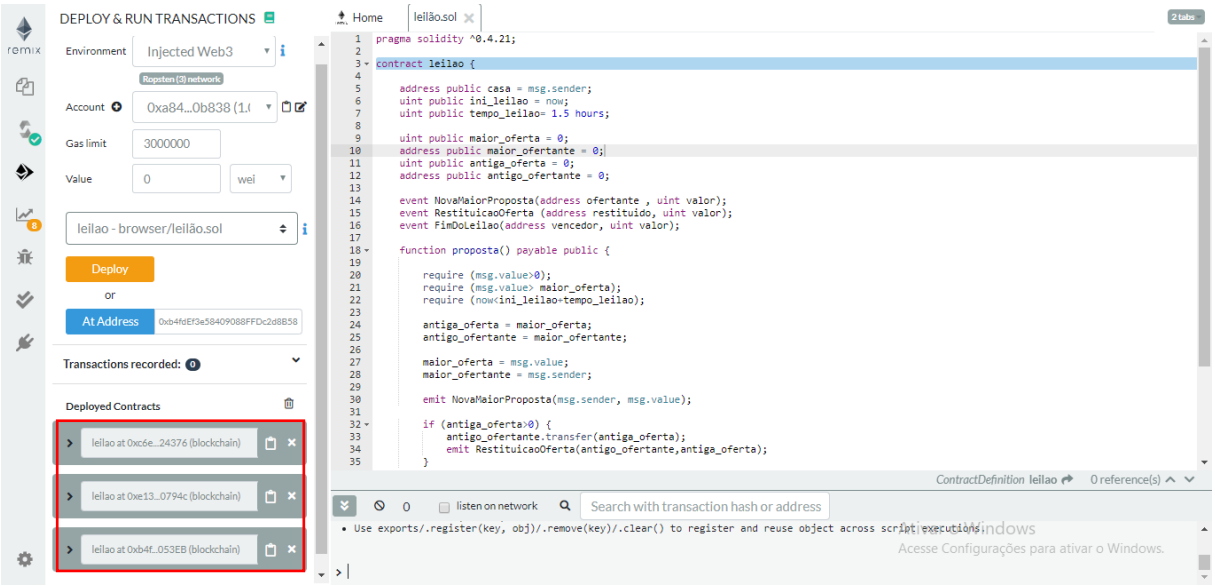
Esses endereços, posteriormente podem ser acessados através da interface WEB Etherscan para visualização de todas as transações existentes e é desta forma que os resultados serão checados.

5.2.2 Criação dos contratos de leilão

Depois de definidas as partes do leilão, os contratos de cada casa são criados. A publicação é feita através da IDE Remix conforme os passos explicitados no Capítulo 4. É realizada a compilação na versão adequada, neste caso, a 0.4.21. Em seguida é feito o "*deploy*" no ambiente "*Injected Web3*" utilizando a conta do MetaMask referente a casa em que se deseja criar o contrato, pois, como visto na lógica do código, o endereço da casa se torna o beneficiário do leilão.

Após a transação ser confirmada na janela do MetaMask, o contrato é publicado na *Blockchain* e recebe um endereço para que seja usado na interface WEB, como é visto na Figura 5.9. Desta forma, os três contratos foram criados e seus endereços podem ser vistos na Tabela 5.2.

Figura 5.9 – Criação do Contrato em linguagem Solidity



Fonte: Própria autora.

Tabela 5.2 – Endereços dos contratos de cada casa

Casa	Endereço do contrato de leilão
1	0xc6e833225B8965F1927696eaA76549c0EFa24376
2	0xe134801AC6a4aD202c65954961962115E4e0794c
3	0xb4fdEf3e58409088FFDc2d8B58aF4B9760b053EB

Fonte: Própria autora.

5.2.3 Ofertas dos compradores

Com os contratos já publicados, a interface WEB está pronta para receber as ofertas, a hora escolhida para a realização das propostas foi aproximadamente às 13h56, neste momento, os dados de produção acumulada e previsão de estoque eram os mostrados na Figura 5.10.

Figura 5.10 – Valores de kWh da interface WEB no momento das ofertas

Fonte: Própria autora.

A partir desses dados, a Tabela 5.3 resume as propostas dos cinco compradores em cada casa, a primeira coluna apresenta o comprador, seguida da casa em que realizará a oferta, o preço que o comprador deseja pagar por kWh e a quantidade de kWh. A última coluna é o resultado em reais da proposta, que no ato do envio será convertido em ETH.

Tabela 5.3 – Ofertas dos compradores

Comprador	Casa	Preço	kWh	Total
Comprador 1	1	R$ 0,60	4,5	R$ 2,72
Comprador 1	2	R$ 0,32	9,4	R$ 3,00
Comprador 1	3	R$ 0,25	4,7	R$ 1,17
Comprador 2	1	R$ 0,61	4,5	R$ 2,77
Comprador 2	2	R$ 0,60	3,0	R$ 1,82
Comprador 2	3	R$ 0,60	3,0	R$ 1,82
Comprador 3	1	R$ 0,40	5,3	R$ 2,12
Comprador 3	2	R$ 0,40	5,3	R$ 2,12
Comprador 3	3	R$ 0,40	5,3	R$ 2,12
Comprador 4	1	R$ 1,00	1,5	R$ 1,51
Comprador 4	2	R$ 1,00	1,5	R$ 1,51
Comprador 4	3	R$ 1,00	1,5	R$ 1,51
Comprador 5	1	R$ 0,28	6,1	R$ 1,69
Comprador 5	2	R$ 0,30	6,1	R$ 1,82
Comprador 5	3	R$ 0,25	4,7	R$ 1,17

Fonte: Própria autora.

No fim do leilão, a partir dessas ofertas, espera-se que os vencedores do leilão de cada casa sejam os apresentados na Tabela 5.4.

Tabela 5.4 – Vencedores do leilão

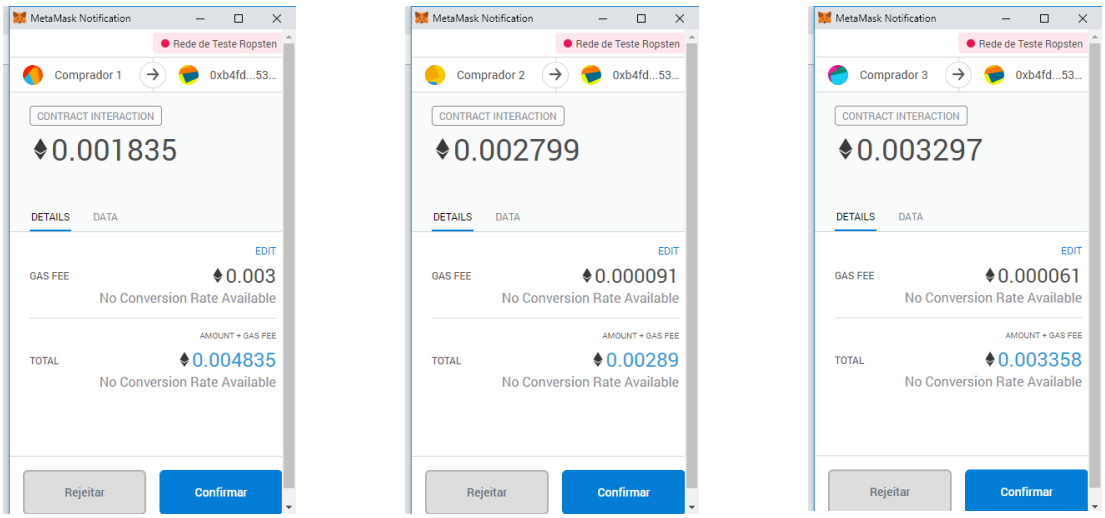
Casa	Vencedor do leilão
1	Comprador 2
2	Comprador 1
3	Comprador 3

Fonte: Própria autora.

De acordo com os valores estabelecidos na Tabela 5.3, as Figuras 5.11, 5.12 mostram a simulação das ofertas dos cinco dos compradores na casa 3. Os valores de kWh e quantidade de kWh são inseridos, de onde o valor total é calculado. Em seguida, é clicado o botão “Submeter” que estabelece uma transação com o contrato referente a casa através do MetaMask.

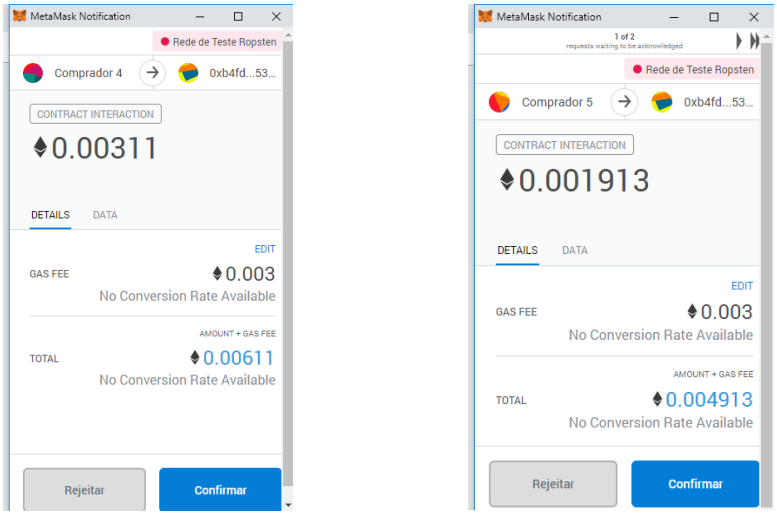
No cabeçalho de cada transação pode ser observado de onde ela parte, no caso, de cada comprador, para onde ela vai, o endereço do contrato da casa 3. As ofertas de cada comprador nas casas foram realizadas da mesma maneira e, para fins de simplificação, não são apresentadas neste item.

Figura 5.11 – Oferta na casa 3 dos compradores 1, 2 e 3



Fonte: Própria autora.

Figura 5.12 – Oferta na casa 3 dos compradores 4 e 5

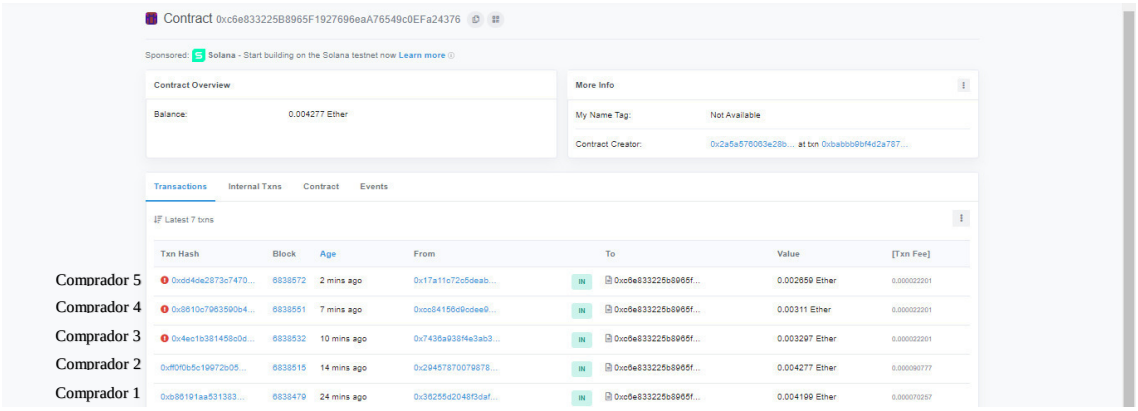


Fonte: Própria autora.

5.2.4 Resultados dos leilões

Para verificar os resultados, as transações envolvendo todos os contratos são verificadas. Para o contrato referente à casa 1 é esperado que o comprador 2 seja o vencedor, como pode ser visto abaixo, a transação que corresponde ao comprador 1 é aceita e logo após o comprador 2 faz um lance superior, os avisos em vermelho ao lado das outras transações querem dizer que foram automaticamente revertidas, como é esperado, pois não cumpriram os requisitos para concluírem a proposta, de acordo com o código do contrato.

Figura 5.13 – Transações ocorridas no contrato 1

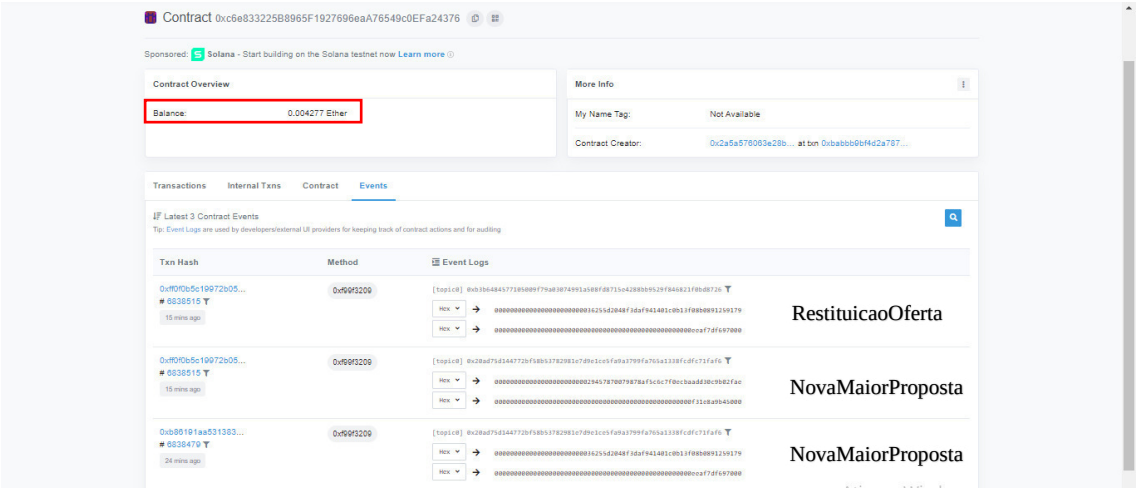


Fonte: Própria autora.

Na Figura 5.14 podem ser observados os eventos que foram ativados durante o funcionamento do contrato. “NovaMaiorProposta” é emitido após a primeira oferta, que logo em seguida é superada, emitindo dois novos eventos. São eles, novamente, um referente a “NovaMaiorProposta” seguido de um “RestituicaoOferta”, que acontece pela devolução do

valor da maior proposta antiga. Pode ser observado também o valor que o contrato carrega ao final das propostas em "*Balance*" no canto superior esquerdo das Figuras 5.13 e 5.14.

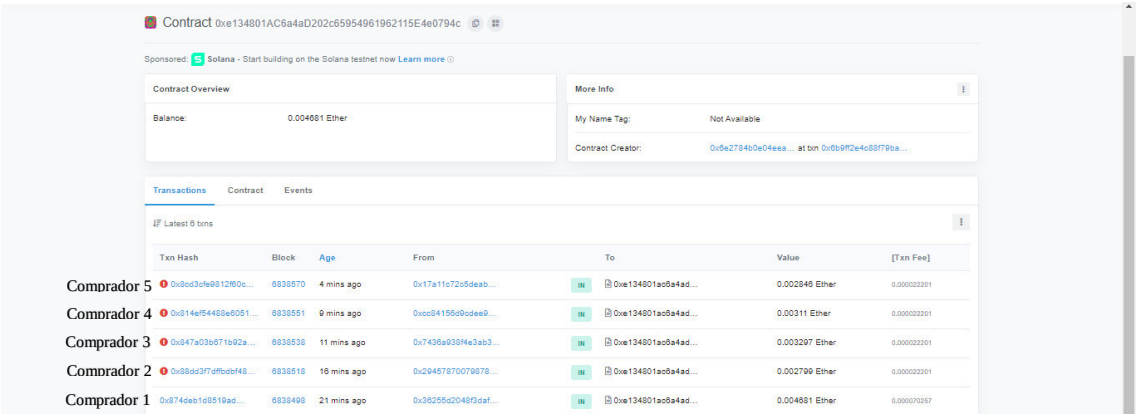
Figura 5.14 – Eventos disparados no contrato 1



Fonte: Própria autora.

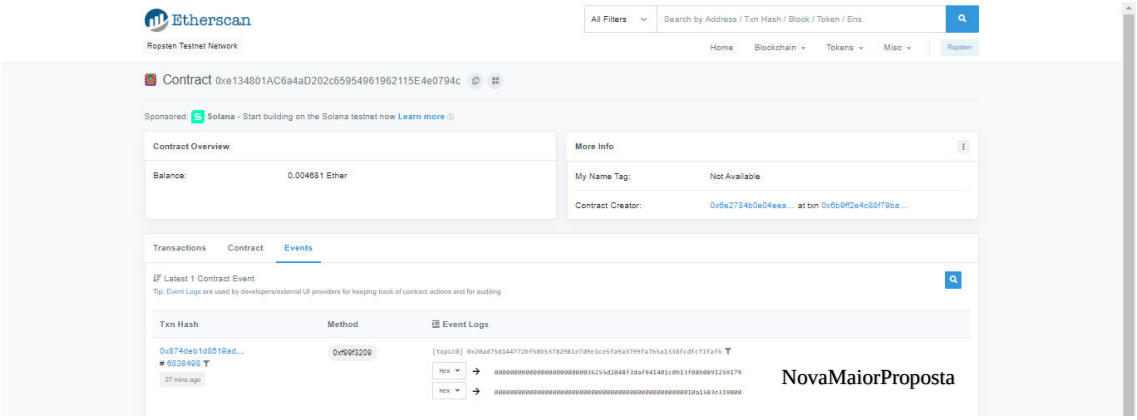
De forma similar, as Figuras 5.15 e 5.16 mostram as transações que ocorreram referentes à casa 2. Particularmente, a primeira oferta foi a vencedora e todas as demais foram imediatamente revertidas. Assim, apenas um evento de "NovaMaiorProposta" foi emitido.

Figura 5.15 – Transações ocorridas no contrato 2



Fonte: Própria autora.

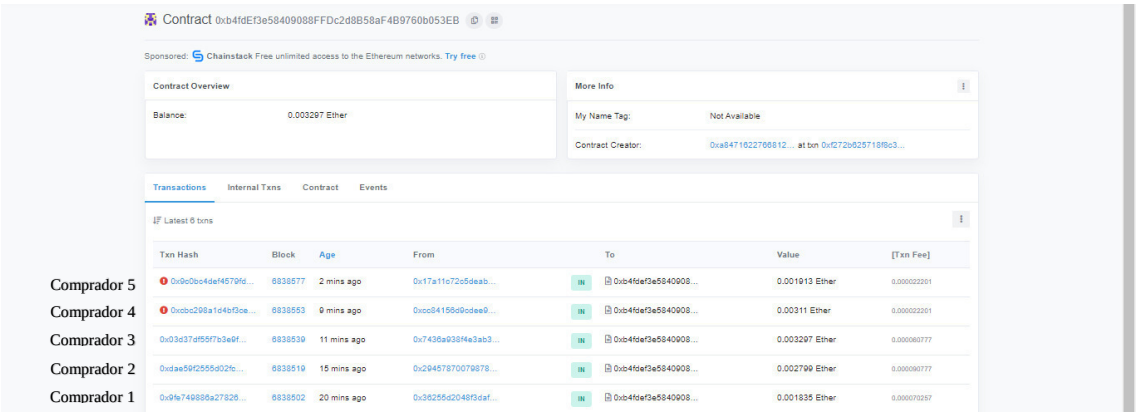
Figura 5.16 – Eventos disparados no contrato 2



Fonte: Própria autora.

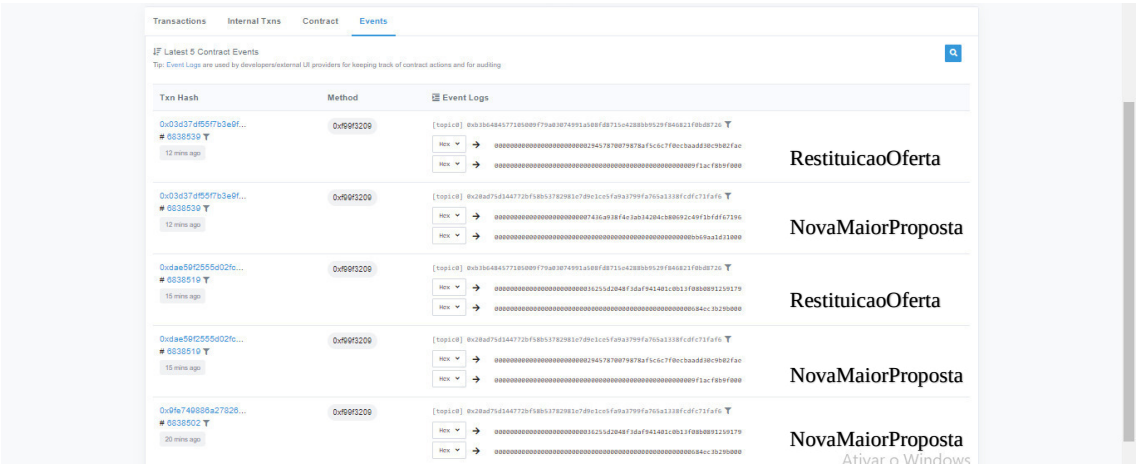
Na casa 3, na Figura 5.17 pode ser observado que houveram três transações bem-sucedidas, ou seja, as ofertas foram superadas três vezes seguidas. Deste modo, é possível ver na Figura 5.18 que cinco eventos foram disparados, três referentes à “NovaMaiorProposta” e dois que se referem as restituições realizadas para os compradores 1 e 2 (“RestituicaoOferta”).

Figura 5.17 – Transações ocorridas no contrato 3



Fonte: Própria autora.

Figura 5.18 – Eventos disparados no contrato 3

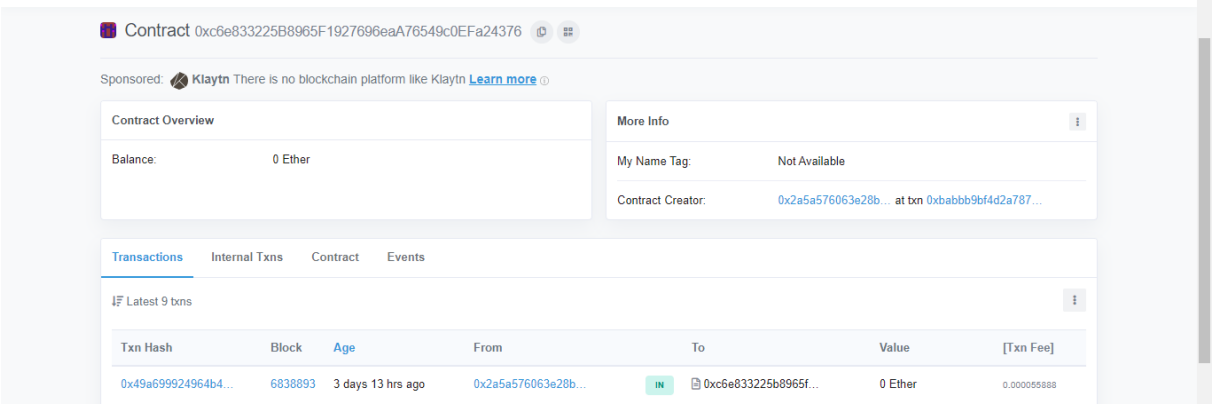


Fonte: Própria autora.

Ao fim do tempo de leilão, as casas disparam a função “Fim_Leilao” e então um novo evento é mostrado na página de acompanhamento do contrato, essa transação, por definição, não consome Ether.

Pode-se observar, após sua execução, que o contador “*Balance*” está zerado, decorrente da transferência de Ether para o beneficiário. Como simplificação, apenas a transação da casa 1 é mostrada na Figura 5.19.

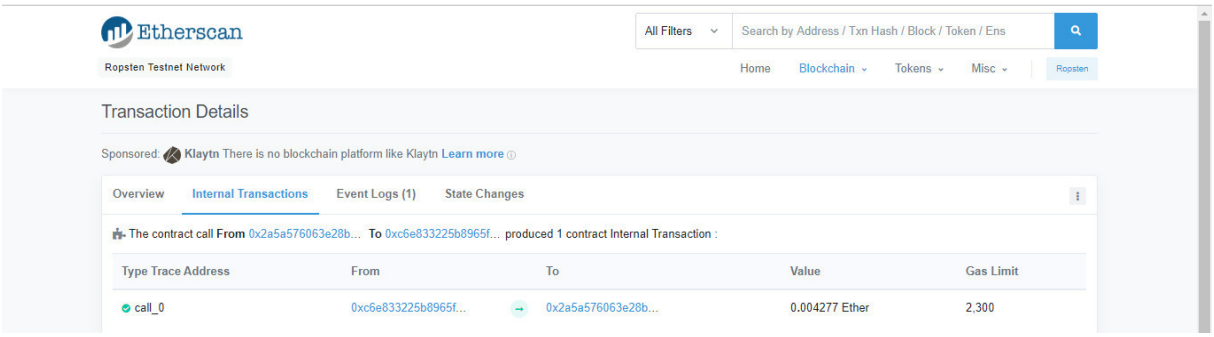
Figura 5.19 – Transação que dispara a função “Fim_Leilão”



Fonte: Própria autora.

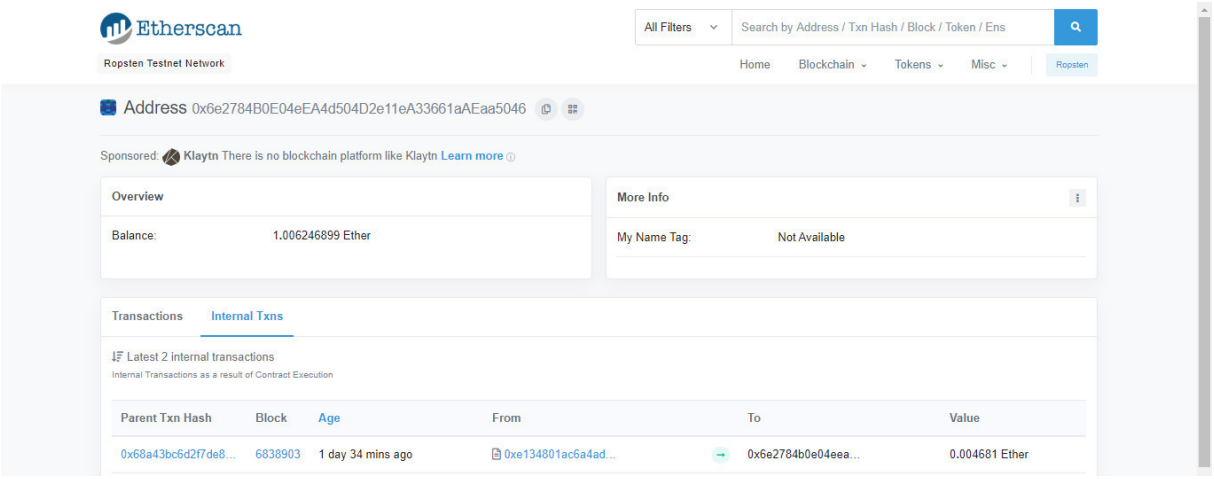
Desta forma, o valor da maior oferta é transferido para a casa de origem. As transações de recebimento, respectivamente, das casas 1, 2 e 3 podem ser vistas nas Figuras 5.20, 5.21 e 5.22. A transação de recebimento da oferta está disponível na aba de “Internal Transations”, isso indica que as transações ocorrem internamente e têm essa característica pois o endereço da casa é o criador do contrato.

Figura 5.20 – Transação da maior oferta para o beneficiário (casa 1)



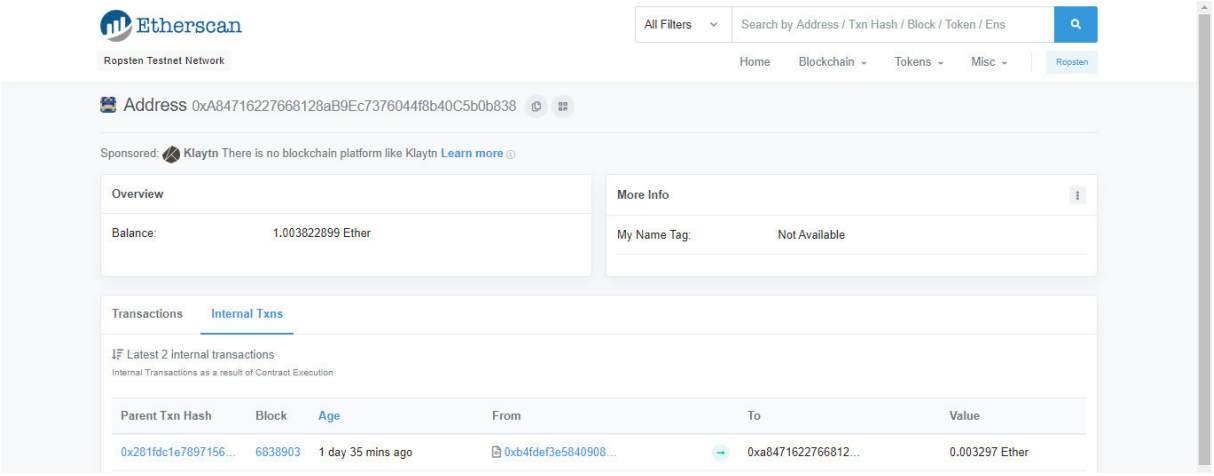
Fonte: Própria autora.

Figura 5.21 – Transação da maior oferta para o beneficiário (casa 2)



Fonte: Própria autora.

Figura 5.22 – Transação da maior oferta para o beneficiário (casa 3)



Fonte: Própria autora.

6 CONCLUSÕES E TRABALHOS FUTUROS

6.1 Conclusões

O mercado de energia passa por grandes transformações com a entrada de novos agentes. A geração distribuída vem crescendo e muito em breve, a forma como os consumidores se relacionam com a energia deverá se adaptar a uma rede mais autônoma, assim como os distribuidores de grande e pequena escala.

Contemporâneo a esse crescimento, a tecnologia do *Blockchain* surge como uma maneira de propor soluções descentralizadas, inicialmente surgindo com a criptomoeda Bitcoin, mas provando gradativamente seu valor como meio de estabelecer contratos confiáveis entre partes sem a necessidade de intermediários.

Este trabalho propões uma ferramenta que proporcione a troca de energia entre prosumidor e consumidor de forma simples e descentralizada. Assim, uma aplicação WEB é criada para que prosumidores tenham uma maneira de comercializar sua energia de maneira simplificada.

No modelo proposto, existem três casas produtoras de energia fotovoltaica e cinco potenciais compradores, que fizeram suas ofertas em cada uma delas para a compra da energia oferecida ao longo do dia. No fim do leilão, os compradores que propuseram uma maior oferta ganharam e a casa produtora recebeu o valor da proposta.

Através dos resultados monitorados das transações dos contratos e dos endereços dos beneficiários, é possível comprovar a lógica do contrato e a confiança que ele traz, proporcionando uma maneira transparente de negociação e um contrato imutável.

6.2 Trabalhos futuros

Para o aprimoramento do modelo proposto por esse trabalho, melhorias são citadas a seguir:

- A simulação da entrega da energia para o comprador que vencer o leilão através do uso de periféricos que simulem uma bateria e medição inteligente;
- A criação de contratos que sejam iniciados e renovados automaticamente;
- A criação de contratos que possam segmentar o envio da energia para mais de um vencedor, a depender da quantidade de energia produzida;

- A simulação de compradores para que as ofertas não precisem ser feitas de forma manual;
- A criação de elementos na interface WEB que apresentem mais informações relativas a produção de energia histórica das residências.

REFERÊNCIAS

7ª REGIÃO, T. Cartilha do Arrematante. 2010.

AHSAN, U.; BAIS, A. **Distributed big data management in smart grid**. IEEE, 2017

ANDONI, M. et al. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. **Renewable and Sustainable Energy Reviews**, v. 100, n. October 2018, p. 143–174, 2019.

ANDONI, M.; ROBU, V.; FLYNN, D. Crypto-control your own energy supply. **Nature**, v. 548, n. 7666, p. 158, 2017.

ANEEL. **Informações sobre Geração Distribuída no Brasil**. Disponível em: <https://www.aneel.gov.br/outorgas/geracao/-/asset_publisher/mJhnKIi7qcJG/content/registro-de-central-geradora-de-capacidade-reduzida/655808?inheritRedirect=false&redirect=http%3A%2F%2Fwww.aneel.gov.br%2Foutorgas%2Fgeracao%3Fp_p_id%3D101_INSTANCE_mJhnKIi7q>. Acesso em: 15 out. 2019.

ANEEL. **Geração Distribuída**. Disponível em: <https://www.aneel.gov.br/geracao-distribuida>. Acesso em: 18 nov. 2019.

AZIZ. **CRYPTO MAINNET VS TESTNET: WHAT IS THE DIFFERENCE?** Disponível em: <https://masterthecrypto.com/mainnet-vs-testnet-whats-the-difference/>. Acesso em: 10 nov. 2019.

BOOTSTRAP. **About - Bootstrap**. Disponível em: <https://getbootstrap.com/docs/4.3/about/overview/>. Acesso em: 12 nov. 2019.

BURGER, C. et al. Blockchain in the energy transition. A survey among decision-makers in the German energy industry. **German Energy Agency**, 2016a.

BURGER, C. et al. Blockchain in the energy transition. A survey among decision-makers in the German energy industry. **German Energy Agency**, p. 41, 2016b.

BUTERIN, V. Ethereum White Paper. 2013.

CANTO, D. D. **Blockchain: which use cases in the energy industry**. Blockchain: which use cases in the energy industry. **Anais**...Glasgow, Scotland: CIRED, 2017

COINMARKETCAP. **Historical data for bitcoin**. Disponível em: <https://coinmarketcap.com/currencies/bitcoin/historical-data/?Start=20161206&end=20171206>. Acesso em: 15 set. 2019.

DHILLON, V. et al. The DAO Hacked. In: **Blockchain Enabled Applications**. [s.l: s.n.].

DIFFIE, W.; HELLMAN, M. New directions in cryptography. **IEEE Transactions on Information Theory**, v. 22, n. 6, p. 644–654, 2016.

DWORK, C.; NAOR, M. **Pricing via processing or combatting junk mail**. 12th Annual International Cryptology Conference. **Anais**...1992

DWYER, G. P. The economics of Bitcoin and similar private digital currencies. **Journal of Financial Stability**, 2015.

EID, C. et al. Managing electric flexibility from Distributed Energy Resources: A review of incentives for market design. **Renewable and Sustainable Energy Reviews**, v. 64, p. 237–247, 2016.

ENEL. **Taxas, Tarifas e Impostos**. Disponível em: <https://www.enel.com.br/pt-ceara/Tarifas_Enel.html>. Acesso em: 15 nov. 2019.

ETHEREUM. **Solidity by Example**. Disponível em: <https://solidity-portuguese.readthedocs.io/pt/latest/solidity-by-example.html>. Acesso em: 1 ago. 2019.

ETHEREUM. **Solidity — Solidity 0.5.2 documentation**.

EURELETRIC. **Eurelectric launches expert discussion platform on blockchain**. Disponível em: <http://www.eurelectric.org/news/2017/eurelectric-launches-expert-discussionplatform-%0Aon-blockchain%0A>. Acesso em: 12 set. 2019.

FOXBIT. **O que é o Ethereum?** Disponível em: <https://foxbit.com.br/o-que-e-ethereum/>. Acesso em: 1 nov. 2019.

FRONI, A. A.; MEULEN, R. VAN DER. **Gartner Identifies Three Megatrends That Will Drive Digital Business Into the Next Decade**.

GONÇALVES, A. **O que é CSS? Guia Básico para Iniciantes**. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>. Acesso em: 12 nov. 2019.

GREGORY. **Intro to Web3.js**. Disponível em: <https://www.dappuniversity.com/articles/web3-js-intro>. Acesso em: 12 nov. 2019.

GREWAL-CARR, V.; STEPHEN, M. Blockchain Enigma. Paradox. Opportunity. **Deloitte**, 2016.

HENRIQUE, D. **O que é o Ethereum?** Disponível em: <https://acriptomoeda.com/guia-ethereum/>. Acesso em: 1 nov. 2019.

JARADAT, M. et al. **The internet of energy: Smart sensor networks and big data management for smart grid**. Procedia Computer Science. **Anais**...2015

LEE, J. A view of cloud computing. **International Journal of Networked and Distributed Computing**, v. 1, n. 1, p. 2–8, 2013.

MATOS, M. **Como funciona o Proof of Work na blockchain do Bitcoin**. Disponível em: <https://livecoins.com.br/proof-of-work-blockchain-bitcoin/>. Acesso em: 1 nov. 2019.

MATTILA, J. **The Blockchain Phenomenon – The Disruptive Potential of Distributed Consensus Architectures**. [s.l: s.n.].

MERCADOBITCOIN. **API de Dados**. Disponível em: <https://www.mercadobitcoin.com.br/api-doc/>. Acesso em: 12 out. 2019.

MUFTIC, S. **Overview and Analysis of the Concept and Applications of Virtual Currencies**. [s.l: s.n.].

NAKAMOTO, S. Bitcoin: A Peer-to-Peer Electronic Cash SyNakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Consulted, 1–9. doi:10.1007/s10838-008-9062-0stem. **Consulted**, 2008.

PACKAGE, E. U. **Energy Union Package - A Framework Strategy for a Resilient Energy Union with a Forward-Looking Climate Change PolicyCOM(2015) 80 final**. [s.l: s.n.].

PETER, V. Blockchain meets Energy. 2019.

PEYROTT, S. **An introduction to Ethereum and smart contracts: a programmable blockchain learn about verified. distributed computations in the cloud using Ethereum**. Disponível em: <https://auth0.com/blog/an-introduction-to-ethereum-and-smart-contracts-part-2/>. Acesso em: 1 set. 2019.

PINSON, P. et al. The Emergence of Consumer-centric Electricity Markets. **PierrerPinson**, p. 1–5, 2017.

PRADO, J. **O que é blockchain? [indo além do bitcoin]**. Disponível em: <https://tecnoblog.net/227293/como-funciona-blockchain-bitcoin/>. Acesso em: 1 set. 2019.

PVOUTPUT. **PVOutput**. Disponível em: <https://pvoutput.org/>. Acesso em: 30 ago. 2019.

RISTESKA STOJKOSKA, B. L.; TRIVODALIEV, K. V. **A review of Internet of Things for smart home: Challenges and solutionsJournal of Cleaner Production**, 2017.

SEPPÄLÄ, J. A. The role of trust in understanding the effects of blockchain on business models. **Working Paper**, 2016.

SWAN, M. **Blockchain: Blueprint for a New Economy**. [s.l: s.n.]. v. 2

SZABO, N. Smart Contracts: Building Blocks for Digital Free Markets. **Extropy Journal of Transhuman Thought**, 1996.

TAPSCOTT, D.; TAPSCOTT, A. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money ... **Sage Publications, Inc.**, 2016.

TAR, A. **Smart Contracts, Explained**. Disponível em: <https://cointelegraph.com/explained/smart-contracts-explained>. Acesso em: 20 out. 2019.

WALPORT, M. Distributed ledger technology: Beyond block chain. **Government Office for Science**, p. 1–88, 2015.

WOOD, G. Ethereum: a secure decentralised generalised transaction ledger. **Ethereum Project Yellow Paper**, 2014.

WOOD, G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. EIP-150 REVISION. **2017**, n. August 1, 2017, p. 33, 2017.

ZHOU, S.; BROWN, M. A. Smart meter deployment in Europe: A comparative case study on the impacts of national policy schemes. **Journal of Cleaner Production**, 2017.

APÊNDICE A – CONTRATO DO LEILÃO DESENVOLVIDO EM LINGUAGEM SOLIDITY

```
1  pragma solidity ^0.4.21;

2  contract leilao {
3  //Parâmetros iniciais do leilão
4  address public casa = msg.sender;
5  uint public ini_leilao = now;
6  uint public tempo_leilao= 15 hours;

7  uint public maior_oferta = 0;
8  address public maior_ofertante = 0;
9  uint public antiga_oferta = 0;
10 address public antigo_ofertante = 0;

11 //Eventos que serão disparados quando as funções forem realizadas
12 event NovaMaiorProposta(address ofertante , uint valor);
13 event RestituicaoOferta (address restituido, uint valor);
14 event FimDoLeilao(address vencedor, uint valor);

15 //Função de submissão da proposta
16 function proposta() payable public {
17       require (msg.value>0); //Valor pago deve ser superior a zero
18       require (msg.value> maior_oferta); //Valor pago deve ser superior a maior_oferta
19       require (now<ini_leilao+tempo_leilao); //Tempo de leilão não deve ter chegado ao fim

20 //Armazena o valor e o endereço da maior oferta que foi superada
21 antiga_oferta = maior_oferta;
22 antigo_ofertante = maior_ofertante;

23 //Armazena o valor e o endereço da nova maior oferta
24 maior_oferta = msg.value;
25 maior_ofertante = msg.sender;

26 //Dispara o evento de NovaMaiorProposta
27           emit NovaMaiorProposta(msg.sender, msg.value);
```

```
28  //Restitui o valor para o endereço que teve sua oferta superada
29  if (antiga_oferta>0) {
30          antigo_ofertante.transfer(antiga_oferta);
31          emit RestituicaoOferta(antigo_ofertante,antiga_oferta);
32      }
33  }
34  // Função que envia o valor da maior oferta após o fim do leilão para a casa
35  function fim_leilao() public {
36          require (msg.sender==casa); //Apenas o endereço da casa pode executar a função
37          require (now >= ini_leilao + tempo_leilao); //O tempo de leilão já deve ter chegado ao fim
38          casa.transfer(maior_oferta); // Transfere a maior oferta para a casa
39          emit FimDoLeilao(maior_ofertante, maior_oferta);
40      }
41  }
```