UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Vitor Albuquerque de Paula

Venda de Energia por Blockchain

São Carlos

2019

Vitor Albuquerque de Paula

Venda de Energia por Blockchain

Monografia apresentada ao Curso de Engenharia de Computação, da Escola de Engenharia de São Carlos e Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro de Computação.

Orientador: Prof. Dr. Carlos Dias Maciel

São Carlos

2019

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues
Fontes da EESC/USP

D419v

De Paula, Vitor Albuquerque
Venda de energia por *Blockchain* / Vitor Albuquerque De Paula; orientador Carlos Dias Maciel. -- São Carlos, 2019.

Monografia (Graduação em Engenharia de Computação) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2019.

1. *Blockchain*. 2. Energia. 3. IoT. 4. Internet das coisas. 5. Contratos inteligentes. 6. Rede elétrica. I. Titulo.

Elaborado por Elena Luzia Palloni Gonçalves – CRB 8/4464

FOLHA DE APROVAÇÃO

Nome: Vitor Albuquerque de Paula

Título: "Venda de energia por Blockchain"

Trabalho de Conclusão de Curso defendido em 29/11/2019.

Comissão Julgadora:	**Resultado:**
Prof. Associado Carlos Dias Maciel - SEL/EESC/USP - Orientador	APROVADO
Mestre Talysson Manoel de Oliveira Santos Doutorando - SEL/EESC/USP	Aprovado
Mestre Jordão Natal de Oliveira Júnior Doutorando - SEL/EESC/USP	Aprovado

Coordenador do Curso Interunidades Engenharia de Computação pela EESC/USP :

Prof. Dr. Maximiliam Luppe

Este trabalho é dedicado aos alunos da USP, como uma contribuição das Bibliotecas do Campus USP de São Carlos para o desenvolvimento e disseminação da pesquisa científica da Universidade.

AGRADECIMENTOS

Agradeço a meu Prof. Orientador, Carlos Maciel, por não apenas me orientar nesse trabalho, mas por me orientar durante a graduação. Gostaria de agradecer a toda minha família, pelo carinho e ajuda durante todos esses anos, em especial a minha mãe Andrea, que sempre me apoiou nos momentos difíceis. Sua força me ajudou a seguir em frente. Também quero deixar um espaço especial a meu avô Adelino, por ser meu melhor amigo e nunca medir esforços para que eu chegasse até esta etapa de minha vida. Gostaria de agradecer aos meus amigos, que compartilharam momentos comigo nessa etapa tão desafiadora da vida, vocês fizeram toda a diferença, especialmente meu amigo Ribeiro, que sempre me ajudou quando eu precisava.

Minha eterna gratidão. Esse trabalho é nosso!

"I read somewhere... how important it is in life not necessarily to be strong, but to feel strong... to measure yourself at least once."

"Eu li em algum lugar ... como na vida é importante não necessariamente ser forte, mas sentir-se forte... para se testar pelo menos uma vez."

Jon Krakauer

RESUMO

Paula, V.A. **Venda de Energia por Blockchain**. 2019. 64p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

Nesse trabalho foi desenvolvido um sistema de venda de energia por Blockchain, usando um contrato inteligente. O projeto tem cunho prático, e é funcional. Também será apresentada uma contextualização do atual mercado de energia, e como a tecnologia Blockchain pode mudá-lo. Em seguida a tecnologia Blockchain é apresentada, bem como seu contexto e tecnologias específicas utilizadas no desenvolvimento do trabalho, como o solidity. É proposto então, uma combinação de hardware que envolve o módulo ESP8266 e uma raspberry pi, para ligar uma rede elétrica real a uma rede Blockchain. Finalmente, um contrato inteligente para gerenciar a venda de energia é proposto.

Palavras-chave: Blockchain. Energia. IoT. Internet das Coisas. Contratos Inteligentes. Rede elétrica.

ABSTRACT

Paula, V.A. **Energy sale by Blockchain**. 2019. 64p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

In this work, a blockchain energy sales system will be developed using a smart contract. The project is practical and functional. The work starts presenting an overview of the current energy market context and how blockchain technology can change it. Then, the blockchain technology and other specific technologies used in the development of this work, such as solidity, are presented. Next, a hardware combination that involves the ESP8266 module and raspberry pi is proposed to connect a real power grid to a blockchain network. Finally, a smart contract to manage energy sales is proposed.

Keywords: Blockchain. Energy. IoT. Internet of Things. Smart Contracts. Power Grid.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

FER	Fonte de Energia Renovável
RED	Recurso Energético Distribuído
REI	Rede Elétrica Inteligente
IoT	Internet das Coisas; Internet of Things
P2P	Ponto a Ponto; Peer-to-Peer
WH	Watt.Hora
KWH	KiloWatt.Hora

SUMÁRIO

1 INTRODUÇÃO

1.1 Relevância do tópico e contextualização do problema

Poluição ambiental e crise energética são pautas que ganharam destaque na última década (GUAN et al., 2017) ao mesmo tempo que sistemas de energia estão passando por grandes mudanças, devido a pressão política por Fontes de Energia Renováveis (FER), iniciativas de políticas energéticas (SCLOVE, 1983) e avanço tecnológico. Redes elétricas inteligentes (REIs), ou Smart Grids, com acesso a recursos de energia renovável prometem ajudar a resolver esses problemas. Além das mudanças causadas pelos recursos energéticos distribuídos (REDs) e pelas FERs, os sistemas de energia estão prestes a se integrar com a era digital, como mostra a implantação em alta escala de medidores inteligentes em vários países (ZHOU; BROWN, 2017). Nas REIs, novos modelos, tecnologias e soluções flexíveis devem ser desenvolvidos para o gerenciamento ágil e inteligente de energia, de forma a facilitar estratégias ideais de fluxo de energia. É importante se atentar ao fato de que o mercado de energia atual tem problemas como alta centralização, baixa eficiência, alto custo KWH, problemas de privacidade (GUAN et al., 2019), entre outros, fazendo com que sua expansão fique presa a um gargalo. Esses motivos são catalizadores para uma mudança do mercado centralizado de energia para o mercado distribuído. No mercado tradicional de energia, os consumidores de energia só podem comprar eletricidade das concessionárias, enquanto nas REIs os seus membros desempenham um papel como consumidores e fornecedores de energia. A geração excessiva de energia renovável (oriunda de painéis fotovoltaicos ou turbinas eólicas) pode ser negociadas com a concessionária e com outros usuários com déficit no fornecimento de energia para benefícios mútuos.

A comunicação avançada e as trocas de dados entre diferentes partes da rede de energia são cada vez mais necessárias, tornando o gerenciamento e a operação central cada vez mais desafiadores. Técnicas de controle e gerenciamento distribuídos locais são necessárias para acomodar essas tendências de descentralização e digitalização (AHSAN; BAIS, 2018). Para isso é necessário considerar questões de eficiência, demanda, custo e emissão para evoluir o modelo de rede elétrica tradicional para um modelo de REI (FANG et al., 2011). A Blockchain é uma tecnologia emergente que visa descentralização, apresenta alto potencial disruptivo e tem sido amplamente adotada em diversas áreas, se encaixa perfeitamente a situação-problema aqui descrita (YANG et al., 2017). A Blockchain é uma tecnologia de registro distribuído, projetada para facilitar transações distribuídas ao retirar o gerenciamento central. Com isso, a Blockchain se torna um componente essencial para enfrentar os desafios enfrentados pelas REI.

As Blockchain são estruturas ou bancos de dados compartilhados e distribuídos que podem ser atualizados por qualquer participante da rede *peer-to-peer* (P2P), armazenar

transações digitais com segurança sem usar um ponto de autoridade central. Isso é feito baseado no consenso entre os participantes, e assegurados por algoritmos de prova, melhores descritos no capitulo 2, que tem a função de prevenir ataques maliciosos que comprometeriam a integridade da informação. Ainda mais importante no contexto de REIs, é a capacidade das Blockchain de execução automatizada de contratos inteligentes em redes P2P (KYPRIOTAKI; ZAMANI; GIAGLIS, 2015; SWAN, 2015). Hawk propõe em um modelo que aborda a questão da privacidade das transações, mostrando como sistemas de contratos inteligentes permitem o compilador gerar protocolos de criptografia eficientes (KOSBA et al., 2016). Uma visão didática da Blockchain, é imaginar um banco de dados que permite que vários usuários façam alterações simultaneamente, o que pode resultar em várias versões do banco de dados. Em vez de gerenciar o banco por um único órgão confiável, cada membro da rede mantém uma cópia da cadeia de registros e alterações e chegam a um acordo sobre o estado válido do banco de dados através de consenso. A metodologia exata de como o consenso é alcançado é uma área contínua de pesquisa e pode diferir para se adequar a uma ampla gama de domínios de aplicação. Novas transações estão vinculadas a transações anteriores por criptografia, o que torna as redes Blockchain resilientes e seguras. Todo usuário de rede pode verificar autonomamente se as transações são válidas, o que fornece transparência, registros confiáveis e à prova de fraudes (MOUGAYAR, 2018).

Criptomoedas foram os principais usos da tecnologia em seu início, e tiveram um aumento sem precedentes, se tornando pauta de políticas a nível nacional, como o caso do Japão que em 2017 reconheceu o Bitcoin como meio de pagamento legal e o caso da China que proibiu transações em Bitcoin. (BAJPAI, 2019; UOL, 2018). Os casos de uso da tecnologia Blockchain são vastos e incluem economia, serviços sociais públicos, diplomas digitais, documentos, sistema de reputação, votações, entre outros e é particularmente interessante no nicho de IoT, onde se pode criar soluções em que não se depende de confiança e facilitar transações maquina-maquina, permitindo o compartilhamento de serviços e recursos e automatizar vários fluxos de trabalho demorados existentes com a autenticação criptográfica (ZHENG et al., 2018; REYNA et al., 2018; CHRISTIDIS; DEVETSIKIOTIS, 2016). A tecnologia Blockchain também possa ser empregada para um sistema REI descentralizado, confiável e seguro, como mostra (ZHIYI; SHAHIDEHPOUR; XUAN, 2018).

A possibilidade de aplicações é tão grande que diversos autores colocam a Blockchain como uma das grandes inovações tecnológicas da humanidade. O potencial está no fato de que as tecnologias Blockchain podem redefinir a confiança digital e remover intermediários. A natureza disruptiva reside no potencial de substituir o controle centralizado por consenso e também nas filosofias inerentes ao consenso distribuído, código aberto, transparência e de tomada de decisões com base na comunidade.

A integração da tecnologia Blockchain no comércio de energia é uma área nova e promissora de pesquisas, e muitos estudos tem sido realizados (PARK; YONG, 2017).

REIs apoiadas por Blockchain podem fornecer sinais de resposta de demanda de alta precisão, reduzir custos e melhorar a velocidade (POP et al., 2018). Além disso resolvem o problema de escalabilidade e mobilidade da rede elétrica criando um mercado competitivo de energia nunca visto antes, aprimorando o papel dos consumidores no sistema e suas escolhas. Seu uso permite aos participantes vender o excedente energético de suas próprias FERs (como células solares) ou escolher sua fonte energética ideal baseada em estratégias pré-definidas, como preço, impacto ambiental, e combinações das diversas variáveis envolvidas (HWANG et al., 2017). Outros benefícios são a alta capacidade de computação com baixo custo, mecanismos de consenso garantindo a solução ideal, impedimento de transações falsas e estabelecimento de um sistema de crédito aberto e transparente . A aplicação da criptografia no Blockchain pode resolver vários problemas de segurança do sistema (RAHMADIKA; RAMDANIA; HARIKA, 2018).

Os primeiros desenvolvedores de Blockchain já começaram a criar plataformas digitais de trocas que podem ser completamente descentralizadas e podem permitir o comércio de energia P2P (KHORASANY; MISHRA; LEDWICH, 2018).

1.2 Proposta

Com a situação descrita em mente, esse trabalho de conclusão de curso se propõe a estudar a questão. Iremos modificar um dispositivo medidor de energia para que esse possa se conectar a IoT e estabelecer uma interface simples de venda de energia. A venda de energia será feita por Blockchain, através da rede ethereum, e gerenciada por um contrato inteligente, provendo todos os benefícios já mencionados da tecnologia Blockchain. O dispositivo só conduz energia caso o contrato permita, em outras palavras, o dispositivo irá operar até ter sido consumido o total de energia comprada.

2 BLOCKCHAIN

Essa sessão irá descrever como a tecnologia Blockchain funciona e quais os seus principais elementos.

2.1 Definições gerais

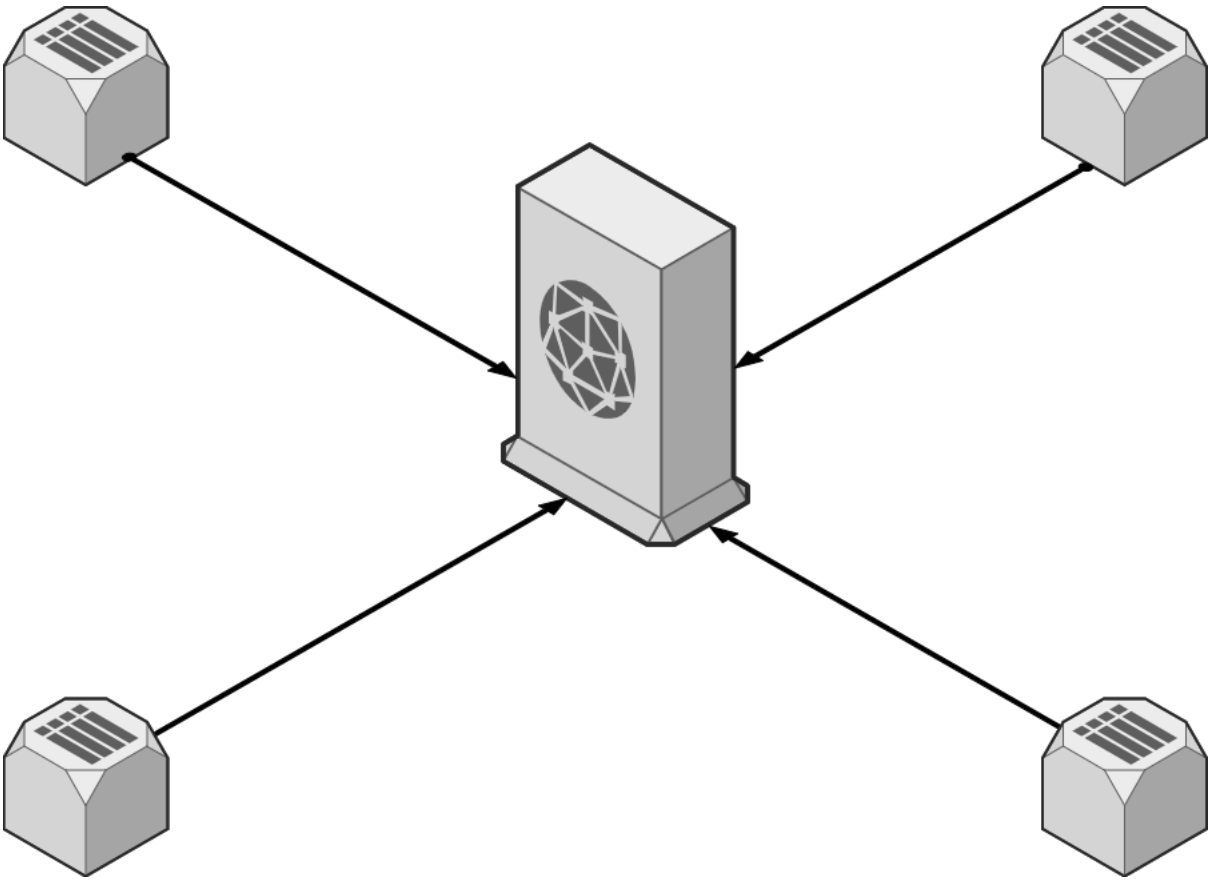
É difícil classificar o Blockchain em uma única definição. O Blockchain pode ser lido e apresentado a partir de diferentes pontos de observação e de diferentes perspectivas. A seguir, é apresentada uma primeira definição relacionada à família de tecnologias na qual essa tecnologia é colocada.

A Blockchain é uma subfamília de tecnologias na qual o registro é estruturado como uma ¨cadeia de blocos¨, uma estrutura de dados digitais, banco de dados compartilhado e distribuído contendo um histórico de todas as transações, em expansão contínua e em ordem cronológica, cuja validação é confiada a um mecanismo de consenso distribuído em todos os pontos da rede (sistemas públicos) ou em todos pontos que estão autorizados a participar do processo de validação de transações a serem incluídas no registro no caso de Blockchains permitidos ou privados. Essa estrutura de dados, em outras palavras, é análoga a um livro de registros que pode conter transações digitais, registros de informação e programas executáveis. Essas transações são organizadas em estruturas maiores, chamadas blocos, criptograficamente carimbadas, com data e hora, de forma a determinar a ordem de sequência dos eventos em uma cadeia de blocos, motivo do nome ¨Blockchain¨.

As principais características das tecnologias Blockchain são a imutabilidade do registro, a transparência, a rastreabilidade das transações e a segurança com base em técnicas criptográficas. Um exemplo simples de uma transação é copiar dados de um local para outro, como no caso das criptomoedas, onde copiamos moedas da carteira de um usuário para outra. O problema se encontra no fato de que o sistema tem que garantir que a moeda seja gasta apenas uma vez, sem haver gasto duplo. Tradicionalmente, iriamos usar uma entidade central com autoridade máxima para fazer esse intermédio; essa solução necessita que ambos os lados confiem no intermediário, além de apresentar grandes desvantagens em caso de falhas, sendo assim vulneráveis a ataques maliciosos (ARMBRUST et al., 2015).

O Blockchain é baseado em uma rede e permite gerenciar um banco de dados de maneira distribuída. Sob o ponto de vista operacional, é uma alternativa as entidades e arquivos centralizados permitindo o gerenciamento de atualizações de dados com a colaboração dos participantes da rede, onde todos os membros podem ter e acessar esses dados, que são distribuídos entre todos os participantes. A figura 1 ilustra um sistema

Figura 1: Sistema de Transação Centralizado

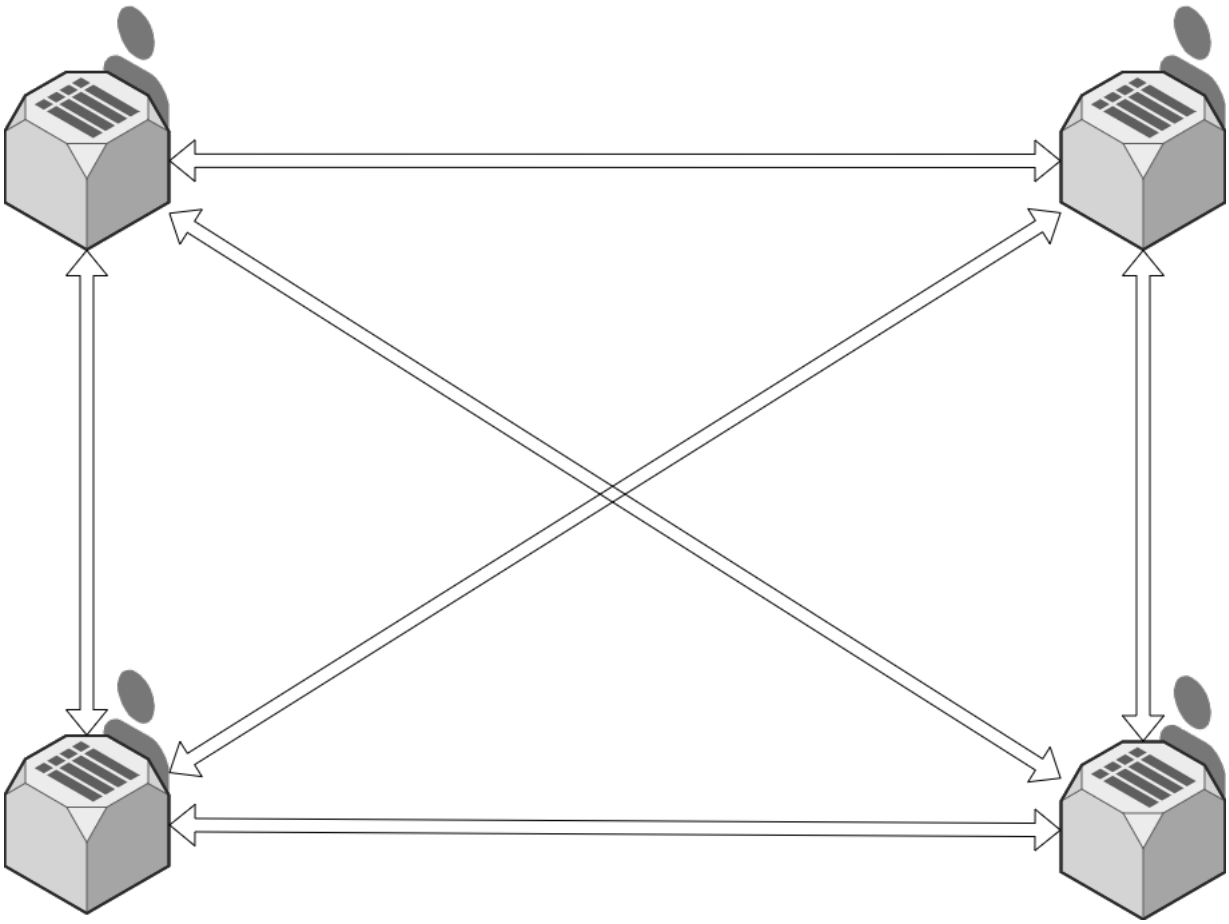


de transação centralizado. De fato, é valido enfatizar: o objetivo principal das tecnologias Blockchain é remover a necessidade desses intermediários e substituí-los por uma rede distribuída de usuários digitais que trabalham em parceria para verificar transações e garantir a integridade dos registros, permitindo um alto nível de transparência. Se o gerenciamento central for removido, o desafio reside em encontrar uma maneira eficiente de consolidar e sincronizar as várias cópias dos registros. O processo exato de validação e consolidação dos registros varia para diferentes tipos de Blockchains; no entanto, em princípio, os membros da rede comparam suas versões dos registros através de um processo bastante semelhante ao voto distribuído, onde é possível chegar a um consenso. (MATTILA, 2016). Isso torna os registros em um sistema Blockchain difíceis de serem adulterados sem a conivência da maior parte da rede, fazendo os sistemas Blockchain robustos e seguros. A figura 2 ilustra um sistema de transação distribuído.

2.2 Protocolo

Vamos ver concretamente em que consiste a Blockchain. O Blockchain é um protocolo de comunicação, que identifica uma tecnologia baseada na lógica do banco de dados distribuído (um banco de dados no qual os dados não são armazenados em um único computador, mas em várias máquinas conectadas entre si, chamadas nós).

Figura 2: Sistema de Transação Distribuído



O Blockchain é uma série de blocos que armazenam um conjunto de transações validadas e correlacionadas por um Marcador de Tempo (Timestamp). Cada bloco inclui o hash (uma função algorítmica do computador não invertível que mapeia uma cadeia de comprimento arbitrário em uma cadeia de comprimento predefinida) que identifica exclusivamente o bloco e permite a conexão com o bloco anterior, identificando o bloco anterior. Os componentes básicos da Blockchain:

Nós são os participantes da Blockchain e são fisicamente constituídos pelos servidores de cada participante da rede.

Transação são os dados que representam os valores de "troca"e que precisam ser verificados, aprovados e depois arquivados.

Bloco é representado pelo agrupamento de um conjunto de transações que são combinadas para serem verificadas, aprovadas e depois arquivadas pelos participantes da Blockchain.

Registro é o registro público no qual todas as transações realizadas de maneira ordenada e sequencial são "anotadas"com a máxima transparência e de maneira imutável.

Figura 3: Transação

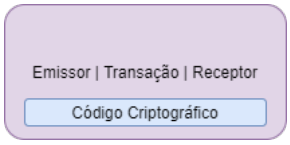
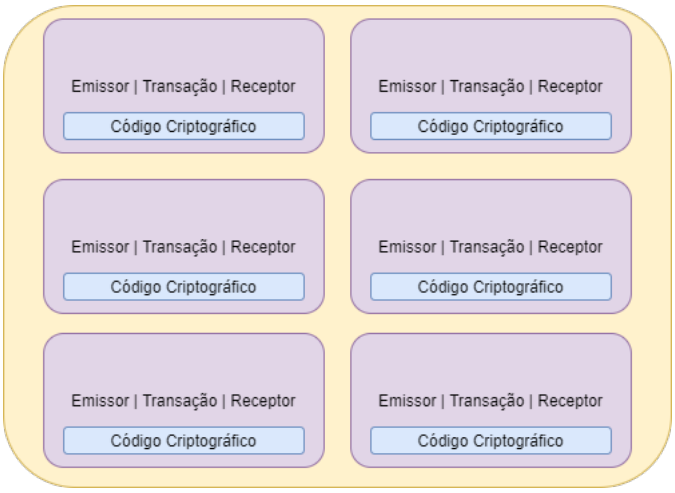


Figura 4: Bloco de transações



Consiste no conjunto de blocos que são encadeados entre si por uma função de criptografia e graças ao uso de hash.

Hash é uma operação não reversível que permite mapear um texto e / ou sequência numérica de comprimento variável em uma sequência única e exclusiva de um determinado comprimento. O Hash identifica de forma exclusiva e segura cada bloco. Um hash não pode ser usado para rastrear o texto que o gerou. Podemos enxergar o hash como uma assinatura digital do bloco.

2.3 Funcionamento

Para um fácil entendimento do funcionamento do Blockchain, serão mostradas passo a passo as etapas que definem o funcionamento de uma transação baseada em Blockchain. O objetivo é gerenciar uma transação, uma troca ou a venda de um ativo. Iremos usar um exemplo aplicado ao tema do trabalho: Duas pessoas desejam fazer uma transação: João vende 1000 KWH de energia excedente para Maria. Surge a necessidade de gerenciar uma transação comercial entre duas partes. Uma transação é criada, consistindo em uma série de elementos, como endereço público do destinatário, quantidade de energia, chaves criptográficas e qualquer outra informação necessária para concluir a estrutura de venda e compra de energia.

As chaves criptográficas de João e Maria são criadas em preparação para a transação. A transação começa com a assinatura digital e a chave pública de cada participante. A figura 3 representa como uma transação é constituída.

A transação entre João e Maria se torna parte de um bloco que será verificado e "resolvido"pelos participantes do Blockchain. Isso cria um novo bloco como mostra a figura 4 com todos os dados relacionados da transação entre João e Maria, dados relacionados com a venda e os novos saldos de João e Maria. O bloco, que também inclui outras transações, agora está preparado para ser submetido à verificação e aprovação dos participantes do Blockchain. A transação é trazida para a rede para ser verificada pelos participantes da Blockchain. Se as informações forem consideradas corretas, a transação é autorizada, validada e realizada. Assim que bloco que inclui a transação entre João e Maria, juntamente com outras transações, é verificado e aprovado pela rede Blockchain o bloco é adicionado à cadeia de blocos criptograficamente encadeadas, que formam o Blockchain. A transação é concluída e é armazenada em todos os nós da Blockchain. Essa cadeia é acessível a todos os participantes, e ninguém pode alterá-la. Temos então uma referência permanente, imutável e transparente dessa transação específica.

2.4 Banco de dados distribuído

Vamos entender melhor o que é um banco de dados distribuído. Nos encontramos diante de um banco de dados que não é encontrado fisicamente apenas em um servidor, mas em vários computadores ao mesmo tempo, todos perfeitamente sincronizados com os mesmos arquivos. Dessa maneira, as informações podem ser encontradas muito rapidamente, pois o poder da computação explora a capacidade conjunta de todos os computadores conectados.

Existem basicamente dois processos que permitem que os bancos de dados distribuídos funcionem corretamente e não percam dados no caminho. Esses processos são:

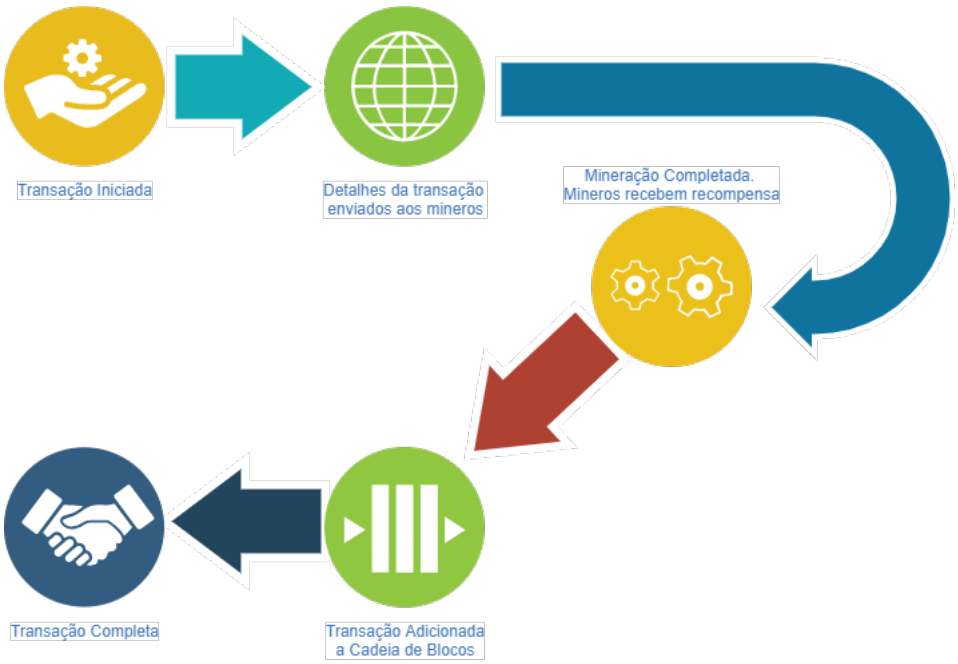
Replicação de dados - um software é responsável por analisar o banco de dados para identificar alterações. Depois que essas alterações são identificadas, o software garante que essas alterações sejam replicadas e que todos os bancos de dados sejam idênticos.

Duplicação - processo que garante que todos os bancos de dados tenham os mesmos dados. Na prática, identifica um banco de dados mestre, que é duplicado em todos os outros bancos de dados, para torná-los iguais. Os usuários podem modificar apenas o banco de dados mestre, garantindo que os dados locais não sejam substituídos por engano.

2.5 Mineração

Para que um novo bloco de transações seja adicionado ao Blockchain é necessário que ele seja verificado, validado e criptografado. Somente com esta etapa ele poderá se tornar ativo e ser adicionado ao Blockchain. Para executar esta etapa é necessário que toda vez que um bloco seja composto, seja resolvido um problema matemático complexo, que

Figura 5: Etapas de uma transação



exija um comprometimento considerável também em termos de potência e capacidade de processamento. Esta operação é definida como "Mineração"e é realizada pelos "Mineiros".

O trabalho dos mineiros é absolutamente fundamental na economia do gerenciamento de Blockchain. Qualquer um pode se tornar um "Mineiro"e pode competir para ser o primeiro a resolver o complexo problema matemático relacionado à criação de cada novo bloco de transação de uma maneira válida e criptografada que pode ser adicionada ao Blockchain. Como esse é um compromisso importante, como mencionado com considerável desperdício de energia, a mineração de criptomoedas é um compromisso que precisa ser remunerado e incentivado. Em Blockchain "Privado"ou com permissão este papel é desempenhado, como uma função de autoridade de governança que ativa o mesmo Blockchain. Nas Blockchains públicas ou sem permissão, esse papel pode ser desempenhado por qualquer participante da Blockchain e o Mineiro é incentivado com formas de remuneração que dependem do tipo de regras ou governança definida por cada Blockchain.

Na maioria dos casos, o primeiro mineiro que cria um bloco válido e o adiciona à cadeia é recompensado com a soma das comissões por suas transações. As comissões referem-se a valores unitários para cada transação individual, mas os blocos são adicionados regularmente e podem conter milhares de transações; portanto, o valor da mineração também pode ser muito alto. Os mineradores também podem receber novas moedas criadas e colocadas em circulação como um mecanismo de inflação, como no caso do

Bitcoin Blockchain. Uma análise mais completa pode ser encontrada em (MUFTIC et al., 2016).

2.6 Imutabilidade

A imutabilidade é o outro grande vantagem da Blockchain que obviamente também se relaciona à segurança dos dados. Em um sistema com uma autoridade central, quando essa fosse violada seria possível alterar, danificar ou destruir os dados armazenados, enquanto no caso da Blockchain isso é virtualmente impossível, pois seria necessário violar cópias dos registros dos participantes da rede Blockchain em quantidade suficiente para vencer o algoritmo de consenso e seria necessário fazer isso com perfeita sincronia, algo que pode-se dizer impossível devido ao tamanho da Blockchain, embora este deva ser avaliado em termos de participantes ou nós. Aqui chegamos ao conceito de Confiança e Controle. A confiança e o controle das transações passam da autoridade central para todos os participantes. As transações baseadas em Blockchain não são centralizadas, ocultas ou fechadas, mas sim descentralizadas, transparentes e abertas a todos.

3 CONTRATOS INTELIGENTES

3.1 Bitcoin

A primeira criptomoeda do mundo, Bitcoin, foi criada em 2009, após o lançamento público de um artigo de Nakamoto (NAKAMOTO et al., 2008), um autor cuja identidade real permanece desconhecida. Ele propôs um sistema de pagamento eletrônico em dinheiro distribuído que usa comunicação P2P de usuários anônimos e desconhecidos da Internet. O dinheiro digital trocado entre usuários não é emitido ou controlado por um banco central, mas por uma rede de computadores que operam em colaboração e usam criptografia para garantir a segurança.

Cada usuário no sistema Bitcoin possui uma carteira digital, onde as moedas são armazenadas, uma chave privada e uma chave pública. A carteira só pode ser acessada pela chave privada secreta do usuário. O endereço da carteira ou o endereço Bitcoin é derivado da chave pública de um usuário e é usado para identificar um usuário, oferecendo pseudônimo. Da inicialização à finalização, uma transação de Bitcoin segue o procedimento geral de transações Blockchain descrito no capítulo 2.

3.2 Ethereum

Embora o Bitcoin represente o maior e mais estabelecido aplicativo de Blockchain até o momento, o Ethereum dominou os aplicativos de Blockchain ao lado das criptomoedas. O Ethereum é uma máquina virtual inovadora baseada em Blockchain e a plataforma Cloud 2.0 que vem com uma linguagem de programação incorporada que permite aos usuários criar seus próprios aplicativos que são executados em cima de arquiteturas Blockchain. O Ethereum permite contratos inteligentes criados pelo usuário e tem como objetivo construir uma plataforma de tecnologia para todos os fins, na qual conceitos de aplicativos baseados em transações possam ser construídos. Conforme descrito em seu site oficial (ETHEREUM, 2019) o Ethereum "permite que os desenvolvedores criem o Marketplace, armazenem registros, movimentem fundos de acordo com instruções específicas (como no caso de contratos de vontade ou futuros) e muitas outras coisas que ainda não foram inventadas, tudo sem intermediários".

3.3 Definição de Contratos Inteligentes

Um Contrato Inteligente ou Smart Contract é a tradução no código de um contrato que pode verificar automaticamente o cumprimento de certas condições (verificação de dados básicos do contrato) e executar ações automaticamente (ou permitir que certas ações possam ser executadas) quando as condições específicas entre as partes são atingidas

e verificadas (CHRISTIDIS; DEVETSIKIOTIS, 2016). Em outras palavras, o Contrato Inteligente é baseado em um código que "lê"as cláusulas e sanções que foram acordadas e as condições operacionais nas quais as condições acordadas devem ocorrer da mesma maneira que um contrato tradicional, mas podem executar automaticamente certos eventos quando os dados referentes às situações reais se aplicam aos dados das condições e cláusulas acordadas.

Para simplificar, o contrato inteligente precisa de suporte legal para a elaboração, mas não para verificação e ativação.

Através do Smart Contract, é possível trocar dinheiro, bens, ações ou qualquer coisa de valor de forma transparente e sem conflitos, evitando os serviços de um intermediário, pois é o conjunto de participantes da Blockchain que valida o contrato, portanto, não há necessidade de intermediários.

Por serem baseados em Blockchain, os contratos inteligentes herdam algumas características interessantes como:

- Imutabilidade
- Segurança
- Nenhuma parte pode alterar os termos do contrato por si só.
- Todas as ações resultantes do contrato inteligente são automáticas e ocorrem sem intermediários.
- Todas as transações são registradas na Blockchain e são irreversíveis.
- Quando as condições predefinidas não são atendidas, as transações não ocorrem.

3.4 Solidity

Solidity é uma linguagem de programação relativamente nova lançada no Ethereum em 2015 e foi projetada para desenvolvedores de Blockchain. A programação em solidity permite que os desenvolvedores escrevam e implementem contratos inteligentes no Blockchain Ethereum.

É executada na máquina virtual Ethereum (EVM), que permite o desenvolvimento de sistemas de contrato inteligentes. A linguagem de codificação Ethereum permite que um contrato interaja com outros contratos e atualize os termos quando necessário. A programação faz com que os contratos sejam executados literalmente, de forma que qualquer ambiguidade possa bloquear transações. Com contratos tradicionais, as partes podem trabalhar juntas e chegar a um acordo. Mas isso não acontece em uma Blockchain onde as transações são irreversíveis. Como os contratos inteligentes são auto executáveis, os

programadores devem prestar atenção especial aos dados inseridos no código (DANNEN, 2017).

Apesar de nova, Solidity é a linguagem de programação de contratos inteligentes com mais recursos e bibliotecas disponíveis no momento da escrita desse trabalho, e por tal razão foi escolhida para programar os contratos inteligentes nesse trabalho.

3.5 Metamask

Com o termo Metamask, nos referimos a uma carteira Ethereum, que possui as faculdades certas para oferecer suporte a Ether e até outros tokens através do sistema de tecnologia ERC20 . Ao contrário da maioria das outras carteiras, isso pode ser feito simplesmente baixando uma extensão do Chrome, ou de seu navegador de preferência. Possui as mesmas funções, recursos e facilidade de acesso que as carteiras normais da Ethereum e além disso, permite que interação com aplicativos distribuídos e Contratos Inteligentes sem precisar baixar o Blockchain inteiro ou instalar qualquer software secundário. O plugin Metamask também pensa em desenvolvedores e permite fácil conexão às principais redes de teste Ethereum, incluindo Ropsten, Kovan e Rinkedy.

4 AMBIENTE

4.1 Desenvolvimento

Para o desenvolvimento, optou-se pelo sistema operacional Linux, distribuição Ubuntu, versão 18.04 LTS, pois apresentava total compatibilidade com as suítes, ferramentas e bibliotecas utilizadas. Primeiramente iniciou-se o desenvolvimento no Windows 10, porém muitos problemas de compatibilidade com bibliotecas e suítes tornavam o desenvolvimento mais complicado, em especial a biblioteca web3.js. Assim, o desenvolvimento foi migrado para o Ubuntu 18.04 que se mostrou mais compatível e com menos configurações necessárias para desenvolvimento e testes.

4.2 ESP8266

O módulo ESP8266 Wireless, baseado em um SoC MCU de 32 bits foi criado para adicionar conectividade sem fio a sistemas eletrônicos e projetar sistemas de IoT de baixo custo. Este módulo possui uma grande capacidade de processamento e graças ela é possível conectar vários periféricos, incluindo sensores e outros dispositivos compatíveis, através de seus pinos GPIO. Além disso, seu pequeno tamanho e a quantidade de documentação disponível on-line tornam o seu uso mais fácil e perfeito para ser usado na construção de dispositivos compactos. Além de poder ser programado como uma placa de desenvolvimento real, pode ser usado como um adaptador serial sem fio com uma conexão SPI/SDIO. Dada a possibilidade de personalizar tanto a parte referente ao software quanto ao hardware, é um módulo com muitas utilidades e usos, tornando-o excelente para aplicações em rede graças à sua alta eficiência energética, com a possibilidade de definir três modos: modo ativo, modo de sono e sono profundo. Quando o módulo minimiza suas funções desativando as não essenciais, ele consome pouco menos de 12uA.

4.2.1 Configuração

Para medição de energia elétrica, foi utilizado um dispositivo chamado Sonoff Pow R2, que nada mais é que um chip ESP8266 com alguns sensores já soldados. Esses sensores provém uma serie de dados sobre a rede a qual ele está conectado, como tensão da rede, corrente e energia gasta em um determinado período. O firmware padrão que vem com o dispositivo não é conveniente para programação, então usaremos um firmware chamado TASMOTA, feito pela ITEAD (SONOFF, 2019), que nos fornece 3 interfaces de comunicação com o dispositivo:

- Serial

- Chamadas HTTP
- MQTT

Para troca do firmware foi feito procedimento com auxílio de um chip conversor USB-SERIAL FTD1232 , conectado a interface GPIO do dispositivo. (MAT, 2019) descreve o processo de forma muito didática.

4.3 Raspberry Pi

A raspberry pi utilizada foi o modelo 3b+; configurada para funcionar como uma ponte, que pode prover comunicação entre com a rede Blockchain e com o modulo 8266. Ela rodava um Node.Js e com o auxílio da robusta biblioteca web3.js (WEB3..., 2019) se comunicava com a rede de testes Rinkeby (RINKEBY..., 2019), que funciona como a rede principal. Assim, dada as condições auto executáveis do contrato, como desligar o fornecimento de energia caso a energia consumida fosse superior a energia comprada, a raspberry comunicava o status ao Modulo ESP8266 e desliga o fornecimento energético.

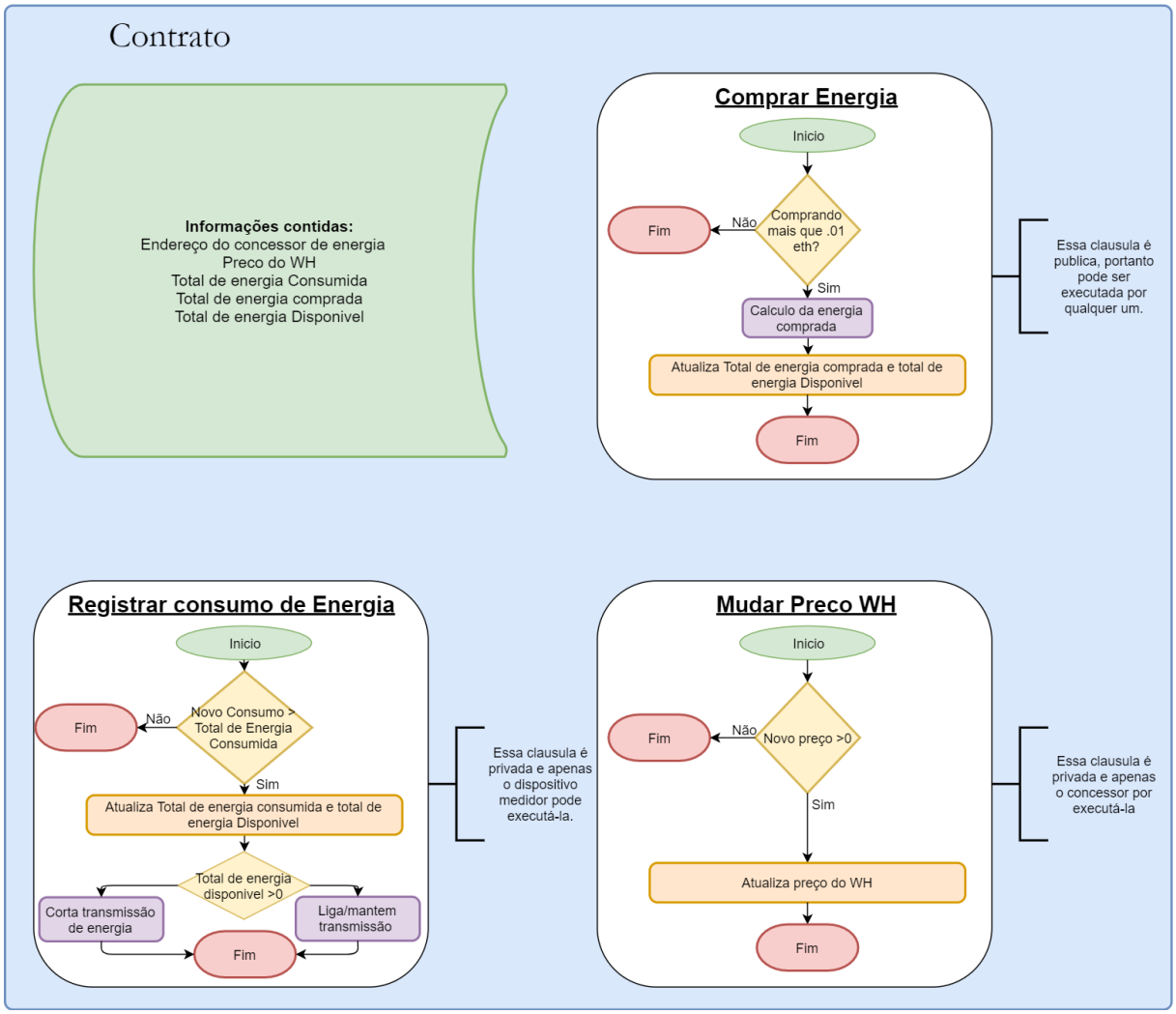
4.4 Interface web

Foi criada uma interface web (figura 7) com o intuito de auxiliar o usuário na compra de energia. A interface apresenta informações atualizadas do contrato, e pode ser usada para comprar energia. O processo é descrito no capíttulo 5. A interface Web foi feita em React Native também era auxiliada pela biblioteca web3.js. A parte principal do código da interface é apresentada no apêndice C.

5 CONTRATO DE VENDA DE ENERGIA

Aqui falaremos um pouco sobre o contrato inteligente de venda de energia criado durante este trabalho. O código do contrato pode ser encontrado no Apêndice A. O contrato foi escrito em solidity. Os campos do banco de dados (do contrato) são: o endereço do concessor de energia, o preço do WH, o total de energia comprada, total de energia consumida, total de energia disponível e uma variável booleana que indica que o balanço está positivo e o sistema pode transmitir energia. A figura 6 apresenta o modelo de contrato simplificado. Podemos encarar o contrato como um acordo onde determinadas informações são acessíveis a todos e com cláusulas que tem suas próprias regras. Na figura 6, as informações estão representadas no quadro verde, e as cláusulas estão representadas nos quadros brancos. Falaremos melhor delas a seguir.

Figura 6: Contrato Simplificado



A ideia, é que a energia seja pré-paga. Assim, o consumidor que deseja comprar energia de um concessor pode verificar o custo do WH no momento da compra, e optar por quanto de energia comprar. Podemos observar que essa informação está no quadro verde da figura 6, sendo acessível a todos participantes da rede Blockchain. Uma das formas mais fáceis para o usuário realizar essa função é usar a interface web criada para esse contrato, apresentada na figura 7. A interface mostra informações referentes ao contrato em tempo real, com valores lidos diretamente da Blockchain. Para comprar a energia pela interface web, o usuário deve acessa-la, colocar o quanto deseja gastar e clicar em comprar. Uma janela do Metamask irá surgir, confirmando a transação; a figura 8 mostra a etapa. Quando o usuário clicar em comprar, o Metamask irá se comunicar com o contrato, enviando o valor inserido pelo usuário. O contrato irá processar a solicitação na rede Blockchain conforme descrito no capítulo 2, então as informações serão atualizadas e uma mensagem de sucesso na compra será apresentada caso todas as cláusulas do contrato sejam cumpridas (figura 9).

Figura 7: Interface web de compra de Energia

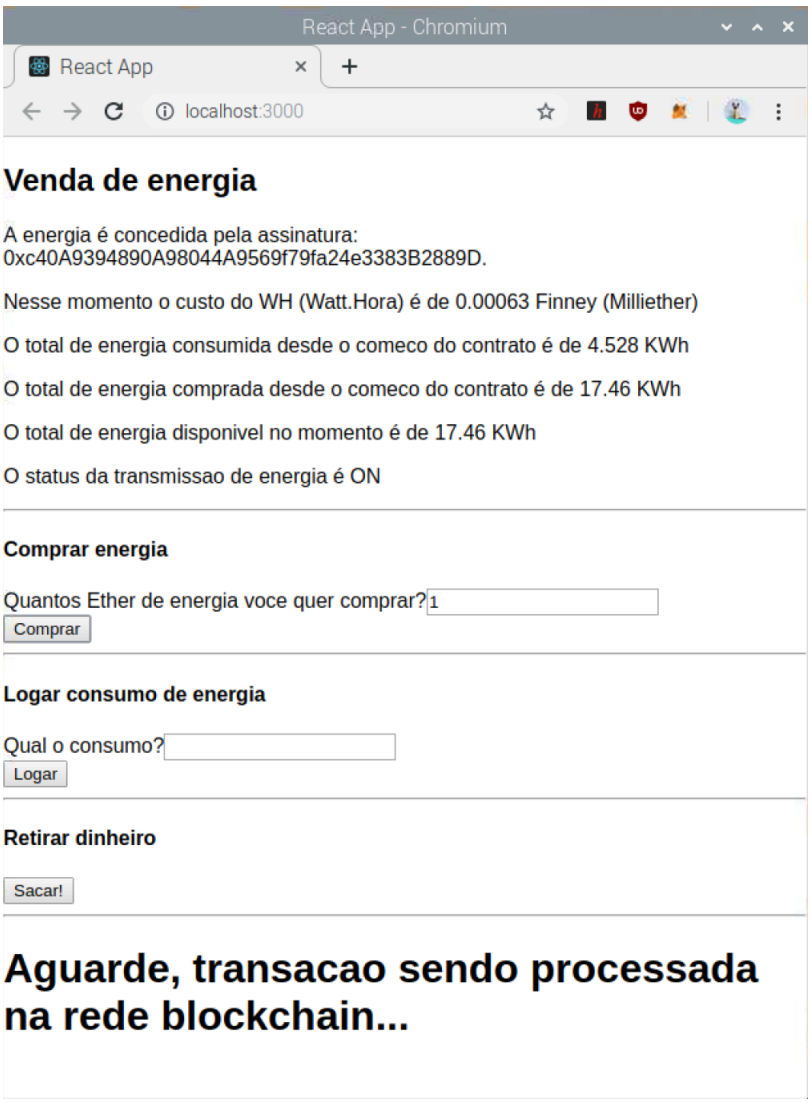


Figura 8: Confirmação de transação Metamask

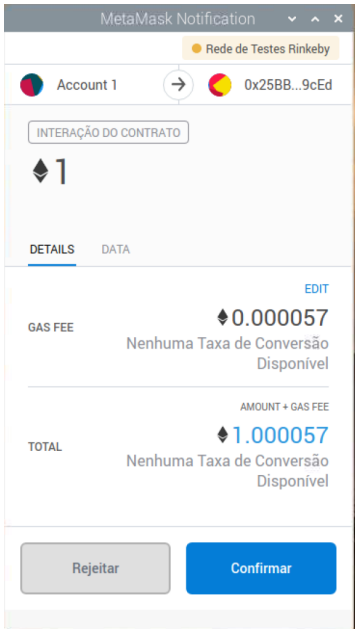
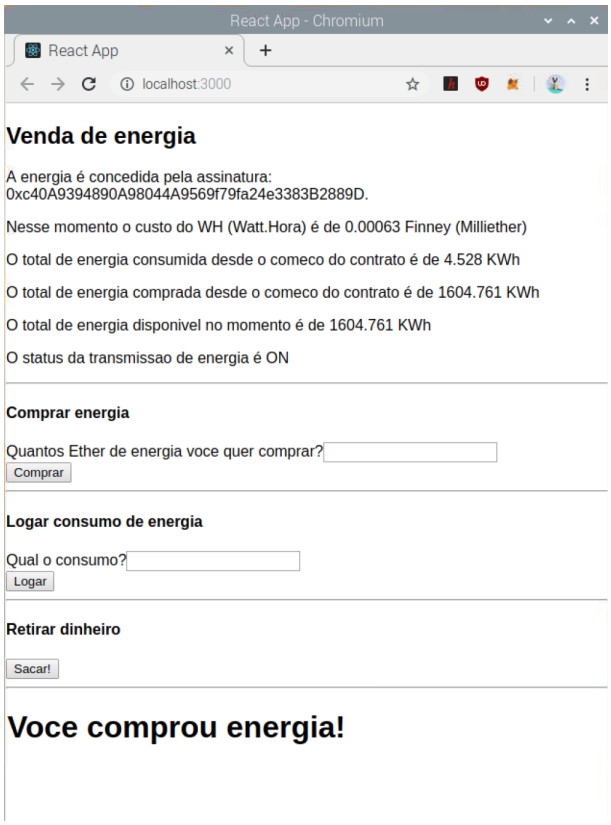


Figura 9: Interface web com informações atualizadas



Cada contrato, é vinculado a um concessor e a um dispositivo medidor e controlador de energia. Podemos pensar na compra de energia, como uma cláusula do contrato. Essa cláusula tem seu fluxo descrito no quadro branco intitulado Compra de Energia apresentado na figura 6. O trecho do código do contrato referente a compra de energia é apresentado

abaixo. É desejado que qualquer participante da rede possa comprar energia para um determinado contrato, pois isso facilita o uso para o usuário convencional, que poderia comprar energia através de qualquer carteira, ou poderia comprar energia para outros usuários, em outros contratos semelhantes, se assim desejar. Então, é necessária uma cláusula que permite acesso público para execução. Essa cláusula pode ser vista na linha 1 do trecho de código apresentado abaixo. A palavra-chave public, representa que é uma cláusula de acesso público, e a palavra-chave informa que é esperado que seja enviado dinheiro na chamada desse método. É desejado estabelecer um valor mínimo em dinheiro para efetuar uma compra, então uma restrição foi aplicada, vide linha 2 do código apresentado abaixo e parada de decisão do fluxograma da figura 6: o valor mínimo para compra é de 0.01 ether, aproximadamente 8 reais em outubro de 2019. Caso o valor enviado seja inferior, a operação é cancelada e o solicitante recebe seu dinheiro de volta. Caso contrário o cláusula continuará a ser operada. A linha 3 calcula a quantidade de energia comprada, usando o valor enviado na chamada e o preço do WH. A linha 4 atualiza a quantidade de energia disponível e a linha 5 verifica se o balanço de energia é positivo. Caso seja, o dispositivo medidor/controlador, irá escutar essa mudança e irá voltar a conduzir, no caso de estar cortado anteriormente a compra. O dinheiro enviado na chamada fica armazenado no contrato e apenas o concessor tem direito de solicitar um saque (linha 45, apêndice A).

```
1     function comprarEnergia() public payable {
2         require(msg.value > .01 ether);  // aproximadamente 8
              reais
3         energiaComprada = energiaComprada + (msg.value/precoWh);
4         energiaDisponivel = energiaComprada - energiaConsumida;
5         if (energiaComprada - energiaConsumida > 0){
6             balancoPositivo = true;
7         }else {
8             energiaDisponivel =0;
9         }
10    }
```

Listing 5.1: cláusula Compra de Energia

O dispositivo, e apenas o dispositivo, pode registrar o consumo de energia até o momento, pois a regra da cláusula contratual estabelece isso. Essa regra pode ser vista na linha 30 do código no apêndice A, e na definição apresentada a cláusula Registrar consumo de energia, mostrada na figura 6. Assim que o dispositivo loga o consumo, o saldo total de energia consumida e de energia disponíveis são atualizados no contrato. Quando o Saldo de energia disponível não for positivo, a transmissão de energia é automaticamente cortada pelo dispositivo. A raspberry faz leituras no contrato a cada 200 ms e, quando identifica

essa condição faz uma chamada HTTP ao ESP8266, que por sua vez corta a energia. Caso mais energia seja comprada, uma nova chamada é feita e o dispositivo volta a conduzir energia.

O concessor, e apenas o concessor, tem direito de mudar o preço do WH conforme estabelecido nas cláusulas contratuais. O processo simplificado é descrito na cláusula Mudar Preço WH da figura 6. Quando o concessor altera o preço da energia, as novas compras de energia passam a ter um valor WH correspondente a alteração (linha 40, apêndice A).

Como é possível ver, com poucas linhas de código é possível fazer um sistema de venda que herda todas vantagens da Blockchain, como segurança, descentralização, e desnecessidade de confiança, ou seja, não é necessário confiar na outra parte para ter certeza que o contrato será cumprido.

Contudo, é necessário ter bastante cuidado ao fazer um contrato Blockchain, pois ele não é modificável e por tal razão se houverem falhas, essas poderão ser facilmente exploradas, tendo em vista que normalmente os códigos fontes são disponibilizados em conjunto ao contrato, para que ambas partes possam conferir. Dito isso, foram desenvolvidas uma série de testes em Javascript, com auxílio da suíte Truffle (SUITE, 2019), para garantir que o contrato não possa ser explorado de forma maliciosa e que todas as regras e condições desejadas estejam funcionando corretamente. Os testes podem ser conferidos no apêndice B.

6 CONCLUSÃO

A tecnologia Blockchain podem claramente beneficiar as operações, mercados e consumidores do sistema de energia. Através da Blockchain conseguimos a desintermediação, transparência e transações à prova de adulteração, além do mais importante: é que as cadeias de blocos oferecem novas soluções para capacitar consumidores e pequenos geradores renováveis. Com poucas linhas de código é possível obter um sistema robusto, permitindo aplicações de economia compartilhada no setor de energia.

Esse trabalho contribui ao oferecer uma contextualização e perspectiva sobre o futuro mercado de energia, bem como trazer uma visão geral da tecnologia Blockchain e demais envolvidas no processo de venda de energia por Blockchain. O trabalho por fim apresenta um modelo simplificado de venda de energia por Blockchain gerido por um contrato inteligente, criado na linguagem solidity, bem como seus testes de integridade.

REFERÊNCIAS

AHSAN, U.; BAIS, A. Distributed smart home architecture for data handling in smart grid. **Canadian Journal of Electrical and Computer Engineering**, IEEE, v. 41, n. 1, p. 17–27, 2018.

ARMBRUST, M. et al. Above the clouds: A view of cloud computing. **Berkeley Reliable Adaptive Distributed systems Laboratory (RADLab)**, 2015.

BAJPAI, P. **Countries Where Bitcoin Is Legal Illegal (DISH, OTSK)**. Investopedia, 2019. Disponível em: <https://www.investopedia.com/articles/forex/041515/countries-where-bitcoin-legal-illegal.asp>.

CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and smart contracts for the internet of things. **Ieee Access**, Ieee, v. 4, p. 2292–2303, 2016.

DANNEN, C. **Introducing Ethereum and Solidity**. [S.l.]: Springer, 2017.

ETHEREUM. 2019. Disponível em: <https://www.ethereum.org/>.

FANG, X. et al. Smart grid—the new and improved power grid: A survey. **IEEE communications surveys & tutorials**, IEEE, v. 14, n. 4, p. 944–980, 2011.

GUAN, Z. et al. Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid. **IEEE Internet of Things Journal**, IEEE, v. 4, n. 6, p. 1934–1944, 2017.

_____. Effect: An efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid. **Science China Information Sciences**, Springer, v. 62, n. 3, p. 32103, 2019.

HWANG, J. et al. Energy prosumer business model using blockchain system to ensure transparency and safety. **Energy Procedia**, Elsevier, v. 141, p. 194–198, 2017.

KHORASANY, M.; MISHRA, Y.; LEDWICH, G. Market framework for local energy trading: a review of potential designs and market clearing approaches. **IET Generation, Transmission & Distribution**, IET, v. 12, n. 22, p. 5899–5908, 2018.

KOSBA, A. et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In: IEEE. **2016 IEEE symposium on security and privacy (SP)**. [S.l.], 2016. p. 839–858.

KYPRIOTAKI, K.; ZAMANI, E.; GIAGLIS, G. From bitcoin to decentralized autonomous corporations. In: **International conference on enterprise information systems**. [S.l.: s.n.], 2015. p. 284–290.

MAT. **Flashing Tasmota on Sonoff POW R2**. 2019. Disponível em: <https://notenoughtech.com/home-automation/esp/flashing-tasmota-on-sonoff-pow-2/>.

MATTILA, J. **The Blockchain Phenomenon – The Disruptive Potential of Distributed Consensus Architectures**. [S.l.], 2016. Disponível em: <https://ideas.repec.org/p/rif/wpaper/38.html>.

MOUGAYAR, W. **Blockchain para negócios: promessa, prática e aplicação da nova tecnologia da internet**. [S.l.]: Alta Books Editora, 2018.

MUFTIC, S. et al. Overview and analysis of the concept and applications of virtual currencies. **Joint Research Centre, Italy Raspoloživo na: http://publications. jrc. ec. europa. eu/repository/handle/JRC105207**, 2016.

NAKAMOTO, S. et al. Bitcoin: A peer-to-peer electronic cash system. Working Paper, 2008.

PARK, C.; YONG, T. Comparative review and discussion on p2p electricity trading. **Energy Procedia**, Elsevier, v. 128, p. 3–9, 2017.

POP, C. et al. Blockchain based decentralized management of demand response programs in smart energy grids. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 18, n. 1, p. 162, 2018.

RAHMADIKA, S.; RAMDANIA, D. R.; HARIKA, M. Security analysis on the decentralized energy trading system using blockchain technology. **Jurnal Online Informatika**, v. 3, n. 1, p. 44–47, 2018.

REYNA, A. et al. On blockchain and its integration with iot. challenges and opportunities. **Future Generation Computer Systems**, Elsevier, v. 88, p. 173–190, 2018.

RINKEBY Testnet. 2019. Disponível em: <https://www.rinkeby.io/>.

SCLOVE, R. E. Energy policy and democratic theory. In: **Uncertain Power**. [S.l.]: Elsevier, 1983. p. 37–65.

SONOFF. 2019. Disponível em: <https://www.itead.cc/wiki/Sonoff_Pow>.

SUITE, T. **Truffle: Truffle Quickstart: Documentation**. 2019. Disponível em: <https://www.trufflesuite.com/docs/truffle/quickstart#testing>.

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

UOL. **Mercado de bitcoin causa furor no Japão**. UOL, 2018. Disponível em: <https://economia.uol.com.br/noticias/afp/2018/01/05/mercado-de-bitcoin-causa-furor-no-japao.htm>.

WEB3.JS - Ethereum JavaScript API. 2019. Disponível em: <https://web3js.readthedocs.io/en/v1.2.2/#>.

YANG, T. et al. Applying blockchain technology to decentralized operation in future energy internet. In: IEEE. **2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)**. [S.l.], 2017. p. 1–5.

ZHENG, Z. et al. Blockchain challenges and opportunities: A survey. **International Journal of Web and Grid Services**, Inderscience Publishers (IEL), v. 14, n. 4, p. 352–375, 2018.

ZHIYI, L.; SHAHIDEHPOUR, M.; XUAN, L. Cyber-secure decentralized energy management for iot-enabled active distribution networks. **Journal of Modern Power Systems and Clean Energy**, Springer, v. 6, n. 5, p. 900–917, 2018.

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**, Elsevier, v. 144, p. 22–32, 2017.

Apêndices

APÊNDICE A – CONTRATO DE VENDA DE ENERGIA EM SOLIDITY

```
1  pragma solidity ^0.4.17;
2
3  contract VendaEnergia {
4      address public concessor;
5      uint public precoWh;
6      uint energiaComprada;
7      uint energiaConsumida;
8      uint energiaDisponivel;
9      bool balancoPositivo;
10
11
12     function VendaEnergia(uint precoWHInicial) public { //funcao
           construtora  // preco em wei
13         require(precoWHInicial > 0);  // preco em nanoether //
               preco no estado de sp = 630 gwei (nano ether)
14         concessor = msg.sender;
15         precoWh = precoWHInicial*(10**9);
16         balancoPositivo = false;
17     }
18
19     function comprarEnergia() public payable {
20         require(msg.value > .01 ether);  // aproximadamente 8
               reais
21         energiaComprada = energiaComprada + (msg.value/precoWh);
22         energiaDisponivel = energiaComprada - energiaConsumida;
23         if (energiaComprada - energiaConsumida > 0){
24             balancoPositivo = true;
25         }else {
26             energiaDisponivel =0;
27         }
28     }
29
30     function logarConsumo(uint novoConsumo) public restricted {
31         require(novoConsumo > energiaConsumida);
32         energiaConsumida = novoConsumo;
33         energiaDisponivel = energiaComprada - energiaConsumida;
34         if (energiaComprada <= energiaConsumida){
```

```
35              balancoPositivo = false;
36              energiaDisponivel = 0;
37          }
38      }
39
40      function mudarPrecoWh(uint novoPreco) public restricted { //
           preco em gwei
41          require(novoPreco > 0);
42          precoWh = novoPreco*(10**9);
43      }
44
45      function receberDinheiro() public restricted { //preco em
           gwei
46          concessor.transfer(this.balance);  // manda o dinheiro no
                contrato para o concessor.
47      }
48
49      function getEnergiaConsumida() public view returns (uint) {
50          return energiaConsumida;
51      }
52      function getEnergiaComprada() public view returns (uint) {
53          return energiaComprada;
54      }
55      function getEnergiaDisponivel() public view returns (uint) {
56          return energiaDisponivel;
57      }
58
59      function getBalanco() public view returns (bool){
60          return balancoPositivo;
61      }
62
63      modifier restricted() {
64          require(msg.sender == concessor);
65          _;
66      }
67  }
```

Listing A.1: Contrato de Venda de Energia em Solidity

APÊNDICE B – TESTES

```
1  const assert = require("assert");
2  const ganache = require("ganache-cli");
3  const Web3 = require("web3");
4  const web3 = new Web3(ganache.provider());
5
6  const { interface, bytecode } = require("../compile");
7
8  let vendaEnergia; // variavel para aramazenar instancia do
       contrato
9  let accounts; // variavel para aramazenar contas geradas pelo
       ganache para testes
10
11 beforeEach(async () => {
12   accounts = await web3.eth.getAccounts(); // passa as contas
         criadas pelo ganache para a variavel accounts
13
14   vendaEnergia = await new web3.eth.Contract(JSON.parse(interface
         ))
15     .deploy({ data: bytecode, arguments: ["630"] })
16     .send({ from: accounts[0], gas: "1000000" });
17 });
18
19 describe("Contrato de venda de energia", () => {
20   it("Realiza deploy do contrato", () => {
21     assert.ok(vendaEnergia.options.address);
22   });
23
24   it("Permite comprar energia", async () => {
25     await vendaEnergia.methods.comprarEnergia().send({
26       from: accounts[1],
27       value: web3.utils.toWei("0.02", "ether")
28     });
29   });
30
31   it("Permite comprar energia de qualquer contas", async () => {
32     await vendaEnergia.methods.comprarEnergia().send({
33       from: accounts[2],
```

```
34        value: web3.utils.toWei("0.02", "ether")
35      });
36    });
37
38    it("Nao permite fazer uma compra com valor inferior a 10
         finneys", async () => {
39      try {
40        await vendaEnergia.methods.comprarEnergia().send({
41          from: accounts[2],
42          value: web3.utils.toWei("0.0099", "ether")
43        });
44        assert(false);
45      } catch (err) {
46        assert(err);
47      }
48    });
49
50    it("Permite qualquer um verificar o preco do WH", async () => {
51      await vendaEnergia.methods.precoWh().send({
52        from: accounts[2]
53      });
54    });
55
56    it("Permite o concessor mudar o preco do WH", async () => {
57      await vendaEnergia.methods.mudarPrecoWh("640").send({
58        from: accounts[0]
59      });
60    });
61
62    it("Exige que um valor seja passado quando solicitar a mudanca
         do preco do WH", async () => {
63      try {
64        await vendaEnergia.methods.mudarPrecoWh().send({
65          from: accounts[0]
66        });
67        assert(false);
68      } catch (err) {
69        assert(err);
70      }
71    });
72
73    it("Apenas o concessor pode mudar o preco do WH", async () => {
```

```
74      try {
75        await vendaEnergia.methods.mudarPrecoWh("640").send({
76          from: accounts[1]
77        });
78        assert(false);
79      } catch (err) {
80        assert(err);
81      }
82    });
83
84    it("Concessor pode receber os fundos do contrato ", async () =>
        {
85      await vendaEnergia.methods.comprarEnergia().send({
86        from: accounts[2],
87        value: web3.utils.toWei("1", "ether")
88      });
89      const saldoInicial = await web3.eth.getBalance(accounts[0]);
90      await vendaEnergia.methods.receberDinheiro().send({
91        from: accounts[0]
92      });
93      const saldoFinal = await web3.eth.getBalance(accounts[0]);
94      const diferenca = saldoFinal - saldoInicial;
95      assert(diferenca > web3.utils.toWei("0.8", "ether"));
96    });
97
98    it("Apenas o concessor pode receber os fundos do contrato",
       async () => {
99      try {
100       await vendaEnergia.methods.receberDinheiro().send({
101         from: accounts[2]
102       });
103       assert(false);
104     } catch (err) {
105       assert(err);
106     }
107   });
108
109   it("Permite o concessor logar o consumo", async () => {
110     await vendaEnergia.methods.logarConsumo("10000").send({
111       from: accounts[0]
112     });
113   });
```

```
114
115   it("Apenas o concessor pode logar o consumo", async () => {
116     try {
117       await vendaEnergia.methods.logarConsumo("10000").send({
118         from: accounts[2]
119       });
120       assert(false);
121     } catch (err) {
122       assert(err);
123     }
124   });
125
126 });
```

APÊNDICE C – INTERFACE WEB

```
import React, { Component } from "react";
import "./App.css";
import web3 from "./web3";
import vendaEnergia from "./vendaEnergia";

class App extends Component {
  state = {
    concessor: "",
    precoWh: "",
    energiaComprada: "",
    energiaDisponivel: "",
    value: "",
    message: "",
    pwr: { POWER: "" },
    obj: {
      StatusSNS: {
        Time: "",
        ENERGY: {
          TotalStartTime: "",
          Total: -1,
          Yesterday: 0,
          Today: 0,
          Power: 0,
          ApparentPower: 0,
          ReactivePower: 0,
          Factor: 0,
          Voltage: 0,
          Current: 0
        }
      }
    }
  };

  async componentDidMount() {
    try {
      setInterval(async () => {
        let dataPOWER;
```

```
38          const res = await fetch("http://192.168.1.148/cm?cmnd=
               Status%208");
39          const dataObj = await res.json();
40          if (dataObj.StatusSNS.ENERGY.Total > this.state.
               energiaComprada) {
41            const res2 = await fetch("http://192.168.1.148/cm?cmnd=
                 Power%200");
42            dataPOWER = await res2.json();
43          } else {
44            const res3 = await fetch("http://192.168.1.148/cm?cmnd=
                 Power%201");
45            dataPOWER = await res3.json();
46          }
47
48          this.setState({
49            obj: dataObj,
50            pwr: dataPOWER
51          });
52
53          console.log(dataPOWER);
54        }, 3000);
55      } catch (e) {
56        console.log(e);
57      }
58
59      const concessor = await vendaEnergia.methods.concessor().call
           ();
60      const precoWh = await vendaEnergia.methods.precoWh().call();
61      const energiaComprada = await vendaEnergia.methods
62        .getEnergiaComprada()
63        .call();
64      const energiaConsumida = await vendaEnergia.methods
65        .getEnergiaConsumida()
66        .call();
67      const energiaDisponivel = await vendaEnergia.methods
68        .getEnergiaDisponivel()
69        .call();
70      const balancoPositivo = await vendaEnergia.methods.getBalanco
           ().call();
71      this.setState({
72        concessor,
73        precoWh,
```

```
74        energiaComprada,
75        energiaConsumida,
76        energiaDisponivel,
77        balancoPositivo
78      });
79    }
80
81    onSubmit = async event => {
82      event.preventDefault();
83
84      const accounts = await web3.eth.getAccounts();
85
86      this.setState({
87        message: "Aguarde, transacao sendo processada na rede
              blockchain..."
88      });
89
90      await vendaEnergia.methods.comprarEnergia().send({
91        from: accounts[0],
92        value: web3.utils.toWei(this.state.value, "ether")
93      });
94
95      this.setState({ message: "Voce comprou energia!" });
96    };
97
98    onLogar = async event => {
99      event.preventDefault();
100
101     const accounts = await web3.eth.getAccounts();
102
103     this.setState({
104       message: "Aguarde, Log sendo processado na rede blockchain
              ..."
105     });
106
107     await vendaEnergia.methods.logarConsumo(this.state.
           inputConsumo).send({
108       from: accounts[0]
109     });
110
111     this.setState({ message: "Voce comprou energia!" });
112   };
```

```
113   onClick = async () => {
114     const accounts = await web3.eth.getAccounts();
115
116     this.setState({ message: "Aguarde, sacando dinheiro" });
117
118     await vendaEnergia.methods.receberDinheiro().send({
119       from: accounts[0]
120     });
121
122     this.setState({ message: "O dinheiro foi enviado" });
123   };
124
125   render() {
126     return (
127       <div>
128         <h2>Venda de energia</h2>
129         <p>A energia    concedida pela assinatura: {this.state.
              concessor}.</p>
130         <p>
131           Nesse momento o custo do WH (Watt.Hora)    de{" "}
132           {web3.utils.fromWei(this.state.precoWh, "finney")}
                Finney (Milliether)
133         </p>
134         <p>
135           O total de energia consumida desde o comeco do contrato
                  de{" "}
136           {this.state.obj.StatusSNS !== undefined
137             ? this.state.obj.StatusSNS.ENERGY.Total
138             : 0}{" "}
139           KWh
140         </p>
141         <p>
142           O total de energia comprada desde o comeco do contrato
                 de{" "}
143           {this.state.energiaComprada / 1000} KWh
144         </p>
145         <p>
146           O total de energia disponivel no momento    de{" "}
147           {this.state.energiaDisponivel / 1000} KWh
148         </p>
149         <p>
150           O status da transmissao de energia   {" "}
```

```
151           {this.state.pwr.POWER !== undefined ? this.state.pwr.
                  POWER : ""}
152         </p>
153         <hr />
154         <form onSubmit={this.onSubmit}>
155           <h4>Comprar energia</h4>
156           <div>
157             <label>Quantos Ether de energia voce quer comprar?</
                    label>
158             <input
159               value={this.state.value}
160               onChange={event => this.setState({ value: event.
                      target.value })}
161             />
162           </div>
163           <button>Comprar</button>
164         </form>
165         <hr />
166         <form onSubmit={this.onLogar}>
167           <h4>Logar consumo de energia</h4>
168           <div>
169             <label>Qual o consumo?</label>
170             <input
171               value={this.state.inputConsumo}
172               onChange={event =>
173                 this.setState({ inputConsumo: event.target.value
                        })
174               }
175             />
176           </div>
177           <button>Logar</button>
178         </form>
179         <hr />
180         <h4>Retirar dinheiro</h4>
181         <button onClick={this.onClick}>Sacar!</button>
182         <hr />
183         <h1>{this.state.message}</h1>
184       </div>
185     );
186   }
187 }
188
```

```
189 export default App;
```