

# Uso da Tecnologia Blockchain no Armazenamento de Dados de Geração de Energia Alternativa

Kelven Klein Rocha<sup>1</sup>, Rodrigo Lange<sup>1</sup>

<sup>1</sup>Instituto Federal de Ciência e Tecnologia do Rio Grande do Sul - Câmpus Ibirubá  
Rua Nelsi Ribas Fritsch, 1111 – CEP: 98200-000 – Ibirubá – RS – Brasil

**Abstract.** *The National Electric Energy Agency (ANEEL) created the rules for energy production by own means, which are not favorable for domestic energy producers that produce less than 75 kw, as they cannot sell electricity in public auctions and are not encouraged to produce more than they consume. The objective of this work is to show the blockchain technology (A decentralized and independent database) the potential to help in the sector of small energy production and consumption networks, such as condominiums. The use of blockchain ensures that energy producers have their data secured and measured, getting the chance to be reclassified in ANEEL regulation. For this, it is used a smart contract (A program written on a blockchain), for this purpose, two frameworks (Common codes for use in various software projects) were theoretically compared to a blockchain (A blockchain framework is composed of the infrastructure and the application) and the best one was chosen and a prototype was built using the most suitable tool to compensate energy producers. The increased use of solar panels exposes the potential of renewable energy for use by condominiums that use bi-directional meters for each house, showing consumption or production in kw/h, the utility standard. Due to the fact that energy collection information is dispersed across several producers, using the blockchain system user information can be grouped with a smart contract written on the prototype, which has identification variables of: kw/h, description, functions for purchase, sale and consultation. Blockchain brings the advantage of not depending on a third party to mediate the relationship between energy producer and consumer, eliminating mutual distrust between the parties. Producers can register their data on the network and thus have the means to market, causing a purchase and sale stimulus, making it possible to change the classification of their producers in the regulation, from the moment that a continuous production of 75 kw can be grouped.*

**Resumo.** *A Agência Nacional de Energia Elétrica (ANEEL) criou as regras para produção de energia por meios próprios, que não são favoráveis para produtores de energia domésticos que produzem menos de 75 kw, pois não podem vender energia elétrica em leilões públicos e não são incentivados a produzir mais que consomem. Objetivo nesse trabalho é mostrar a tecnologia blockchain (Um banco de dados descentralizado e independente) o potencial de ajudar no setor de pequenas redes de produção e consumo de energia, como condomínios. O uso de blockchain garante que os produtores de energia tenham seus dados assegurados e mensurados obtendo a chance de serem reclassificados na regulação da ANEEL. Para isso, é usado um contrato inteligente (Um programa escrito em uma blockchain), para isso foi comparado teoricamente dois*

*framework (Códigos comuns para uso em vários projetos de software) para blockchain (Um framework para blockchain é composto pela infraestrutura e a aplicação) decidiu-se pelo melhor e foi montado um prototipo usando a ferramenta mais adequada para compensar os produtores de energia. O aumento do uso de painéis solares expõe o potencial da energia renovável para uso de condomínios que usam medidores bidirecionais para cada casa, mostrando o consumo ou produção em kw/h, o padrão das concessionárias conforme o protocolo que é o ABNT-CODI na norma NBR14522(ABNT, 2008), este protocolo também define a comunicação remota entre os medidores e os sistemas (Sistemas para controle de consumo para emissão de faturas). Devido ao fato das informações de coleta de energia serem dispersas em vários produtores, usando o sistema blockchain as informações dos utilizadores podem ser agrupadas com um contrato inteligente escrito no prototipo, que possui variáveis de identificação de: kw/h, descrição, funções para compra, venda e consulta. A blockchain traz a vantagem de não depender de uma terceira parte para mediar a relação do produtor e do consumidor de energia, eliminando a mutua desconfiança entre as partes. Os produtores podem registrar seus dados na rede e assim ter meios para comercializarem, causando um estímulo de compra e venda fazendo com que haja possibilidade de mudar a classificação de seus produtores na regulação, à partir do momento que possa ser agrupado uma produção contínua de 75 kw.*

## **1. Introdução**

A geração de energia elétrica por meios alternativos tem se tornado cada vez mais popular em diversos países, e o Brasil não é exceção. É fácil observar o crescimento do número de painéis solares instalados tanto em empresas de todos os portes quanto em ambientes domiciliares.

Devido ao crescimento da geração de energia por meios próprios, a Agência Nacional de Energia Elétrica (ANEEL) criou, em 2012, regras para compensação de produção de energia. Tais regras definem a relação entre distribuidoras de energia e clientes, explicitando como a energia excedente gerada pode ser vendida no mercado energético nacional (ANEEL, 2015).

Entretanto, as regras da ANEEL estabelecem que geradores com capacidade de geração de até 5MW somente podem participar do sistema de compensação, isto é, não podem vender energia elétrica em leilões públicos (ANEEL, 2015). Estas regras são bastante desfavoráveis para para produtores domésticos individuais, que geralmente se encaixam no perfil de microgeradores de energia com capacidade instalada de até 75kW (ANEEL, 2015).

Para o caso de um condomínio residencial com diversos geradores individuais (cuja carga somada ultrapasse os 5MW), uma solução para evitar as restrições impostas pela ANEEL pode ser a criação de um ambiente cooperado cujo excesso de geração é vendido em leilões de energia.

Um problema relacionado à criação de um ambiente cooperado é o armazenamento dos dados dos geradores individuais, tanto de geração quanto de consumo. Normalmente estes dados são armazenados em banco de dados (seja localmente ou na nu-

vem), mas esse tipo de solução exige a disponibilização de uma estrutura de Tecnologia da Informação.

Uma opção para o armazenamento dos dados surge na forma do uso de tecnologias de *blockchain*, que podem ser considerados bancos de dados descentralizados (SWAN, 2015).

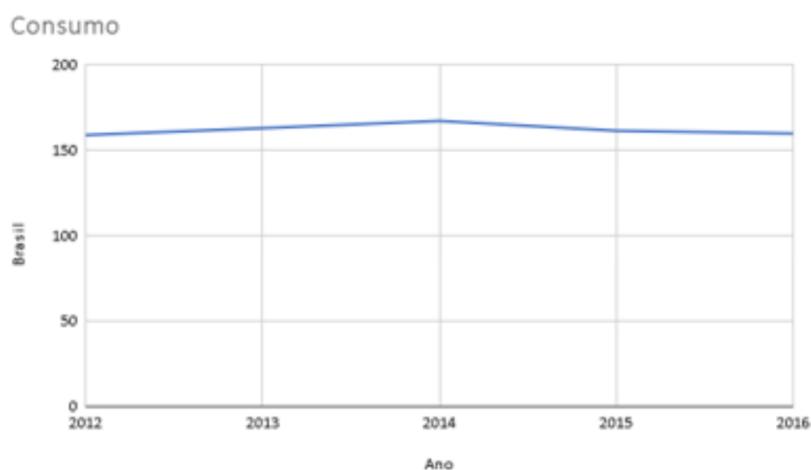
Do exposto acima vem o principal objetivo deste trabalho: demonstrar a possibilidade de uso da tecnologia *blockchain* para o armazenamento de dados de geração e de consumo de energia elétrica por pequenos condomínios de produtores individuais.

No restante deste capítulo serão apresentadas as motivações, justificativas e delimitações do trabalho.

No Capítulo 2 são apresentadas as tecnologias de blockchain utilizadas nesta proposta. No Capítulo 3 são discutidas propostas correlatas. O Capítulo 4 descreve o que foi realizado e por fim, no Capítulo 5 são apresentadas as conclusões e considerações finais do trabalho.

### 1.1. Panorama de microgeração de energia no Brasil

Conforme norma da ANEEL resolução normativa - REN nº 687-2015, os produtores de energia que produzem menos de 75 kw contínuo, fazem parte do Ambiente de Contratação Regulada (ACR) (EPE, 2017), não podendo comercializar energia elétrica em leilões públicos. A ACR não atende às necessidades de grande parte da população, pois a média de consumo de energia elétrica per capita no Brasil raramente ultrapassa 175 kWh/mês, como ilustrado no gráfico da figura 1.



**Figura 1. Consumo médio residencial do Brasil (kWh/mês versus ano), mostrando uma média de consumo baixa em nível nacional. Fonte: (EPE, 2017).**

Como mostrado no gráfico 1, o consumo de energia de 2 ou 3 pessoas em uma casa de produtores de energia fotovoltaicos, automaticamente entra na regulação da ACR. Nessa regulação, um produtor só consegue vender energia para a concessionária. A produção é mensurada através de medidores bidirecionais (ALVES, 2019) e o excedente é

emprestado à distribuidora, que devolve em até 60 meses através de descontos na fatura. Logo, não existem incentivos para que o produtor instale uma capacidade de produção superior ao seu consumo mensal (SIQUEIRA, 2015).

Para os produtores serem incentivados a instalar sistemas e poderem vender energia livremente (WU, 2018), é necessário o registro de dados em um sistema terceiro para que produtores de energia tenham seus conjuntos de dados mensurados a quantidade de energia produzida e assim poder reclassificar seus utilizadores em grupos.

Conforme verificado, as perspectivas futuras para utilização de geração própria para grupos ou em condomínios são positivas, melhoram a qualidade de vida e o senso de comunidade, aumentando o senso de cidadania e organização das populações gerando um bom impacto social (MELO, 2018).

## **1.2. Objetivo**

O presente trabalho tem como objetivo o estudo teórico da de dois frameworks para blockchain, a utilização de um contrato inteligente baseado em no framework de blockchain escolhido para o registro de dados de produção de energia de pequenas redes de produção e consumo de energia, como condomínios. Desta forma, este estudo objetiva comparar dois principais tipos de framework para blockchain, os quais serão abordados neste trabalho: a blockchain Hyperledger e a Ethereum.

## **1.3. Objetivos específicos**

Analisar teoricamente o panorama atual de dois frameworks para blockchain, detalhando a composição, assim como, suas características.

Implementar em um framework escolhido, um sistema que utilize um contrato inteligente que suporte os dados de produção da base geradora.

Desenvolver um protótipo em contrato inteligente, voltado ao registro de conhecimento que, de maneira distribuída, consiga manipular os dados de geração.

Registrar os dados de geração fictícios, e futuramente atuar em um modelo de negócio mais complexo como trocar para outras modalidades nas normas da ANEEL, a partir do conhecimento dos dados.

## **1.4. Justificativa**

Devido à dificuldade dos geradores individuais se encaixarem na regulação como foi mostrado na seção 1.1, esse trabalho se justifica através do registro dos dados dos geradores de energia serem guardados em uma blockchain. Esses dados são assegurados devido às transações de compra e venda de energia serem registradas por criptografia e replicadas várias cópias em estações diferentes e em diversos lugares. Essas são características básicas das redes blockchain, tornando os sistemas blockchain mais resilientes e seguros (MELO, 2018). A proposta é melhor que um banco de dados tradicional devido ao fato de ser baseada em infraestrutura existente: Internet e descentralização (WU, 2018). Um exemplo é a Brooklyn Microgrid, pioneira plataforma de comércio de energia usando blockchain (WANG, 2019).

## **1.5. Delimitação**

O tema apresentado foca essencialmente na comparação teórica entre os frameworks para blockchain, e a respectiva escolha de um framework para montagem de um

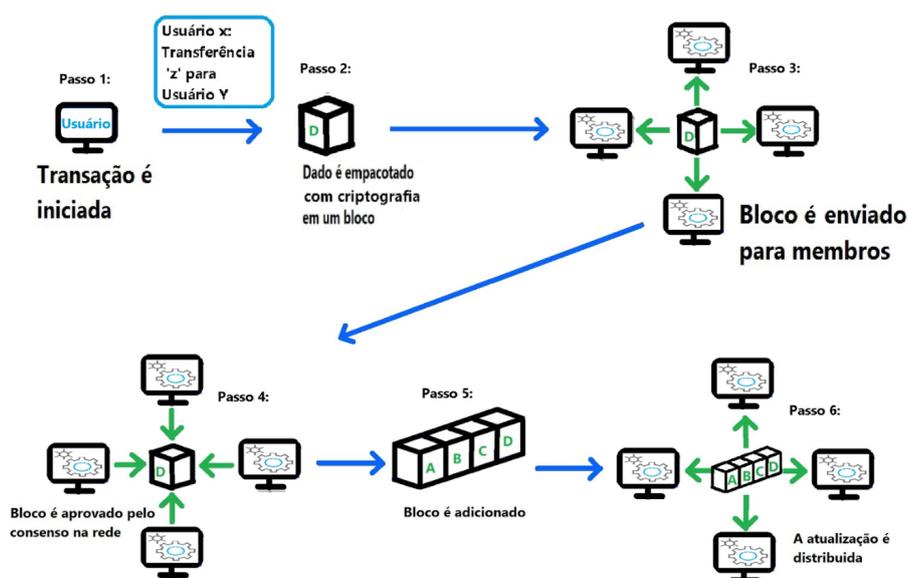
protótipo fazendo o registro de dados fictícios de: produtores, consumidores e transmissores de energia (individuais ou em grupo) em um contrato inteligente em uma blockchain. Não fez parte do escopo deste trabalho propor algo dependente de outros fatores, como questões comerciais e versões das ferramentas mencionadas.

No restante deste documento serão mostradas a definição de uma blockchain, a definição de um framework para blockchain e o seu funcionamento (Capítulo 2), seguido de uma discussão sobre propostas semelhantes às já existentes (Capítulo 3) que inspiraram a linha de pensamento sobre o trabalho. Por fim, no Capítulo 4 será apresentada como foi abordada a solução da proposta em si, contendo diagramas e as atividades que foram desenvolvidas nesta segunda etapa do trabalho.

## 2. Blockchain

Blockchain é um sistema de registro distribuído que mantém uma lista ordenada, chamada de bloco. Os usuários só podem editar a parte da lista que eles possuem, com a chave privada (Somente o dono à possui) necessária para gravar o arquivo. Sua segurança é baseada em criptografia e a replicação em milhares de nós (Pontos de verificação) na rede (NAKAMOTO, 2008).

Uma blockchain funciona da seguinte forma: todo dado de uma parte para outra é registrado no sistema, e o empilhamento dos dados forma um bloco que tem um endereço (Hash) para referência ao bloco seguinte. Após o bloco ser concluído, é distribuído e comparado na rede (CROSBY, 2018). É feito um consenso entre os nós da rede então o bloco é validado, salvo e novamente distribuído nos nós, e encaixado acima do bloco anterior, formando mais um elo na cadeia. Por fim, salvo em toda a rede, todo o processo acima descrito é observado na Figura 2.



**Figura 2.** O usuário inicia a transação ao acionar sua chave privada, o registro é feito por criptografia no bloco e replicado para os membros, quando o bloco for aprovado por todos os participantes da rede é adicionado e distribuído.

## 2.1. Definição de framework para blockchain

No desenvolvimento de programas para computadores, um framework é uma abstração que fornece funcionalidades genéricas, podendo ser alterado seletivamente pelo desenvolvedor. Ele fornece um padrão de construção de aplicativos e é um ambiente de reutilização numa plataforma de software maior para facilitar produtos e soluções. As estruturas de software podem incluir programas de suporte, compiladores, bibliotecas de código, conjuntos de ferramentas e interfaces de programação de aplicativos (APIs) para o desenvolvimento de um projeto ou sistema.

Um framework para blockchain é composto pela infraestrutura e a aplicação (GARRIGA MAXMILIANO ARIAS, 2018). A infraestrutura é a rede que é formada pelos nós da rede e pelo software executado neles. Os nós da rede podem ser máquinas físicas ou máquinas virtuais que fornecem recursos e funcionamento dos usuários (identidade ou contas), criação e processamento de transações, protocolo de consenso e exclusividades, como por exemplo: token para as de rede pública, e gerenciamento para as permissionadas.

A aplicação no framework é o código executado na infraestrutura, sendo também conhecido como contrato inteligente. É formado por um aplicativo cliente que interage com a infraestrutura. Este último serve como ponto de acesso do mundo externo. Uma boa estrutura de framework para blockchain permite o desenvolvimento de aplicativos fora da implantação real (Blockchain para uso em redes ou máquinas locais), e também na rede real.

Uma das desvantagens de um blockchain público é o grande poder computacional necessário para manter o livro caixa distribuído. Outra desvantagem é a abertura do blockchain público, o que implica pouca privacidade para transações.

## 2.2. Definição Blockchain pública e permissionada

O tipo de uma blockchain pode ser definido por público e permissionada, que possuem muitas semelhanças entre elas:

São redes descentralizadas, onde cada nó mantém uma réplica do livro caixa.

Mantêm as réplicas sincronizadas por meio de um protocolo de consenso.

Fornecem certas garantias sobre a imutabilidade do livro caixa, mesmo quando alguns participantes são defeituosos ou maliciosos.

A única distinção entre os dois tipos está relacionada a quem tem permissão para participar da rede, executar o protocolo de consenso e manter o livro caixa compartilhado (GARRIGA MAXMILIANO ARIAS, 2018). Uma rede blockchain pública é completamente aberta e qualquer um pode participar. A rede geralmente possui um mecanismo de incentivo (Comércio ou ágio de moedas, dependendo da rede) para ingressarem mais participantes.

Já uma rede blockchain permissionada à partir do momento que é implementada, requer um convite emitido pelo iniciador da rede ou por um conjunto de regras implementadas. Isso impõe restrições sobre quem pode participar da rede ou em determinadas transações. Pode ser implementado o como vai ser feito o ingresso dos futuros participantes: Uma autoridade reguladora pode emitir licenças para participação; ou um consórcio poderia tomar as decisões.

## 2.3. Descrição da blockchain Ethereum

A blockchain Ethereum é uma plataforma descentralizada que executa programas e contratos inteligentes, é de acesso público, isto devido ao fato de qualquer um poder

configurar uma máquina para executar o sistema distribuído, se tornando um nó da rede, sem nenhuma restrição.

Ethereum pode utilizar o Truffle como framework (MUKHOPADHYAY, 2018). Truffle têm um ambiente de desenvolvimento e de teste para usar a Solidity, que é a linguagem de Ethereum, usada para a implementação de contratos inteligentes.

Solidity foi implementado usando C++, Python e JavaScript e é projetado para utilizar a Ethereum Virtual Machine (EVM), nela é possível criar contratos para votação, crowdfunding, leilões, carteiras com várias assinaturas (DANNEN, 2017).

Um contrato inteligente utilizando ferramenta Truffle têm seu código em Solidity residindo em um endereço específico na blockchain Ethereum. Sua estrutura é feita para mapear facilmente os contratos que possuem classes, atributos e funções (HEGEDŰS, 2019). No seu algoritmo apresenta o Proof-of-Stake (PoS), que premia os nós da rede com ágio, após os nós processarem os blocos do sistema (PORAT, 2017).

Seu código apresenta vantagens nas propriedades de herança, podendo criar classes compartilhadas. Usa variáveis pré-definidas (MUKHOPADHYAY, 2018), que podem ser aplicadas a várias funções, reduzindo a carga de código dentro dos contratos. Também possui como vantagem: a Interface Binária de Aplicação (ABI), que facilita as interações entre aplicativos binários (MODI, 2018).

Uma das desvantagens na rede Ethereum é o fato de ser impossível de adquirir dados nos sistemas, exceto por meio de transações (HUKKINEN, 2019).

#### **2.4. Descrição da blockchain Hyperledger**

Fabric é o framework de um dos projetos da Hyperledger que utiliza blockchain privado, só pode ser utilizado por membros que contenham a permissão que contenha dos requisitos mínimos para participação. Permite a utilização com contratos inteligentes, suportando que o sistema conecte usando tomadas de decisões, com um protocolo de confiança consensual modular (BLUMMER, 2018). No Hyperledger os nós de validação funcionam para validar transações, manter o livro caixa e executar o protocolo de consenso.

Sem dependência de uma criptomoeda nativa, seu código é escrito em Go, Java e Node Js (MANEVICH ARTEM BARGER, 2019). Foca o uso em aplicações para padrão de uso industrial e corporativo (ANDROULAKI, 2018). Têm capacidade de consulta em dados particionados com escalabilidade e usa software livre.

Como o Fabric foi lançado em julho de 2017, uma desvantagem é o fato de o framework ser novo. Não tendo um token criptográfico, não há incentivo para os nós manterem a rede, quanto os validadores fazem. Uma outra desvantagem é de ser obrigado a se identificar na rede.

#### **2.5. Análise dos frameworks para blockchain**

Os dois frameworks atendem os aspectos de auditoria, acessibilidade aos dados, confiabilidade e rapidez. Porém vale salientar, que ao precisar de uma ferramenta que poderia compensar os produtores de energia, mostra-se que o Ethereum é mais capacitado. A desvantagem do Hyperledger é não ter um token e principalmente de precisar de permissão de terceiros para inserção de usuários (Conforme a Tabela 1). Já o Ethereum traz a vantagem de ter o seu uso anônimo, o que é muito bom para aqueles que estão tentando fugir do ambiente altamente regulado. Como pode ser checado na tabela de características

seguinte, o Ethereum é de uso público então traz uma maior vantagem, por ser de mais fácil acesso, então por esse motivo será trabalhado neste trabalho. Além disso, o Ethereum foi escolhido por usar Javascript para escrever os códigos, que é muito bem conhecido.

Tabela de comparação.		
Características	Ethereum	Hyperledger
Linguagem:	Solidity (C++, Python e JavaScript)	Escrito em Go, Java, Node Js
Consenso:	Consenso de PoS	Conectável
Origem:	Uso público	Uso permissio-nado
Parceiros:	IC3, Microsoft, Accenture, J.P. Morgan, Consensus, Intel, Santander e CME Group	S. Accenture, AirBus, American Express, Cisco, Daimler, J.P. Morgan, Intel, IBM e SAP
Moeda:	Possui o Ether (ETH)	Não possui token
Aplicabilidade:	Público e privado, com uso genérico e trocas B2C	Ideal para B2B, participação permissiva

### 3. Análise de propostas correlatas

O artigo Blockchain technology in the energy sector: A systematic review of challenges and opportunities (ANDONIA, 2018) foi o principal artigo que contribuiu para o estudo, ao mostrar o conteúdo em seguida: aumento do uso de painéis solares e geradores eólicos no Reino Unido, também expondo o potencial do mercado de energia renovável e seus números de crescimento.

No artigo há vários casos potenciais de uso de blockchain para microgrid (pequenas redes de produção e consumo de energia, como condomínios), como o Brooklyn Microgrid, que exibe as muitas potencialidades para o uso da tecnologia para a indústria, investimentos, mercado de energia descentralizado de médio e pequeno porte e principalmente projetos para microgrids.

Já o artigo When Energy Trading Meets Blockchain in Electrical Power System: The State of the Art (WANG, 2019) destaca a racionalização do consumo e produção de energia com o uso da tecnologia de blockchain, comparada com os meios centralizados. Além disso, o artigo mostra a possível potencialidade do mercado peer-to-peer (P2P) de produção de energia. O artigo é interessante ao expor o problema quando há um sistema com alta disponibilidade, com participantes em contato sempre.

O artigo *Application of Blockchain Technology in Sustainable Energy Systems: An Overview* (WU, 2018) trouxe força ao estudo, pois no artigo, a empresa de energia dos Estados Unidos da América, LO3 Energy e a empresa de blockchain ConsenSys lançaram o projeto TransActive Grid no bairro do Brooklyn. Esta é uma plataforma de negociação de micro-potência baseada em blockchain composta por 10 casas de famílias (Em comparação aos testes mostrados nas próximas seções, foram simulados testes com 8 contas). No artigo o equipamento de geração de energia fotovoltaica é instalado no telhado de cada casa, formando um pequeno sistema blockchain com 10 nós. Cada casa pode ser um consumidor ou um produtor, tendo um medidor bidirecional que mostra o consumo ou produção em kwh, o mensuramento padrão das concessionárias, até nos medidores que são usados no Brasil (ALVES, 2019).

Semelhante ao trabalho, o artigo expõe que cada nó da rede conseguiu vender o excesso de energia diretamente aos vizinhos, e o processo de transação não teve a participação de empresas de energia. No artigo também cita a fornecedora de eletricidade alemã RWE e a empresa blockchain Slock colaboraram no projeto de carregamento elétrico de veículos com um sistema pagamento com contratos inteligentes de blockchain Ethereum (A mesma que usei no trabalho) que foram usados no lugar de empresas ou indivíduos terceiros. O método que usam para o artigo é qualitativo (textos, quadros explicativos e relações, que ajudam a explicar os achados (MARTINS, 2004)), o mesmo que foi usado no trabalho.

Destacando várias possibilidades de uso da blockchain para indústria, comércio, e consumo, o artigo conclui com os desafios como a quantidade de transações por segundo, a segurança e o consumo crescente de energia dos mineradores de tokens.

Todos os artigos mostram o potencial do uso da tecnologia blockchain para o setor de energia. Os autores abordam o uso voltado para o registro das informações de produtores descentralizados.

Os artigos também destacam que os dados no setor de energia estão dispersos e o sistema blockchain consegue trazer essas informações muito facilmente para as mãos dos utilizadores. Na próxima seção será detalhada a forma como foi implementada a proposta apresentada neste documento.

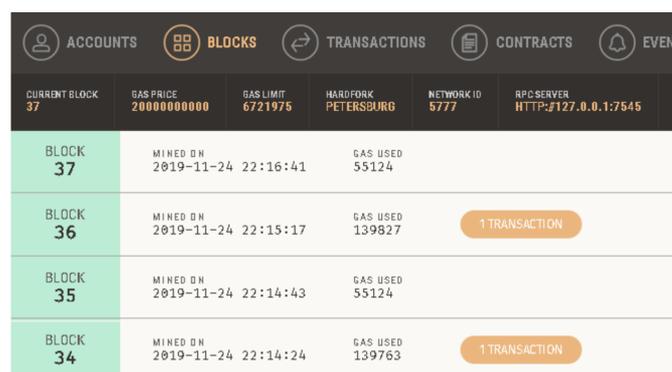
## **4. Metodologia**

A implementação qualitativa (Explicar uma tecnologia ainda não conhecida e pouco estudada (MARTINS, 2004)) da proposta foi realizada em uma máquina de uso local, desenvolvendo uma aplicação da blockchain escolhida na seção 2.5, a Ethereum. Para coleta, análise e visualização foi utilizada uma interface diretamente no navegador para os dados registrados em somente 8 contas (Representando um condomínio pequeno, como foi feito no projeto TransActive Grid (WU, 2018)) com quantidade de energia gerada, valor do token e uma descrição.

### **4.1. Ambiente de testes**

Para os registros das transações e dos contratos foi utilizada a rede blockchain de uso pessoal de testes da Ethereum: o Ganache (GOULART, 2021), que conforme a Figura 3, mostra os blocos que foram registrados, se houve transações com contrato e o custo computacional da rede Ethereum (GAS).

Para gerenciamento das contas, das transações e dos contratos, foi utilizada a extensão Metamask, no navegador Google Chrome, essa extensão é amigável ao usuário

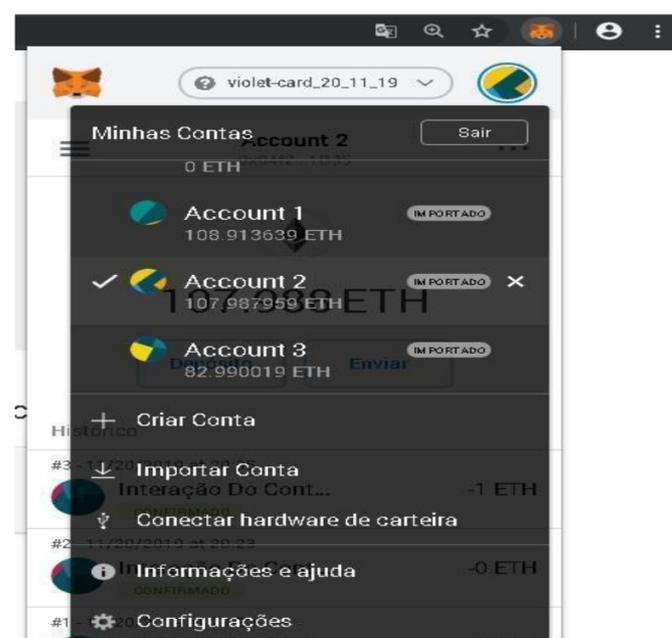


ACCOUNTS	BLOCKS	TRANSACTIONS	CONTRACTS	EVENTS	
CURRENT BLOCK	GAS PRICE	GAS LIMIT	HARDFORK	NETWORK ID	RPC SERVER
37	2000000000	6721975	PETERSBURG	5777	HTTP://127.0.0.1:7545
BLOCK 37	MINED ON	GAS USED			
37	2019-11-24 22:16:41	55124			
BLOCK 36	MINED ON	GAS USED		1 TRANSACTION	
36	2019-11-24 22:15:17	139827			
BLOCK 35	MINED ON	GAS USED			
35	2019-11-24 22:14:43	55124			
BLOCK 34	MINED ON	GAS USED		1 TRANSACTION	
34	2019-11-24 22:14:24	139763			

**Figura 3. Blockchain Ganache.**

e permite se conectar com a rede Ethereum com a biblioteca web3.js (Biblioteca que se conecta via HTTP, executa toda a ação de envio, leitura, gravação referente aos contratos inteligentes).

Não foi feito nenhum projeto ou análise de sistema, o único requisito era poder armazenar o valor de kwh usado por medidores de energia bidirecionais remotos usados por concessionárias de energia, seguindo os padrões da norma NBR14522 (ABNT, 2008). Os medidores com comunicação remota, podem se conectar via API (Código que permite dois sistemas se comunicarem) em uma rede blockchain. Os testes foram realizados enviando as informações fictícias de produção de energia através de um contrato inteligente que interage com uma das 8 contas em uso na extensão do navegador Google Chrome Metamask, tudo gerenciado pelo Metamask (EBEL, 2021) como pode ser verificado na Figura 4, com contas criadas com ethers fictícios.



**Figura 4. Gerenciamento das contas na extensão MetaMask, nesse exemplo são mostradas 3 contas, podendo fazer trocas entre elas, criar ou até adicionar contas em uso.**

A criação do ambiente de implementação foi realizada em um laptop Acer Aspire E1-571, utilizando o sistema operacional Windows 10. Nesse setup o processamento do programa Ganache, não ultrapassou os 30 por cento de uso do cpu dual-core de 1.3ghz da máquina de teste, não impactando na sua utilização, conforme a Figura 5 abaixo, mostrando um bom desempenho:

Nome	Status	47% CPU	56% Memória	0% Disco	0% Rede
Ganache (5)		29,0%	400,4 MB	0,1 MB/s	0,1 Mbps
Gerenciador de Janelas da Área ...		1,8%	19,7 MB	0 MB/s	0 Mbps
Antimalware Service Executable		1,3%	166,1 MB	0,1 MB/s	0 Mbps
Opera Internet Browser (12)		0,8%	1.065,9 MB	0 MB/s	0 Mbps
Gerenciador de Tarefas		0,7%	22,9 MB	0 MB/s	0 Mbps

**Figura 5. Ganache em uso no gerenciador de tarefas.**

Para implementar o código foi utilizado o Visual Studio Code, que disponibiliza uma extensão específica para a linguagem: Ethereum Solidity Language for Visual Studio Code, que consegue ler a sintaxe e mostrar o código com destaque, facilitando o uso. O projeto é iniciado pelo Truffle framework, na fase de desenvolvimento, em que deve-se informar o endereço da rede (Mostrada na Figura 6), que é configurada para o uso no front-end (A parte do sistema que o usuário interage). Logo após instalar o ambiente, é usada a rede no Ganache.

```

4  module.exports = {
5    networks: {
6      ganache: {
7        host: "localhost",
8        port: 7545,
9        network_id: "*", // Padroniza para qualquer id de rede
10       gas: 4700000
11      }
12    },
13  };

```

**Figura 6. Configuração do endereço de teste rede blockchain.**

O contrato inteligente foi escrito na linguagem Solidity, e compilado para a rede da Ethereum Virtual Machine (EVM), em alto nível (EBEL, 2021), o contrato gerencia as transações de energia. O contrato possui variáveis de identificação, nome, kwh e descrição, também possui funções para compra, venda e consulta do que está disponível para venda, conforme a Figura 7 e figura posterior, a Figura 8.

#### 4.2. Funcionamento da aplicação

Primeiramente, para inserir os dados na rede o usuário deve selecionar uma das contas em uso no Metamask, conforme a Figura 9. A conta selecionada, deve conter o saldo no token Ether. Nessa tela também é exibida as informações das interações dos contratos.

Em seguida, para efetuar a venda da quantidade de energia, o usuário deve inserir os dados na interface do protótipo, conforme a Figura 10, que também mostra a interação do contrato com a quantidade de ether e a taxa gas fee na account 2 com o hash do endereço.

Por fim, as informações enviadas à página no front-end vão para o arquivo web3.js, que é transcrito para o formato .sol (Da linguagem de programação Solidity). Se há alguma interação, o resultado binário do contrato é compilado para o nó da rede Ethereum.

```

ChainList.sol
contract NomeDoContrato []
    struct venda {
        uint256 id;
        address vendedor;
        address comprador;
        string kwh;
        string descricao;
        uint256 preco;
    }

    function venderEnergia(
        string memory _kwh,
        string memory _descricao,
        uint256 _preco
    ) public {
        contadorVenda++;
        venda[contadorVenda] = venda(
            contadorVenda,
            msg.sender,
            address(0),
            _kwh,
            _descricao,
            _preco
        );
        emit LogvenderEnergia(contadorVenda, msg.sender, _kwh, _preco);
    }
}

```

Figura 7. Declaração das variáveis do ambiente e a função de venda.

```

function comprarEnergia(uint256 _id) public pagador {
    require(venda.preco == msg.value, "Não há saldo suficiente."); //
    venda.vendedor.transfer(msg.value); // o comprador consegue compra
    emit LogcomprarEnergia( // trigger
        _id,
        venda.vendedor,
        venda.comprador,
        venda.kwh,
        venda.preco
    );
}

//consultar kwh disponiveis para venda
function soConsultarSaldo() public view returns (uint256[] memory) {
    if (vendaCounter == 0) {
        return new uint256[](0);
    }
    uint256[] memory vendaIds = new uint256[](vendaCounter);
    uint256 consultaSaldo = 0;
    for (uint256 i = 1; i <= vendaCounter; i++) {
        //mantem o id de uma conta que tenha kwh
        if (vendas[i].comprador == address(0)) {
            vendaIds[consultaSaldo] = vendas[i].id;
            consultaSaldo++;
        }
    }
    uint256[] memory paraVenda = new uint256[](consultaSaldo);
    for (uint256 j = 0; j < consultaSaldo; j++) {
        paraVenda[j] = vendaIds[j];
    }
    return paraVenda;
}

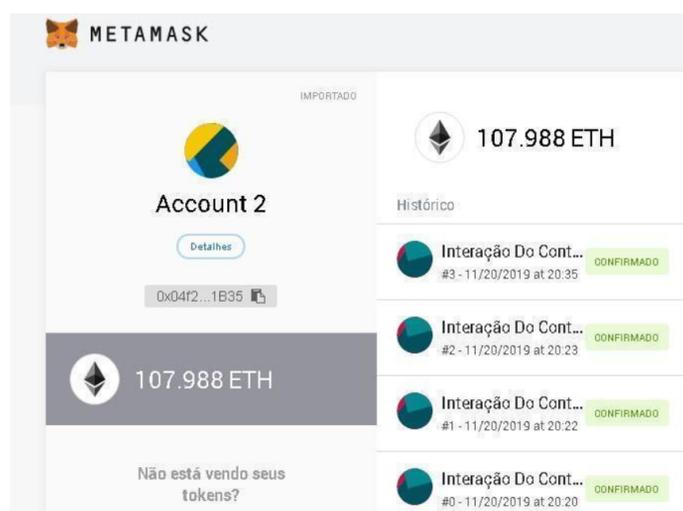
```

Figura 8. Função de compra e de consulta.

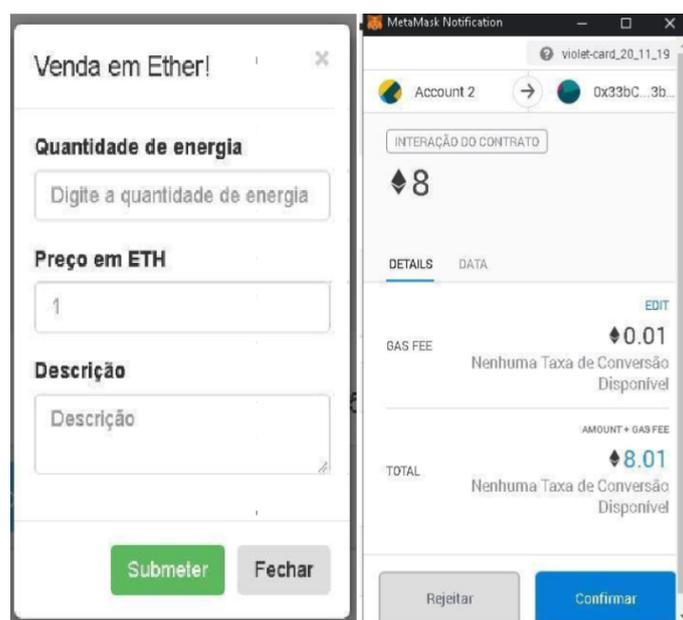
Após o contrato ser compilado, pode ser chamado cada vez que for necessário para verificar as informações ou modificar o contrato na blockchain no nó privado de uso local da blockchain Ethereum. o Ganache (Nó local) exibe todos os logs das transações, conforme a figura, mostrando os hashes das contas de origem e destino, valor enviado e taxa, conforme verificado na Figura 11.

## 5. Contribuição e conclusão

O trabalho desenvolvido constatou que a blockchain traz a vantagem de não depender de uma terceira parte para mediar a relação de consumo e produção no setor de energia, devido às análises de dificuldades encontradas nas regulações deste setor, acrescentando que ao utilizar essa tecnologia avaliada auxilia nos registros das informações de dados de produção de energia, mostrando no ambiente de testes do Ganache todos os logs necessários.



**Figura 9. Informações de saldo e interações da Conta 2 e registro das interações que a conta já teve no Metamask.**



**Figura 10. Tela da interface do protótipo com a quantidade de energia a ser vendida, o preço (Em ETH) e uma descrição e a interação do contrato.**

Avaliando que em geral, os produtores podem registrar seus dados na rede e assim ter meios para comercializarem, causando um estímulo para a produção de energia e os recursos derivados, mas como a maioria dos produtores utiliza o sistema regulado, percebe-se a falta de incentivo, o que faz com que muitas pessoas possam perder o interesse, mesmo com o barateamento do hardware de produção de energia, o que limita perfis diferenciados dos produtores.

Diante da implementação e do uso das ferramentas da blockchain Ethereum, ficou evidente que os objetivos foram alcançados. A interface front-end que utilizou o navegador para ser acessível, possibilitando a visualização das ofertas e a troca entre contas simulando a compra e venda e o registro em contratos e também a verificação através dos logs.

TX HASH		CONTRACT CALL	
<b>0x3edd40b402bb048d6b8581656820b31d405076fefac50e</b>			
FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0x04f2E4b63E9339cE57DFF7	0x33bC60a26fEAD0B1259a584bD6	55124	800000000000000000
TX HASH		CONTRACT CALL	
<b>0x397fd0dc8aa4eaf10358e803a0c9cfd34a618e7d74436</b>			
FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0x4B86549e4Df68E05475FFF	0x33bC60a26fEAD0B1259a584bD6	139827	0
TX HASH		CONTRACT CALL	
<b>0x1f65748cf04f8362f09ef137ef1ec2cf27710f2c49983f</b>			
FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0x4B86549e4Df68E05475FFF	0x33bC60a26fEAD0B1259a584bD6	55124	800000000000000000
TX HASH		CONTRACT CALL	
<b>0x32375bb3ac5058024541d427f31e585fb84f190527c5b9</b>			
FROM ADDRESS	TO CONTRACT ADDRESS	GAS USED	VALUE
0x04f2E4b63E9339cE57DFF7	0x33bC60a26fEAD0B1259a584bD6	139763	0

**Figura 11. Log de transações do Ganache.**

A atividade desenvolvida de registro em um contrato inteligente simulando compra e venda no protótipo têm potencial para fornecer um ambiente de negócios seguro e simples, permitindo entender visualizando o conceito das transações e os registros feitos na blockchain da rede Ganache.

Com esse tema de importância, é necessário o desenvolvimento de um protótipo de sistema que seja de registro mensal das informações (Padrão das concessionárias), se a produção for mais de 300 kwh/mês, normalmente um consumo de uma casa de três ou quatro pessoas, podendo atender um grande grupo de usuários que se encaixa nessa faixa, também mediando a interação entre várias casas em um condomínio e em potencial, se a base geradora registrada tiver somada a quantidade de kwh suficiente, reclassificar sua modalidade na ANEEL.

O uso da blockchain foi compreendida e tem potencial para registro de dados de produção e consumo de energia, permitindo visualizar a mediação entre o processo de compra e venda e registro de contrato inteligente de uma forma mais direta sem depender de terceiros, motivando a investir nessa tecnologia, fazendo com que a matriz energética nacional possa ser mais diversificada e que a soma comprovada da produção de energia de várias casas, possa mudar a classificação de seus produtores.

Dada à importância do tema, torna-se necessário o desenvolvimento de projetos futuros que possam desencadear competências e habilidades de um sistema para blockchain, que atendam às diferentes necessidades dos seus utilizadores e, assim, efetivar uma prática de programação diferenciada. Projetos futuros, indicar que o próximo estagiário continue trabalhando e faça o resto.

## Referências

ABNT. *NBR14522: Intercâmbio de informações para sistemas de medição de energia*

- elétrica*. [S.l.], 2008. Disponível em: <<https://www.abnt.org.br/>>.
- ALVES, M. de O. L. *Estudo da geração de energia elétrica através dos sistemas fotovoltaicos on-grid e off-grid*. [S.l.], 2019. Disponível em: <<https://www.ufop.br/>>.
- ANDONIA, e. a. M. *Blockchain technology in the energy sector: A systematic review of challenges and opportunities*. [S.l.], 2018. Disponível em: <<https://www.sciencedirect.com/>>.
- ANDROULAKI, e. a. E. *Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains*. [S.l.], 2018. Disponível em: <<https://arxiv.org/>>.
- ANEEL. *Micro e Minigeração Distribuída Sistema de Compensação de Energia Elétrica REM 587:2015*. [S.l.], 2015. Disponível em: <<https://www2.aneel.gov.br/cedoc/ren2015687.pdf>>.
- BLUMMER, e. a. T. *An Introduction to Hyperledger Creative Commons*. [S.l.], 2018. Disponível em: <<https://www.hyperledger.org/>>.
- CROSBY, e. a. M. *BlockChain Technology - Beyond Bitcoin*. [S.l.], 2018. Disponível em: <<https://scet.berkeley.edu/>>.
- DANNEN, C. *Solidity Programming. In Introducing Ethereum and Solidity*. [S.l.], 2017. Disponível em: <<http://the-eye.eu/>>.
- EBEL, J. *Augmenting the MetaMask-Wallet with Domain Name Based Authentication of Ethereum Accounts*. [S.l.], 2021. Disponível em: <<https://www.matthes.in.tum.de/>>.
- EPE. *Anuário Estatístico de Energia Elétrica*. [S.l.], 2017. Disponível em: <<http://www.aneel.gov.br/>>.
- GARRIGA MAXMILIANO ARIAS, A. D. R. M. *Blockchain and Cryptocurrency: A comparative framework of the main Architectural Drivers*. [S.l.], 2018. Disponível em: <<https://www.arxiv.org/>>.
- GOULART, B. B. *Aplicação da tecnologia blockchain no armazenamento de documentos*. [S.l.], 2021. Disponível em: <<http://repositorio.unesc.net/>>.
- HEGEDŪS, P. *Towards Analyzing the Complexity Landscape of Solidity Based Ethereum Smart Contracts*. [S.l.], 2019. Disponível em: <<https://www.mdpi.com/>>.
- HUKKINEN, e. a. T. *Skimping on Gas – Reducing Ethereum Transaction Costs in a Blockchain Electricity Market Application*. [S.l.], 2019.
- MANEVICH ARTEM BARGER, Y. T. Y. *Endorsement in Hyperledger Fabric via Service Discovery*. [S.l.], 2019. Disponível em: <<https://www.researchgate.net/>>.
- MARTINS, H. H. de S. *Metodologia qualitativa de pesquisa*. [S.l.], 2004. Disponível em: <<https://www.scielo.br/>>.
- MELO, e. a. Lucas Santos de. *Impactos Sociais da Geração Distribuída Fotovoltaica no Programa Minha Casa Minha Vida em Juazeiro/BA*. [S.l.], 2018. Disponível em: <<https://anaiscbens.emnuvens.com.br/>>.
- MODI, R. *Solidity Programming Essentials*. [S.l.], 2018. Disponível em: <<https://www.mdpi.com/>>.

MUKHOPADHYAY, M. *Ethereum Smart Contract Development*. [S.l.], 2018. Disponível em: <<http://www.davidpascal.com/>>.

NAKAMOTO, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. [S.l.], 2008. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>.

PORAT, e. a. A. *Blockchain Consensus: An analysis of Proof-of-Work and its applications*. [S.l.], 2017. Disponível em: <<http://www.scs.stanford.edu/>>.

SIQUEIRA, L. M. de. *Estudo do Dimensionamento e da Viabilidade Econômica de Microgerador Solar Fotovoltaico Conectado à Rede Elétrica*. [S.l.], 2015. Disponível em: <<http://www.ufjf.br/>>.

SWAN, M. *Blueprint for a New Economy*. [S.l.], 2015. Disponível em: <<https://books.google.com.br/>>.

WANG, e. a. N. *When Energy Trading Meets Blockchain in Electrical Power System: The State of the Art*. [S.l.], 2019. Disponível em: <<https://www.mdpi.com/>>.

WU, N. K. T. J. *Application of Blockchain Technology in Sustainable Energy Systems: An Overview – Sustainability*. [S.l.], 2018. Disponível em: <<https://www.mdpi.com/>>.

## Anexo I: Códigos desenvolvidos

```

1 //Contrato dono
2 pragma solidity >0.4.99 <0.6.0; //vers o do Solidity
3 contract Dono {
4     // estado da variavel
5     address pagavel dono;
6     // modificador
7     modifier apenasDono() {
8         require(msg.sender == Dono, "Essa fun o s pode ser chamada
9             pelo dono do contrato.");
10    }
11    // construtor
12    constructor() public {
13        Dono = msg.sender;
14    }
15 //Contrato das migra es
16 pragma solidity >0.4.99 <0.6.0; //vers o do Solidity
17 contract Migracoes {
18     address public Dono;
19     uint public ultima_migracao_completa; //carrega a ultima migra o
20     //completada
21     modifier restricted() { //modificador restrito somente ao dono
22         if (msg.sender == Dono) _;
23     }
24     constructor() public { //
25         Dono = msg.sender;
26     }
27     function setcompletada(uint migracao_completada) public restricted {
28         ultima_migracao_completa = migracao_completada;
29     }
30     function atualizar(address new_address) public restricted {
31         Migracoes atualizar = Migracoes(new_address);

```

```

30     atualizar.setcompletada(ultima_migracao_completa);
31     }
32 }
33
34 //Contrato Dono
35 pragma solidity >0.4.99 <0.6.0;
36 import "../Dono.sol";
37 contract ChainList is Dono {
38     struct energia {
39         uint id;
40         address pagavel vendedor;
41         address comprador;
42         string nome;
43         string descricao;
44         uint256 preco;
45     }
46     // Variaveis de estado
47     mapping(uint => energia) public Energia;
48     uint ContadorderMedicoes;//
49     address vendedor;//endere o do vendedor
50     address comprador;//endere o do comprador
51     string nome;//nome
52     string descricao;//descricao
53     uint256 preco;//preco
54     // Eventos de log compra e venda
55     event Logvenderenergia (
56         uint indexed _id,
57         address indexed _vendedor,
58         string _nome,
59         uint256 _preco);
60     event Logcomprarenergia (
61         uint indexed _id,
62         address indexed _vendedor,
63         address indexed _comprador,
64         string _nome,
65         uint256 _preco);
66     // destruir contrato
67     function destruir() public apenasDono {
68         selfdestruct(Dono);
69     }
70     // vender an energia
71     function venderenergia(string memory _nome, string memory
72         _descricao, uint256 _preco) public {
73         //contador de energia
74         ContadorderMedicoes++;
75         // armazenar as medicoes
76         Energia[ContadorderMedicoes] = energia(
77             ContadorderMedicoes,
78             msg.sender,
79             address(0),
80             _nome,
81             _descricao,
82             _preco
83         );
84         // gatilho do evento

```

```

84     emit Logvenderenergia(ContadorderMedicoes, msg.sender, _nome,
85         _preco);
86 }
87 // comprar an energia
88 function comprarenergia(uint _id) public pagavel {
89     // verifica se ha uma quantidade de energia
90     require(ContadorderMedicoes > 0, "Deve haver pelo menos alguma
91         energia.");
92     // verifica se a energia existe
93     require(_id > 0 && _id <= ContadorderMedicoes, "Energia com
94         est identifica o n o existe.");
95     // pega o id da medicao de energia
96     energia storage energia = Energia[_id];
97     // verifica se a energia j foi vendida
98     require(energia.comprador == address(0), "Energia j foi
99         vendida.");
100    // bloqueia o vendedor de comprar sua propria energia
101    require(energia.vendedor != msg.sender, "Vendedor n o pode
102        comprar sua pr pria energia.");
103    //verifica se h saldo de energia
104    require(energia.preco == msg.value, "N o h saldo de energia
105        suficiente.");
106    // mantem a informa o do comprador
107    energia.comprador = msg.sender;
108    // o comprador pode comprar a energia
109    energia.vendedor.transfer(msg.value);
110    // gatilho para o evento
111    emit Logcomprarenergia(_id, energia.vendedor, energia.comprador
112        , energia.nome, energia.preco);
113 }
114 // pesquisa o n mero de energia no contrato
115 function getNumerodeEnergia() public view returns (uint) {
116     return ContadorderMedicoes;
117 }
118 // encontra e retorna a quantidade de energia disponivel para venda
119 function getEnergiaparaVenda() public view returns (uint[]memory) {
120     // verifica se h pelo menos uma medida de energia
121     if(ContadorderMedicoes == 0) {
122         return new uint[](0);
123     }
124     // prepara a saida dos vetores
125     uint[] memory energiaIds = new uint[](ContadorderMedicoes);
126     uint NumerodeEnergiaparaVenda = 0;
127     // conta quantas medidas da energia foram feitas
128     for (uint i = 1; i <= ContadorderMedicoes; i++) {
129         // mantem o id da energia ainda n o vendida
130         if (Energia[i].comprador == address(0)) {
131             energiaIds[NumerodeEnergiaparaVenda] = Energia[i].id;
132             NumerodeEnergiaparaVenda++;
133         }
134     }
135     //copia o id de energia do vetor para o outro vetor paraVenda
136     uint[] memory paraVenda = new uint[](NumerodeEnergiaparaVenda);
137     for (uint j = 0; j < NumerodeEnergiaparaVenda; j++) {
138         paraVenda[j] = energiaIds[j];
139     }

```

```
133     return paraVenda;  
134 }  
135 }
```

**Listing 1. Código implementado**