



UNIVERSIDADE D
COIMBRA

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Eletrotécnica e de Computadores

**GESTÃO LOCAL DE ENERGIA BASEADA
EM SMART CONTRACTS E BLOCKCHAIN**

Diogo Miguel Pardal Monteiro

Mestrado em Engenharia Eletrotécnica e de Computadores

Coimbra, fevereiro de 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

GESTÃO LOCAL DE ENERGIA BASEADA EM SMART CONTRACTS E BLOCKCHAIN

Presidente:

Prof. Doutor Carlos Henggeler

Orientador:

Prof. Doutor Álvaro Gomes

Vogal:

Prof. Doutor Pedro Moura

Mestrado em Engenharia Eletrotécnica e de Computadores

Coimbra, fevereiro de 2023

ÍNDICE

Lista de Figuras	vii
Lista de Acrónimos	ix
Resumo	xi
Abstract	xii
1. Introdução	14
2. Estado da Arte	17
2.1 Blockchain e Smart Contracts	19
2.1.2 Definição e revisão da tecnologia <i>Blockchain</i>	19
2.2 Smart Contracts	24
2.2.1 Relação entre blockchain e smart contracts	24
2.2.2 Definição e ciclo de vida de um contrato inteligente	25
2.2.3 Áreas de aplicação dos smart contracts e da blockchain no setor energético	27
2.3.4 Outras aplicações	32
2.3.5 Projetos académicos e industriais	34
3. Descrição do problema	38
3.1 HEMS convencional	38
3.2 HEMS Utilizando <i>Smart Contracts</i> e <i>Blockchain</i>	39
3.2.1 Segurança	41
3.2.2 Custos	42
3.2.3 Resiliência e Fiabilidade	43
3.2.4 Facilidade de implementação	43
4. Procedimentos e Implementação	45
4.1 Arquitetura	45
4.1.1 Escolha das plataformas a utilizar	45
4.2 Simulação	54
5. Análise de Resultados	61
5.1 Cyber-segurança e privacidade	63
5.2 Riscos de implementação e comunicação	63
5.3 Gastos computacionais dos smart contracts	64
5.4 Requisitos de software	64
5.5 Regulação	65
6. Conclusão e trabalhos futuros	67
Bibliografia	71
Apêndices	77

Lista de Figuras

1. *Diferenças entre sistemas centralizados e descentralizados*
2. *Mecanismo de hashing*
3. *Cadeia de blocos representativa da blockchain*
4. *Etapas do smart contract*
5. *Exemplo de uma transação de energia peer-to-peer*
6. *Sistema HEMS convencional*
7. *Estrutura de um sistema de gestão de uma casa inteligente utilizando smart contracts e blockchain*
8. *Sistema HEMS utilizando smart contracts e blockchain*
9. *Exemplo de um smart contract implementado na plataforma Remix IDE*
10. *Exemplo do pagamento de gas fees de uma transação com a rede de teste Goerli na carteira digital Metamask*
11. *Plataforma Goerli Faucet*
12. *Carteira digital metamask conectada à mainnet da Ethereum*
13. *Alteração da rede na carteira digital metamask*
14. *Plataforma metamask na rede de teste Goerli Faucet*
15. *Demonstração do contrato "ownable"*
16. *Demonstração da utilização de mappings*
17. *Interface de interação com o smart contract na plataforma Remix IDE*
18. *Representação dos consumos, da produção e da bateria no cenário sem controlo*
19. *Representação dos consumos, da produção e da bateria no cenário com controlo*
1. *Tabela: Características das blockchains públicas e privadas*
2. *Tabela: Caracterização dos equipamentos utilizados*

Lista de Acrónimos

BC – Blockchain

DAG – Direct Acyclic Graph

DLT – Distributed Ledger Technology

ETH – Ethereum

HEMS – Home Energy Management System

IoT – Internet Of Things

P2P – Peer-to-peer

PoA – Proof-of-authority

PoS – Proof-of-stake

PoW – Proof-of-work

SC – Smart Contracts

SOC – State of charge

VE's – Veículos elétricos

Resumo

O presente trabalho visa perceber de que forma os *smart contracts* a par com a tecnologia *blockchain* podem ou não ter um impacto positivo na gestão local de recursos. Tratando-se de tecnologias disruptivas que permitem fazer o registo de dados e transações de uma forma segura, imutável e transparente, as mesmas podem ser uma mais valia no setor energético.

Com o intuito de perceber os prós e os contras destas tecnologias, foi implementado um *smart contract* na *blockchain* da *Ethereum* com o objetivo de simular um cenário de gestão/transação de energia dentro de uma casa de um utilizador residencial. O cenário simulado utilizou diversos equipamentos como máquinas de lavar roupa, máquinas de lavar loiça, painéis solares e baterias. Foi implementado um algoritmo de gestão de forma a minimizar os custos da fatura a pagar pelo utilizador no final de cada dia. O algoritmo implementado conseguiu obter uma diminuição na fatura de cerca de 20% face ao preço da fatura no cenário sem controlo. Este valor apesar de estar dentro do range de diminuição de custos dos sistemas de gestão convencionais (HEMS), ainda pode ser bastante otimizado através da implementação de algoritmos e técnicas mais complexas como por exemplo a utilização de *machine learning*.

É de salientar que apesar da diminuição no preço da fatura de 20%, o *smart contract* exigiu o pagamento de taxas de utilização muito altas devido aos congestionamentos da rede *Ethereum* e a ainda à forma como o código foi escrito. Como tal, é crucial que em implementações futuras se tenha bastante atenção à forma como o *smart contract* é escrito, quais são os dados e cálculos que devem ser efetivamente alocados na *blockchain* e quais podem ser utilizados *off-chain* de forma a se maximizar a eficiência do código e se possa tirar o máximo proveito destas tecnologias.

Palavras-chave: *Blockchain, smart contracts, HEMS, gestão local de recursos, gas fees*

Abstract

This work aims to understand how smart contracts together with blockchain technology can or cannot have a positive impact on local energy management. Being disruptive technologies that allow the registration of data and transactions in a secure, immutable and transparent way, they can be an asset in the energy sector. In order to understand the pros and cons of these technologies, a smart contract was implemented in the Ethereum blockchain with the objective of simulating a energy management/transaction scenario inside a residential user's house. The simulated scenario contained various equipment such as washing machines, dishwashers, solar panels and batteries. A management algorithm was implemented in order to minimize the costs of the bill to be paid by the user at the end of each day. The implemented algorithm managed to obtain a decrease in the bill of about 20% compared to the price of the bill in the scenario without control. This value, despite being within the range of cost reduction of conventional management systems (HEMS), can still be greatly optimized through the implementation of more complex algorithms and techniques, such as the use of machine learning.

It should be noted that despite the 20% decrease in bill price, the smart contract required the payment of very high gas fees due to Ethereum network congestion and the way the code was written. As such, it is crucial that in future implementations a lot of attention is paid to how the smart contract is written, which data and calculations should be effectively allocated on the blockchain and which can be used off-chain in order to maximise the efficiency of the code and get the most out of these technologies.

Keywords: Blockchain, smart contracts, HEMS, gas fees

Capítulo 1

Introdução

Em novembro de 2019, o Parlamento Europeu declarou um estado de emergência climática e ambiental, reforçando o objetivo comum de alcançar a neutralidade carbónica até 2050. O sector energético é a área mais poluente da atividade humana, representando cerca de 33% das emissões de gases com efeito de estufa na atmosfera [1]. As tecnologias de energias renováveis são a chave para reduzir as emissões. Segundo a agência internacional de energia, estima-se que a produção global a partir de energias renováveis triplique até 2030 e aumente oito vezes até 2050. O grande crescimento das energias limpas tem-se conseguido como resultado de políticas energéticas de apoio, incentivos económicos e mudanças no sector, assim como o estabelecimento de comunidades energéticas e micro redes [2]. A própria invasão russa na Ucrânia também tem sido um fator impulsionador do aumento deste tipo de energias [1]. No entanto, o maior volume de fontes de energia renováveis (FER) de pequena escala está a alterar a gestão e o funcionamento das redes elétricas nacionais e locais, uma vez que aumenta o risco de desequilíbrio do sistema. Além disso, os ativos de produção descentralizada em pequena escala não são atualmente coordenados pelo operador do sistema para equilibrar a rede e são frequentemente demasiado pequenos para participar nos mercados de energia ou de serviços auxiliares, a menos que a sua participação seja permitida por um agregador. Por conseguinte, o sistema requer novas técnicas de gestão e distribuição a nível local.

A tendência atual de descentralização e digitalização está a impulsionar a procura de soluções inovadoras que permitam a troca de dados e energia de forma fiável, eficiente e à prova de adulteração para aumentar o autoconsumo nas comunidades energéticas locais, e apoiar a implementação de soluções de controlo mais distribuídas. As casas inteligentes, equipadas com vários sensores e dispositivos, são capazes de gerir, armazenar e consumir a energia de uma forma eficiente e sustentável, o que pode ser uma ajuda na integração de renováveis em pequena escala. No entanto, os sistemas de gestão atuais são centralizados e carecem de transparência e segurança. De forma a tentar dar resposta a estes problemas, os *smart contracts* e a tecnologia *blockchain*, têm sido cada vez mais utilizados no setor energético. Os *smart contracts* em conjunto com a tecnologia *blockchain* apresentam uma oportunidade para solucionar estes problemas. Estas tecnologias fornecem um sistema descentralizado, seguro e transparente que permite coordenar as transações de energia. Os *smart contracts* são contratos que se executam de forma automática de acordo com as condições implementadas em código, com eles é possível automatizar a gestão da energia. A tecnologia *blockchain*, é um livro razão descentralizado e distribuído, que pode fornecer uma forma segura e transparente de registar e verificar estas transações de energia [4].

A utilização de *blockchain* e *smart contracts* na gestão local de energia oferece vários benefícios. Em primeiro lugar, devido à alta segurança proporcionada pela tecnologia *blockchain*, garante-se a integridade dos registos energéticos. Para além disso, os *smart contracts* permitem a automatização de processos de negociação e pagamento de energia, reduzindo os custos administrativos e aumentando a eficiência. Outra vantagem importante da gestão local baseada nestas tecnologias, é a capacidade das mesmas promoverem a descentralização da produção. Isto significa que as comunidades locais de energia podem produzir e controlar a sua própria energia, aumentando a segurança, a transparência e a resiliência em caso de falhas na rede elétrica. Além disso, estas tecnologias também podem contribuir para a promoção de fontes de energia renovável uma vez que permitem a integração de pequenas fontes de geração de energia, como painéis solares, na rede elétrica, o que é importante para a transição para uma economia de energia mais limpa e sustentável.

Posto isto, os objetivos desta tese são identificar o papel dos *smart contracts* e da tecnologia *blockchain* no setor energético e avaliar de que forma os mesmos podem ser utilizados para fazer a gestão/transação local de recursos num edifício. Para isto, será desenvolvido um *smart contract* na plataforma *Remix IDE* utilizando a *blockchain* da *Ethereum* que fará a simulação de um cenário de gestão local envolvendo diversos recursos. Dado que se vão utilizar tecnologias relativamente recentes e com um grau de desenvolvimento relativamente baixo, a implementação do *smart contract* terá maioritariamente um caráter exploratório onde se procura perceber quais os prós e contras deste tipo de implementação.

Capítulo 2

Estado da Arte

Neste capítulo são apresentados os principais conceitos relevantes para esta dissertação de mestrado. Inicialmente, esta revisão abrangerá os antecedentes teóricos das tecnologias utilizadas, *blockchain* e *smart contracts*. Serão discutidas definições centrais, soluções existentes, e conclusões relevantes para cada uma delas. Para além disso, serão apresentadas as principais aplicações da *Blockchain* (BC) e dos *Smart Contracts* (SC) no setor energético, assim como alguns exemplos de projetos académicos e industriais.

Distributed Ledger Technologies

As *distributed ledger technologies* (DLT's) são tecnologias distribuídas e descentralizadas de registo eletrónico de dados. Por outras palavras, uma DLT é um tipo de base de dados que está dispersa por uma rede de vários *nodes*, em que os *nodes* são os utilizadores da rede [fig.1]. Ao contrário de uma base de dados centralizada, numa DLT cada utilizador tem uma cópia do registo dos dados, o que torna o registo de informação seguro e transparente [5].

Esta tecnologia utiliza protocolos de acesso, validação e armazenamento através de redes de comunicação bilaterais de computadores distribuídos por várias localizações. Numa base de dados centralizada os dados são armazenados em um único computador (servidor), ao qual é necessário conectarmo-nos para ter acesso às informações. Numa base de dados descentralizada não há necessariamente uma única localização física onde os dados estejam concentrados, os mesmos estão distribuídos em diversos servidores conectados entre si [6].

Devido ao facto das DLT's permitirem o registo de transações/dados de uma forma segura e transparente, fazem com que a necessidade de elementos terciários (bancos, agências de certificação) no processamento e na utilização de dados seja reduzida ou eliminada. Algumas das principais DLT's existentes são as *DAG (Directed Acyclic Graph)*, as *Tangle*, e a *Blockchain* que irá ser abordada com mais detalhe na próxima secção.

Componentes principais

De acordo com [5], as DLT's contêm três elementos comuns. Em primeiro, a base de dados/registos distribuído, em segundo a rede *peer-to-peer* (P2P), e em terceiro o mecanismo de *consensus*.

- 1) A base de dados distribuída tem a função de armazenar todos os dados, transações e aplicações.

- 2) A rede P2P faz a manutenção da rede e partilha todas as transações que ocorrerem na rede com todos os *nodes*.
- 3) O mecanismo de *consensus* faz a gestão da inserção de transações na rede, verificando se pelo menos 51% dos *nodes* da rede está em concordância.

O conceito de um **base de dados distribuída** foi introduzido nos anos 90 quando alguns investigadores começaram a explorar o potencial dos sistemas descentralizados na gestão de dados e transações. No entanto, só em 2008, aquando da criação da *blockchain* da *Bitcoin* através de uma pessoa/entidade com o pseudónimo de Satoshi Nakamoto é que as DLT's se tornaram amplamente reconhecidas e adotadas. Como mencionado, numa DLT, cada *node* da rede contém uma cópia de toda a informação presente na base de dados, e todos os *nodes* em conjunto estabelecem um acordo/consenso (*consensus*) sobre o estado da informação contida na base de dados. Numa base de dados tradicional, a manutenção e validação dos registos é feita por uma entidade centralizada. Por exemplo, o livro razão na contabilidade de uma empresa é algo onde se guarda o registo das transações efetuadas nessa mesma empresa, no entanto, como a cópia desse registo está na posse de alguém, ou seja, os dados estão registados apenas num "*node*", os mesmos podem ser facilmente corrompidos. Utilizando uma base de dados distribuída, a cópia desse registo estará distribuída por milhares ou milhões de participantes sendo praticamente impossível de corromper, uma vez que se surgisse alguma alteração no registo, esta seria facilmente identificada. Em [7], efetuou-se uma análise entre bases de dados centralizadas e descentralizadas no sistema financeiro e concluiu-se que as DLT's vão provocar grandes alterações à banca e aos sistemas financeiros atuais, dado que a vantagem da descentralização supera o sistema centralizado atual.

Em arquitetura de software, o termo rede *peer-to-peer* designa uma arquitetura de rede distribuída onde os *nodes* da rede atuam em simultâneo como servidor e como cliente. Isto significa que cada *node* pode enviar e receber dados, sem a necessidade de um servidor central ou um intermediário, o que torna as redes P2P mais rápidas e eficientes [8]. Para além disso, permitir a comunicação direta e a partilha de recursos entre *nodes*, torna a rede mais resiliente, flexível e escalável [2]. Esta infraestrutura de rede pode ser benéfica na introdução de recursos renováveis nos sistemas energéticos, nomeadamente em relação aos pagamentos, à produção de eletricidade e aos mercados energéticos [9]. Já as redes com arquiteturas tradicionais, uma vez que agregam os dados num servidor central, têm um único ponto de falha, o que as torna mais suscetíveis a ciberataques.

De forma a se poder registar uma nova transação na base de dados distribuída, a mesma deve ser verificada pelos participantes da rede. A função destes participantes é provar a veracidade da informação que está a ser armazenada. Depois de se ter validado a informação, cada *node* da rede atualiza a sua base de dados, adicionando a nova transação inserida.

Este processo de validação é designado por *consensus*. Este estado só é alcançado quando pelo menos 51% dos *nodes* da rede estiverem de acordo com a veracidade da informação. Em [10] é referido que para uma DLT funcionar em larga escala, é necessário um algoritmo de *consensus* que seja eficiente, seguro e funcional, fornecendo resiliência a falhas nos nós, atrasos de mensagens, partições na rede e corrupção.

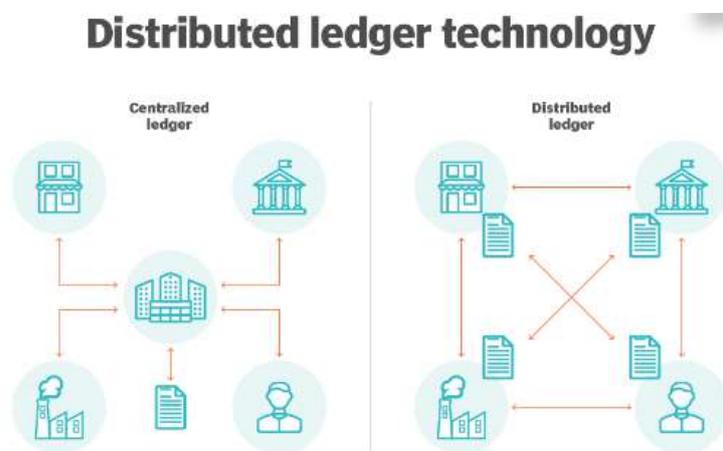


Figura 1 - Diferenças entre sistemas centralizados e descentralizados [11]

2.1 Blockchain e Smart Contracts

2.1.2 Definição e revisão da tecnologia *Blockchain*

A *blockchain* é um tipo de DLT que permite armazenar e registar transações de uma forma segura, transparente e imutável, o que faz com que seja o tipo de DLT mais conhecida e utilizada. A *blockchain* é essencialmente uma base de dados que contém uma série contínua de transações, ordenadas cronologicamente, armazenadas em blocos [12,13]. Através da figura 2, é possível visualizar uma imagem representativa do que é a estrutura *blockchain*. Os blocos da cadeia estão todos conectados entre si através de um mecanismo de *hashing* e cada bloco contém um certo número de transações que também são codificadas através deste mesmo mecanismo. O *hashing* é um processo matemático que permite transformar uma entrada de dados numa saída de tamanho fixo, chamada de '*hash*'. O *hash* é utilizado para garantir a integridade dos dados uma vez que a alteração na entrada dos mesmos resulta numa saída completamente diferente. Isto significa que o bloco mais recente a ser criado na cadeia da *blockchain* contém um *hash* que recebe o *input* de todos os blocos com todas as transações anteriores. Esta conexão de forma cronológica forma uma sequência ordenada de blocos, daí o nome "*Block* (Bloco) + *Chain* (Cadeia)". Todas as

transações que são registadas na cadeia de blocos são visíveis a todos os participantes da rede, é isto que a torna transparente.

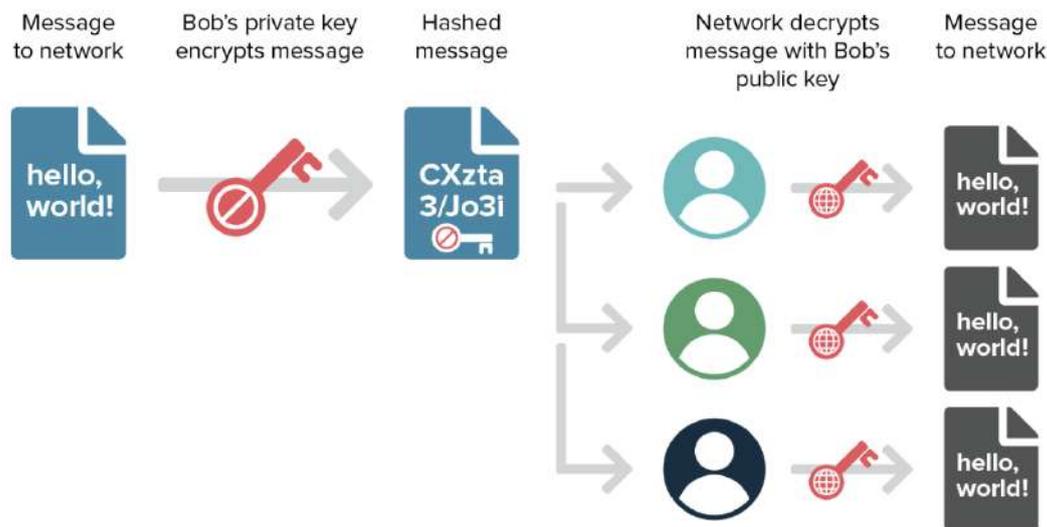


Figura 2 - Mecanismo de hashing [83]

As DLT's como a *blockchain* criam uma rede distribuída de *nodes* que trabalham em parceria para verificar as transações e proteger a integridade da base de dados. Numa *blockchain* 'convencional' (*blockchains* públicas), nenhuma entidade tem o controlo sobre a escrita da informação, são os próprios *nodes* que se governam a si próprios, criando um sistema mais democrático. Cada *node* contém uma cópia dos dados presentes na cadeia de blocos e, em conjunto com os restantes *nodes*, concordam sobre a informação a ser adicionada à *blockchain*, num processo chamado *consensus*. Ou seja, sempre que uma transação quer dar entrada na cadeia de blocos, tem de ser aprovada pelos *nodes* da rede. Por exemplo, no caso de uma troca *peer-to-peer*, em que a pessoa 'A' quer comprar a energia produzida pela pessoa 'B', como os *nodes* da cadeia têm acesso às transações efetuadas por cada participante, é possível comprovar que a pessoa 'A' efetuou o pagamento e recebeu a energia, e a pessoa 'B' recebeu o dinheiro e enviou a energia. Se pelo menos 51% dos *nodes* da rede estiverem de acordo que as trocas ocorreram conforme planeado, a transação é registada na *blockchain* permanecendo lá para sempre sem poder ser alterada [14].

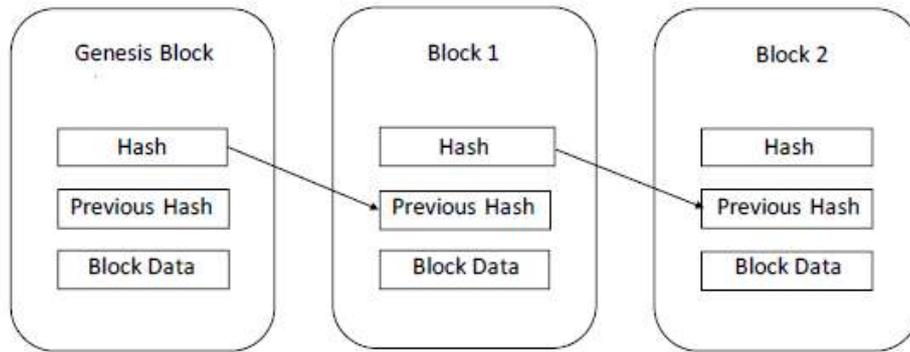


Figura 3 - Cadeia de blocos representativa da blockchain

Uma questão-chave em sistemas blockchain é o método de alcançar o *consensus* entre os nós. Existem muitas variantes de consenso implementados ou propostos [15,16], sendo os principais:

1. **Proof of Work (PoW):** Esta é a forma de *consensus* mais utilizada. No *consensus* PoW, o *node* que tem direito a adicionar o próximo bloco à cadeia é determinado pela resolução de um puzzle criptográfico, ou seja, um puzzle que é extremamente difícil de resolver, mas fácil de verificar. Adicionar um novo bloco é frequentemente referido como "mineração", e os *nodes* que realizam esta atividade são conhecidos como mineradores (*miners*), que são recompensados com uma certa quantidade de moeda criptográfica nativa ou subunidade da mesma (cada *blockchain* tem uma moeda nativa que é utilizada para fazer pagamentos dentro da rede) por cada novo bloco que extraem com sucesso. O puzzle criptográfico é necessário para garantir que os *nodes* da rede têm gastos, neste caso gastos energéticos, ao participar na rede. Isto porque, caso contrário, haveria uma enorme facilidade de sabotar a rede com *nodes* pertencentes a piratas informáticos. Como tal, a resolução do puzzle exige a utilização de hardware especializado que gasta uma quantidade de energia bastante grande.

Esta grande quantidade de energia necessária para efetuar cálculos de PoW é, popularmente, uma das características mais conhecidas e marcantes das cadeias de blocos PoW, uma vez que excede atualmente o consumo de energia de vários países. A sustentabilidade do elevado consumo de energia tem sido questionada, com a maioria das *mining pools* (conjuntos grandes de hardware dedicado à mineração) a serem estabelecidas em locais com energia muito barata. Embora isto aconteça frequentemente em áreas com excesso de geração a partir de energias renováveis, em muitos casos utilizam-se fontes de energia barata em alguns países/regiões (que são frequentemente baseadas no carvão ou outros combustíveis fósseis) [17].

2. ***Proof of Stake (PoS)***: Ao contrário do *proof-of-work*, que requer que os mineradores resolvam problemas matemáticos complexos para poderem adicionar blocos à rede, o *proof-of-stake* é um mecanismo de *consensus* que seleciona validadores de forma semi-aleatória para validar as transações. Esta seleção é considerada semi-aleatória uma vez que esta seleção é baseada no número de moedas nativas da *blockchain* em causa que os mesmos possuem [17]. Neste mecanismo, os validadores são incentivados a agir de forma correta, uma vez que podem perder as moedas que possuem caso os mesmos participem em atividades ilícitas. Como referido, a probabilidade de um validador ser selecionado está diretamente relacionada com o número de moedas que o mesmo possua. Depois de haver a seleção da entidade a validar o bloco, o mesmo é recompensado com mais moedas. As vantagens principais deste mecanismo são o facto de ser muito mais eficiente em termos energéticos, mais rápido, e mais escalável. No entanto, a segurança é reduzida em comparação com o mecanismo *proof-of-work*.

3. ***Proof of Authority (PoA)***: Este mecanismo de *consensus* pode ser encarado como uma variante do protocolo de *proof of stake*, em que a 'stake' é a identidade do validador. O PoA baseia-se num número (relativamente pequeno) de contas de validadores pré-approvadas ou "autoridades", que têm o direito de validar transações e adicionar novos blocos. Os *nodes* de autoridade são obrigados a passar por um processo de pré-seleção, revelar a sua identidade e registar-se numa base de dados pública e cumprir uma série de regras para permanecerem dignos de confiança. Uma vez que são recompensados por o fazerem e têm um incentivo para permanecer dignos de confiança, e evitarem ser comprometidos por ataques. Os protocolos PoA revelaram-se especialmente populares em blockchains privadas (empresariais), incluindo aplicações de energia (por exemplo, o sistema de cadeias de bloqueio Energy Web Foundation). Isto deve-se à elevada taxa de transações que é possível alcançar em sistemas baseados em PoA, e a custos gerais e de energia muito mais baixos do que, por exemplo, os sistemas PoW. No entanto, ter um pequeno número de nós de autoridade pode ser visto como indo contra os princípios da descentralização subjacentes às blockchains, pelo que esta é uma alternativa menos adequada para as blockchains públicas [17].

4. Outros protocolos: Foram propostos outros protocolos como: *proof of elapsed time*, *proof of activity*. Pode-se consultar em [18] ou em [19] discussões detalhadas.

Para diferentes tipos de aplicações, existem diferentes tipos de *blockchains*. As principais são:

A. *Blockchains* sem permissão ou *blockchains* públicas

Blockchains sem permissão ou públicas, são *blockchains* que operam sem a necessidade de uma entidade central ou um intermediário para validar as transações. Neste tipo de *blockchains*, qualquer participante se pode juntar à rede como *node* e validar transações, assegurando a integridade e segurança da rede. *Blockchains* públicas como a *Bitcoin* e a *Ethereum* são exemplo de *blockchains* onde os participantes podem enviar e receber transações de forma direta sem qualquer tipo de interveniente [14,17].

B. *Blockchains* de autorização ou *blockchains* privadas

Este tipo de rede também é conhecido como *blockchain* privadas ou empresariais. Neste tipo de *blockchain* nem todas as partes podem aderir, existe uma autoridade central que concede acesso de acordo com regras pré-acordadas. As *blockchains* privadas são utilizadas por exemplo num esquema de comércio entre prosumer, micro-redes, entidades que comercializam energia mediante um dado protocolo, etc. Estas *blockchains* podem por vezes ser criticadas por não seguirem o que alguns autores veem como o “verdadeiramente descentralizado”. No entanto, vale a pena salientar que este tipo de *blockchain* é ainda muito diferente das bases de dados centralizadas na medida em que existe um processo de verificação e concessão de acesso e a informação é ainda armazenada e escrita de forma descentralizada entre *nodes* num livro-razão distribuído.

	<i>Blockchain</i> Pública	<i>Blockchain</i> Privada
Controlo de acesso	Sem permissão – Toda a gente pode aceder	Autorização – Apenas utilizadores identificados podem aceder
Segurança	Todas as transações validadas são públicas	Apenas entidades autorizadas têm acesso à informação
Velocidade	Velocidade de transações lentas devido ao mecanismo global de consenso	Velocidade de transações mais rápida
Mecanismo de <i>Consensus</i>	<i>Proof-of-Work, Proof-of-stake</i>	<i>Proof-of-authority</i>
Desafios	Confirmação dos blocos (seg – min)	Perda de descentralização

Tabela 1 - Características das *blockchains* públicas e privadas

Como é possível identificar na tabela apresentada em cima, existem evidentemente vantagens e desvantagens que podem ser destacadas para cada tipo. As *blockchains* públicas são descritas em algumas fontes como as únicas que são "verdadeiramente" descentralizadas, ou seja, é impossível para qualquer parte alterar a informação ou regras armazenadas, a menos que ganhem o controlo de 51% do poder computacional, o que é improvável num grande sistema como o *Bitcoin*. Embora alguns autores tenham levantado preocupações sobre a crescente concentração de 'mining pools', que são grupos de *miners* que juntam as suas capacidades computacionais para aumentar as probabilidades de validar um bloco e receber a recompensa correspondente. As *mining pools* levantam preocupações, uma vez que, se o poder computacional estiver agregado no mesmo sítio ou estiver a ser controlado por apenas uma pessoa/grupo de pessoas ou entidade/s, perde-se a descentralização [15].

Apesar das *blockchains* públicas garantirem uma segurança e transparência nas transações, estas não são tão escaláveis sobretudo devido ao tempo de confirmação dos blocos. No caso das *blockchains* privadas, apesar da segurança e transparência não ser tão efetiva, a escalabilidade e a velocidade são maiores [3].

O setor energético tem explorado a utilização de ambos os tipos de *blockchain* para diversas aplicações como por exemplo o rastreamento de energias renováveis, os créditos de carbono, a gestão de energia e o comércio de energia. As *blockchains* privadas (como por exemplo a *blockchain* da *Hyperledger Fabric*) são mais utilizadas no setor devido à sua capacidade de processar as transações de uma forma mais rápida, o que é crucial neste setor devido à sensibilidade dos dados. Por outro lado, as *blockchains* públicas têm sido exploradas devido ao facto das mesmas aumentarem a transparência, reduzirem a fraude e aumentarem a segurança.

2.2 Smart Contracts

2.2.1 Relação entre blockchain e smart contracts

A criação e o surgimento da primeira *blockchain* (*Bitcoin*) surgiu com o intuito de registar as transações da sua moeda nativa (*Bitcoin*). No entanto, com o surgimento da *blockchain* da *ethereum* passou a ser possível fazer-se o registo e o armazenamento de código de software. Ou seja, da mesma forma que na *blockchain* da *bitcoin* todas as trocas da moeda são registadas na rede, na *blockchain* da *ethereum*, é possível armazenar na *blockchain* código programável. Surgiram assim os *smart contracts*. Os *smart contracts* são código de software que são executados quando as condições estipuladas no contrato são atingidas. Todo o código implementado no *smart contract*, nomeadamente, o código compilado e as informações específicas, tais como a lista de funções a executar, são enviadas de uma carteira digital (*node*) para a *blockchain*. Este código e informação devem então ser incluídos num bloco que é acrescentado à base de dados através do

mecanismo de *consensus*, altura em que o código do contrato inteligente será executado para estabelecer o estado inicial do contrato inteligente. À semelhança das transações monetárias, o ‘*hashing*’ criptográfico protege o *smart contract* contra tentativas de alteração ou adulteração do mesmo.

2.2.2 Definição e ciclo de vida de um contrato inteligente

Os contratos inteligentes foram introduzidos pela primeira vez pelo cientista informático Nick Szabo em 1996, com a visão de utilizar código informático par automatizar contratos legais fazendo uso da criptografia para os tornar seguros e à prova de adulteração [20]. Szabo define os contratos inteligentes como "um conjunto de promessas, especificadas em formato digital, incluindo protocolos dentro dos quais as partes executam estas promessas" [21]. Quando implementados numa blockchain, os contratos inteligentes podem automaticamente alcançar e aplicar acordos que resultam num processo mais rápido e de custos reduzidos. Isto é especialmente benéfico para acordos/transações recorrentes onde não haja confiança entre as entidades e onde a troca comercial seja de baixo valor, como acontece com o comércio de energia P2P. O princípio central de um smart contract baseia-se na lógica “*if ... then*”, que requer a programação do resultado/ação desejada e das condições. Por exemplo, o resultado de um contrato inteligente pode ser uma ação como a descarga de uma bateria, enquanto que a condição para esta ser desencadeada pode ser o preço de exportação de eletricidade acima de um certo valor ou a transferência bem sucedida dos fundos necessários de um comprador.

Embora possa ser completamente automatizado, algumas partes do contrato inteligente podem ser programadas para ter ação e controlo humano. A figura 4 descreve concisamente o ciclo de vida de um contrato inteligente em quatro etapas fundamentais que são: (1) acordo entre as partes, (2) estabelecimento do contrato inteligente, (3) verificação do cumprimento dos critérios e (4) execução da transferência de valor (por exemplo, troca de dinheiro e energia). A etapa (3), a verificação dos critérios alcançados, proporciona uma nova vantagem para os sistemas energéticos e especialmente para os mercados locais de energia. Os contratos inteligentes podem permitir o comércio automático de energia entre entidades. Os dados dos contadores inteligentes podem ser utilizados para verificar as transações de energia e desencadear o processo de faturação de um contrato inteligente. Isto resultaria numa liquidação mais precisa e rápida, aumentando o benefício tanto para o comprador (consumidor) como para o vendedor (produtor). O principal objetivo dos contratos inteligentes é proporcionar transações mais seguras em comparação com os métodos tradicionais de contratação e diminuir os custos e o tempo de processamento e verificação. Embora estas características tornem os contratos inteligentes muito adequados para transações financeiras utilizando moedas criptográficas [23], a utilização de contratos inteligentes

no setor da energia ainda se encontra em fase de desenvolvimento, uma vez que existem várias preocupações relacionadas com segurança, privacidade, escalabilidade e faturação [24,25].



Figura 4 - Etapas do smart contract [3]

Principais características dos *smart contracts*

1. Auto-execução e automatização

Os *smart contracts* são pedaços de código que tomam decisões com base em inputs específicos. Este código é executado de forma automática num ambiente virtual que é partilhado pelos *nodes* da *blockchain* assim que as condições são verificadas.

2. À prova de adulteração

Os *smart contracts* são pedaços de software armazenados na *blockchain*. Como tal, herdam as propriedades da mesma, entre elas o facto da mesma ser imutável.

3. Fidedignidade

O facto de o código ser à prova de adulteração garante que não pode ser alterado por ninguém. Garante-se assim que o *smart contract* é original e corresponde exatamente ao que o seu criador fez.

4. Transparência e acessibilidade

Fazendo parte da *blockchain*, o *smart contract* é transparente e acessível a todos os membros da *blockchain*. Numa *blockchain* pública todos podem ter acesso ao conteúdo

do contrato inteligente, já no caso de uma blockchain de autorização só alguns utilizadores é que têm acesso ao mesmo.

5. Segurança

A *blockchain* faz uso de um alto nível de criptografia e as suas características tornam-na numa tecnologia extremamente segura. Os *smart contracts* ao estarem integrados na cadeia de blocos, herdaram um elevado nível de segurança. O seu conteúdo não pode ser alterado por ninguém e a sua execução é automática.

6. Rapidez e fiabilidade

Este é um aspeto chave dos *smart contracts*, como são executados num ambiente virtual partilhado (no caso da *ethereum* é a EVM – *ethereum virtual machine*) entre os *nodes* da rede, o código é executado quando o ambiente virtual identifica que a condição foi verificada. Isto assegura uma fiabilidade alta na execução, uma vez que a execução do código não depende de um único servidor como aconteceria numa arquitetura centralizada.

7. Desempenho e despesas

Sempre que alguém faz a implementação de um *smart contract*, o mesmo requer poder computacional que é assegurado pelos *nodes* da rede (criando uma espécie de supercomputador). Como tal, quem faz a implementação do contrato, tem de assegurar os custos da utilização deste poder computacional. Na blockchain da *Ethereum*, o ‘*gas*’ é a unidade de medição para a execução de operações na EVM. Em [26], apresenta-se uma tabela com as diferentes necessidades de *gas* por operação. Quanto maior for a complexidade do *smart contract* maior será a quantidade de *gas* necessário. Por exemplo, uma transação de *Ether* (moeda nativa da *blockchain* da *ethereum*) a outra entidade custa 21000 *gas*, enquanto a implementação de um *smart contract* custa pelo menos 32000 *gas*. Nas redes públicas o custo do *gas* é determinado pela oferta e pela procura, onde os *nodes* oferecem o seu poder computacional a um grande número de agentes, fazendo flutuar os preços da utilização da rede. No caso das redes privadas os custos são geralmente negligenciados uma vez que os *nodes* da *blockchain* não necessitam de um incentivo financeiro para participar na rede. Por norma, os *nodes* das *blockchains* privadas têm outros benefícios, tais como o fornecimento de energia verde e barata. [27,28]

2.2.3 Áreas de aplicação dos smart contracts e da blockchain no setor energético

Depois de resumir as vantagens e características principais dos *smart contracts* e da tecnologia *blockchain*, nesta secção é fornecida uma revisão sistemática das aplicações dos mesmos no setor energético. Os *smart contracts* já foram propostos e utilizados em muitas aplicações, desde o comércio de energia à coordenação de ativos distribuídos como os veículos

elétricos e geração renovável. O tipo de aplicações dos *smart contracts* pode ser categorizado em duas categorias principais: o comércio e a flexibilidade de energia e o controlo distribuído.

A. Comércio de energia e flexibilidade

A flexibilidade energética refere-se à capacidade que os sistemas energéticos têm de responder de forma rápida às mudanças que ocorrem na oferta e na procura de energia. Esta flexibilidade é conseguida através de tecnologias e sistemas de gestão que permitem ajustar a produção de energia e a sua utilização de forma rápida e eficiente, conforme as necessidades vão variando ao longo do tempo. Esta flexibilidade é muito importante porque permite aos sistemas energéticos lidarem de forma eficiente com a crescente incorporação de fontes de energia renováveis que são intermitentes, como a energia solar e eólica. Para além disso, a flexibilidade energética é essencial para garantir a estabilidade e a segurança dos sistemas energéticos [3].

No setor energético, os *smart contracts* são principalmente utilizados no contexto de aplicações de comércio de energia ou de flexibilidade. O principal objetivo do *smart contract* é facilitar as trocas entre consumidores e prosumidores utilizando um mecanismo de pagamento de confiança. Os *smart contracts* têm sido utilizados para as seguintes aplicações:

1) Transações *Peer-to-peer*

Em transações comerciais, tipicamente os SC recebem primeiro as propostas e ofertas dos diferentes intervenientes (produtores, prosumidores e consumidores), o que normalmente exige um depósito dos compradores. Depois deste depósito o *smart contract* verifica se alguma das propostas do consumidor da corresponde com alguma proposta feita por um vendedor, e em caso afirmativo, procede-se à compra/venda de energia. Para as trocas entre o comprador e o vendedor as abordagens variam desde métodos heurísticos a abordagens mais complexas que incluem leilões duplos e validações de fluxo de energia [29], como mostra a figura 5. No método heurístico, o *smart contract* estabelece o acordo entre compradores e vendedores e valida uma transação à medida que as ofertas chegam. Depois do *smart contract* validar uma transação, que consiste num preço, numa quantidade de energia e num tempo de entrega, o *smart contract* analisa e monitoriza a produção e o consumo reais através dos *smart meters* [30]. Esta análise pode desencadear automaticamente a liquidação do contrato, a fim de distribuir recompensas e penalizações de acordo com as condições estipuladas. Quando as trocas P2P não cobrem todas as necessidades dos utilizadores, os *smart contracts* podem facilitar as transações com a rede elétrica.

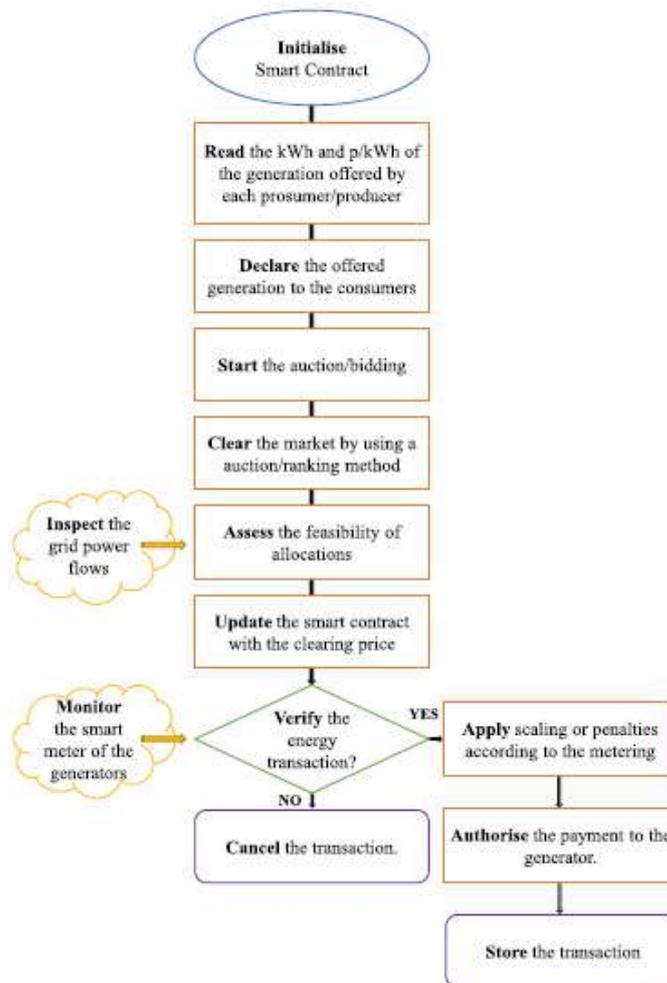


Figura 5 - Exemplo de uma transação de energia peer-to-peer [3]

2) Mercado Retalhista

Os *smart contracts* também podem ser utilizados para aplicações no mercado retalhista, de forma a permitir que os consumidores escolham um fornecedor, assinem um contrato com o mesmo, mas também para armazenar em segurança os dados recolhidos pela infraestrutura de monitorização de energia e os dados de faturação associados [32,33]. Isto é conseguido em primeira instância, permitindo ao operador da rede de distribuição registar cada *smart meter* no *smart contract*. Depois disso, os fornecedores podem transmitir as ofertas de energia através do contrato inteligente, que autenticará os clientes interessados utilizando o endereço do *smart meter* e exigindo um depósito de dinheiro. O pagamento é executado pelo *smart contract* após o período de monitorização e liquidação ser validado [34,35]. Do mesmo modo, os SC podem ser utilizados em conjunto com os *smart meters* de forma a medir em tempo real a quantidade de energia produzida ou consumida e ajustar automaticamente a oferta e a procura. Os SC podem ainda ajudar a automatizar certas atividades como a definição de custos de eletricidade por um dado intervalo de tempo, políticas de pagamento, tempos de compra e venda de eletricidade. Ao

aproveitar as características dos *smart contracts* pode-se efetivamente aumentar a velocidade, a fiabilidade, a escalabilidade e a segurança do mercado energético [37,38].

3) Gestão do Lado da Procura (*'Demand-side Response'*)

O *demand response* é um programa que visa ajustar a procura de energia elétrica através de acordos realizados com os consumidores. A ideia é incentivar os consumidores a reduzirem ou a aumentarem o seu consumo de energia consoante haja muita ou pouca procura, respetivamente.

O *demand response* ajuda a garantir a estabilidade do sistema elétrico evitando desperdícios de energia. Para além disso, também pode ser uma forma de reduzir os custos para os consumidores, uma vez que os mesmos são incentivados a consumir em períodos em que os preços da energia são mais baixos. O facto da maioria das fontes de energia, como por exemplo as centrais a carvão ou centrais nucleares, não conseguirem aumentar de forma rápida a sua produção, pode fazer com que hajam interrupções no fornecimento de energia. Como tal, o *demand response* é crucial para que se evitem essas interrupções. Para além disso, as variações imprevisíveis das fontes renováveis como a energia solar e eólica também podem ser compensadas com este mecanismo, daí a sua importância.

Os *smart contracts* podem ser utilizados em diferentes fases deste processo. Em primeiro, podem calcular e armazenar o perfil de carga previsto e o perfil solicitado [38], ou podem definir periodicamente a flexibilidade disponível, o perfil de energia do consumidor e calcular o balanço energético da rede. Para além disso, de forma semelhante às transações P2P, os *smart contracts* podem ser utilizados para estabelecer um contrato específico entre consumidores interessados e o agregador, especificando a aceitação do pedido de *'demand response'*, com o perfil de carga requerido. Por último, a faturação e o pagamento podem também ser gerados pelo SC, a fim de recompensar ou penalizar os participantes que cumpriram ou não o perfil de carga [39].

B. Controlo Distribuído

1) Gestão de veículos elétricos (VE's)

Os veículos elétricos estão a ganhar cada vez mais popularidade no mercado automóvel, e com isto, vem a necessidade de uma gestão eficiente da infraestrutura de carregamento e da distribuição da energia elétrica. É aqui que entram os *smart contracts*, oferecendo soluções inovadoras para a gestão destes veículos. Os *smart contracts* podem ser utilizados para automatizar processos como pagamentos depois da utilização de estações de carregamento, bem como garantir a segurança e a confiança das transações envolvidas. Outra aplicação bastante interessante dos SC é o facto de conseguirem gerir as baterias dos veículos elétricos, possibilitando o estabelecimento de acordos de compra e venda da energia armazenada nas

baterias dos VE's, criando assim uma rede de armazenamento descentralizada, o que pode ser uma ajuda crucial no equilíbrio do sistema elétrico [40].

2) Gestão da rede elétrica

A gestão da rede elétrica é um setor crítico e complexo que requer soluções inovadoras para garantir a eficiência, a segurança e a confiança da distribuição de energia. Os *smart contracts*, oferecem soluções digitais que podem transformar a forma como a rede opera atualmente. Os *smart contracts* podem ser utilizados para automatizar processos de pagamento, assim como garantir a segurança e a confiabilidade das transações envolvidas. Para além disso, os SC podem ser utilizados para coordenar de forma automática atuadores ou tomar decisões de controlo [41,42]. Devido às características de segurança inerentes aos SC, estes também podem ser utilizados para conceder acesso aos dados da rede, como por exemplo dados de mercado [43]. Outra aplicação interessante dos SC é na gestão da geração de energias renováveis, como a solar e a eólica [44,45]. Os SC permitem estabelecer acordos de compra e venda de energia gerada por utilizadores com painéis solares, tornando-os fornecedores ativos para a rede elétrica, ajudando assim a equilibrar o sistema elétrico e reduzindo o custo de energia.

3) Auditoria e certificação da cadeia de abastecimento

A natureza inviolável e de auto-execução dos *smart contracts* pode ser utilizada para estabelecer uma cadeia de fornecimento transparente. Em [46] utilizaram-se SC com o propósito de calcular e atribuir créditos de energia renovável e de carbono. Observaram-se reduções significativas no tempo despendido e também no custo, uma vez que os SC eliminam a necessidade de uma entidade externa para fazer a auditoria. Em [47] utilizou-se uma abordagem semelhante, mas neste caso o foco está na emissão de garantias de origem e certificação verde.

4) Internet of things (IoT)

Outra aplicação dos *smart contracts* passa pela *Internet of things* (IoT). À medida que a IoT começa a ser cada vez mais utilizada no setor energético para monitorização e controlo de cidades inteligentes e ativos distribuídos, há cada vez mais preocupação com o controlo e a segurança dos dados recolhidos por estes dispositivos, especialmente quando estes são geridos de forma central por um único sistema. De forma a abordar esta questão, foi feito um estudo intitulado “PrivySharing”, onde se utilizou a blockchain para preservar a privacidade e segurança dos dados das cidades inteligentes [48]. Foi utilizado um *token*, denominado “PrivyCoin” de forma a se poder fazer a partilha de dados com partes externas. Estes dados podem incluir a localização geográfica de um *prosumer*, ou outras informações pessoais que devem ser protegidas. Os autores deste artigo demonstraram que a utilização da tecnologia *blockchain* é uma

forma viável de proteger os dados recolhidos pelos sensores IoT e de manter a privacidade do utilizador.

5) Casas inteligentes e sistemas de gestão de energia

Ultimamente, os *smart contracts* também têm sido utilizados para sistemas de gestão de energia no setor residencial. Os *smart contracts* permitem fazer a coordenação de diferentes cargas de utilização final, tipicamente com o objetivo de minimizar as faturas e/ou reduzir a pegada carbónica. Os contratos inteligentes são utilizados para coordenar as cargas, tomar decisões de controlo de forma automática dependendo do estado de algumas variáveis, assegurando que o canal de comunicação é seguro [49,50]. Ao contrário dos sistemas convencionais, os *smart contracts* permitem que se automatizem os processos e aumente-se a transparência e a segurança das transações. Na prática, em termos de *hardware*, os SC podem ser ‘instalados’ em qualquer dispositivo que possua capacidade de processamento e armazenamento, como computadores, dispositivos de IoT, etc. A comunicação com os sensores e atuadores pode ser feita de várias maneiras, dependendo da tecnologia utilizada. Em sistemas baseados em IoT, os sensores e atuadores podem conectar-se diretamente ao dispositivo que contém o *smart contract* através de protocolos de comunicação como o Wi-fi e o Zigbee. As informações coletadas pelos sensores são enviadas para o *smart contract*, que irá processá-las e tomar as ações necessárias.

2.3.4 Outras aplicações

1) Gestão de ativos e utilizadores

Uma das funções mais básicas que os SC podem executar no setor energético corresponde ao registo dos utilizadores (consumidores, produtores e *prosumer*) ou dos ativos. Este registo permite aos utilizadores autenticarem-se a si próprios, utilizando, por exemplo, o endereço do *smart meter* correspondente. Os SC podem também ser utilizados para conceder acesso ao fluxo de dados. A gestão dos utilizadores finais envolve, entre outras, funções que atualizam a lista de utilizadores finais e funções que registam estatísticas relacionadas com cada agente, entre elas a qualidade da eletricidade fornecida pelos *prosumer*. À semelhança da gestão de utilizadores, alguns contratos inteligentes implementam funções de gestão de ativos, tais como o carregamento/descarregamento de sistemas de armazenamento de energia [51].

2) Operações de contratação

Os SC podem ser utilizados para estabelecer um acordo entre duas entidades. Em contratos inteligentes como aqueles apresentados em [32], transmite-se a lista de fornecedores disponíveis a todos os utilizadores finais, a fim de os ajudar a encontrar um fornecedor adequado. Concluída a correspondência entre um produtor e um consumidor, o contrato inteligente estabelece de forma automática um contrato assinado entre eles quer para a aplicação no mercado retalhista, quer para eventos de *demand response*.

Foram também implementadas funções de faturação em *smart contracts*, com o propósito de determinar de forma automática a fatura diária ou mensal entre utilizadores finais e o fornecedor de energia [52].

3) Gestão de ofertas e licitações de energia

Uma das principais interações dos utilizadores finais com os *smart contracts* é a apresentação de ofertas de energia. Assim, a maioria dos *smart contracts* utilizados para comércio de energia P2P, implementam uma função que recebe e guarda as ofertas ou exigências dos utilizadores finais, tal como foi implementado em [53,54]. Estas ofertas podem incluir a quantidade de energia, os instantes de tempo em que a energia é necessária ou está disponível e o preço a que se deseja comprar ou vender a quantidade de energia proposta. O *smart contract* desempenha ainda o papel de assegurar a viabilidade da oferta, assegurando que o utilizador final efetuou um depósito correspondente à quantidade de energia solicitada, ou assegurando que a oferta de um *prosumer* possa ser honrada dada a energia restante numa bateria. Dito de outra forma os SC são um expediente muito útil para implementar leilões, que é uma das formas vulgares para transação de energia/flexibilidade.

4) Armazenamento de dados

Os *smart contracts* podem ser utilizados para armazenar diversos dados referentes ao setor energético. Entre eles, registos de transações energéticas, compromissos contratuais acordados, quantidades de energia a negociar, o preço, as partes envolvidas (no caso de transações P2P), o tempo de entrega, etc. [55]. Apesar de os SC armazenarem a informação relativa aos compradores, vendedores e as respetivas propostas e ofertas de uma forma segura, é boa prática limitar a quantidade de informação armazenada nos *smart contracts* (consultar os capítulos 4 e 5 para mais detalhes) [56,57].

2.3.5 Projetos acadêmicos e industriais

A capacidade de processamento automático de forma descentralizada e segura utilizando *smart contracts* tem motivado a criação de um grande número de projetos relacionados com o setor energético em diferentes áreas, tais como os mercados de energia, o armazenamento de dados, a gestão local de energia, a rastreabilidade de CO₂, etc. Como referido anteriormente, a gestão local de energia é um tópico de grande importância e interesse, especialmente com o aumento da produção de energia renovável. A combinação de *blockchain* e *smart contracts* pode ser uma ferramenta poderosa para a gestão da geração e consumo de energia. Nos últimos anos, com o crescer destas tecnologias, foram implementados vários projetos no âmbito da gestão local de energia baseada em *blockchain* e *smart contracts*. Alguns destes projetos foram desenvolvidos para melhorar a eficiência energética e aumentar a participação da energia renovável. Outros projetos, foram concebidos para permitir que os produtores de energia renovável conseguissem vender a energia excedente diretamente aos consumidores. Em [2], foi desenvolvido uma plataforma que visou desenvolver uma plataforma digital integrada para comunidades de energias renováveis industriais e comerciais, cujo objetivo passou por gerir as interações entre vários membros da comunidade e os seus recursos físicos. O autor desenvolveu duas arquiteturas de plataformas que permitiam a troca de energia entre comunidades. A primeira abordagem utilizou *blockchain* para criar *smart contracts* que armazenam os dados e a lógica do mercado. A segunda abordagem seguiu um método tradicional, onde se utilizou uma base de dados relacional. O desenho do mercado local de energia foi testado com dados de um cenário real. Na simulação as empresas englobadas na comunidade comercializaram energia com base no custo de oportunidade relativamente aos retalhistas. Os resultados revelaram um potencial de poupança de custos individuais e coletivos de 10%. No entanto, o autor comparou as duas tecnologias, chegando à conclusão de que as bases de dados convencionais são melhores para um desenvolvimento e uma implementação mais rápida. No entanto, o autor salienta que a utilização de *smart contracts* pode vir a ser algo a implementar no futuro quando os problemas de escalabilidade forem resolvidos. De seguida serão apresentados outros projetos implementados no setor energético, nomeadamente na gestão local de energia.

LO3 Energy

Um exemplo de projeto de gestão local de energia baseado em blockchain, desenvolvido pela LO3 Energy é o "Brooklyn Microgrid". Fundada em 2012, a LO3 energy tem o objetivo de melhorar a geração e o comércio local de energia. A *Brooklyn microgrids* [59] foi desenvolvida pela LO3 energy utilizando o comércio *peer-to-peer* na infraestrutura elétrica existente. Através do projeto em *Brooklyn* foi possível desenvolver uma plataforma de comércio de energia chamada *Exergy* [60], que permite negociar de forma P2P, e ainda a plataforma *Pando* [62] que pode ser usado pelo operador da rede para reunir recursos locais e estabelecer um mercado de energia, baseado em leilões entre empresas e prosumer. Em 2019, a LO3 Energy, em conjunto com a *Green Mountain Power*, implantou um mercado de energia piloto chamado *Vermont Green* [62] que foi o primeiro mercado autorizado nos Estados Unidos.

O projeto foi bem sucedido e mostrou ser possível criar uma rede de energia descentralizada e autónoma que permita aos seus participantes ter mais controlo sobre a gestão de energia e ainda reduzir o impacto ambiental. No entanto, ainda existem algumas questões técnicas e regulatórias que devem ser resolvidas antes que projetos similares possam ser implementados. É crucial garantir que haja uma boa integração da *microgrid* com a rede elétrica existente, assim como garantir que o projeto cumpra as normas e regulamentos aplicáveis.

Energy Web Foundation (EWF)

É uma organização sem fins lucrativos fundada pela *Grid Singularity* e pelo instituto *Rocky Mountain*. A missão da EWF é criar uma visão do sistema elétrico centrada no cliente, utilizando a *blockchain* para facilitar a implementação de tecnologias e aplicações descentralizadas [58]. O projeto é baseado na *blockchain* da *Ethereum*, mas utiliza um mecanismo de *consensus* público e autorizado 'proof-of-authority' (PoA), o que segundo a organização, aumenta a capacidade de transação em 30x e uma diminuição do consumo de energia em 2 a 3 vezes em relação à *Ethereum*.

Power Ledger

É uma empresa australiana fundada em 2016, focada no comércio de energia *peer-to-peer*. A *power ledger* implementa um ecossistema utilizando o mecanismo de *proof-of-authority* de forma a diminuir o consumo de energia e controlar o acesso à rede. A plataforma permite que o operador da rede ou 'prosumers' gerenciem uma micro-rede com um mercado de energia em tempo real, gerenciem certificados de energia renovável e gerenciem picos de energia ou escolham o tipo e a qualidade da energia.

Prosume.io

Fundada em 2016, a *prosume.io* [63] propõe uma plataforma baseada em *smart contracts*, dispositivos IoT e *tokens* com múltiplas aplicações, incluindo o comércio de energia P2P, a faturação inteligente, o balanceamento da rede e a otimização de processos de comercialização de eletricidade e gás, de acordo com as leis locais de cada país.

IBM

Em outubro de 2016, a IBM lançou o *Hyperledger Fabric* [64], que é uma blockchain de permissão *opensource*. Em associação com a IBM, a *Energy Blockchain Lab* [65] criou uma plataforma descentralizada de gestão de crédito de carbono na China que espera reduzir entre 20% a 50% o ciclo médio de produção de carbono de alguns ativos. Outras aplicações relevantes baseadas na *Hyperledger* são a *Car eWallet* [66], *Sunchain* [67] e *Tennet* [68].

Share&Charge

É uma fundação alemã focada em mobilidade elétrica [69]. Promovem a *Open Charging Network* (OCN) como uma solução descentralizada para serviços de carregamento de VE's. Estão incluídos diferentes serviços para estações de carregamento, como os certificados verdes, pagamento instantâneo e contratos *eRoaming*.

Capítulo 3

Descrição do problema

A rede elétrica está a passar por uma grande transformação, impulsionada pela crescente necessidade de descarbonizar a sociedade e pelas mudanças no comportamento dos consumidores. Os edifícios residenciais são responsáveis por uma parte significativa do consumo total de energia em todo o mundo. De acordo com a Agência Internacional de Energia (AIE), os edifícios são responsáveis por cerca de 30% do consumo global de energia, e espera-se que este número continue a aumentar nos próximos anos, à medida que a população global cresce e a procura aumenta, que não estará dissociada da crescente eletrificação de alguns usos finais (por exemplo, o transporte) e a utilização mais intensiva de alguns equipamentos. Em paralelo, assiste-se a um crescente recurso à geração de eletricidade a partir de fontes renováveis variáveis. Uma das características menos interessantes destas fontes é frequentemente, não estarem disponíveis em quantidade nas horas de maior consumo de energia elétrica. Há, pois, necessidade de, por um lado, reduzir o consumo e/ou a taxa de crescimento de consumo de energia e, por outro lado, de alterar os normais padrões de consumo, por forma a terem uma maior coincidência com a disponibilidade das fontes renováveis variáveis. Como tal, as cidades e casas/edifícios do futuro equipadas com sistemas inteligentes de gestão de energia têm o papel fulcral de reduzir o consumo energético global e permitirem acautelar mais facilmente a crescente disseminação das renováveis variáveis [70].

Nesta secção abordar-se-á o papel dos sistemas de gestão de energia residencial (Home energy management systems - HEMS) na gestão ativa de energia ao nível do consumidor final, delineando as suas vantagens e desvantagens. Da mesma forma, será feita uma análise aos sistemas de gestão de energia utilizando *smart contracts* e *blockchain* com o objetivo de delinear as principais diferenças entre as duas abordagens.

3.1 HEMS convencional

Um sistema de gestão de energia pode ser designado como um conjunto de tecnologias e ferramentas, tanto de hardware como software, que permitem aos proprietários de casas monitorizar e controlar o consumo de energia em tempo real. Os HEMS podem ajudar os proprietários a compreenderem melhor os seus consumos de energia, e a tomarem decisões mais informadas e mais conscientes aquando da utilização da mesma, sendo esperado que os HEMS implementem ações de gestão de recursos energéticos que o consumidor implementaria se

estivesse a gerir os recursos de forma manual, assumindo que teria disponibilidade e o conhecimento para fazer essa gestão de forma otimizada. A componente de hardware inclui tipicamente sensores e atuadores que são instalados na residência, fornecendo dados sobre a utilização de diferentes aparelhos, e uma unidade de controlo que comunica com os dispositivos na casa. A componente de software, para além da capacidade de decisão sobre a gestão dos diversos recursos que possam existir (cargas controláveis, produção local, baterias estáticas, VE's), inclui uma interface para o utilizador com ferramentas analíticas, que permitem aos proprietários das habitações aceder em tempo real a dados sobre a utilização de energia, criar alertas, e controlar dispositivos de forma remota.

Estes sistemas permitem que o utilizador minimize custos de eletricidade, reduza a pegada carbónica, facilite o autoconsumo ou maximize vendas, mantendo ao mesmo tempo os níveis desejados de conforto [71].

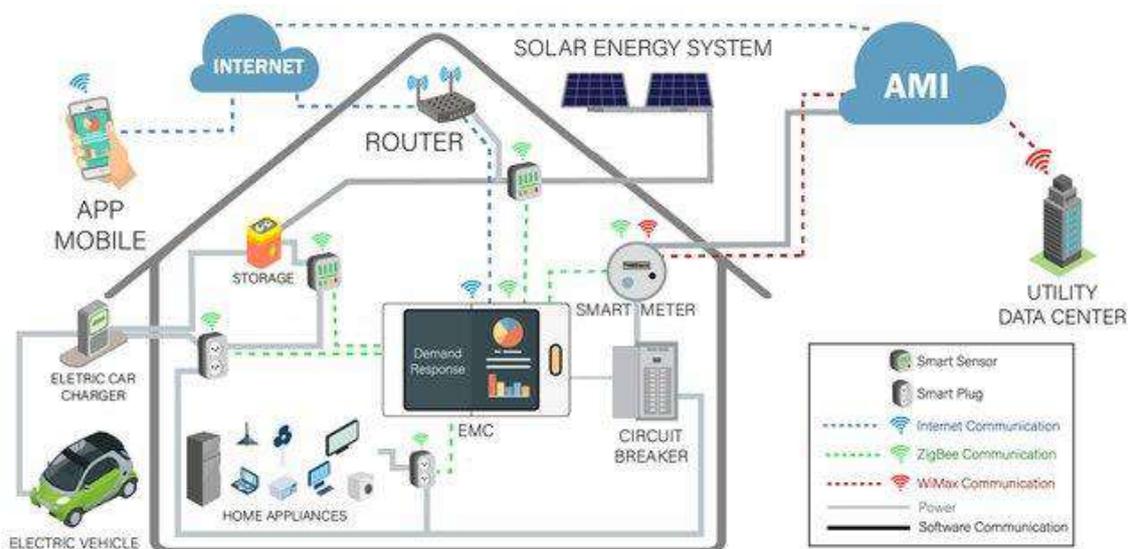


Figura 6 – Sistema HEMS convencional [84]

3.2 HEMS Utilizando *Smart Contracts* e *Blockchain*

Apesar das vantagens de um HEMS convencional, a utilização de *smart contracts* e *blockchain* oferece uma solução mais avançada de gestão de energia residencial onde os proprietários das casas podem automatizar determinados processos de negociação e gestão de energia, assim como manter a segurança e privacidade dos dados e reduzir o risco de fraude ou manipulação dos dados. De forma a perceber a integração do sistema *blockchain* na casa inteligente que contém o HEMS, são apresentadas as figuras 7 e 8. A figura 7 apresenta uma infraestrutura em camadas para o uso da tecnologia *blockchain* e dos *smart contracts*. A primeira

camada “*smart home IoT systems*” representa os dispositivos de IoT presentes na casa inteligente, entre eles os sensores e atuadores do sistema HEMS que coletam e transmitem dados sobre o uso de energia, temperatura luz, etc. A segunda camada é onde a tecnologia *blockchain* é implementada, fornecendo uma base segura e distribuída para armazenar os dados coletados pelos dispositivos IoT. Para além disso, é nessa camada onde os *smart contracts* se encontram presentes permitindo a regulação das interações entre os diversos componentes do sistema. A terceira camada, é onde são implementadas as aplicações baseadas em *blockchain* e *smart contracts*, permitindo aos utilizadores realizar determinadas ações e aceder aos dados referentes à produção, ao consumo e à compra e venda de energia. Por fim, a quarta camada, é responsável por fornecer acesso às aplicações e aos dados armazenados na *blockchain*.

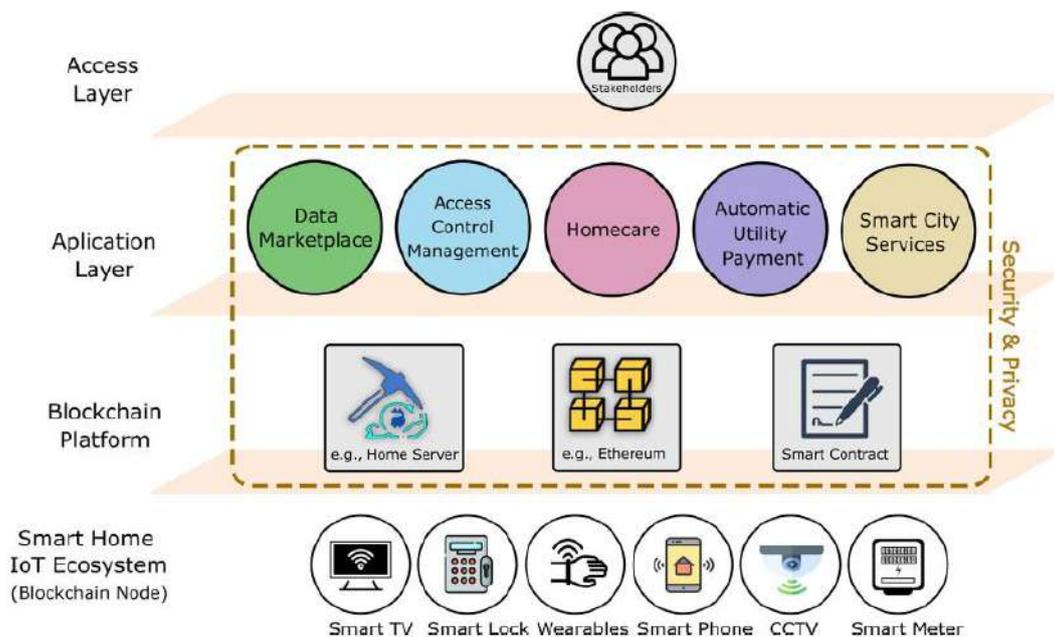


Figura 7 - Estrutura de um sistema de gestão de uma casa inteligente utilizando smart contracts e blockchain [83]

Num sistema HEMS que utilize *smart contracts* (figura 8) os sensores e atuadores comunicam com o mesmo através de dispositivos intermediários como *gateways*. Estas *gateways* são responsáveis por coletar os dados dos sensores, processá-los e enviá-los para o *smart contract*. Para além disso, também são responsáveis por enviar comandos para os atuadores baseados nas decisões tomadas pelo *smart contract*. A comunicação entre os *gateways* e o *smart contract* é feita através de transações na *blockchain*. Estas transações são executadas de forma descentralizada, o que significa que as operações de processamento de dados realizadas por múltiplos *nodes* da rede, o que pode tornar a velocidade das transações mais rápidas.

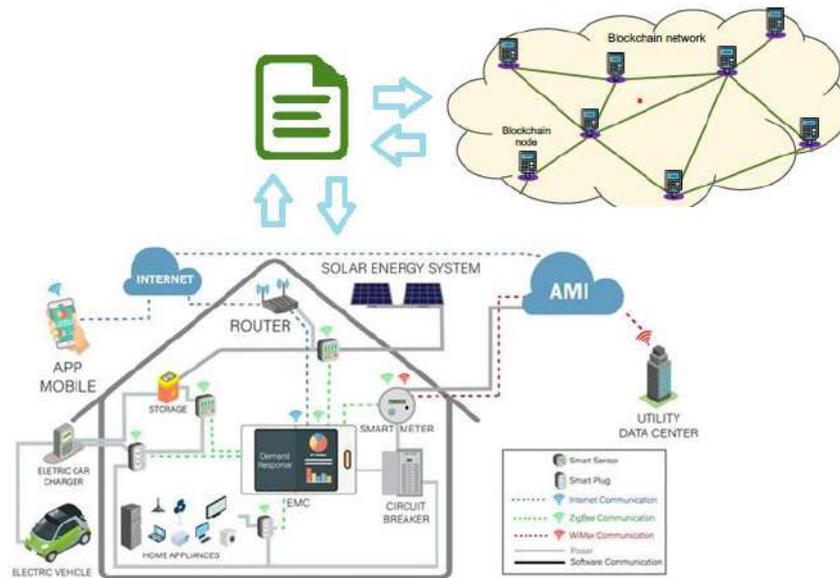


Figura 8 - Sistema HEMS utilizando smart contracts e blockchain

Algumas das vantagens da implementação de *smart contracts* e *blockchain* num sistema de gestão de energia residencial são:

3.2.1 Segurança

Os sistemas de gestão de energia residencial dependem da recolha, transmissão e armazenamento de informações sensíveis, tais como os dados de consumo de energia, padrões de utilização pessoais, e mesmo a localização de dispositivos em casa. Por conseguinte, a segurança e a privacidade são preocupações significativas. Os dispositivos e sistemas HEMS podem ser vulneráveis ao *hacking* e a outras formas de acesso não autorizado, o que poderia permitir que os atacantes ganhassem controlo do sistema e roubassem informações pessoais ou até mesmo que interrompessem o fornecimento de energia. A fuga de dados pode revelar hábitos pessoais e padrões de utilização que os proprietários podem não querer partilhar com outros. Para além disso, a informação recolhida por este tipo de sistemas pode revelar a presença ou não de pessoas em casa.

Têm sido relatados vários casos de HEMS a serem pirateados. A *Symantec*, uma empresa de ciber-segurança, descobriu que um utilizador tinha sido explorado por piratas informáticos através de uma vulnerabilidade do sistema de gestão. Com isto, os piratas fizeram aumentar o consumo energético, o que fez com que os utilizadores pagassem faturas mais elevadas [73]. Num outro exemplo, um grupo de piratas informáticos conseguiu ter acesso a um HEMS, e através disso fez aumentar a temperatura de forma significativa colocando o ar condicionado a funcionar de forma contínua. Estes são apenas alguns exemplos, mas realçam a importância de assegurar

que este tipo de sistemas previna o acesso não autorizado e proteja os utilizadores contra tentativas de *hacking*. Embora os dispositivos HEMS possam à partida contribuir para reduzir os custos de eletricidade do cliente, os ciberataques e preocupações com a privacidade podem levar a que haja receio por parte dos utilizadores em participar em mecanismos de DR [74].

Utilizar *smart contracts* e *blockchain* para fazer a gestão local de energia pode melhorar de forma significativa a segurança nas casas inteligentes. Com a utilização de *smart contracts*, a comunicação entre dispositivos é assegurada pela *blockchain* que utiliza criptografia e algoritmos de *consensus* de forma a proteger contra o acesso não autorizado. Os *smart contracts* podem ser programados com controlos de acesso que restringem que dispositivos e utilizadores são capazes de comunicar com o HEMS. Isto é, um *smart contract* pode ser configurado para permitir apenas a ligação de dispositivos autorizados ao HEMS e permitir apenas aos utilizadores autorizados o controlo desses dispositivos. Para além disso, num sistema de gestão que utilize *smart contracts* e *blockchain*, mesmo que alguém tenha sucesso em comprometer os dados do sistema, os mesmos não serão apagados ou alterados uma vez que há cópias da informação em todos os *nodes*. Os *smart contracts* também proporcionam transparência num HEMS, permitindo a monitorização em tempo real da utilização de energia e pagamentos e liquidações automáticas com as empresas de serviços. Esta transparência assegura que toda a utilização de energia é rastreada, registada e pode ser auditada, o que torna difícil a ocorrência de qualquer utilização não autorizada ou fraude.

Em suma, a utilização de *smart contracts* e *blockchain* num HEMS pode melhorar significativamente a segurança ao proporcionar uma comunicação segura, controlos de acesso, transparência e autonomia, o que pode ajudar a proteger contra acesso não autorizado, *hacking* e outros tipos de ataques.

3.2.2 Custos

Os sistemas convencionais envolvem tipicamente a instalação de um *hub* central que comunica com vários dispositivos na residência. Estes sistemas podem ser relativamente caros, uma vez que requerem tipicamente a compra e instalação de equipamento especializado e software. Para além disso, estes sistemas também envolvem custos de manutenção e operação (eletricidade, internet e salários). No entanto, os custos associados podem variar dependendo do tamanho e complexidade do sistema, bem como do número de dispositivos e aparelhos que precisam de ser ligados.

Os sistemas de gestão que utilizam *smart contracts* e *blockchain*, por outro lado, utilizam os *smart contracts* para gerir a utilização e distribuição de energia. Neste tipo de abordagem, cada dispositivo é equipado com um microcontrolador que é ligado à rede da *blockchain*. Estes sistemas podem ser menos dispendiosos de implementar, uma vez que não requerem um *hub* central [2,3]. No entanto, apesar dos custos de implementação poderem ser mais baixos, os *smart*

contracts poderão exigir maiores custos de manutenção, uma vez que para ocorrer a execução do contrato é necessário pagar taxas de utilização ('gas fees'). Os custos podem variar bastante dependendo do tipo de plataforma que é utilizada e do número de dispositivos e aparelhos que precisam de ser ligados. [2,3].

3.2.3 Resiliência e Fiabilidade

O facto de os sistemas convencionais dependerem de um único *hub* central para gerir e controlar a utilização de energia na casa faz com que estes tipos de sistemas sejam mais suscetíveis a pontos de falha únicos, tais como o mau funcionamento ou o comprometimento do sistema. A utilização de *smart contracts* torna o sistema mais resiliente devido à natureza descentralizada da *blockchain*, o que significa que não há um ponto único de falha. Em termos de fiabilidade, um HEMS centralizado está dependente da fiabilidade do servidor central. Por outro lado, o HEMS baseado em *smart contracts* e *blockchain* é mais fiável devido ao facto de ser descentralizado. Os *smart contracts* são autoexecutáveis e os dados armazenados na *blockchain* são imutáveis, o que significa que os mesmos estão sempre disponíveis e acessíveis. Isto reduz a probabilidade de ocorrerem erros e faz aumentar a confiança no sistema. No entanto, é importante lembrar que a fiabilidade dos sistemas descentralizados depende em muito da plataforma que se está a utilizar. De modo a maximizar as potencialidades destes sistemas é crucial escolher uma *blockchain* que seja robusta, capaz de lidar com um número elevado de transações e que tenha uma boa escalabilidade e segurança.

3.2.4 Facilidade de implementação

A implementação do *hub* central nos sistemas centralizados pode ser um processo relativamente simples desde que o sistema seja compatível com os dispositivos e aparelhos da casa. Por outro lado, a implementação de um sistema com *smart contracts* pode ser mais complexa. A implementação de um sistema que utilize *smart contracts* exige um conhecimento aprofundado da tecnologia *blockchain*, assim como a capacidade de configurar de forma adequada cada dispositivo e ainda o conhecimento necessário para desenvolver o *smart contract*. [41]

Capítulo 4

Procedimentos e Implementação

4.1 Arquitetura

De acordo com o que foi abordado no capítulo anterior, nesta secção, discutir-se-á a implementação de um *smart contract* que terá o objetivo de simular o comportamento de um HEMS. Como tal, o *smart contract* é concebido de forma a gerir o consumo, a produção e o armazenamento de energia num ambiente residencial, permitindo fazer o rastreio e a distribuição de energia entre diferentes equipamentos. Isto inclui a capacidade de monitorizar a utilização de energia, estabelecer limites de utilização e estabelecer regras impostas pelo utilizador. O *smart contract* foi implementado na plataforma *Remix IDE* utilizando a linguagem de programação *solidity*.

4.1.1 Escolha das plataformas a utilizar

Em primeira instância, para proceder à implementação do *smart contract* foi preciso ter em conta diversos fatores como o tipo de linguagem de programação a ser utilizada, o ambiente/plataforma de desenvolvimento do *smart contract*, a plataforma *blockchain*, as ferramentas de teste, etc. Para além disso, como referido anteriormente, a implementação de um *smart contract* envolve o pagamento de ‘gas fees’, como tal, isso também teve de ser levado em linha de conta. Serão apresentados nos tópicos seguintes, os principais motivos pela escolha dos diferentes fatores mencionados.

Plataforma *Blockchain*

Existem várias plataformas para escrever e implementar *smart contracts*, como por exemplo a *Ethereum*, *Hyperledger*, *EOS*, entre outras. A *Ethereum* é uma *blockchain* pública que há muito pouco tempo funcionava com o mecanismo de *consensus proof-of-work*, mas que recentemente passou a adotar o *proof-of-stake*, que como mencionado no capítulo 2, é um mecanismo mais sustentável e amigo do ambiente, uma vez que reduz significativamente o consumo de energia necessário para o funcionamento da rede (redução de energia de aproximadamente 99,95%) [4]. Desta forma, os mineradores dos *nodes* utilizam muito menos energia do que no mecanismo ‘proof-of-work’ que necessita de resolver cálculos matemáticos complexos. A *blockchain* da *Ethereum* é a plataforma mais utilizada no setor energético. Para além de ser das *blockchains* mais

seguras, a blockchain da *Ethereum*, possibilita a criação de aplicações descentralizadas (dApps - aplicações que utilizam a tecnologia *blockchain* para funcionar de forma autónoma, sem a intervenção de intermediários) e fornece uma vasta gama de ferramentas para o desenvolvimento, teste, e implementação dos *smart contracts*, como por exemplo o *Remix IDE* e o *Truffle* [26, 47]. Apesar das mais valias desta *blockchain*, um dos aspetos negativos é o facto de apenas conseguir processar em média 15 transações por segundo, o que pode ser um problema se o nível de utilização da rede for muito grande [26]. Contudo, por todas as vantagens mencionadas anteriormente, e como as desvantagens não têm impacto negativo na aplicação que se pretende efetuar, foi esta a *blockchain* escolhida para a implementação do *smart contract*.

Ambiente de desenvolvimento

Quanto ao ambiente de desenvolvimento, a plataforma escolhida foi o *Remix IDE* [76]. O *Remix IDE* é um ambiente de desenvolvimento integrado que permite escrever e testar *smart contracts* na *blockchain* da *Ethereum*. Os *smart contracts* podem ser implementados utilizando a linguagem de programação da *Ethereum*, o *solidity*. O *Remix IDE* fornece uma interface ‘*user-friendly*’, que simplifica o processo de desenvolvimento e teste de *smart contracts*. Inclui um editor de código integrado que foi concebido especialmente para a linguagem de programação *solidity*. Para além disso, a plataforma é bastante utilizada pela comunidade de programadores e tem uma grande comunidade de colaboradores que estão constantemente a trabalhar para melhorar o software. Em suma, o *Remix IDE* é uma plataforma simples e fácil de usar em comparação com outras plataformas como o *Truffle* que têm funcionalidade e capacidades mais avançadas e personalizáveis, mas que não são necessárias no nosso trabalho. Na figura em baixo, pode-se ver um *smart contract* implementado na plataforma *Remix IDE*.

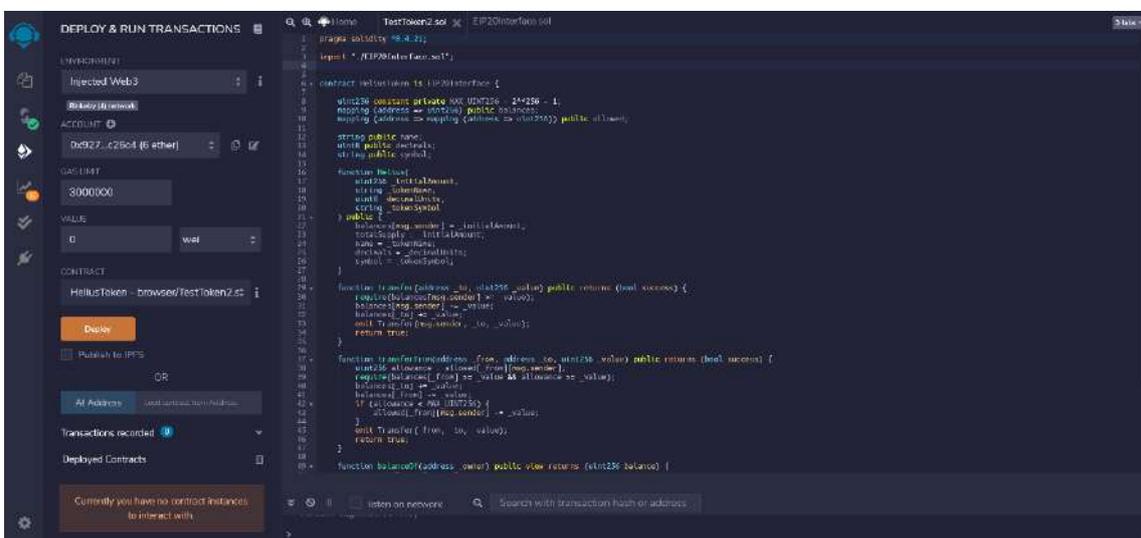


Figura 9 - Exemplo de um smart contract implementado na plataforma *Remix IDE*

Gas fees

Como mencionado anteriormente, de forma a implementar os *smart contracts* é necessário pagar as taxas de utilização da rede *Ethereum* (*gas fees*). Não é possível executar um *smart contract* sem pagar as taxas de utilização. Esta taxa, neste caso, é paga em *ether* (ETH) que é a moeda nativa da blockchain da *Ethereum*.

Quando a *Ethereum* era *proof-of-work* e fazia uso de poder computacional (máquinas ligadas à rede) para manter a rede segura, esta taxa era utilizada para pagar aos *miners*. No entanto, em *proof-of-stake*, os validadores (conhecidos em *proof-of-work* como *miners*) são incentivados a proteger a rede ao ‘apostarem’ na moeda nativa da *blockchain*. Ou seja, os *nodes* que validam os blocos e as transações na *blockchain* da *Ethereum* (ou outra *blockchain* que utilize o mecanismo de *consensus proof-of-stake*) têm uma certa quantidade de moeda nativa em *stake*, o que significa que se a rede da *Ethereum* for afetada, eles também são afetados porque possuem uma certa quantidade de moeda nativa. Como tal, os validadores têm interesse em manter a integridade da rede, dado que isso ajuda a moeda que eles possuem a ter valor. Desta forma, as *gas fees* numa *blockchain* baseada em *proof-of-stake* ajudam a prevenir ataques à rede ao incentivar os participantes a proteger a rede, pois, se a rede for comprometida, o valor da moeda apoiada pode ser afetado. Para além disso, como é necessário o pagamento de um *gas fee* para cada transação enviada para a rede da *blockchain*, sairia muito dispendioso a um pirata informático sobrecarregar a rede com transações (conhecido como ataque de congestão de rede - DDoS (*Distributed Denial of Service*)) [26].

Os *gas fees* são medidos em Gwei que é uma divisão do *ether* (1 Gwei = 0,000000001 eth). Se 1 ETH valer 1650€, 1 Gwei equivale a 0,0000165€ (neste caso como vamos utilizar a rede de teste da Goerli, o pagamento dos *gas fees* é medido em GoerliETH, sendo que 1 GoerliETH corresponde a 1 ETH). A implementação de um *smart contract* depende muito do congestionamento da rede no momento da implementação, como tal, o preço de implementação de um *smart contract* pode ir de umas décimas de cêntimo até umas dezenas ou até mesmo centenas de euros.

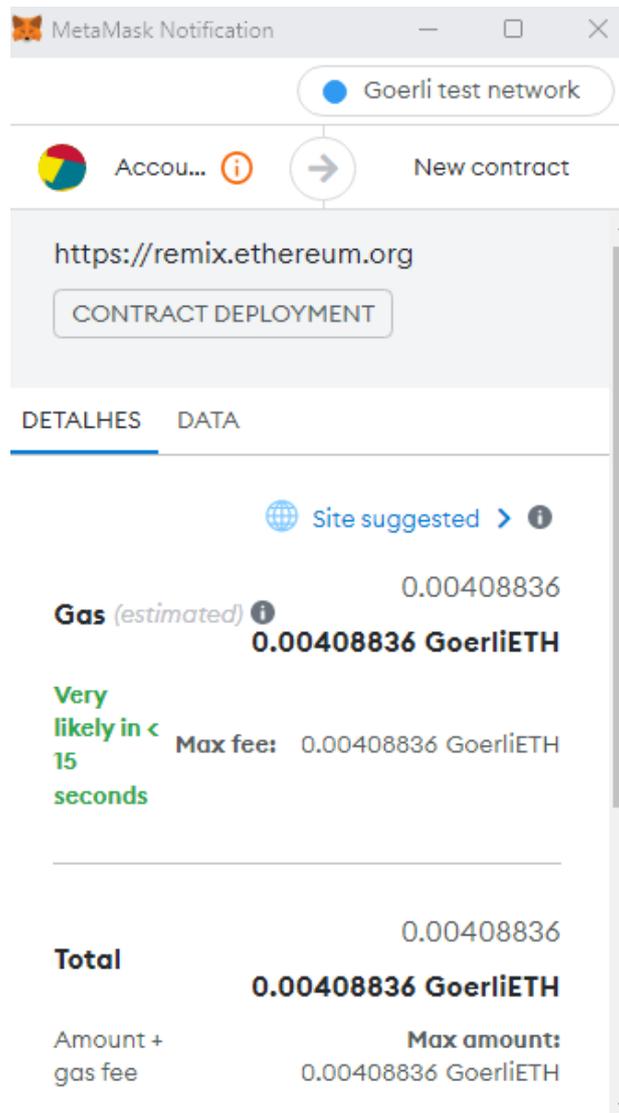


Figura 10 - Exemplo do pagamento de gas fees de uma transação com a rede de teste Goerli na carteira digital Metamask

'Testnet'

Como o objetivo da implementação do *smart contract* desta dissertação é fazer apenas uma simulação, utilizou-se uma rede de teste (*test net*). Esta rede é essencialmente uma cópia da *blockchain* principal utilizada para testar *smart contracts*, aplicações descentralizadas e outras funcionalidades antes de serem lançadas na rede principal ('mainnet'). Esta rede de teste permite executar *smart contracts* sem pagar taxas, o que é uma ajuda enorme aos programadores (sairia muito caro a um programador implementar *smart contracts* caso esta alternativa não existisse).

É possível utilizar uma 'testnet' como por exemplo o *Ganache* ou ainda obter 'tokens experimentais' através do *Goerli Faucet*. Desta forma, é possível correr o código sem gastar

dinheiro real. No entanto, é de salientar que para implementar o *smart contract* na rede principal da *blockchain* ('mainnet') é sempre necessário pagar utilizando 'tokens reais' (neste caso, como se trata da *blockchain* da *Ethereum* o pagamento é feito em *ether*). No âmbito desta dissertação, foi usado o *Goerli Faucet* [77], que é uma *testnet* que fornece de forma gratuita *tokens* de teste (GoerliETH) aos utilizadores com a finalidade de fazer implementações na rede de teste. A plataforma *Goerli Faucet* foi concebida para criar um ambiente de teste mais estável e fiável para os programadores testarem os *smart contracts*. No entanto, apesar dos *tokens* serem gratuitos, os mesmos também são limitados de forma a prevenir abusos no sistema. Desta forma, a comunidade que gere o *Goerli Faucet* só envia 0,2 GoerliETH para cada carteira digital ou endereço IP. Apesar de ser uma medida que previne o abuso por parte de entidades que quisessem sobrecarregar o sistema, é um ponto negativo que esta *testnet* tem, uma vez que muitas das vezes a execução de um *smart contract* exige mais do que 0,2 GoerliETH. Como tal, em implementações futuras, aconselha-se a utilização de uma alternativa ao *Goerli Faucet*, como por exemplo o *Ganache*, para evitar esta barreira de ter *tokens* limitados em cada dia. De forma a perceber melhor as diferenças entre a rede de teste do *Goerli Faucet* e do *Ganache*, e a ajudar o leitor a escolher qual rede se adequa melhor à aplicação desejada, foi criado um ficheiro em [78], com o nome 'Goerli vs. Ganache' que evidencia as principais diferenças entre as duas redes.

Na imagem imediatamente em baixo, é possível observar a plataforma *Goerli Faucet*. De forma a obter os *tokens* experimentais, é necessário colocar o endereço da carteira digital para onde queremos enviar os mesmos (o endereço da carteira é representado na imagem pela *string* que começa em '0xdE').

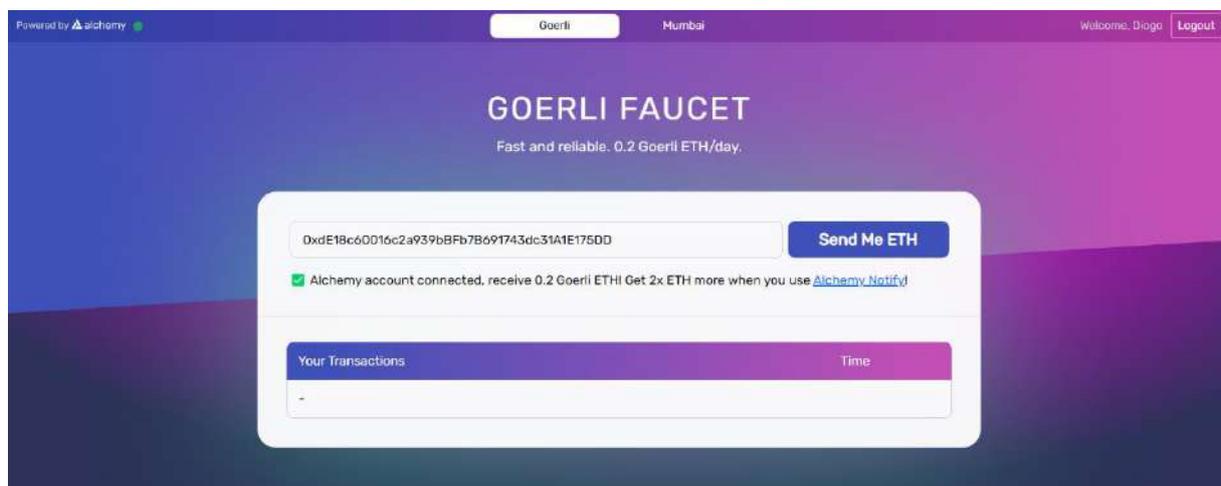


Figura 11 - Plataforma Goerli Faucet

Carteira digital

De forma a poder interagir com o *smart contract* é necessário criar uma carteira digital. No âmbito desta dissertação foi utilizada a *Metamask* como uma extensão do navegador ‘Google Chrome’ de forma a poder interagir com o *smart contract* implementado no *Remix* IDE. O *metamask* é uma carteira digital que permite aos utilizadores interagir de forma fácil com aplicações descentralizadas. Uma das principais vantagens da utilização do *metamask* é o facto de permitir o acesso de forma simplificada à rede da *Ethereum* e a capacidade de interagir com *smart contracts* sem ter de executar um *node* completo. Isto significa que os utilizadores podem testar e interagir com os seus *smart contracts* sem terem de criar e manter a sua própria infraestrutura. Como referido anteriormente, cada *node* contém a cópia integral da *blockchain* instalada no seu dispositivo computacional, e participa na rede verificando as transações, mantendo a integridade da mesma. No entanto, o *metamask* permite que um utilizador consiga interagir com a *blockchain* enviando transações para a mesma, mesmo sem o utilizador em questão ‘ter’ um *node* completo. Essencialmente, o *metamask* interage com a *blockchain* através da utilização de um *gateway* ou um *node* intermediário, que funciona como uma ponte entre o *browser* do utilizador e a *blockchain*. Sempre que um utilizador precisa de enviar uma transação ou aceder a informações da *blockchain*, o *metamask* envia essa requisição para o *gateway*, que se encarrega de aceder à *blockchain* e efetuar as ações desejadas. Desta forma, o utilizador não precisa de executar um *node* completo, que requer recursos computacionais consideráveis ou uma larga quantidade de moeda nativa em *stake*, e pode interagir com a *blockchain* de forma mais fácil e conveniente.

Para além disso, o *metamask* também oferece uma interface de fácil utilização para gerir contas, enviar e receber *tokens*, e interagir com aplicações. Isto torna-o uma ferramenta excelente para o desenvolvimento e teste de *smart contracts*. O *metamask* proporciona uma forma fácil de trocar entre diferentes redes, como a ‘testnet’ e a ‘mainnet’, o que é bastante útil para testar e implementar *smart contracts*. Em suma, o *metamask* provou ser uma ferramenta bastante valiosa na interação com o *smart contract* no *remix* e no desenvolvimento e teste na rede *Ethereum*.

Na figura 12 é possível visualizar a carteira digital *metamask*. Neste momento a carteira está conectada à rede principal da *blockchain* da *Ethereum* (‘Ethereum Mainnet’), o que significa que qualquer transação que se efetue com a rede, implica o pagamento de *tokens* reais (ETH). De forma a proceder à alteração para a rede de teste, é necessário fazer uma alteração (figura 13).

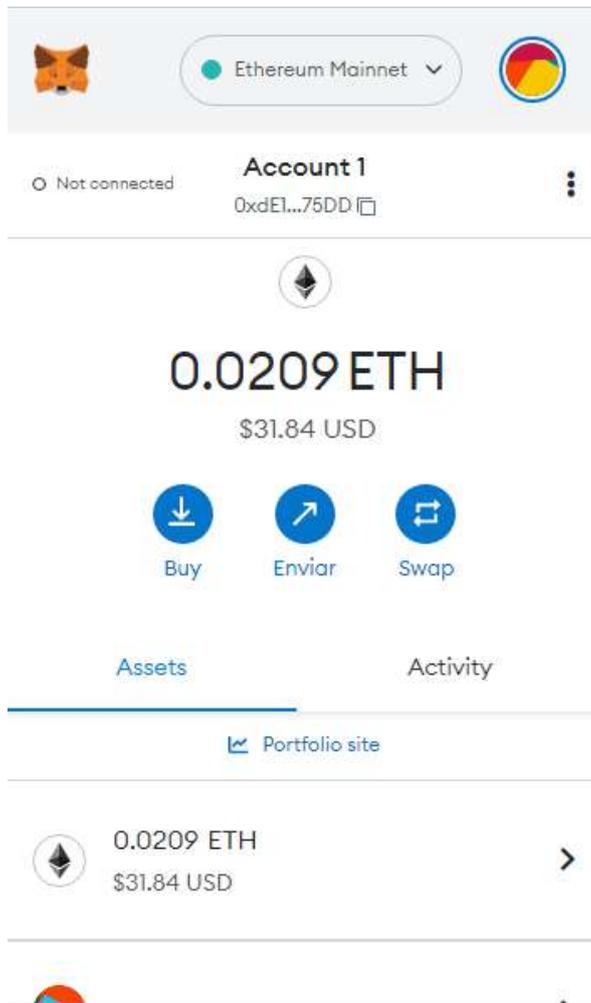


Figura 12 - Carteira digital metamask conectada à mainnet da Ethereum

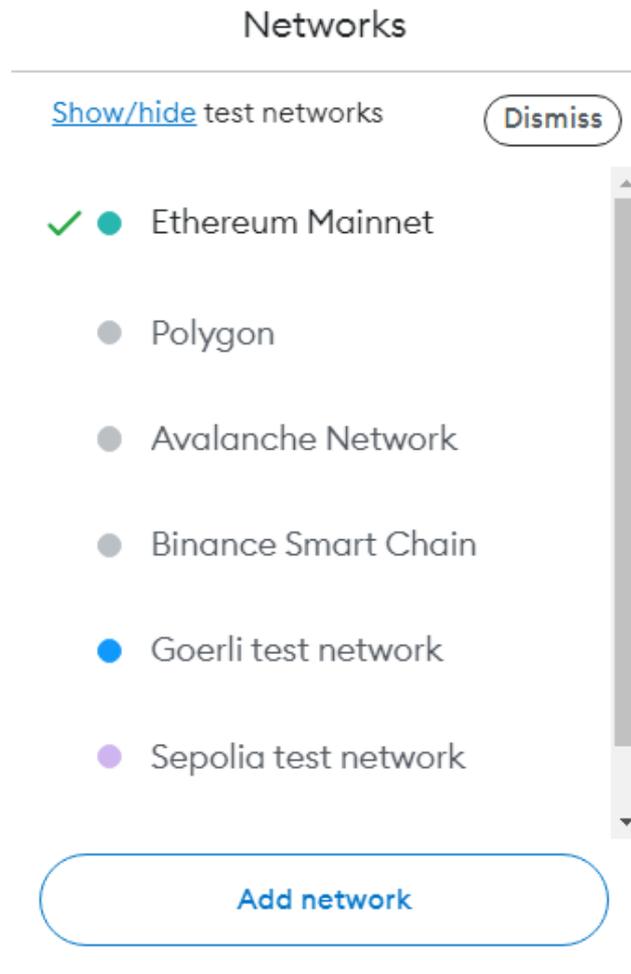


Figura 13 - Alteração da rede na carteira digital metamask

Depois de se proceder à alteração da rede principal para a rede de teste, a carteira digital está pronta para ser utilizada utilizando os *tokens* virtuais (Goerli ETH). Como é visível na figura 14 a carteira tem na sua posse 0,6 Goerli ETH. Estes *tokens* vão ser utilizados para fazer a submissão de transações na plataforma *Remix IDE*.

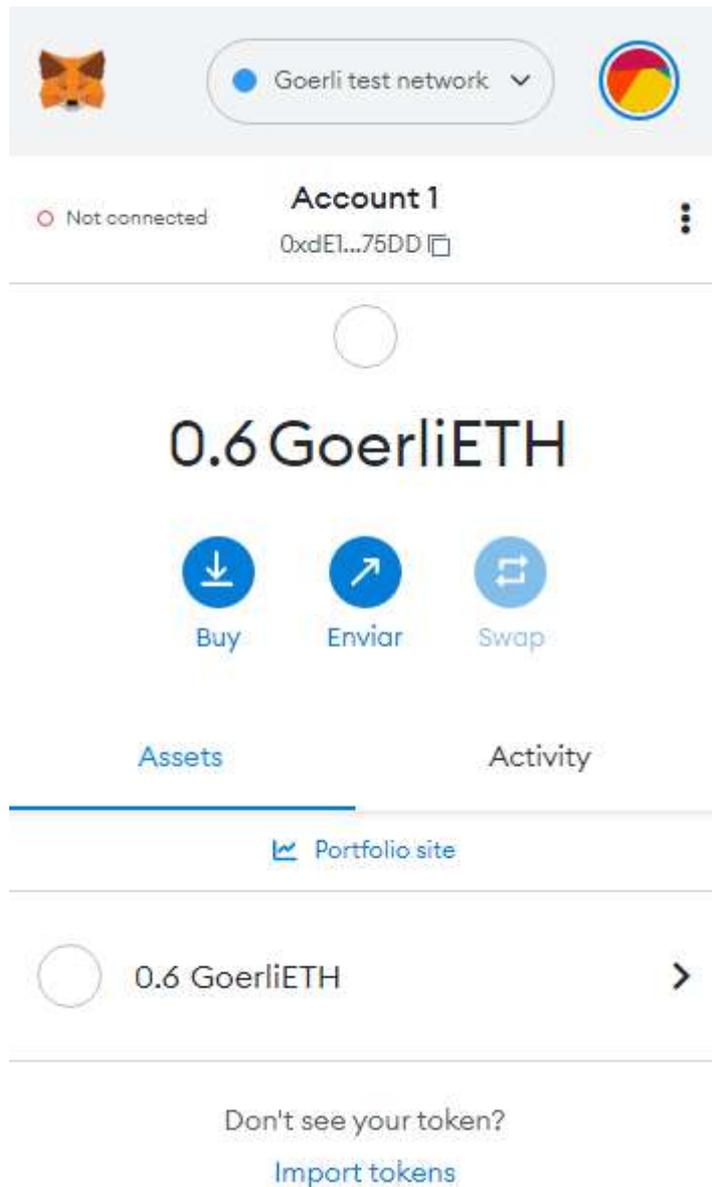


Figura 14 - Plataforma metamask na rede de teste Goerli Faucet

4.2 Simulação

Após a escolha das plataformas e ferramentas necessárias para a simulação do *smart contract*, procedeu-se à implementação do algoritmo. O objetivo do algoritmo passa por simular a gestão de energia de um dia numa habitação que contém diversos equipamentos entre eles, iluminação, máquinas de lavar loiça, máquinas de lavar e secar roupa, baterias e painéis solares. Para inicializar a simulação foram definidos os padrões de consumo dos diversos aparelhos, sendo que o dia foi dividido em intervalos de 15 minutos (96 intervalos) [ver anexos]. Foi definido também a previsão solar para esse dia, assim como a capacidade máxima da bateria e o nível de carga da mesma no início do dia. De acordo com [82], os diferentes tipos de cargas podem ser classificados como cargas não controláveis, cargas ajustáveis, cargas interrompíveis e cargas deslocáveis.

No entanto, no âmbito desta dissertação, apesar de se ter considerado a utilização de cargas ajustáveis e de um algoritmo/conjunto de regras mais complexo, optou-se por fazer uma versão mais simples de forma a simplificar a simulação e torná-la mais fácil de compreender. Para além disso, como referido anteriormente, a utilização de um algoritmo mais complexo iria fazer com que os *gas fees* fossem superiores, e, uma vez que a rede de teste (Goerli Faucet) só atribui 0,2 Goerli ETH por dia, ficaria difícil de implementar o *smart contract*. Como tal, foram apenas utilizadas cargas deslocáveis e inflexíveis. Na tabela imediatamente em baixo é possível ver o tipo de equipamentos utilizados, assim como o tipo de carga e a potência de funcionamento de cada equipamento. Foram tomados como referência os valores da potência dos equipamentos apresentados em [82], que tem por base os equipamentos presentes em casa do utilizador residencial comum, assim como o respetivo padrão de consumo diário de cada equipamento.

A nível dos preços de compra de energia à rede, foi utilizado como base os preços da tarifa bi-horária da EDP (opção ciclo semanal – Hora legal de inverno), sendo que as horas de vazio são da [00:00h às 07:00h] e as horas foras de vazio são das [07:00h às 00:00h], com um preço de compra de 0,20620€ e 0,26120, respetivamente. No caso de haver venda de energia à rede, os preços utilizados foram 0,06€ e 0,1€, nos períodos de vazio e fora de vazio, respetivamente. Foi utilizada uma potência contratada de 6,9 kVA.

Equipamentos	Tipo de carga	Potência (W)
Painel Solar	-	1500
Máquina de lavar loiça	Deslocável	3000
Máquina de lavar roupa	Deslocável	2000
Máquina de secar roupa	Deslocável	2500
Bateria	-	*
Carga base	Inflexível	-

Tabela 2 - Caracterização dos equipamentos utilizados

*no caso da bateria foi utilizada uma capacidade máxima de 5 kWh com uma potência de carga/descarga de 1 kW

De forma a perceber qual a melhor estratégia de implementação do algoritmo, em primeiro lugar efetuou-se uma análise de diferentes cenários simulados. Tendo em conta, o consumo típico dos equipamentos, a previsão solar, o estado da bateria e os preços de energia, chegou-se à conclusão de que de forma a minimizar os custos os equipamentos devem ser colocados a funcionar durante o período de tempo em que a produção solar atinge os pontos máximos. Para além disso, se houver uma sobreposição da utilização dos equipamentos a eficiência de utilização de energia também é afetada. Como tal, sabendo que os equipamentos devem ser colocados a funcionar nos períodos de maior produção e que os equipamentos não devem ser colocados a funcionar de forma simultânea, utilizou-se uma função que calcula as médias de consumo dos diferentes equipamentos. Sabendo qual dos equipamentos tem maior média, calculou-se (de acordo com o período de funcionamento do equipamento em questão) a maior média de produção solar e colocou-se o respetivo equipamento a funcionar durante esse período. O mesmo foi feito para os seguintes equipamentos. No caso da bateria, de forma a conservar o tempo de vida útil, apenas foi utilizada quando o *soc* fosse maior do que 30%. Pode-se encontrar o código implementado em *solidity* numa pasta no *github* [78]. O código foi partilhado em conjunto com um ficheiro de texto que explica o passo a passo da implementação do *smart contract* no remix IDE. Isto foi feito para que o mesmo possa ser visualizado por alguém com interesse na área, e, quiçá possa contribuir de alguma forma para algum projeto futuro.

De modo a se compreender melhor o funcionamento do código implementado, é representada de seguida a lista de etapas sequenciais que o código executa, desde a introdução dos inputs necessários, ao resultado final.

Algoritmo

1. Inicialização: considerando o intervalo de planeamento
 - a) Estado inicial de carga da bateria
 - b) Consumo dos equipamentos
 - c) Produção solar
 - d) Preços da energia

2. Deslocamento das cargas deslocáveis

3. Para cada intervalo de tempo:
 - a) Calcular o consumo energético de todos os equipamentos
 - b) Calcular a produção solar
 - c) Calcular a energia de carregamento ou descarregamento da bateria
 - d) Atualizar o estado da bateria
 - e) Calcular o custo energético
 - f) Atualizar a fatura a pagar no final do dia

O algoritmo implementado é demonstrado imediatamente em baixo:

SOC – ‘State of charge’ da bateria

P_{DES} – potência máxima de descarga da bateria

$P_{DES}(t)$ – potência de descarga da bateria num dado instante t

P_{CAR} – potência máxima de carregamento da bateria

$P_{CAR}(t)$ – potência de carga da bateria num dado instante t

$P_G(t)$ – potência pedida à rede no intervalo t

$P_L(t)$ – potência pedida pelas cargas no intervalo de tempo t

$P_{PV}(t)$ – potência dos PV

E_{ML} – Consumo máquina lavar roupa

E_{MS} – Consumo máquina secar roupa

E_{LL} – Consumo máquina lavar loiça

E_{PV} – Energia produzida pelos painéis num dado Δt

Algoritmo de gestão das cargas

Para i in range (Prod.Solar)

Calcula média E_{ML}

Calcula média E_{MS}

Calcula média E_{LL}

Se (média $E_{ML} >$ média E_{MS} AND média $E_{ML} >$ média E_{LL})
Define-se (média E_{ML}) como média máxima

Senão

...

(Faz-se o mesmo para as outras médias)

Calcula média E_{PV}

Faz

Insere média máxima consumo na média máxima da produção

Algoritmo de gestão da bateria

Se $(P_{PV} - P_L) \leq 0$ então

Faz

Se $(SOC > 30\%)$

{o que se vai buscar à bateria}

$$P_{DES}B(t) = \min (P_L(t) - P_{PV}(t); P_{DES});$$

{alimenta a carga com o PV e com a bateria e o resto da rede}

$$P_G(t) = P_L(t) - P_{PV}(t) - P_{DES}B(t);$$

$$SOC(t) = SOC(t-1) - (P_{DES}B(t));$$

Senão

{se $(SOC < 30\%)$ }

{alimenta a carga com o PV e o resto a partir da rede}

$$P_G(t) = P_L(t) - P_{PV}(t);$$

Se $(P_{PV} - P_L) \Rightarrow 0$ então

Faz

$$P_{CAR}B(t) = \min (P_{PV}(t) - P_L(t); P_{CAR});$$

{quantidade de energia enviada para a rede}

$$P_G(t) = - (P_{PV}(t) - P_L(t) - P_{CAR}B(t));$$

$$SOC(t) = SOC(t-1) + (P_{CAR}B(t));$$

Características do código

De forma a atribuir a posse do *smart contract* a apenas um utilizador, o contrato utiliza a biblioteca “OpenZeppelin” que contém um contrato “ownable”. Este contrato providencia um único “dono” ao *smart contract* de forma a restringir o acesso a certas funções. Como tal, o *smart contract* é propriedade de apenas um endereço, que tem autoridade para adicionar aparelhos, ler os consumos energéticos, a produção e os preços, e com isso processar os dados de acordo com o algoritmo demonstrado anteriormente. Apenas este endereço tem o direito de alterar estes dados, ou seja, qualquer entidade que os tentasse manipular não o conseguiria fazer porque não tem acesso à carteira com o respetivo endereço. Isto faz com que este sistema de gestão de energia seja praticamente impossível de alterar uma vez que está armazenado na *blockchain* da *Ethereum*, reforçando assim a segurança dos dados e garantindo a imutabilidade.

```
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/access/Ownable.sol";
import "@openzeppelin/contracts/utils/Counters.sol";

contract Introspection is Ownable {
    using Counters for Counters.Counter;
    uint256 public constant BATTERY_MAX_CAPACITY = 500;

    struct Appliance {
        uint8 id;
        mapping(uint256 => uint256[96]) energyConsumption;
    }
}
```

Figura 15 - Demonstração do contrato "ownable"

São também utilizados *mappings*, que são uma estrutura de dados típica da linguagem da programação em *solidity*. Os *mappings* são uma tabela de *hash* que armazenam dados de forma criptográfica utilizando pares de chaves, onde a chave pode ser qualquer tipo de dados. Desta forma, apenas o utilizador com o endereço da carteira correto pode aceder aos dados na tabela *hash*, reforçando a segurança e a privacidade dos dados.

```
mapping(uint256 => uint256[96]) public _solarProduction;
mapping(uint256 => uint256[96]) public _batteryStateOfCharge;
mapping(uint256 => uint256[96]) public _gridEnergyBuyPrice;
mapping(uint256 => uint256[96]) public _solarEnergySellPrice;
mapping(uint256 => int256[96]) public _expenditure;
```

Figura 16 - Demonstração da utilização de mappings

De forma a executar o smart contract é necessário inserir os inputs mencionados anteriormente. Para fazer isto, utilizam-se as funções do contrato demonstradas na figura 17. Cada vez que ocorre a introdução de um input é necessário pagar os *gas fees* à rede e, como tal, é necessário utilizar a carteira digital para fazer o pagamento dessas mesmas taxas (figura 10). Após a introdução dos inputs e do pagamento das taxas o contrato está pronto a ser executado.

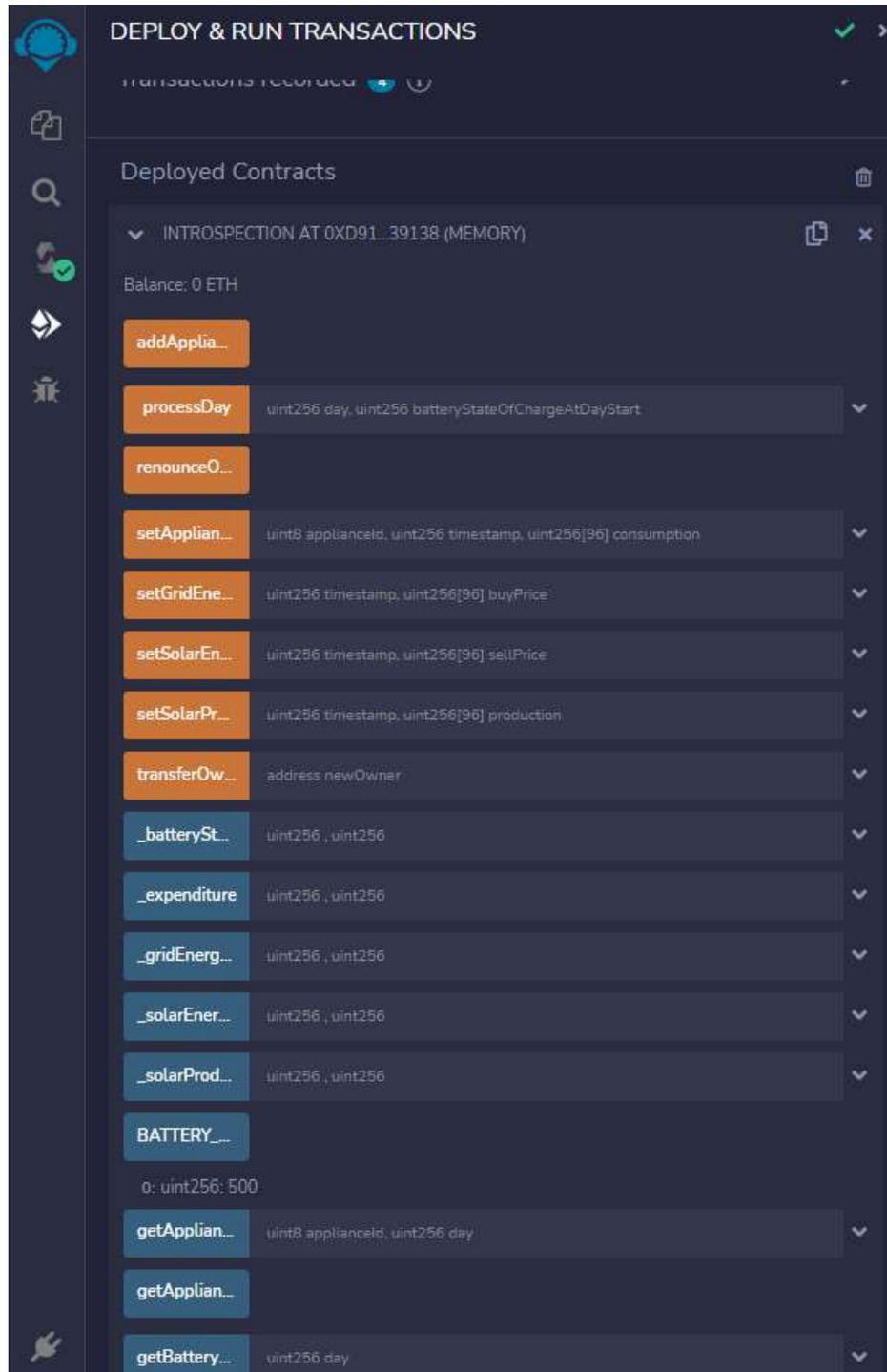


Figura 17 - Interface de interação com o smart contract na plataforma Remix IDE

Capítulo 5

Análise de Resultados

Este capítulo apresenta os resultados obtidos com a implementação do algoritmo/conjunto de regras de gestão de energia local utilizando *smart contracts* e *blockchain*. O objetivo principal desta implementação foi avaliar a eficácia da tecnologia blockchain na gestão da energia numa casa inteligente, com o objetivo de minimizar o custo da fatura de energia.

Os resultados apresentados incluem uma comparação do consumo e produção de energia antes e depois da implementação do algoritmo, bem como uma avaliação do impacto da utilização da tecnologia *blockchain* e dos *smart contracts* na gestão da energia. Ao longo deste capítulo, será possível avaliar se o algoritmo implementado atingiu os objetivos propostos e se a tecnologia blockchain se mostrou eficaz na gestão da energia em uma casa inteligente.

Cenário sem controlo

Na figura 18 é possível ver o consumo e a produção de energia da casa ao longo do dia antes da implementação do algoritmo implementado no *smart contract*. Este gráfico apresenta o padrão de consumo de energia, mostrando os momentos em que há mais necessidade de energia como por exemplo entre as 17:00h e as 22:00h, o que é compreensível dado que correspondem aos períodos horários em que as pessoas chegam a casa depois do trabalho. No cenário sem controlo apenas 52,89% da energia produzida é utilizada pelas cargas e pela bateria. Como tal, 47,11% da energia produzida é vendida à rede.

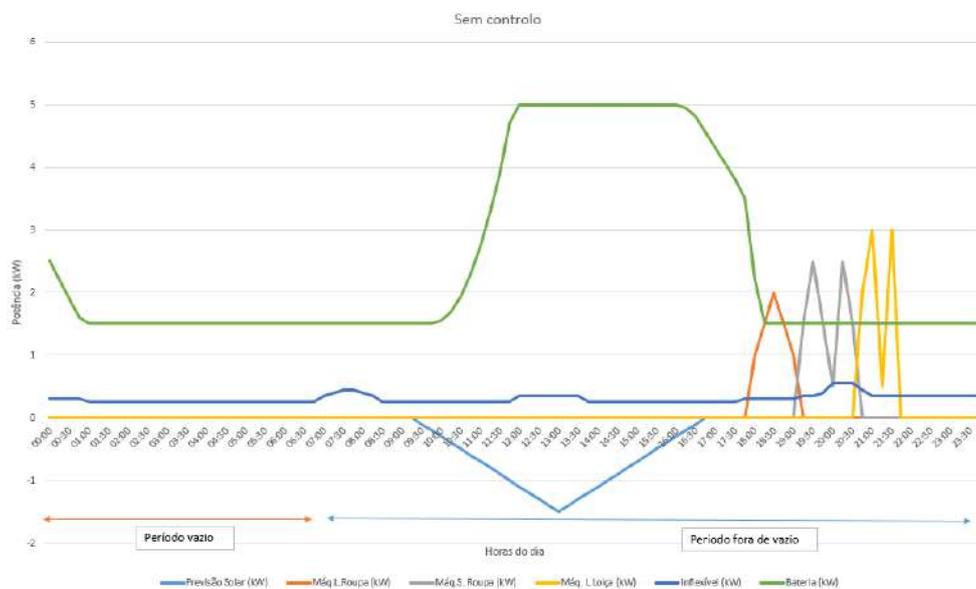


Figura 58 - Representação dos consumos, da produção e da bateria no cenário sem controlo

Cenário com controlo

Na figura 19 é apresentado o diagrama de carga da casa com o algoritmo implementado. Como mencionado anteriormente, faz sentido colocar as cargas flexíveis a funcionar quando a produção de energia é superior, como tal, o algoritmo implementado no *smart contract* faz com que as cargas entrem em funcionamento nesses mesmos períodos. No cenário com controlo, toda a energia produzida pelo painel solar é utilizada ou pelas cargas ou para armazenar energia na bateria, ou seja, há uma rentabilização da energia produzida de 100%.



Figura 19 - Representação dos consumos, da produção e da bateria no cenário com controlo

Após a implementação do algoritmo, chegou-se à conclusão de que houve uma poupança na fatura de energia de cerca de 20%. De acordo com [79,80,81], a implementação de um HEMS pode reduzir o preço da fatura energética num range entre 4% a 40%. Como tal, o algoritmo implementado, apesar de não estar otimizado consegue entrar no range de poupanças face aos sistemas convencionais utilizados. No entanto, é importante notar que a implementação do *smart contract* exigiu o pagamento de *gas fees*.

Apesar da redução de 20% na fatura energética, a introdução dos inputs e a posterior implementação do *smart contract* na *blockchain* ocasionaram um custo aproximadamente 0,0319 ETH. Tendo em conta o preço atual do *ether* (1583,22\$, de acordo com o *coinmarketcap*), a implementação do contrato custaria ao utilizador cerca de 50\$. Como tal, a implementação do *smart contract* não é justificada face aos preços exigidos pela rede da *Ethereum*. Para além disso,

o *smart contract* exige que sempre que se pretende inserir um input, haja a necessidade de cobrar um *gas fee*. Como tal, desta forma o algoritmo não poderia ser implementado conforme está num cenário real. De acordo com alguma pesquisa feita, uma possível solução para este problema passa pela instalação de um *bot* que automatiza os pagamentos de acordo com certas regras estipuladas. No entanto, esta questão deve ser analisada e deve ser tida em linha de conta em aplicações futuras.

5.1 Cyber-segurança e privacidade

Nesta dissertação foi proposta uma solução para a gestão local de energia numa casa utilizando *smart contracts*. Desta forma, proporciona-se uma maior transparência e confiança no processo de gestão de energia e automatização de certas tarefas, uma vez que todas as transações efetuadas no âmbito do sistema de gestão podem ser vistas (apesar de serem encriptadas) por toda a gente. Contudo, há também questões de segurança e privacidade que precisam de ser consideradas antes da implementação de *smart contracts*.

Apesar dos *smart contracts* estarem armazenados na *blockchain* é possível que os mesmos contenham erros ou vulnerabilidades que podem ser exploradas por piratas informáticos. Uma vez que os *smart contracts* contêm informações sensíveis, se estes não forem devidamente protegidos, poderão ser comprometidos, o que pode levar a violações de privacidade ou outras consequências negativas. Para além disso, o facto dos *smart contracts* serem imutáveis pode ser visto como uma desvantagem em certos casos, na medida em que se houver erros no código podem ocorrer resultados inesperados.

De forma a mitigar estes problemas, é crucial que os *smart contracts* sejam concebidos com medidas de segurança robustas, tais como práticas de codificação seguras e auditorias de segurança regulares.

Outra forma de piratas informáticos ganharem acesso aos dados passa pelo roubo da chave privada da carteira digital do utilizador. Como tal, é crucial que o titular da conta esteja ciente da importância de manter a chave privada segura e não a partilhar com ninguém em circunstância alguma.

5.2 Riscos de implementação e comunicação

Há certos riscos que acompanham a implementação e a comunicação dos *smart contracts* no setor energético. Apesar de muitos estudos serem apenas simulações, alguns investigadores já implementaram *smart contracts* em *blockchain* operacionais, principalmente em *Ethereum* e *blockchains* privadas. Também há alguns estudos que utilizaram *raspberry pi* para simular uma *blockchain* privada de forma a tornar a experiência mais real. No entanto, estas experiências podem não ter totalmente em conta os impactos da latência da rede no mundo real ou os erros de

comunicação. As questões de comunicação podem desempenhar um papel significativo, especialmente em *smart contracts* que têm uma implementação em tempo real.

5.3 Gastos computacionais dos smart contracts

A execução dos *smart contracts* exige a utilização de poder computacional e de espaço de armazenamento na rede da *blockchain*. Como referido anteriormente, a utilização deste poder computacional e deste armazenamento é medida em termos de ‘gas’. Quanto mais complexo for o *smart contract*, quanto maiores forem o número de operações que ele tiver de desenvolver, quanto maior for a quantidade de dados que o mesmo tiver de aceder, maior será o ‘gas’ que o mesmo vai consumir. *Smart contracts* que exijam alto poder computacional vão ser difíceis de ser eficientes e vantajosos, uma vez que os gastos associados à implementação do mesmo serão bastante elevados. Como tal, é crucial que se diminua a quantidade de recursos computacionais necessários para executar o *smart contract*. Uma das formas para minimizar estes recursos é através da otimização da escrita do código. Isto pode ser atingido reduzindo o número de operações, diminuindo a quantidade de informação armazenada, e utilizar algoritmos mais eficientes. Outra solução pode passar por utilizar poder computacional fora da *blockchain* (‘off-chain’), desta forma, só a parte do código crucial estará implementada dentro da *blockchain*, reduzindo a quantidade de ‘gas’ necessária para a implementação do *smart contract*. Para além disso, a utilização de uma *blockchain* mais eficiente também pode contribuir em muito para reduzir os gastos computacionais do *smart contract*.

5.4 Requisitos de software

Como referido anteriormente, para a implementação de um *smart contract* são necessários alguns requisitos. Nomeadamente, um ambiente de implementação do *smart contract*, a linguagem de programação, a *blockchain* a utilizar, uma carteira digital que permita interagir com o contrato e o conhecimento dos protocolos de consenso, segurança e escalabilidade. A maioria dos investigadores não estão familiarizados com a utilização de linguagens de programação de *smart contracts* o que afeta a investigação nesta área (apesar de já haver algumas tentativas de criar protocolos de comunicação com ferramentas como o simulink). Para além disso, a linguagem de programação mais utilizada – *solidity*, tem algumas restrições e limitações como por exemplo:

- a) Utilização de funções recursivas: Apesar do *solidity* permitir a escrita de funções recursivas, é importante notar de que as mesmas devem ser escritas com o máximo cuidado, uma vez que podem levar a que o contrato gaste uma grande quantidade de ‘gas’.
- b) Utilização de decimais: a utilização de *smart contracts* no setor energético necessitam de realizar cálculos com números decimais para calcular consumos energéticos e preços, por

exemplo. Contudo, o *solidity* não permite a escrita e a leitura de números decimais o que pode tornar mais difícil a implementação do código.

- c) Alocação de memória: *smart contracts* no setor energético poderão precisar de armazenar grandes quantidades de dados como dados de consumo ao longo do tempo. A linguagem de programação *solidity* pode tornar pouco eficiente o acesso a grandes quantidades de dados devido à quantidade de ‘gas’ utilizada.
- d) Comunicação com outros contratos: pode haver necessidade de interagir com outros *smart contracts* ou sistemas externos (faturação, contagens, gestão de dispositivos) o que pode ser complicado de realizar em *solidity* devido a problemas de interoperabilidade, resultando em erros inesperados e vulnerabilidades.

5.5 Regulação

A regulação dos *smart contracts* no setor energético é uma questão importante a ter em conta de modo a se garantir a eficiência e a eficácia da sua utilização. A regulação adequada pode ajudar a garantir a segurança, a transparência e a integridade dos contratos, para além de promover a confiança nas transações de energia através de *smart contracts*. As questões regulatórias incluem a responsabilidade dos participantes em casos de erros ou falhas, a definição de regras para a resolução de disputas, etc. Para além disso, a regulação ajuda com a garantia de interoperabilidade entre diferentes sistemas. Em suma, a regulação dos *smart contracts* é uma área que precisa de ser abordada para promover a aceleração da integração dos mesmos no setor energético.

Capítulo 6

Conclusão e trabalhos futuros

A realização de toda a pesquisa, investigação e implementação levada a cabo no âmbito desta dissertação, faz com que seja possível concluir que, apesar de serem tecnologias relativamente recentes, os *smart contracts*, a par com a tecnologia *blockchain*, têm o potencial de transformar de uma forma positiva o setor energético.

De acordo com a bibliografia explorada, a tecnologia *blockchain* reduz a necessidade de elementos terciários, aumenta a segurança e a transparência das transações o que pode ser uma mais valia muito grande no ramo das energias. A aplicação de *smart contracts* num sistema de gestão local de energia, onde haja a troca de informações com o exterior, pode ser fulcral para prevenir que haja a adulteração de dados e também para prevenir que os mesmos possam ser utilizados por terceiros para benefício próprio. Para além disso, a utilização destas tecnologias num cenário de gestão/transação local pode vir a reduzir os custos de manutenção e implementação face aos sistemas atuais. Em suma, a utilização de *smart contracts* e *blockchain* podem abrir o leque de oportunidades existentes no setor, permitindo o comércio local de energia (comércio P2P) gerada por fontes descentralizadas, assim como a automatização das transações de energia, etc.

O algoritmo/conjunto de regras implementado no *smart contract* teve um carácter maioritariamente exploratório, ou seja, um dos principais objetivos foi o de perceber quais os prós e contras das tecnologias utilizadas neste tipo de cenário. No entanto, apesar da simplicidade do algoritmo implementado, conseguiu-se uma redução no preço da fatura energética de 20%. Sendo que esta redução se enquadra no range médio dos resultados obtidos por sistemas de gestão de energia convencionais. A implementação do *smart contract* foi feita na *blockchain* da *Ethereum*, utilizando a plataforma Remix IDE, na rede de teste Goerli. Através desta implementação foi possível concluir que, em termos de cenários simulatórios (redes de teste), a rede Goerli não é a melhor opção a ser utilizada, dado que restringe o programador na quantidade de transações que pode fazer por dia. Como tal, aconselha-se a utilização de por exemplo um *local hardhat node* que permite conectar a *blockchain* da *Ethereum* ao computador do programador. Desta forma, elimina-se o problema da restrição do número de testes diários. Quanto à plataforma *blockchain* utilizada, a escolha recai sobre diversos fatores a ter em linha de conta, entre eles: o tipo de mecanismo de *consensus* a ser utilizado, a escalabilidade, a segurança, a interoperabilidade, o custo e a adoção. Nesta dissertação foi utilizada a *blockchain* da *Ethereum*, que ganha pontos no mecanismo de *consensus*, na segurança e na adoção. No entanto, seria extremamente interessante,

em trabalhos futuros, utilizar-se uma *blockchain* como por exemplo a *Hyperledger Fabric*, que de acordo com a literatura analisada é uma *blockchain* que foi desenhada para ser altamente escalável e que consegue processar um grande número de transações por segundo (ao contrário da *Ethereum* que apenas consegue processar 15 transações por segundo). Este aumento de escalabilidade, apesar de ser acompanhado de uma redução da segurança e na adoção, parece ser uma mais valia. Em sistemas de gestão local de energia, a troca de dados e o número de transações acontece praticamente a todo o instante, como tal, pode ser preferível optar por escolher uma plataforma que consiga processar mais transações ainda que se limite a segurança. Para além das alterações nas plataformas utilizadas, em trabalhos futuros podem ser investigadas formas de melhorar o algoritmo, nomeadamente através de mecanismos de inteligência artificial e técnicas de *machine learning*. Isto permitirá, obter um controlo/gestão dos recursos de uma forma otimizada e adaptada a cada utilizador.

Posto isto, de forma a terminar a análise feita até então, acredita-se ser construtivo apresentar algumas recomendações práticas ou pontos de atenção para a implementação futura de *smart contracts* no domínio da energia.

1. Primeiro, é importante perguntar se os *smart contracts* são a ferramenta certa para o problema específico em consideração. Embora os *smart contracts* possam ser ferramentas poderosas na construção de sistemas de energia descentralizados, esta necessidade de descentralização deve sempre ser equiparada com as potenciais desvantagens, como os elevados riscos de segurança, o elevado consumo de energia e a alta complexidade de implementação. Para muitas aplicações, como foi o caso implementado nesta dissertação, uma das potenciais soluções poderia passar por considerar quais os cálculos, e quais os dados, devem ser feitos/armazenados dentro do contrato inteligente e executados na *blockchain*, e quais o cálculo (como por exemplo algoritmos complexos/*machine-learning*) devem ser executados fora da *blockchain*. Fazer estes ajustes pode ajudar a reduzir bastante as taxas de utilização da rede, uma vez que se diminui o poder computacional necessário.
2. Ao realizar a implementação de um *smart contract*, é crucial considerar aspetos de *cyber*-segurança e ter uma compreensão completa do código implementado. Como referido anteriormente, uma vez que o *smart contract* é executado de forma automática e é à prova de adulteração, devem ser analisadas cuidadosamente as ameaças de *cyber*-segurança de forma a não haverem (ou pelo menos serem reduzidas) possíveis explorações do código por parte de *hackers*.
3. Outro fator muito importante a ser levado em linha de conta são os custos de implementação do *smart contract*. Isto é, custos financeiros (em termos de *gas fees*) e custos ambientais e energéticos, na execução dos protocolos *proof-of-work*. Até agora, a

maioria dos projetos no setor energético, têm sido implementados em *blockchains* privadas. No entanto, à medida que os projetos se tornam mais comerciais podem vir a ser executados em *blockchains* públicas. Como tal, é crucial avaliar quais os custos da implementação dos *smart contracts* nesse cenário, uma vez que se houver uma poupança global de energia inferior com a energia adicional necessária para o funcionamento do *smart contract*, a utilização destas tecnologias torna-se muito questionável.

4. Por último, é de notar que a utilização de um *smart contract* num projeto/plataforma de energia, não tornará o projeto, por si só “inteligente”. Serão precisas técnicas sofisticadas de *machine learning* para uma utilização eficiente dos recursos e uma gestão/controlo inteligente. Muitas vezes a inovação provém das técnicas de aprendizagem computacional, e não da implementação do *smart contract*. No entanto, conclui-se que, se os *smart contracts* forem implementados de forma correta em combinação com outras técnicas, os mesmos são uma tecnologia potencialmente poderosa que pode desempenhar um papel crucial na viabilização dos sistemas energéticos verdadeiramente descentralizados do futuro.

Bibliografia

- [1] - Friedlingstein, P., Jones, M. W., O'sullivan, M., Andrew, R. M., Hauck, J., Peters, G. P., ... & Zaehle, S. (2019). Global carbon budget 2019. *Earth System Science Data*, 11(4), 1783-1838.
- [2] - Marques, T. M. (2022). *Development and analysis of a local energy market using blockchain* (dissertation).
- [3] - Kirli, D., Couraud, B., Robu, V., Salgado-Bravo, M., Norbu, S., Andoni, M., ... & Kiprakis, A. (2022). Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. *Renewable and Sustainable Energy Reviews*, 158, 112013.
- [4] - Korkmaz, A., Kılıç, E., Türkay, M., Çakmak, Ö. F., Arslan, T. Y., & Erdoğan, U. (2021). A Blockchain Based P2P Energy Trading Solution for Smart Grids. *Researchgate. net*
- [5] - Hrga, A., Capuder, T., & Žarko, I. P. (2020). Demystifying distributed ledger technologies: Limits, challenges, and potentials in the energy sector. *IEEE Access*, 8, 126149-126163.
- [6] - Foxbit. (2022, February 17). *O que são bases de dados centralizadas, descentralizadas E distribuídas?* Foxbit. Retrieved February 14, 2023, from <https://foxbit.com.br/blog/diferenca-entre-as-bases-de-dados-blockchain/>
- [7] - Rejeb, A., Rejeb, K., & Keogh, J. G. (2021). Centralized vs. decentralized ledgers in the money supply process: a SWOT analysis. *Quantitative Finance and Economics*, 5(1), 40-66.
- [8] - Schollmeier, R. (2001, August). A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings first international conference on peer-to-peer computing* (pp. 101-102). IEEE.
- [9] - Kang, E. S., Pee, S. J., Song, J. G., & Jang, J. W. (2018, April). A blockchain-based energy trading platform for smart homes in a microgrid. In *2018 3rd international conference on computer and communication systems (ICCCS)* (pp. 472-476). IEEE.
- [10] - Shahaab, A., Lidgley, B., Hewage, C., & Khan, I. (2019). Applicability and appropriateness of distributed ledgers consensus protocols in public and private sectors: A systematic review. *IEEE access*, 7, 43622-43636.
- [11] - Troy, S., & Pratt, M. K. (2021, June 3). *Distributed Ledger Technology (DLT)*. CIO. Retrieved February 14, 2023, from <https://www.techtarget.com/searchcio/definition/distributed-ledger>
- [12] - Mattila, J., Seppälä, T., Naucler, C., Stahl, R., Tikkanen, M., Bådenlid, A., & Seppälä, J. (2016). *Industrial blockchain platforms: An exercise in use case development in the energy industry* (No. 43). ETLA Working Papers.
- [13] - Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50-58.
- [14] - Winter, T. (2018). The advantages and challenges of the blockchain for smart grids.
- [15] - Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 21260.
- [16] - Szabo, N. (1996). Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought*, 18(2), 28.
- [17] - Dahlquist, O., & Hagström, L. (2017). Scaling blockchain for the energy sector.
- [18] - Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., ... & Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and sustainable energy reviews*, 100, 143-174.

- [19] - Hardjono, T., Kim, A., & Pentland, A. (2020). 7. Health IT: Algorithms, Privacy, and Data. *Building the New Economy*.
- [20] – Miglani, A., Kumar, N., Chamola, V., & Zeadally, S. (2020). Blockchain for Internet of Energy management: Review, solutions, and challenges. *Computer Communications*, 151, 395-418.
- [21] – Macrinici, D., Cartofeanu, C., & Gao, S. (2018). Smart contract applications within blockchain technology: A systematic mapping study. *Telematics and Informatics*, 35(8), 2337-2354.
- [22] – R3. Barclays smart contract templates. 2021, <https://relayto.com/hub/barclayssmart-contract-templates-582b3a01802d7>. [Accessed 10 March 2021].
- [23] – Han, D., Zhang, C., Ping, J., & Yan, Z. (2020). Smart contract architecture for decentralized energy trading and management based on blockchains. *Energy*, 199, 117417.
- [24] – Macrinici D, Cartofeanu C, Gao S. Smart contract applications within blockchain technology: A systematic mapping study. *Telemat Inform* 2018;35(8):2337–54. <http://dx.doi.org/10.1016/j.tele.2018.10.004>.
- [25] – Zeng, Z., Li, Y., Cao, Y., Zhao, Y., Zhong, J., Sidorov, D., & Zeng, X. (2020). Blockchain technology for information security of the energy internet: Fundamentals, features, strategy and application. *Energies*, 13(4), 881.
- [26] – Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014), 1-32.
- [27] – Park, J., Lee, D., & In, H. P. (2017). Saving deployment costs of smart contracts by eliminating gas-wasteful patterns. *International Journal of Grid and Distributed Computing*, 10(12), 53-64.
- [28] - Chen, T., Li, X., Luo, X., & Zhang, X. (2017, February). Under-optimized smart contracts devour your money. In *2017 IEEE 24th international conference on software analysis, evolution and reengineering (SANER)* (pp. 442-446). IEEE.
- [29] - Van Leeuwen G, AlSkaif T, Gibescu M, van Sark W. An integrated blockchainbased energy management platform with bilateral trading for microgrid communities. *Appl Energy* 2020;263. <http://dx.doi.org/10.1016/j.apenergy.2020.114613>.
- [30] – Mustafa, M. A., Cleemput, S., Aly, A., & Abidin, A. (2019). A secure and privacy-preserving protocol for smart metering operational data collection. *IEEE Transactions on Smart Grid*, 10(6), 6481-6490.
- [31] – Khalid R, Javaid N, Almogren A, Javed M, Javaid S, Zuair M. A blockchainbased load balancing in decentralized hybrid P2P energy trading market in smart grid. *IEEE Access* 2020;8:47047–62. <http://dx.doi.org/10.1109/ACCESS.2020.2979051>.
- [32] –Karuseva, N., Livshits, S., Kotsubinski, A., Yudina, N., Novikova, O., & Tabakova, A. (2019). The impact of innovative technologies on consumers in the power supply market. In *E3S web of conferences* (Vol. 140, p. 04009). EDP Sciences.
- [33] –Robu, V., Vinyals, M., Rogers, A., & Jennings, N. R. (2017). Efficient buyer groups with prediction-of-use electricity tariffs. *IEEE Transactions on Smart Grid*, 9(5), 4468-4479.
- [34] – Kouveliotis-Lysikatos, I., Kokos, I., Lamprinos, I., & Hatziargyriou, N. (2019, September). Blockchain-powered applications for smart transactive grids. In *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)* (pp. 1-5). IEEE.
- [35] – Lu, J., Wu, S., Cheng, H., Song, B., & Xiang, Z. (2021). Smart contract for electricity transactions and charge settlements using blockchain. *Applied Stochastic Models in Business and Industry*, 37(3), 442-453.

- [36] – Kunkel, J. M., Shoukourian, H., Heidari, M. R., & Wilde, T. (2019). Interference of billing and scheduling strategies for energy and cost savings in modern data centers. *Sustainable Computing: Informatics and Systems*, 23, 49-66.
- [37] – Nakayama K, Moslemi R, Sharma R. Transactive energy management with blockchain smart contracts for P2P multi-settlement markets. In: 2019 IEEE power and energy society innovative smart grid technologies conference. 2019, <http://dx.doi.org/10.1109/ISGT.2019.8791652>.
- [38] – Di Silvestre, M. L., Gallo, P., Sanseverino, E. R., Sciume, G., & Zizzo, G. (2019, June). A new architecture for Smart Contracts definition in Demand Response Programs. In *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe)* (pp. 1-5). IEEE.
- [39] – Pop, C., Antal, M., Cioara, T., Anghel, I., Sera, D., Salomie, I., ... & Bertoncini, M. (2019). Blockchain-based scalable and tamper-evident solution for registering energy data. *Sensors*, 19(14), 3033.
- [40] – Ahmed, M. A. A., Silva, F. C. S., Martínez, J. M. M., & Kim, Y. C. K. (2019). Design and Implementation of a Blockchain-Based Energy Trading Platform for Electric Vehicles in Smart Campus Parking Lots.
- [41] – Thomas, L., Zhou, Y., Long, C., Wu, J., & Jenkins, N. (2019). A general form of smart contract for decentralized energy systems management. *Nature Energy*, 4(2), 140-149.
- [42] – Thomas, L., Long, C., Burnap, P., Wu, J., & Jenkins, N. (2017). Automation of the supplier role in the GB power system using blockchain based smart contracts.
- [43] – Wang, H., Li, Q., Huang, H., Qu, H., Wang, W., & Meng, T. (2019, November). Research on power data management based on sovereign blockchain technology. In *Journal of Physics: Conference Series* (Vol. 1346, No. 1, p. 012010). IOP Publishing.
- [45] – Hadi, A. A., Sinha, U., Faika, T., Kim, T., Zeng, J., & Ryu, M. H. (2019, September). Internet of things (IoT)-enabled solar micro inverter using blockchain technology. In *2019 IEEE industry applications society annual meeting* (pp. 1-5). IEEE.
- [46] – Ashley, M. J., & Johnson, M. S. (2018). Establishing a secure, transparent, and autonomous blockchain of custody for renewable energy credits and carbon credits. *IEEE Engineering Management Review*, 46(4), 100-102.
- [47] – Castellanos, J. A. F., Coll-Mayor, D., & Notholt, J. A. (2017, August). Cryptocurrency as guarantees of origin: Simulating a green certificate market with the Ethereum Blockchain. In *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)* (pp. 367-372). IEEE.
- [48] – Makhdoom, I., Zhou, I., Abolhasan, M., Lipman, J., & Ni, W. (2020). PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities. *Computers & Security*, 88, 101653.
- [49] – Kang, E. S., Pee, S. J., Song, J. G., & Jang, J. W. (2018, April). A blockchain-based energy trading platform for smart homes in a microgrid. In *2018 3rd international conference on computer and communication systems (ICCCS)* (pp. 472-476). IEEE.
- [50] – Bindra, L., Eng, K., Ou, Y., Ardakanian, O., & Stroulia, E. (2019, November). Indoor path planning and decentralized access control in commercial buildings. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (pp. 385-386).
- [51] – Liu, H., Zhang, Y., Zheng, S., & Li, Y. (2019). Electric vehicle power trading mechanism based on blockchain and smart contract in V2G network. *IEEE Access*, 7, 160546-160558.

- [52] – Abidin, A., Aly, A., Cleemput, S., & Mustafa, M. A. (2018). Secure and privacy-friendly local electricity trading and billing in smart grid. *arXiv preprint arXiv:1801.08354*.
- [53] – Han, D., Zhang, C., Ping, J., & Yan, Z. (2020). Smart contract architecture for decentralized energy trading and management based on blockchains. *Energy*, 199, 117417.
- [54] – van Leeuwen G, AlSkaif T, Gibescu M, van Sark W. An integrated blockchainbased energy management platform with bilateral trading for microgrid communities. *Appl Energy* 2020;263. <http://dx.doi.org/10.1016/j.apenergy.2020.114613>.
- [55] – Duan, B., Xin, K., & Zhong, Y. (2019). Optimal dispatching of electric vehicles based on smart contract and internet of things. *IEEE Access*, 8, 9630-9639.
- [56] – Dang, T. L. N., & Nguyen, M. S. (2018, November). An approach to data privacy in smart home using blockchain technology. In *2018 International Conference on Advanced Computing and Applications (ACOMP)* (pp. 58-64). IEEE.
- [57] – Zhou, Y., Han, M., Liu, L., Wang, Y., Liang, Y., & Tian, L. (2018, July). Improving iot services in smart-home using blockchain smart contract. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 81-87). IEEE.
- [58] – Hartnett, S., Henly, C., Hesse, E., Hildebrandt, T., Jentzch, C., Krämer, K., ... & Trbovich, A. (2018). The Energy Web Chain-Accelerating the Energy Transition with an Open-Source, Decentralized Blockchain Platform. *Energy Web Foundation*.
- [59] – Mengelkamp, E., Gärtner, J., Rock, K., Kessler, S., Orsini, L., & Weinhardt, C. (2018). Designing microgrid energy markets: A case study: The Brooklyn Microgrid. *Applied energy*, 210, 870-880.
- [60] – Cali, U., Kuzlu, M., Kempf, J., Saha, S. S., Gourisetti, S. N. G., & Hughes, T. (2022, May). Unlocking New Opportunities Towards Smart Contracts for Transactive Energy Systems: Technical and Legislative Perspectives. In *2022 IEEE PES Transactive Energy Systems Conference (TESC)* (pp. 1-5). IEEE.
- [61] – Pando: Engage customers, enable green communities and reshape our energy future with pando, a marketplace solution. 2020, <https://lo3energy.com/pando/>.
- [62] – Energy marketplace: Share with vermont green. 2020, <https://greenmountainpower.com/vermont-green/>.
- [63] - Prosume. The prosume platform. 2020, <https://prosume.io/white-paper/>. [Accessed 16 August 2020].
- [64] – Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Yellick, J. (2018, April). Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference* (pp. 1-15).
- [65] – IBM. Energy blockchain labs inc., . 2020, <https://www.ibm.com/case-studies/energy-blockchain-labs-inc>.
- [66] – ZFCar eWallet GmbH. Car ewallet webpage. 2020, <https://car-ewallet.de/>.
- [67] - Sunchain. Sunchain. 2020, <https://www.sunchain.fr/>.
- [68] - TenneT Holding BV 2020. Tennet unlocks distributed flexibility via blockchain. 2020, <https://www.tennet.eu/news/detail/tennet-unlocksdistributed-flexibility-via-blockchain/>. [Accessed 16 August 2020].
- [69] - Share and Charge. Share and charge. 2020, <https://shareandcharge.com/>.

- [70] - Van Cutsem, O., Dac, D. H., Boudou, P., & Kayal, M. (2020). Cooperative energy management of a community of smart-buildings: A Blockchain approach. *International Journal of electrical power & energy systems*, 117, 105643.
- [71] – Faqiry, M. N., Wang, L., Wu, H., Krishnamurthy, D., & Palmintier, B. (2018, August). ADP-based home energy management system: a case study using DYNAMO. In *2018 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1-5). IEEE.
- [72] – Ahmed, M. S., Mohamed, A., Khatib, T., Shareef, H., Homod, R. Z., & Abd Ali, J. (2017). Real time optimal schedule controller for home energy management system using new binary backtracking search algorithm. *Energy and Buildings*, 138, 215-227.
- [73] - CNBC. (2017, September 6). *Hackers gain entry into US, European Energy Sector, Symantec warns*. CNBC. Retrieved February 14, 2023, from <https://www.cnbc.com/2017/09/06/hackers-gain-entry-into-us-european-energy-sector-symantec-warns.html>
- [74] – Anuebunwa, U. R., Rajamani, H. S., Abd-Alhameed, R., & Pillai, P. (2018, August). Investigating the impacts of cyber-attacks on pricing data of home energy management systems in demand response programs. In *2018 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1-5). IEEE.
- [75] - Cminor. (2021, August 13). *How to assess building energy management system costs*. Frazier Service Company. Retrieved February 14, 2023, from <https://frazierservicecompany.com/assess-energy-management-system-costs/>
- [76] - *Ethereum IDE & Community*. Remix. (n.d.). Retrieved February 14, 2023, from <https://remix-project.org/>
- [77] - *Goerli faucet*. (n.d.). Retrieved February 14, 2023, from <https://goerlifaucet.com/>
- [78] – Diagoosparrow. (n.d.). *Diagoosparrow/HEMS-smart-contract*. GitHub. Retrieved February 14, 2023, from <https://github.com/diagoosparrow/HEMS-Smart-Contract>
- [79] – Tuomela, S., de Castro Tomé, M., Iivari, N., & Svento, R. (2021). Impacts of home energy management systems on electricity consumption. *Applied Energy*, 299, 117310.
- [80] – Ruth, M., Pratt, A., Lunacek, M., Mittal, S., Wu, H., & Jones, W. (2015). *Effects of home energy management systems on distribution utilities and feeders under various market structures* (No. NREL/CP-6A20-63500). National Renewable Energy Lab.(NREL), Golden, CO (United States).
- [81] - Partnerships, N. E. E. (2015). Opportunities for home energy management systems (HEMS) in advancing residential energy efficiency programs.
- [82] – Soares, A., Gomes, Á., & Antunes, C. H. (2014). Categorization of residential electricity consumption as a basis for the assessment of the impacts of demand response actions. *Renewable and Sustainable Energy Reviews*, 30, 490-503.
- [83] - Moniruzzaman, M., Khezr, S., Yassine, A., & Benlamri, R. (2020). Blockchain for smart homes: Review of current trends and research challenges. *Computers & Electrical Engineering*, 83, 106585.
- [84] - Veras, J. M., Silva, I. R. S., Pinheiro, P. R., Rabêlo, R. A., Veloso, A. F. S., Borges, F. A. S., & Rodrigues, J. J. (2018). A multi-objective demand response optimization model for scheduling loads in a home energy management system. *Sensors*, 18(10), 3207.

Apêndices

Apêndice A

	Previsão Solar (kW)	Máq.L.Roupa (kW)	Máq.S. Roupa (kW)	Máq. L.Loixa (kW)	Inflexível (kW)	Bateria (kW)
00:00	0	0	0	0	0,3	2,5
00:15	0	0	0	0	0,3	2,2
00:30	0	0	0	0	0,3	1,9
00:45	0	0	0	0	0,3	1,6
01:00	0	0	0	0	0,25	1,5
01:15	0	0	0	0	0,25	1,5
01:30	0	0	0	0	0,25	1,5
01:45	0	0	0	0	0,25	1,5
02:00	0	0	0	0	0,25	1,5
02:15	0	0	0	0	0,25	1,5
02:30	0	0	0	0	0,25	1,5
02:45	0	0	0	0	0,25	1,5
03:00	0	0	0	0	0,25	1,5
03:15	0	0	0	0	0,25	1,5
03:30	0	0	0	0	0,25	1,5
03:45	0	0	0	0	0,25	1,5
04:00	0	0	0	0	0,25	1,5
04:15	0	0	0	0	0,25	1,5
04:30	0	0	0	0	0,25	1,5
04:45	0	0	0	0	0,25	1,5
05:00	0	0	0	0	0,25	1,5
05:15	0	0	0	0	0,25	1,5
05:30	0	0	0	0	0,25	1,5
05:45	0	0	0	0	0,25	1,5
06:00	0	0	0	0	0,25	1,5
06:15	0	0	0	0	0,25	1,5
06:30	0	0	0	0	0,25	1,5
06:45	0	0	0	0	0,25	1,5
07:00	0	0	0	0	0,35	1,5
07:15	0	0	0	0	0,4	1,5
07:30	0	0	0	0	0,45	1,5
07:45	0	0	0	0	0,45	1,5
08:00	0	0	0	0	0,4	1,5
08:15	0	0	0	0	0,35	1,5
08:30	0	0	0	0	0,25	1,5
08:45	0	0	0	0	0,25	1,5
09:00	0	0	0	0	0,25	1,5
09:15	0	0	0	0	0,25	1,5
09:30	-0,1	0	0	0	0,25	1,5
09:45	-0,2	0	0	0	0,25	1,5
10:00	-0,3	0	0	0	0,25	1,55
10:15	-0,4	0	0	0	0,25	1,7
10:30	-0,5	0	0	0	0,25	1,95
10:45	-0,6	0	0	0	0,25	2,3
11:00	-0,7	0	0	0	0,25	2,75
11:15	-0,8	0	0	0	0,25	3,3
11:30	-0,9	0	0	0	0,25	3,95
11:45	-1	0	0	0	0,25	4
12:00	-1,1	0	0	0	0,35	5
12:15	-1,2	0	0	0	0,35	5
12:30	-1,3	0	0	0	0,35	5
12:45	-1,4	0	0	0	0,35	5
13:00	-1,5	0	0	0	0,35	5
13:15	-1,4	0	0	0	0,35	5
13:30	-1,3	0	0	0	0,35	5
13:45	-1,2	0	0	0	0,25	5
14:00	-1,1	0	0	0	0,25	5
14:15	-1	0	0	0	0,25	5
14:30	-0,9	0	0	0	0,25	5
14:45	-0,8	0	0	0	0,25	5
15:00	-0,7	0	0	0	0,25	5
15:15	-0,6	0	0	0	0,25	5
15:30	-0,5	0	0	0	0,25	5
15:45	-0,4	0	0	0	0,25	5
16:00	-0,3	0	0	0	0,25	5
16:15	-0,2	0	0	0	0,25	4,95
16:30	-0,1	0	0	0	0,25	4,8
16:45	0	0	0	0	0,25	4,55
17:00	0	0	0	0	0,25	4,3
17:15	0	0	0	0	0,25	4,05
17:30	0	0	0	0	0,25	3,8
17:45	0	0	0	0	0,3	3,5
18:00	0	1	0	0	0,3	2,2
18:15	0	1,5	0	0	0,3	1,5
18:30	0	2	0	0	0,3	1,5
18:45	0	1,5	0	0	0,3	1,5
19:00	0	1	0	0	0,3	1,5
19:15	0	0	1,5	0	0,35	1,5
19:30	0	0	2,5	0	0,35	1,5
19:45	0	0	1,5	0	0,4	1,5
20:00	0	0	0,5	0	0,55	1,5
20:15	0	0	2,5	0	0,55	1,5
20:30	0	0	1,5	0	0,55	1,5
20:45	0	0	0	2	0,45	1,5
21:00	0	0	0	3	0,35	1,5
21:15	0	0	0	0,5	0,35	1,5
21:30	0	0	0	3	0,35	1,5
21:45	0	0	0	0	0,35	1,5
22:00	0	0	0	0	0,35	1,5
22:15	0	0	0	0	0,35	1,5
22:30	0	0	0	0	0,35	1,5
22:45	0	0	0	0	0,35	1,5
23:00	0	0	0	0	0,35	1,5
23:15	0	0	0	0	0,35	1,5
23:30	0	0	0	0	0,35	1,5
23:45	0	0	0	0	0,35	1,5

Apêndice B

	Previsão Solar (kW)	Máq.L.Roupa (kW)	Máq.S. Roupa (kW)	Máq. L.Loíça (kW)	Inflexível (kW)	Bateria (kW)
00:00	0	0	0	0	0,3	2,5
00:15	0	0	0	0	0,3	2,2
00:30	0	0	0	0	0,3	1,9
00:45	0	0	0	0	0,3	1,6
01:00	0	0	0	0	0,25	1,5
01:15	0	0	0	0	0,25	1,5
01:30	0	0	0	0	0,25	1,5
01:45	0	0	0	0	0,25	1,5
02:00	0	0	0	0	0,25	1,5
02:15	0	0	0	0	0,25	1,5
02:30	0	0	0	0	0,25	1,5
02:45	0	0	0	0	0,25	1,5
03:00	0	0	0	0	0,25	1,5
03:15	0	0	0	0	0,25	1,5
03:30	0	0	0	0	0,25	1,5
03:45	0	0	0	0	0,25	1,5
04:00	0	0	0	0	0,25	1,5
04:15	0	0	0	0	0,25	1,5
04:30	0	0	0	0	0,25	1,5
04:45	0	0	0	0	0,25	1,5
05:00	0	0	0	0	0,25	1,5
05:15	0	0	0	0	0,25	1,5
05:30	0	0	0	0	0,25	1,5
05:45	0	0	0	0	0,25	1,5
06:00	0	0	0	0	0,25	1,5
06:15	0	0	0	0	0,25	1,5
06:30	0	0	0	0	0,25	1,5
06:45	0	0	0	0	0,25	1,5
07:00	0	0	0	0	0,35	1,5
07:15	0	0	0	0	0,4	1,5
07:30	0	0	0	0	0,45	1,5
07:45	0	0	0	0	0,45	1,5
08:00	0	0	0	0	0,4	1,5
08:15	0	0	0	0	0,35	1,5
08:30	0	0	0	0	0,25	1,5
08:45	0	0	0	0	0,25	1,5
09:00	0	0	0	0	0,25	1,5
09:15	0	0	0	0	0,25	1,5
09:30	-0,1	0	0	0	0,25	1,5
09:45	-0,2	0	0	0	0,25	1,5
10:00	-0,3	0	0	0	0,25	1,55
10:15	-0,4	0	0	0	0,25	1,7
10:30	-0,5	0	0	0	0,25	1,95
10:45	-0,6	0	0	0	0,25	2,3
11:00	-0,7	0	0	0	0,25	2,75
11:15	-0,8	0	1,5	0	0,25	1,8
11:30	-0,9	0	2,5	0	0,25	1,5
11:45	-1	0	1,5	0	0,25	1,5
12:00	-1,1	0	0,5	0	0,35	1,75
12:15	-1,2	0	2,5	0	0,35	1,5
12:30	-1,3	0	1,5	0	0,35	1,5
12:45	-1,4	0	0	2	0,35	1,5
13:00	-1,5	0	0	3	0,35	1,5
13:15	-1,4	0	0	0,5	0,35	2,05
13:30	-1,3	0	0	3	0,35	1,5
13:45	-1,2	1	0	0	0,25	1,5
14:00	-1,1	1,5	0	0	0,25	1,5
14:15	-1	2	0	0	0,25	1,5
14:30	-0,9	1,5	0	0	0,25	1,5
14:45	-0,8	1	0	0	0,25	1,5
15:00	-0,7	0	0	0	0,25	1,95
15:15	-0,6	0	0	0	0,25	2,3
15:30	-0,5	0	0	0	0,25	2,55
15:45	-0,4	0	0	0	0,25	2,7
16:00	-0,3	0	0	0	0,25	2,75
16:15	-0,2	0	0	0	0,25	2,7
16:30	-0,1	0	0	0	0,25	2,55
16:45	0	0	0	0	0,25	2,3
17:00	0	0	0	0	0,25	2,05
17:15	0	0	0	0	0,25	1,8
17:30	0	0	0	0	0,25	1,55
17:45	0	0	0	0	0,3	1,5
18:00	0	0	0	0	0,3	1,5
18:15	0	0	0	0	0,3	1,5
18:30	0	0	0	0	0,3	1,5
18:45	0	0	0	0	0,3	1,5
19:00	0	0	0	0	0,3	1,5
19:15	0	0	0	0	0,35	1,5
19:30	0	0	0	0	0,35	1,5
19:45	0	0	0	0	0,4	1,5
20:00	0	0	0	0	0,55	1,5
20:15	0	0	0	0	0,55	1,5
20:30	0	0	0	0	0,55	1,5
20:45	0	0	0	0	0,45	1,5
21:00	0	0	0	0	0,35	1,5
21:15	0	0	0	0	0,35	1,5
21:30	0	0	0	0	0,35	1,5
21:45	0	0	0	0	0,35	1,5
22:00	0	0	0	0	0,35	1,5
22:15	0	0	0	0	0,35	1,5
22:30	0	0	0	0	0,35	1,5
22:45	0	0	0	0	0,35	1,5
23:00	0	0	0	0	0,35	1,5
23:15	0	0	0	0	0,35	1,5
23:30	0	0	0	0	0,35	1,5
23:45	0	0	0	0	0,35	1,5