



UNIVERSIDADE D
COIMBRA

Márcio José Lopes Ribeiro

PROTÓTIPO BLOCKCHAIN PARA CENTRAIS ELÉTRICAS VIRTUAIS

Dissertação no âmbito do Mestrado em Engenharia Informática,
especialização em Engenharia de Software, orientada pelos Professores João
Barata e Paulo Rupino e apresentada ao Departamento de Engenharia
Informática da Faculdade de Ciências e Tecnologia da Universidade de
Coimbra.

Setembro de 2023



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Márcio José Lopes Ribeiro

Protótipo Blockchain para Centrais Elétricas Virtuais

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software, orientada pelos Professores João Barata e Paulo Rupino e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2023

Resumo

Uma Central Elétrica Virtual / *Virtual Power Plant* (VPP) agrega recursos energéticos distribuídos / *distributed energy resources* (DERs) de forma a trabalharem em conjunto como se de uma única central se tratasse tornando-se assim um modelo importante na integração de fontes de energia renovável no setor energético.

Nos últimos anos, a tecnologia *blockchain* tem sido testada para armazenar informações essenciais da VPP, tirando partido das suas características de transparência, imutabilidade, automatização e rastreabilidade. Estas características são promissoras e podem oferecer benefícios à operação deste tipo de centrais.

A primeira fase deste trabalho inclui uma revisão sistemática da literatura conduzida para investigar como é que a tecnologia *blockchain* é utilizada em VPPs. Os resultados da revisão revelam que a tecnologia *blockchain* desempenha um papel fundamental na simplificação de transações de energia e na gestão das operações nas VPPs. As vantagens incluem maior transparência, confiança entre os participantes, maior eficiência energética e redução de custos de energia.

A segunda fase apresenta o desenvolvimento de um protótipo que será utilizado como referência para a implementação de uma VPP. Este protótipo oferece aos seus utilizadores a capacidade de gerir de componentes da VPP, supervisionar as atividades de manutenção, analisar dados dos contadores inteligentes para monitorizar o fluxo de energia e avaliar os dados de sensores dos componentes para monitorizar o estado operacional. A utilização da tecnologia *blockchain* garante a integridade e a transparência dos dados, promovendo a confiança entre as organizações participantes.

Em suma, esta investigação oferece informações valiosas que contribuem para o avanço do panorama energético com *blockchain*, que se encontra atualmente em evolução.

Palavras-Chave

central elétrica virtual, *blockchain*, recurso energético distribuído

Abstract

A Virtual Power Plant (VPP) aggregates distributed energy resources (DERs) so that they work together as if they were a single power plant. This makes it an important model for integrating renewable energy sources into the energy sector.

In recent years, blockchain technology has been tested to store essential VPP information. This is because blockchain has the characteristics of transparency, immutability, automation, and traceability. These characteristics are promising and could offer benefits to the operation of this type of facility.

The first phase of this work includes a systematic literature review conducted to investigate how blockchain technology is used in VPPs. The results of the review revealed that blockchain technology plays a key role in simplifying energy transactions and managing operations in VPPs. The advantages include greater transparency, trust between participants, greater energy efficiency, and reduced energy costs.

The second phase presents the development of a prototype that will be used as a reference for the implementation of a VPP. This prototype offers its users the ability to manage VPP components, supervise maintenance activities, analyze data from smart meters to monitor energy flow, and evaluate data from component sensors to monitor operational status. The use of blockchain technology guarantees data integrity and transparency, promoting trust between the participating organizations.

In summary, this research provides valuable information that contributes to the progress of the energy sector with blockchain, which is currently evolving.

Keywords

virtual power plant, blockchain, distributed energy resource

Índice

Capítulo 1	Introdução	1
1.1	Contexto e Motivação	1
1.2	Objetivos	2
1.3	Estrutura do Documento	2
Capítulo 2	Planeamento	4
2.1	Planeamento	4
2.1.1	Primeiro Semente	4
2.1.2	Segundo Semestre	5
2.2	Riscos	7
Capítulo 3	Conceitos Fundamentais	9
3.1	<i>Blockchain</i>	9
3.2	Virtual Power Plants	14
3.3	<i>Blockchain</i> e Virtual Power Plants	16
Capítulo 4	Revisão Sistemática da Literatura	19
4.1	Metodologia da Revisão	19
4.2	Análise Descritiva dos Artigos	20
4.3	Revisão dos artigos	22
4.4	Casos de Uso	24
Capítulo 5	Requisitos e Arquitetura de Software	31
5.1	Introdução e objetivos do protótipo	31
5.2	Requisitos	32
5.2.1	Requisitos Funcionais	34
5.2.2	Requisitos não funcionais	37
5.3	Arquitetura de Software	38
5.3.1	Diagrama de Contexto	38
5.3.2	Diagrama de Container	39
Capítulo 6	Implementação	42
6.1	Blockchain Hyperledger Fabric	42
6.1.1	Configuração da rede	42
6.1.2	Desenvolvimento dos <i>chaincodes</i>	43
6.2	REST API	48
6.3	Interface gráfica	49
Capítulo 7	Testes	57
7.1	Configuração do ambiente de testes	57
7.2	Testes unitários	57
7.3	Testes de integração	58
7.4	Testes de sistema	59
Capítulo 8	Conclusão	62
Referências	67	

Apêndice A – Casos de uso	75
Apêndice B – Testes unitários	82
Apêndice C – Testes de Integração	91
Apêndice D – Testes de Sistema	96

Acrónimos

ADMM – Alternating Method of Multipliers

API – Application Programming Interface

DER – Distributed Energy Resource

IoT – Internet of Things

P2P – Peer-to-peer

PBFT – Practical Byzantine Fault Tolerance

PoEM – Proof of Energy Market

PoS – Proof of Stake

PoW – Proof of Work

PPoEM – Private Proof of Energy Market

REST – Representational State Transfer

T4IVPP – Tools for Enabling further RES Integration through the adoption of Virtual Power Plants

VPP – Virtual Power Plant

Lista de Figuras

Fig. 1 - Arquitetura geral do projeto T4IVPP.....	2
Fig. 2 - Diagrama de Gantt do primeiro semestre, Fonte: autor.....	4
Fig. 3 - Diagrama de Gantt do segundo semestre (previsto), Fonte: autor.....	5
Fig. 4 - Diagrama de Gantt do segundo semestre (realizado), Fonte: autor.....	6
Fig. 5 - Representação simplificada da <i>blockchain</i> , Fonte: [9].....	9
Fig. 6 - Representação de uma <i>Merkle Tree</i> , Fonte: [6].....	10
Fig. 7 - Estrutura de um bloco, Fonte: [13].....	11
Fig. 8 - Estrutura típica de uma VPP.....	15
Fig. 9 - Exemplo de uma VPP baseada em <i>blockchain</i> , Fonte: [36].....	16
Fig. 10 - Abordagem à pesquisa sistemática, Fonte: autor.....	20
Fig. 11 - Número de artigos por ano de publicação.....	21
Fig. 12 - Distribuição geográfica dos artigos, Fonte: autor.....	22
Fig. 13 - Cenário multi-VPP, Fonte: autor.....	31
Fig. 14 - Casos de uso, Fonte: autor.....	33
Fig. 15 - Diagrama de contexto, Fonte: autor.....	39
Fig. 16 - Container Diagram, Fonte: autor.....	40
Fig. 17 - Ciclo de vida de um componente, Fonte: autor.....	44
Fig. 18 - Ciclo de vida de uma manutenção, Fonte: autor.....	45
Fig. 19 - Ciclo de vida de um alerta, Fonte: autor.....	46
Fig. 20 - Ciclo de vida de uma contagem, Fonte: autor.....	46
Fig. 21 - Dashboard, Fonte: autor.....	49
Fig. 22 - Transferências de energia, Fonte: autor.....	50
Fig. 23 - Registrar componente, Fonte: autor.....	51
Fig. 24 - Lista de componentes, Fonte: autor.....	51
Fig. 25 - Detalhes de um componente, Fonte: autor.....	52
Fig. 26 - Agendar manutenção, Fonte: autor.....	53
Fig. 27 - Lista de manutenções, Fonte: autor.....	54
Fig. 28 - Detalhes de uma manutenção, Fonte: autor.....	55
Fig. 29 - Detalhes de um alerta, Fonte: autor.....	55
Fig. 30 - Resultados de teste de carga, Fonte: autor.....	60
Fig. 31 - Resultados de teste de stress, Fonte: autor.....	61

Lista de Tabelas

Tabela 1 - Comparação entre os tipos de <i>blockchain</i> , Fonte: [13]	11
Tabela 2 - Transações Energéticas	25
Tabela 3 - Gestão VPP	27
Tabela 4 - Caso de uso 9.1.....	34
Tabela 5 - Requisitos funcionais	35
Tabela 6 - Exemplo de teste unitário.....	58
Tabela 7 - Exemplo de teste de integração	59
Tabela 8 - Exemplo de teste de sistema	59

Capítulo 1

Introdução

O presente documento foi desenvolvido no âmbito da dissertação “Protótipo *Blockchain* para Centrais Elétricas Virtuais” elaborada no contexto do Mestrado em Engenharia Informática com especialização em Engenharia de Software da Faculdade de Ciências e Tecnologias da Universidade de Coimbra no ano letivo 2022/2023. Neste primeiro capítulo são descritos o contexto e motivação, os objetivos e a estrutura do documento.

1.1 Contexto e Motivação

Atualmente, no setor energético, existe uma elevada dependência de combustíveis fósseis, como por exemplo em centrais térmicas de produção de energia, sendo estes responsáveis pela elevada emissão de gases de efeito estufa [1]. A produção e o consumo de energia representam mais de 75% das emissões deste tipo de gases [2].

Recentemente, a União Europeia propôs o *European Green Deal* que apresenta um conjunto de ações para tornar a União Europeia mais sustentável e com o objetivo de atingir a neutralidade carbónica até 2050 [2]. Neste acordo, a descarbonização no setor energético é apresentada como um passo crítico para o cumprimento do objetivo.

Alinhados com as metas estabelecidas pela União Europeia, vários países passaram a apresentar planos para reduzir as emissões de gases de efeito estufa através da integração de fontes de energia renovável. Portugal representa um desses países e apresenta o Plano Nacional de Energia e Clima (PNEC) onde traça objetivos até 2030 [3].

Estas mudanças fazem com que apareçam novos desafios no setor energético uma vez que este passa a deixar de estar assente em grandes centrais controláveis e passa a ter várias centrais, mais distribuídas e com maior variabilidade e incerteza.

Uma possível solução para estes novos desafios são as Centrais Elétricas Virtuais / *Virtual Power Plants* (VPP). Uma VPP representa um sistema que agrega recursos energéticos distribuídos, como fontes de energia renovável e sistemas de armazenamento, de forma a trabalharem em conjunto como se fossem uma única central [4].

É neste contexto que nasce o projeto *Tools for Enabling further RES Integration through the adoption of Virtual Power Plants* (T4IVPP). O projeto é promovido por uma união liderada pela entidade VOLTALIA em conjunto com três empresas, a EFACEC, a VISBLUE e a Evolution, e quatro entidades não empresarias sendo estas o Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC), o Instituto Pedro Nunes (IPN), o Instituto de Telecomunicações (IT) e a Universidade de Coimbra (UC). A Fig. 1 apresenta a visão geral do conceito apresentado pelo projeto T4IVPP.

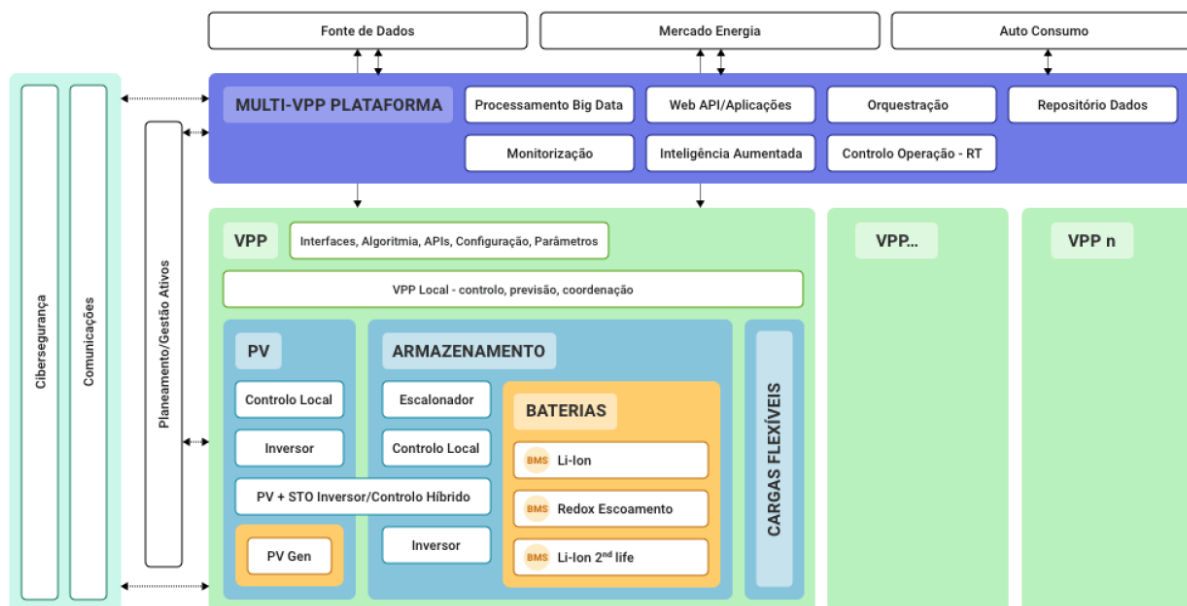


Fig. 1 - Arquitetura geral do projeto T4IVPP

O projeto T4IVPP visa a investigação e desenvolvimento de protótipos de soluções que viabilizem a operação de uma Central Elétrica Virtual / *Virtual Power Plant* (VPP) que possibilite a agregação de diferentes fontes de produção, baseadas em recursos renováveis. O principal objetivo deste projeto, para o qual esta tese contribuirá, é que existam avanços no estado da arte que sirvam de resposta a questões relacionadas com a criação de um sistema distribuído para a supervisão, gestão e o controlo de VPPs não só num modo local, mas também num contexto Multi-VPP onde são agregadas várias centrais elétricas virtuais que se encontram distribuídas geograficamente.

1.2 Objetivos

Enquadrada no contexto do projeto T4IVPP, esta tese tem dois principais objetivos. O primeiro consiste no estudo do potencial da tecnologia *blockchain* para as VPP e a criação de um modelo conceptual para a sua aplicação. O segundo é a criação de um protótipo *blockchain* que permita registar as fontes de produção e armazenamento da VPP, assim como eventos relevantes para a sua operação.

1.3 Estrutura do Documento

O documento encontra-se organizado em diferentes capítulos que são brevemente descritos nos seguintes parágrafos.

O capítulo 2 apresenta o planeamento do trabalho realizado ao longo dos dois semestres. Este planeamento é visualizado através de diagramas de Gantt que fornecem uma visão geral da distribuição das tarefas. Além disso, o capítulo aborda os vários riscos que foram identificados em relação ao cumprimento dos objetivos da dissertação.

No capítulo 3 são apresentados os conceitos fundamentais ao desenvolvimento do trabalho. Em primeiro lugar, é explicado o que é a *blockchain* aprofundando a explicação do conceito às suas características, à sua estrutura, os diferentes tipos de *blockchain*, mecanismos de consenso e

Introdução

smart contracts. De seguida, é apresentado o conceito de VPP e as suas características. Por fim é referida a interseção destes dois conceitos de modo a introduzir o capítulo seguinte.

O capítulo 4 consiste numa revisão sistemática da literatura concebida para responder à seguinte questão: “Como é que a tecnologia *blockchain* é aplicada em *Virtual Power Plants*?”.

No capítulo 5 são apresentados os detalhes de uma solução proposta para apoiar a gestão da operação e manutenção de VPPs auxiliando-se da tecnologia *blockchain*. O capítulo descreve os requisitos e a arquitetura de software previstos para o desenvolvimento do protótipo.

O capítulo 6 apresenta o processo de implementação do protótipo. Em primeiro lugar os detalhes relativos à plataforma Hyperledger Fabric. De seguida, descreve o desenvolvimento de uma REST API que permite a interação com a *blockchain* através de outras aplicações ou de páginas Web. Por fim, apresenta a interface gráfica.

O capítulo 7 apresenta os testes que foram realizados para validar o desenvolvimento da solução proposta. Começa por descrever os testes unitários, passa para os testes de integração e termina com os testes de sistema.

Por fim, o capítulo 8 conclui de forma sucinta o trabalho realizado.

Capítulo 2

Planeamento

Um planeamento cuidadoso é essencial para o sucesso de qualquer projeto. Este capítulo aborda o processo de planeamento do trabalho e oferece informações sobre a forma de como o projeto foi estruturado. O capítulo começa por descrever o trabalho realizado ao longo do primeiro semestre. De seguida, aborda o trabalho planeado para o segundo semestre e fornece uma análise comparativa do trabalho planeado com a execução efetiva. Por fim, o capítulo termina com a descrição dos riscos que foram identificados durante o processo de planeamento e a forma como estes puderam ser mitigados. A definição destes elementos ajuda a compreender como o projeto foi gerido e executado.

2.1 Planeamento

2.1.1 Primeiro Semente

A Fig. 2 apresenta a distribuição temporal do trabalho realizado durante o primeiro semestre.



Fig. 2 - Diagrama de Gantt do primeiro semestre, Fonte: autor

O desenvolvimento do trabalho iniciou-se após uma primeira reunião com os orientadores, em meados de setembro, onde foram apresentados os objetivos da tese. A primeira tarefa consistiu numa pesquisa exploratória de alguns conceitos chave (ex.: VPP, *blockchain*) uma vez que sem o conhecimento e entendimento destes, seria impossível iniciar qualquer tipo de trabalho.

De seguida, deu-se início ao processo de revisão sistemática da literatura que acabou por demorar mais tempo do que era expectável. Foi um processo lento, pois envolveu a aprendizagem da metodologia de revisão sistemática e, posteriormente, o estudo de variados

trabalhos científicos. Inicialmente, houve dificuldade em perceber os passos a seguir e como os efetuar da maneira correta. Isto fez com que a pesquisa da literatura tivesse de ser repetida mais que uma vez. O facto de a revisão exigir a leitura de uma grande quantidade de artigos também contribuiu para a sua demora. Após a recolha de dados relevantes da literatura para a revisão, procedeu-se à escrita do relatório. Começou-se por escrever os capítulos relativos à revisão e aos conceitos fundamentais uma vez que estes são necessários para apoiar as escolhas feitas para o *design* do protótipo. Finalmente foi elaborada uma primeira versão dos requisitos e da arquitetura do sistema que foi implementado no segundo semestre.

2.1.2 Segundo Semestre

Ao contrário do primeiro semestre, em que o planeamento foi menos estruturado, o segundo semestre começou com um plano concebido que delineava as tarefas essenciais necessária para o desenvolvimento bem sucedido do protótipo *blockchain*. Abaixo, a Fig. 3 ilustra um diagrama de Gantt que oferece uma representação visual da distribuição de tarefas ao longo do semestre.



Fig. 3 - Diagrama de Gantt do segundo semestre (previsto), Fonte: autor

O início do projeto foi previsto para a segunda semana de fevereiro. A primeira tarefa da agenda consistia na familiarização com os aspetos práticos das tecnologias essenciais para o desenvolvimento do protótipo. De seguida, foi prevista uma reunião com os parceiros do projeto para alinhar os requisitos funcionais com as necessidades e preferências de todas as partes interessadas. Assim que estivessem adquiridas as capacidades necessárias para trabalhar com as tecnologias de desenvolvimento, o objetivo era iniciar a fase de desenvolvimento do protótipo. Foi previsto que esta fase fosse concluída até ao início de maio. Após a conclusão da fase de desenvolvimento, foi prevista uma fase de testes e melhorias, com uma duração de 3 semanas. Finalmente era esperado que o último mês fosse dedicado à produção do relatório final.

Este seria o planeamento ideal para a concretização dos objetivos da dissertação. No entanto, na realidade as coisas raramente se desenrolam de forma linear e sem falhas, levando a desvios do plano inicial. Abaixo, a Fig. 4 representa o percurso real do trabalho realizado ao longo do projeto.

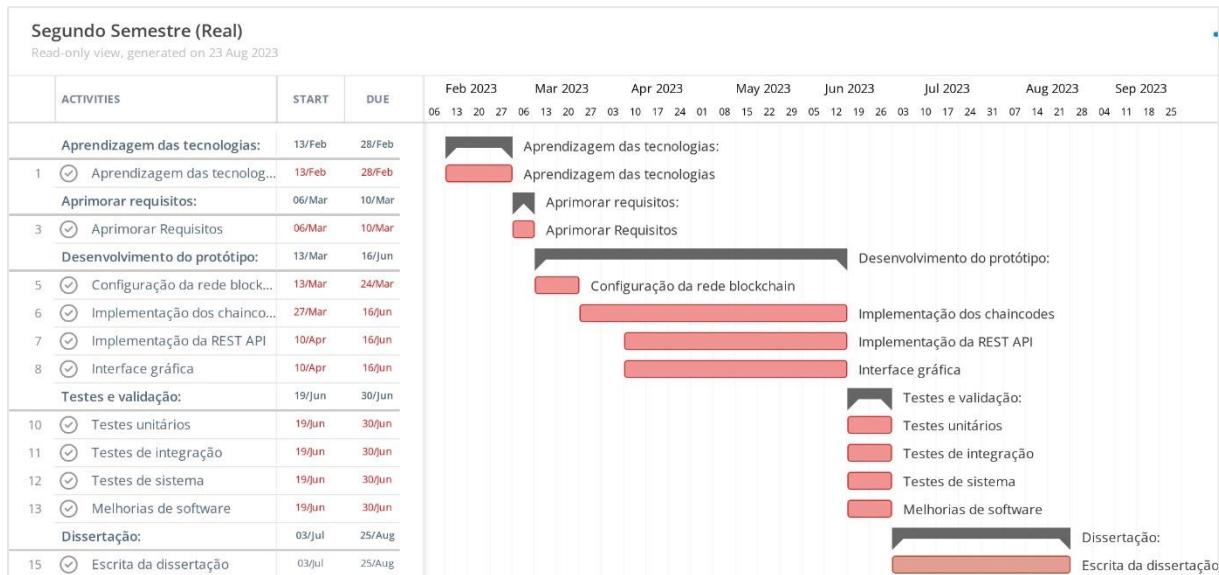


Fig. 4 - Diagrama de Gantt do segundo semestre (realizado), Fonte: autor

Este diagrama ilustra a natureza dinâmica da gestão de projetos, em que desafios imprevistos podem alterar a trajetória do projeto. Segue-se o relato do trabalho realizado ao longo do segundo semestre, acompanhado de uma explicação dos atrasos que ocorreram.

O planejamento teve o seu primeiro desvio em relação ao calendário previsto, logo no início, com um atraso de uma semana no arranque dos trabalhos. Um fator que mais contribuiu para os atrasos do projeto foi a curva de aprendizagem acentuada associada ao domínio de novas tecnologias. Percorrer os pormenores da plataforma Hyperledger Fabric revelou-se uma tarefa mais complexa do que o esperado, que exigiu muita pesquisa e tempo para perceber os básicos do seu funcionamento. Acrescentado a isso, aprender a linguagem de programação Golang, utilizar pela primeira vez Node.js para o desenvolvimento da REST API e reaprender a programar com React, HTML e CSS fez com que a fase de aprendizagem se prolongasse mais do que era inicialmente previsto, principalmente devido à grande quantidade de novas informações que tinham de ser absorvidas.

Após a fase de aprendizagem, surgiu um obstáculo inesperado quando se tornou evidente que não haveria oportunidade para uma reunião de aperfeiçoamento dos requisitos. Esta ausência de informação constituiu um desafio único, uma vez que os requisitos do projeto tiveram de ser estabelecidos sem orientação externa com experiência no setor onde se pretende inserir a solução.

O desenvolvimento do protótipo acabou naturalmente por ter início mais tarde do que foi inicialmente previsto. Neste aspeto, houve uma diferença relativamente ao planeado uma vez que o desenvolvimento dos *chaincodes*, REST API e interface gráfica foi realizado em paralelo. Vários fatores agravaram ainda mais os atrasos. O desenvolvimento de *chaincodes*, em particular, apresentou uma curva de aprendizagem maior do que o esperado. Estes surgiram como um panorama de programação completamente novo, resultando numa quantidade significativa de tentativas e erros durante o seu desenvolvimento. Além disso, o teste das funcionalidades implementadas nos *chaincodes* exigiu inicialmente a necessidade de instalar o *chaincode* na rede *blockchain* sempre que este sofresse alguma alteração, sendo este um processo bastante demorado. Felizmente, cerca de três semanas após o início do desenvolvimento, foi descoberta uma extensão para o Visual Studio Code (Hyperledger Fabric

Debugger) que simplificou bastante este processo, acelerando significativamente o desenvolvimento dos *chaincodes*.

Uma vez concluída a fase de desenvolvimento, foi necessário efetuar testes e melhorar o software. Após a conclusão destas tarefas, restava apenas cerca de uma semana para o processo de escrever a dissertação e a respectiva apresentação. Este prazo confirmou-se muito curto levando ao alargamento do período de escrita até ao final de agosto.

2.2 Riscos

A gestão de riscos é um processo importante para a minimização de impacto negativo que algum tipo de problema possa causar durante o desenvolvimento da tese. Portanto, é necessário que estes sejam identificados, avaliados e que seja gerada a estratégia adequada para a sua mitigação.

Foram identificados os seguintes riscos, relativamente ao desenvolvimento do protótipo *blockchain* realizado no segundo semestre:

1. Falta de experiência no desenvolvimento *blockchain*

Este risco resulta de potenciais dificuldade de adaptação à tecnologia *blockchain* durante as fases iniciais do desenvolvimento do protótipo, o que poderia causar a necessidade de um maior tempo de adaptação e comprometer a realização dos objetivos do projeto atempadamente.

- **Probabilidade:** Alta
- **Impacto:** Alto
- **Estratégia de mitigação:** Procurar tomar decisões tecnológicas já reportadas na literatura e para as quais existe maior suporte documental. Além disso, agendar um período para a aprendizagem da componente prática antes do início do período de desenvolvimento.

2. Ausência de orientação externa com experiência no setor

Este risco está associado à ausência de orientação externa de especialistas no setor das centrais elétricas virtuais, o que poderia resultar no potencial desenvolvimento de uma solução desviada da realidade das operações VPP.

- **Probabilidade:** Alta
- **Impacto:** Alto
- **Estratégia de mitigação:** Investir tempo na pesquisa aprofundada sobre o setor energético, com o foco em cenários VPP. Esta investigação permite obter uma melhor compreensão do setor e minimiza a probabilidade de desenvolver uma solução desfasada.

Este capítulo resume o planeamento que foi realizado para a execução deste trabalho. O próximo capítulo apresenta algumas definições-chave relacionadas com a *blockchain* e VPPs, lançando as bases para uma compreensão abrangente destes tópicos.

Capítulo 3

Conceitos Fundamentais

Neste capítulo é explorada a interseção de duas tecnologias: *blockchain* e *virtual power plant*. Em primeiro lugar é apresentada uma visão geral sobre a tecnologia *blockchain* e os seus conceitos como a sua estrutura de dados, tipos de *blockchain*, mecanismos de consenso e *smart contracts*. De seguida, é explorado o conceito de *Virtual Power Plant* e como esta funciona.

3.1 Blockchain

O conceito “registo digital descentralizado” foi proposto pela primeira vez em 1991 por Stuart Haber e W. Scott Stornetta, que pretendiam criar uma forma segura de registar *timestamps* em documentos sem a necessidade de recorrer a uma autoridade central [5]. Neste artigo, os autores descreveram um sistema que recorre a técnicas criptográficas para criar uma cadeia de blocos de dados inviolável, onde cada bloco contém um *timestamp* e um valor obtido através de uma função matemática (*hash*) que representa o bloco anterior. Apesar destes desenvolvimentos, o conceito de *blockchain* só foi totalmente concretizado em 2008 após a publicação do *white paper* que introduziu a Bitcoin [6].

A *blockchain* é conhecida por ser uma base de dados distribuída [7]. O nome tem origem da concatenação das palavras “*block*” e “*chain*”, ou seja, bloco e cadeia. A palavra bloco refere-se a um conjunto de dados que são guardados e agrupados consecutivamente. A palavra cadeia refere-se ao facto desses blocos estarem ligados [8]. A Fig. 5 apresenta de uma forma simplificada a visualização de como os blocos se unem e formam a cadeia.

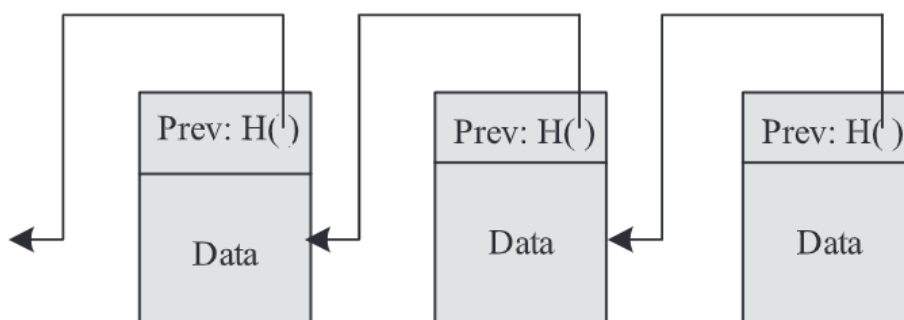


Fig. 5 - Representação simplificada da *blockchain*, Fonte: [9]

De uma forma geral, esta tecnologia pode ser vista então como uma lista ligada de blocos que cresce continuamente e onde cada bloco se encontra ligado ao anterior através de criptografia [9]. Segundo Alexander Savelyev [10], a *blockchain* apresenta quatro características chave: transparência, redundância, imutabilidade e desintermediação. **Transparência** implica que os dados que se encontram guardados nos blocos sejam públicos, não possam ser arbitrariamente adulterados e possam ser facilmente confirmados. **Redundância** deve-se ao facto dos nós manterem e compartilharem uma cópia constantemente atualizada da cadeia de blocos (registo). **Imutabilidade** significa que é muito difícil alterar os dados da cadeia e requer um consenso de

acordo com um protocolo definido de forma inequívoca, sendo assim assegurada a integridade dos registos. A **desintermediação** é conseguida com a eliminação de intermediários, tornando possível a redução de custos das transações [10].

Um **bloco** é o equivalente a um contentor onde são guardados dados. Antes de descrever a estrutura interna dos blocos, é importante explicar o conceito de *Hashing* e *Merkle tree*.

- **Hashing** tem como objetivo principal verificar a integridade dos dados. Uma função *hash* aceita um determinado bloco de dados como entrada e produz um valor de tamanho fixo chamado *hash*. Este tipo de função corresponde a um algoritmo para o qual é computacionalmente impraticável encontrar um conjunto de dados que é mapeado para um resultado pré-especificado (propriedade unidirecional) ou existirem dois conjuntos de dados que sejam mapeados para o mesmo resultado (propriedade livre de colisão). Devido a estas características, as funções *hash* são regularmente utilizadas para determinar se um conjunto de dados foi alterado ou não [11].
- **Merkle tree** é uma árvore binária que guarda valores *hash*. Nesta estrutura, as folhas guardam o *hash* correspondente a um determinado conjunto de dados e os nós intermédios guardam um *hash* que provém da combinação do conteúdo dos seus dois filhos. A raiz guarda o último *hash* que é calculado a partir do conteúdo presente nos dois últimos nós intermédios. Esta conexão faz com que a alteração dos dados correspondentes a uma determinada folha provoque uma alteração em todos os nós intermédios que se encontram ligados a esta, incluindo a raiz da árvore. Dadas estas propriedades, se quisermos verificar se os dados de uma folha foram alterados ou não, apenas é necessário comparar os valores das raízes desta estrutura [12]. A Fig. 6 mostra a representação visual deste tipo de estrutura de dados.

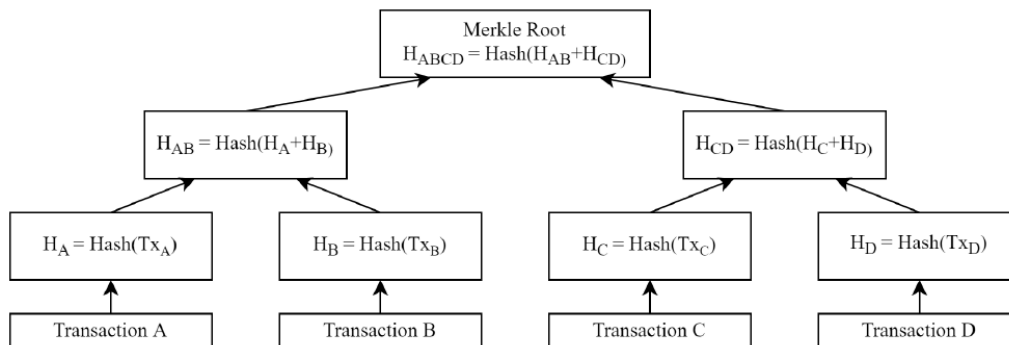


Fig. 6 - Representação de uma *Merkle Tree*, Fonte: [6]

Após o conhecimento destes conceitos, estamos em condições de aprofundar a estrutura de um bloco na blockchain. Cada bloco encontra-se dividido em *header* e *body* como é possível observar na Fig. 7.

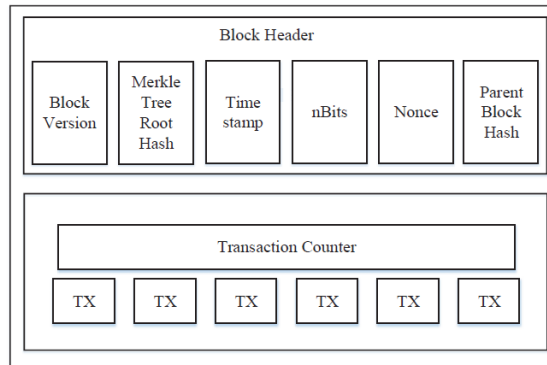


Fig. 7 - Estrutura de um bloco, Fonte: [13]

O *header* é composto por um conjunto de elementos. O **hash da merkle root** (raiz da *merkle tree*) corresponde ao valor que representa todas as transações. Este valor é utilizado para reduzir o esforço na verificação da integridade do bloco. Apenas é necessária a comparação do valor evitando percorrer todas as transações. O **timestamp** guarda a informação sobre quando o bloco foi criado. O **nonce** é utilizado na criação e verificação do bloco. O **hash do bloco anterior** é utilizado para efetuar a ligação ao bloco anterior. [14]. O *body* é composto por um contador de transações e é a divisão onde estas se encontram guardadas [13].

Uma classificação geral faz com que a *blockchain* seja dividida em três categorias distintas: *public blockchain*, *consortium blockchain* e *private blockchain* [15]. A Tabela 1 apresenta a lista de comparação de uma forma generalizada com os *trade-offs* entre os diferentes tipos.

Tabela 1 - Comparação entre os tipos de *blockchain*, Fonte: [13]

Propriedade	Public <i>blockchain</i>	Consortium <i>blockchain</i>	Private <i>blockchain</i>
Determinação de consenso	Todos os mineradores	Conjuntos de nós selecionados	Uma organização
Permissão de leitura	Pública	Pode ser pública ou restrita	Pode ser pública ou restrita
Imutabilidade	Quase impossível de adulterar	Pode ser adulterada	Pode ser adulterada
Eficiência	Baixa	Elevada	Elevada
Centralizada	Não	Parcialmente	Sim
Processo de consenso	Permissionless	Permissioned	Permissioned

- **Public blockchains** são vistas como completamente descentralizadas uma vez que qualquer um se pode juntar à rede e não existe nenhuma autoridade que a controle. O acesso à

informação é público e todos os membros podem participar no processo de aceitação do seu estado. Este tipo de *blockchain* é seguro devido ao seu mecanismo de consenso que alcança o acordo entre todos os participantes. Existem prémios e incentivos de modo a promover a participação no processo de consenso [13].

- **Consortium blockchains** são vistas como parcialmente descentralizadas uma vez que são geridas por mais do que uma organização. A informação pode ser visível para todos os membros, mas a sua alteração e aceitação só é possível ser realizada por determinados grupos de nós [15].
- **Private blockchains:** são consideradas como um tipo de *blockchain* centralizada onde existe uma autoridade central que determina as permissões sobre quem pode ver a sua informação ou efetuar alguma transação. Portanto, o mecanismo de consenso é definido por uma autoridade central [15].

Após analisar e comparar os três tipos de *blockchain* é possível verificar que existe diferença em como é atingido o consenso entre os participantes. Por exemplo na *public blockchain* todos os mineradores determinam o consenso. Por outro lado, na *consortium blockchain* e *private blockchain* a determinação do consenso é realizada por um grupo de nós selecionados ou por uma organização, respetivamente.

A *blockchain*, sendo uma rede descentralizada, não tem um ponto central que verifique todas as transações e que assegure que o registo distribuído por todos os nós seja igual. Portanto, existe a necessidade de algum tipo de protocolo que indique que todas as transações estão válidas e que assegure que o registo guardado nos diferentes nós é o mesmo [16]. A validação e confirmação de transações é efetuada pela decisão coletiva dos nós da rede. Para adicionar um novo bloco à *blockchain* é necessário recorrer a algoritmos de consenso. Estes algoritmos são utilizados para construir confiança e para guardar as transações em blocos de forma apropriada. Um mecanismo de consenso é essencialmente um conjunto de regras que são seguidas por todos os nós. Como não existe confiança universal entre os nós, é necessário um mecanismo deste tipo para que todos os participantes possam concordar com o estado atual do sistema [17]. Existem diversos tipos de algoritmos de consenso, entre eles os mais conhecidos são *proof-of-work* (PoW), *proof-of-stake* (PoS), *practical Byzantine fault tolerance* (PBFT), entre outros [18].

- **PoW** é o algoritmo de consenso mais conhecido uma vez que é utilizado na famosa Bitcoin [6]. Este algoritmo baseia-se na escolha aleatória de um nó para publicar um novo bloco. Esta escolha aleatória é vulnerável a ataques uma vez que poderia ser escolhido um nó com más intenções. De modo a contornar esta vulnerabilidade, se um nó quiser publicar um bloco de transações, este tem que realizar trabalho. Neste caso, trabalho significa que tem de realizar vários cálculos computacionais. Neste algoritmo, os mineradores estão constantemente a procurar um valor para o *nonce* presente no bloco de maneira a que o *hash* desse bloco atinja uma condição definida previamente, como por exemplo, encontrar um *nonce* que fará com que os primeiros dez bits do *hash* sejam iguais a zero [16]. Quando o valor é encontrado, o bloco é transferido para os outros nós que confirmam se o *nonce* de facto produz um valor *hash* que cumpra os critérios. Se isto se verificar, o bloco é adicionado ao registo de cada nó. Este algoritmo apresenta a desvantagem de desperdiçar muita energia uma vez que os mineradores têm de realizar muitos cálculos computacionais [13].
- **PoS** também é um algoritmo muito conhecido uma vez que é atualmente utilizado na *blockchain* Ethereum. Neste algoritmo os nós são escolhidos para criar novos blocos baseados no número de tokens que possuem e não no poder computacional. Este algoritmo

é mais eficiente relativamente ao PoW uma vez que não requer calculos que consumam muita energia [19].

- **PBFT** é um algoritmo conhecido por ser suportado pela *blockchain* Hyperledger Fabric [20] e é utilizado em *permissioned blockchains* pois requer que todos os nós da rede sejam conhecidos [13]. Neste método, todos os nós participam num processo de votação para que um novo bloco seja adicionado. O consenso é atingido quando mais de dois terços dos nós apresentam opinião favorável relativamente ao bloco. Ou seja, o PBFT pode contornar o comportamento malicioso de um terço da plataforma. Deste modo, o consenso é atingido de uma maneira mais rápida e económica comparando com o PoW [16]. De acordo com [21] pode ser utilizado em redes *blockchain* com um baixo número de nós. Este algoritmo garante consistência e disponibilidade mesmo na presença de nós maliciosos ou defeituosos.

Estes três algoritmos estão entre os algoritmos mais conhecidos uma vez que são utilizados nas maiores plataformas blockchain da atualidade. O artigo [18] apresenta uma lista com outros algoritmos de consenso juntamente com informação pretinente sobre cada um.

O lançamento da Bitcoin, em 2009, marca o início da adoção da tecnologia *blockchain* de uma forma generalizada. Nos anos que se seguiram, o conceito *blockchain* começou a alcançar uma maior atenção e outros projetos baseados nesta tecnologia foram desenvolvidos. Em 2015, foi lançada a *blockchain* Ethereum que introduziu o conceito de *smart contracts* que permite aos utilizadores a criação e execução automática de contratos na *blockchain* [22].

O conceito *smart contract* foi proposto por Nick Szabo nos anos 90, como uma maneira de transformar contratos em linhas de código e fazer cumprir os seus termos de forma transparente, segura e digital [23]. Os termos são programados em código e este é executado automaticamente quando forem atingidas certas condições. Contudo, este termo só se tornou popular após o aparecimento da *blockchain* Ethereum, apresentando uma plataforma que permite a desenvolvedores a criação e implementação de *smart contracts* que apresentam várias capacidades [24]:

- **Automatização:** Capacidade de automatizar processos complexos e assim eliminar a necessidade de intermediários. Isto pode reduzir o custo, aumentar a eficiência e ainda reduzir o risco de erros.
- **Transparência:** São guardados na *blockchain* e portanto podem ser vistos por todos os seus participantes. Isto ajuda a construir confiança e assegura que os termos do contrato são seguidos.

A introdução deste conceito abriu portas para uma grande variedade de novos casos de uso para a *blockchain* além de transações financeiras. Nos últimos anos esta tecnologia tem-se expandido e tem sido aplicada em novas áreas como:

- **Sistemas de voto:** A tecnologia *blockchain* está a ser explorada como uma forma de melhorar a segurança e transparência dos sistemas de voto [25].
- **Saúde:** A tecnologia *blockchain* tem a capacidade de por exemplo fornecer segurança na prestação de cuidados de saúde através da eliminação de despesas médicas fraudulentas, devido às propriedades de imutabilidade que a tecnologia oferece [26].
- **Energia:** Um exemplo da aplicação da tecnologia *blockchain* no setor da energia é o desenvolvimento de plataformas de comercio de energia de forma peer-to-peer (P2P) onde os participantes podem comprar e vender energia solar que tenham em excesso entre eles invés de optar por comprar a uma companhia tradicional. A startup Power Ledger é um exemplo de uma plataforma deste tipo [27].

Para concluir, é importante destacar que a tecnologia *blockchain*, embora ofereça muitas vantagens, também enfrenta alguns desafios e vulnerabilidades, como os seguintes:

- **Vulnerabilidades de segurança:** A referência [28] descreve 24 vulnerabilidades de segurança que podem afetar a tecnologia *blockchain*. Entre estas, uma preocupação é o ataque de 51%. Este cenário de ataque ocorre quando um adversário assume o domínio de mais de 50% dos nós da rede ou do seu poder computacional, conferindo-lhe autoridade para adulterar o algoritmo de consenso, o que representa uma ameaça grave à segurança da rede.
- **Desafios de escalabilidade:** As redes *blockchain* debatem-se com problemas de escalabilidade uma vez que apresentam um limite no número de transações que podem processar por segundo. Isto pode resultar em tempos de confirmação lentos e custos elevados durante o período de congestionamento da rede [29].
- **Consumo energético:** Muitas redes *blockchain*, especialmente as redes que utilizam o algoritmo PoW, como a Bitcoin, necessitam de muito poder computacional, o que resulta num elevado consumo de energia para validar as transações. Este facto suscita preocupações relativas ao impacto ambiental [30].

3.2 Virtual Power Plants

O aumento do número de recursos de energéticos distribuídos / *distributed energy resources* (DERs) surge como um desafio para a rede energética. DERs são sistemas de pequena escala que se encontram distribuídos e têm a capacidade de gerar, armazenar ou gerir o fluxo energético. A maioria dos DERs, como os painéis fotovoltaicos e as turbinas eólicas, estão dependentes de condições temporais e portanto a sua produção de energia é intermitente, o que faz com que a quantidade de energia gerada não possa ser completamente controlada. Por outro lado, a maioria dos DERs opera de forma isolada satisfazendo apenas as necessidades locais sem contribuir para a rede energética. Com o objetivo de solucionar estes problemas, surge o conceito VPP [31].

Uma VPP é um agregador virtual de DERs, responsável pela sua coordenação e conexão. As VPP utilizam tecnologias de comunicação e controlo avançadas de modo a otimizar a operação destes recursos em tempo real e a fornecer serviços à rede, como o balanceamento entre a oferta e a procura e tentar manter a estabilidade da rede [32]. A Fig. 8 apresenta um esquema geral do funcionamento de uma VPP.

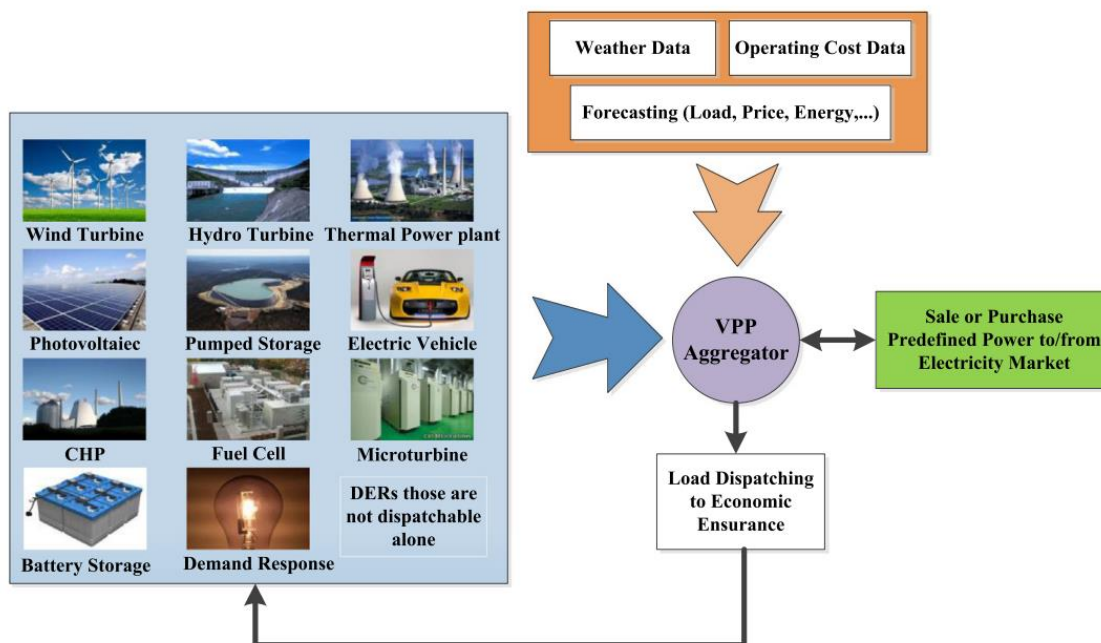


Fig. 8 - Estrutura típica de uma VPP

O estudo realizado por Hossein Rouzbahani [33] identificou as seguintes características numa VPP:

- **Eliminação de restrições geográficas:** Uma das principais características é a capacidade de eliminar as limitações geográficas relativamente à integração de DERs. Isto porque as VPPs utilizam uma configuração virtual que permite que os participantes se integrem no mercado energético independentemente da sua localização física.
- **Fluxo de informação bidirecional e monitorização em tempo real:** A existência de canais de comunicação de alta velocidade também é uma característica importante. Estes são necessários para facilitar a troca de dados energéticos em tempo real. Infraestruturas de comunicação avançadas são essenciais pois permitem a troca bidirecional de dados permitindo assim o comércio de energia entre todos os participantes.
- **Capacidade de mineração de dados:** Outra importante característica é a capacidade de mineração de dados. Este é o processo utilizado para extrair informação útil de streams de dados em tempo real geradas por instrumentos inteligentes e equipamentos instalados na rede. A capacidade de mineração de dados das VPPs permite-lhes lidar de forma eficiente com os desafios de gerir e analisar grandes quantidades de dados, que são necessários para providenciar monitorização em tempo real e para responder à alteração de procura energética.
- **Gestão dinâmica de energia:** Por fim, programas de gestão de energia geram e controlam as trocas energéticas entre os participantes. A gestão de energia é afetada por várias incertezas como a variabilidade das fontes de energia renováveis, restrições dos preços, entre outros fatores. As VPPs têm a capacidade de realizar a gestão de energia de uma forma dinâmica, confiável e económica, coordenando o fluxo de energia entre produtores e consumidores recorrendo a um sistema de gestão de energia. Este sistema é responsável por supervisionar a interação de energia e informação entre os diferentes sistemas na VPP.

3.3 Blockchain e Virtual Power Plants

As VPPs, apesar de apresentarem uma estrutura descentralizada, adotam um modo de gestão centralizado. Num modo de operação tradicional, ou seja, sem utilizar a tecnologia *blockchain*, a distribuição de informação está dependente de um operador de uma agência centralizada. A existência deste intermediário leva à falta de transparência no sistema e consequentemente à falta de confiança por parte dos participantes [34].

Para resolver este problema fundamental, a tecnologia *blockchain* surge como uma solução promissora. Uma rede *blockchain* é por natureza descentralizada uma vez que é composta por nós distribuídos que trabalham em conjunto para coordenar e validar dados. Esta descentralização alinha-se com a estrutura das VPPs onde os DERs se encontram geograficamente dispersos. Esta semelhança de estrutura surge como uma oportunidade para a utilização da tecnologia *blockchain* para a coordenação de VPPs, eliminando a necessidade de um operador central, aumentando a transparência e a confiança entre os participantes [35].

Além disso, a tecnologia *blockchain* pode ser utilizada para criar VPPs descentralizadas que facilitam as transações energéticas P2P sem ter que depender de intermediários. Esta inovação, melhora a eficiência operacional através da utilização *smart contracts*. Adicionalmente, a utilização de algoritmos de encriptação e a característica de imutabilidade da *blockchain* podem aumentar a segurança do registo de dados críticos aumentando ainda mais a credibilidade do sistema [34].

A Fig. 9 apresenta o esquema de uma estrutura VPP baseada em *blockchain*.

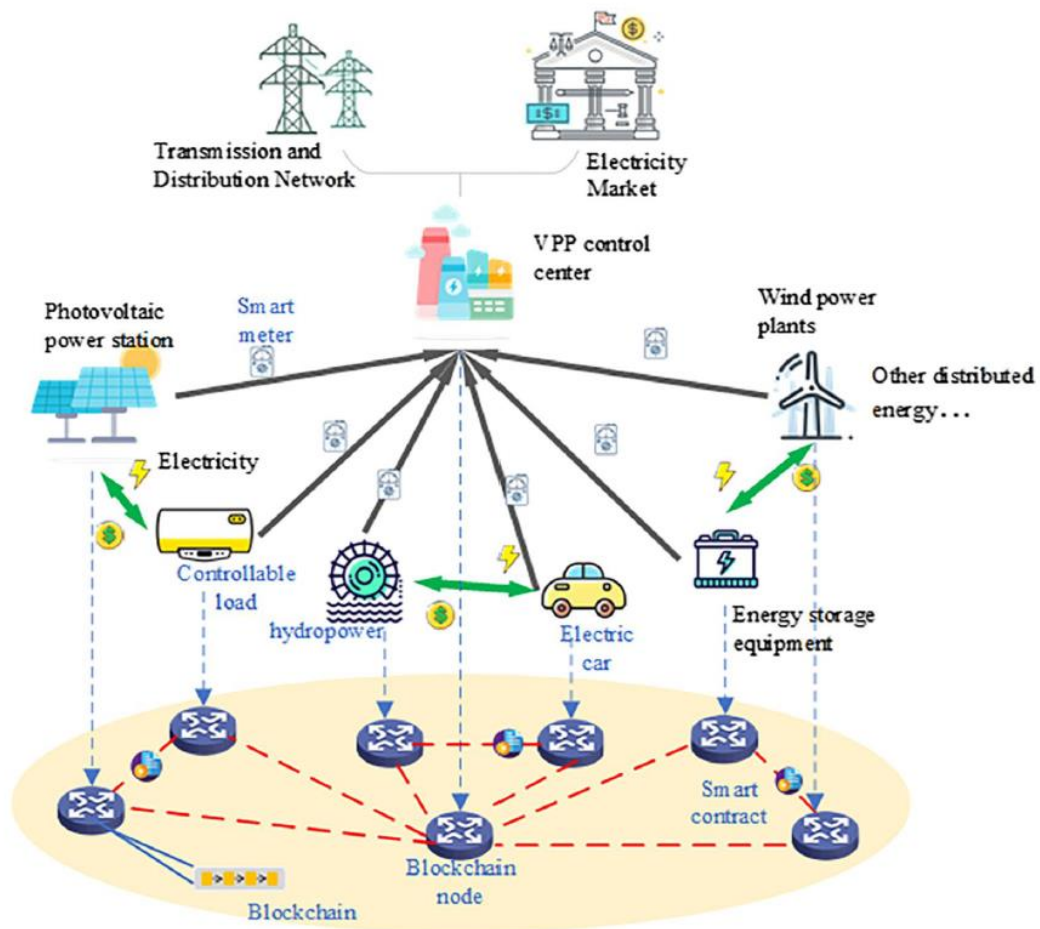


Fig. 9 - Exemplo de uma VPP baseada em *blockchain*, Fonte: [36]

Resumidamente, a aplicação da tecnologia *blockchain* fornece uma estrutura descentralizada, transparente e segura às VPPs. Ao alinhar-se com a descentralização das VPPs e ao possibilitar transações de energia P2P, não só aumenta a confiança dos participantes como abre caminho para um sistema energético mais eficiente.

No capítulo 4, é apresentada uma revisão sistemática da literatura sobre a utilização da tecnologia *blockchain* em VPPs onde este tema será aprofundado e que pretenderá responder à seguinte questão de problema : Como é que a tecnologia *blockchain* é aplicada em *Virtual Power Plants*?

Capítulo 4

Revisão Sistemática da Literatura

Este capítulo apresenta uma revisão sistemática da literatura realizada de acordo com a estrutura definida por Webster e Watson [37]. A integração da tecnologia *blockchain* no setor da energia surgiu como uma perspectiva promissora que promete oferecer o potencial para uma maior eficiência, transparência, descentralização e imutabilidade. Esforços recentes têm explorado novas aplicações desta tecnologia para procurar enfrentar os desafios que surgem no estabelecimento de VPPs. No entanto, apesar do entusiasmo à volta das aplicações da *blockchain* em VPPs, notou-se a ausência de uma revisão sistemática abrangente e atualizada da literatura. Esta lacuna limita a compreensão dos potenciais benefícios e desafios da utilização de *blockchain* em VPPs.

À luz destas considerações, a questão central de investigação que orienta a revisão sistemática é a seguinte:

“Como é que a tecnologia *blockchain* é aplicada em centrais elétricas virtuais?”

Para conseguir responder a esta questão os principais objetivos desta revisão sistemática são dois:

1. Sintetizar e analisar a literatura existente sobre a aplicação da tecnologia *blockchain* em VPPs.
2. Identificar os casos de uso em que está a ser aplicada a tecnologia *blockchain*

Esta revisão explora os diferentes casos de uso e destaca os benefícios apresentados na literatura até à data. Os resultados desta revisão irão fornecer informações valiosas para os demais interessados no desenvolvimento e implementação de VPPs.

O resto do presente capítulo encontra-se estruturado da seguinte forma. A secção 4.1 apresenta a metodologia da revisão que descreve o método de investigação, a secção 4.2 apresenta uma análise do descritiva dos artigos encontrados, a secção 4.3 apresenta uma revisão dos artigos e por fim a secção 4.4 apresenta os casos de uso da tecnologia *blockchain* em cenários VPP, juntamente com os benefícios apresentados.

4.1 Metodologia da Revisão

Com o objetivo de identificar um conjunto relevante de artigos foram selecionadas as bases de dados *Web of Science*, *Scopus*, *EBSCO*, *IEEE Xplore*, *Science Direct*, *Wiley Online Library*, *Taylor & Francis* e *AISel* como fontes de pesquisa. A pesquisa foi realizada em meados de outubro de 2022. Foi utilizado o seguinte conjunto de *keywords* como expressão de pesquisa:

- (“*virtual power plant*” OR *vpp* OR “*virtual power plants*”) AND *blockchain*

Foram considerados os seguintes critérios de inclusão para a seleção dos artigos.

- Os artigos têm de estar escritos em inglês
- Os artigos têm de abordar especificamente a utilização da tecnologia *blockchain* num cenário VPP

A Fig. 10 ilustra o processo de pesquisa realizado para obter os documentos para posterior análise.

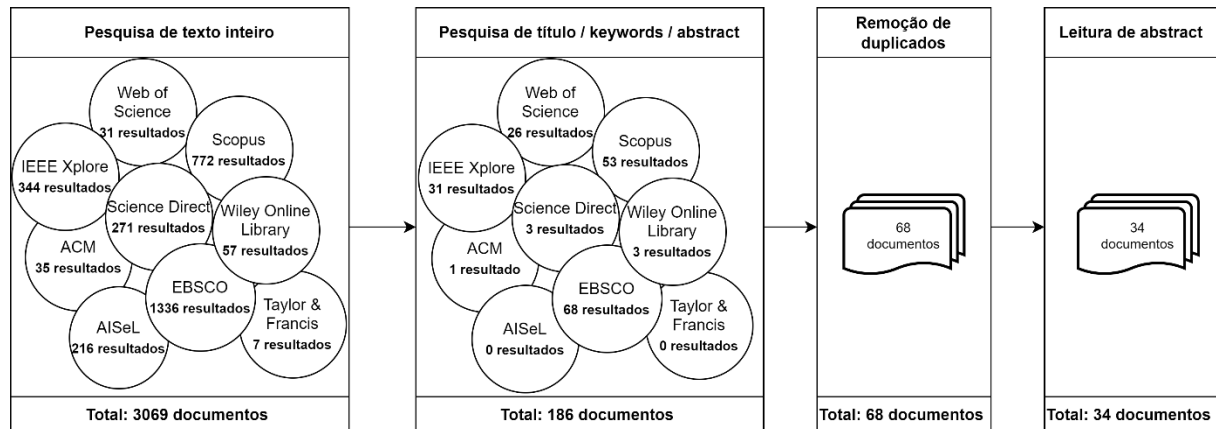


Fig. 10 - Abordagem à pesquisa sistemática, Fonte: autor

O processo foi iniciado por uma primeira pesquisa em que a expressão foi aplicada ao texto inteiro tendo esta retornando 3069 resultados do conjunto de bases de dados selecionado. Com o intuito de reduzir este número, foi feita uma segunda ronda de pesquisa com a mesma combinação de *keywords* restringida ao título, *keywords* e *abstract*. Existiram algumas variações devido à diferença nas interfaces de pesquisa e de opções disponíveis para as diferentes bases de dados. Desta ronda resultaram 168 artigos. Após a eliminação de 118 duplicados sobraram 68 documentos. O último passo consistiu em analisar separadamente o *abstract* de cada artigo e a classificação de cada um de acordo com a sua relevância para a análise (Sim/Não/Talvez). Finalmente, com base na leitura do *abstract* foram eliminados 34 artigos considerados menos relevantes, permanecendo assim um conjunto de 34 para análise do texto integral. Não foi possível obter acesso ao artigo [38] para análise, reduzindo assim a amostra final para um conjunto de 33 artigos. Foi realizada uma leitura de cada artigo com a intenção de retirar informação sobre o uso da tecnologia *blockchain* em VPPs.

4.2 Análise Descritiva dos Artigos

O número de publicações é um indicador importante para avaliar o estado de desenvolvimento de um determinado tema. Analisando o baixo número de artigos obtidos nos últimos anos, é possível interpretar que o tema em discussão é recente.

A Fig. 11 apresenta um gráfico que agrupa o número de artigos por ano de publicação. É possível observar que o interesse no tema, apesar de recente, tem vindo ao crescer e que este apresenta tendência a aumentar nos próximos anos. É também importante referir que segundo os critérios de inclusão, apenas foram escolhidos artigos escritos em inglês sendo, portanto, excluídos artigos escritos em chinês onde foram realizados os primeiros estudos do tema em 2017 e 2018.

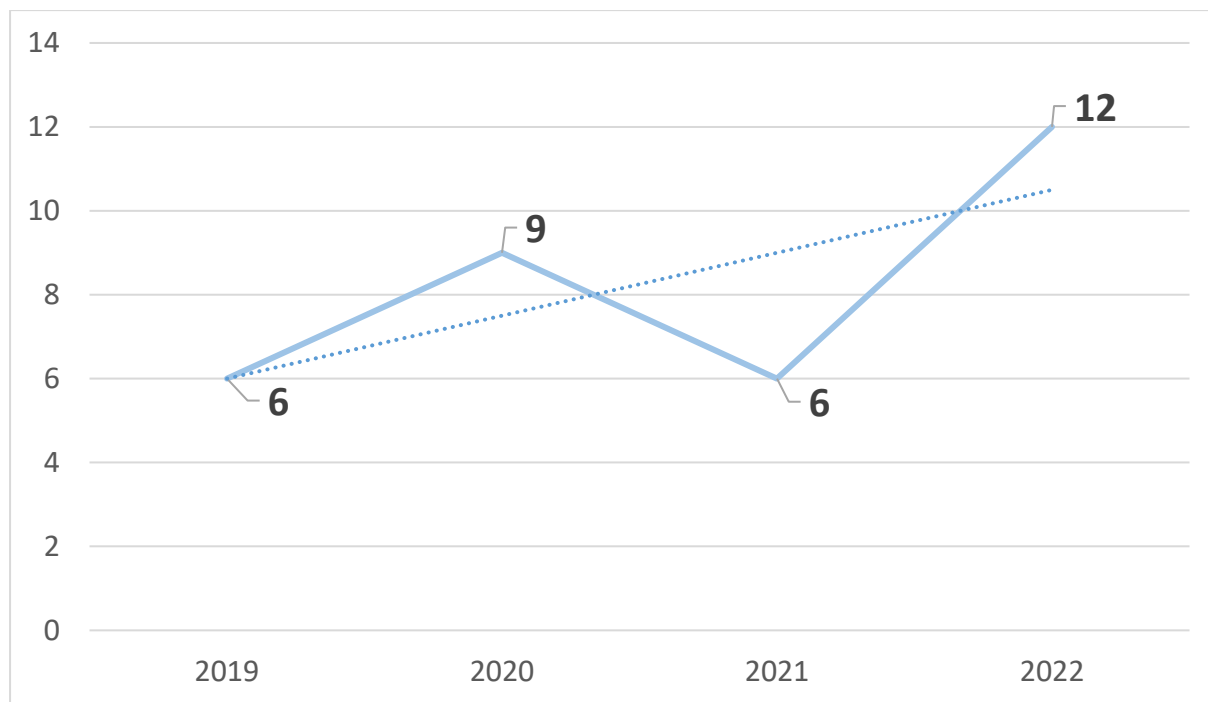


Fig. 11 - Número de artigos por ano de publicação

O número de publicações em cada país reflete o nível de pesquisa que está a ser aplicado a determinado tema nesse mesmo país. Neste caso apenas são analisados os países correspondentes aos artigos que foram analisados em profundidade. A Fig. 12 fornece informação visual relativa à distribuição geográfica das publicações. A esta, é associado o número de artigos agrupados por ano dentro de cada país. Através da análise do gráfico é possível observar que está a ser feito um grande investimento na China relativamente aos outros países e que esta foi pioneira, tendo participado em pesquisas de 2019. Também é possível ver que 6 dos 16 países juntou-se à pesquisa deste tema no ano de 2022, o que apresenta um crescimento de aproximadamente 38%. Isto indica que recentemente o interesse na pesquisa começou a espalhar-se por mais países, nomeadamente pela Europa.

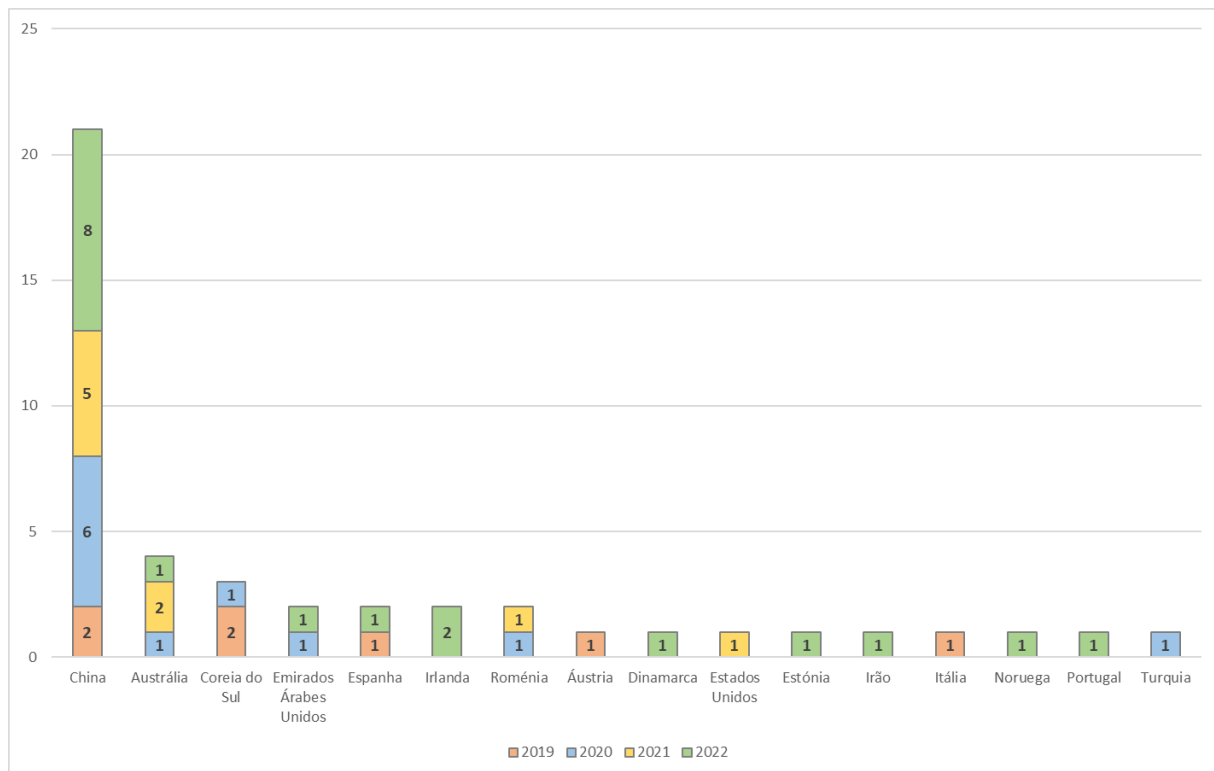


Fig. 12 - Distribuição geográfica dos artigos, Fonte: autor

4.3 Revisão dos artigos

Os *smart contracts* são uma funcionalidade da *blockchain* muito importante. A sua versatilidade faz com que estes possam ser aplicados em vários contextos na operação de uma VPP. A automatização de transações energéticas é uma das capacidades dos *smart contracts*. Em [39] é proposto um processo de transação de energia P2P baseado em *blockchain* que permite a um *prosumer* (participante no mercado que produz e consome energia) comprar energia diretamente a outro *prosumer*. Em [40] a empresa *Korea Telecom* apresenta uma proposta de um serviço de resposta à procura de energia baseado em *blockchain* através do recurso a *smart contracts* para automatização de transações de energia entre clientes. Em [41] é desenvolvido um sistema *blockchain* integrado numa VPP que permite o tratamento automático de relações contratuais através de *smart contracts*. O sistema permite o controlo de DERs de maneira automática baseando-se em termos pré acordados pelo operador VPP e o dono do DER. Quando um DER é utilizado é efetuado o pagamento de forma instantânea através da execução de um *smart contract*. Os autores do artigo [42] investigam a resolução para o desafio que os operadores das VPPs têm na coordenação e organização de transações. Para resolver este problema, é proposta uma estratégia de computação evolucionária para a formação de *smart contracts*. Após vários testes os resultados mostraram que esta nova estratégia tem melhor performance que as existentes. Em [43] é proposta a utilização de *blockchain* em conjunto com *smart contracts* para a construção de VPPs que providenciam a agregação de pequenos *prosumers* de forma descentralizada. Em [44] é apresentada uma investigação sobre a tecnologia *blockchain* para ser aplicada no mercado da eletricidade. Os autores expõem a implementação de acordos para compra e venda de energia solar que são executados automaticamente por meio de *smart contracts*. Em [45] é apresentado um modelo de otimização de energia transitiva para o escalonamento de DERs. O modelo oferece a possibilidade de transações intra-VPP e multi-VPP e tem como função objetiva a minimização do custo total de

todos os *prosumers*. Sobre este modelo, são utilizados *smart contracts* para automatizar e guardar transações de energia. O artigo [46] começa por apresentar um modelo de transações baseado em *smart contracts* onde procura obter um plano de geração de energia ótimo tendo em conta o custo de produção e um índice de proteção ambiental. De seguida apresenta uma estratégia de pré distribuição que tem como função objetiva o custo mínimo.

Vários artigos recorrem a uma estratégia de leilão para efetuar a correspondência entre produtores e consumidores de energia. O artigo [47] propõe uma arquitetura para a negociação de energia P2P numa VPP através de um sistema de licitação que corre sobre a rede *blockchain* através de *smart contracts*. Em [48], muito semelhante ao artigo anterior, é apresentado um esquema de transações de energia baseado em *blockchain* sob a arquitetura de uma VPP onde produtores com excesso de energia podem iniciar um leilão para venda da mesma e consumidores podem licitar para obter os recursos. Outro caso semelhante é apresentado pelo artigo [34] onde o modo de transação baseia-se num mecanismo de leilão duplo contínuo. Neste mecanismo, quando a janela de leilão termina, as licitações por parte de compradores são ordenadas da maior para a menor e as licitações dos vendedores são ordenadas da menor para a maior. Após este ordenamento, é efetuada a combinação de compradores e vendedores e é estabelecido um preço de venda que corresponde à média dos valores licitados por ambos.

Outra funcionalidade dos *smart contracts* é a capacidade de implementar algoritmos matemáticos. O algoritmo *alternating method of multipliers* (ADMM) que resolve problemas de otimização pode ser implementado através de um *smart contract* como parte da *blockchain* permitindo também que os utilizadores preservem privacidade. No artigo [49] é desenvolvido um sistema de coordenação de DERs onde este algoritmo participa na resolução da solução ótima para a minimização de custos dos DERs. Já no artigo [50] o algoritmo é utilizado para atingir a otimização do escalonamento de gestão de energia para todos os utilizadores. O artigo [35] estabelece um modelo para o sistema de controlo de uma VPP baseado na tecnologia *blockchain* onde cada participante tem oportunidade de estabelecer o seu plano de controlo ótimo para a distribuição energética dentro da rede, baseado num modelo matemático estabelecido. Em [51] é proposto um modelo *blockchain* que auxilia a conexão de DERs à rede e a calcular o escalonamento ótimo da VPP baseado na procura de energia. Em [52] é estabelecido um modelo de negociação de energia baseado em *blockchain* e de seguida são comparados três métodos para transações de forma descentralizada. O primeiro método consiste na negociação independente entre participantes. O segundo método, chamado “Estratégia pré distribuída”, é composto por um centro de negociação virtual que tendo em conta as exigências dos consumidores faz os cálculos da energia que é necessário ser alocada. O terceiro e último método consiste na modificação do segundo método e substitui o centro de negócio virtual por um sistema distribuído.

Foi encontrado um conjunto de 2 artigos que além de apresentarem uma proposta de estratégia para transações energéticas exploram a necessidade da criação de novos algoritmos de consenso orientados a estas. Em [53] os autores focam-se em desenvolver um mecanismo de incentivo para os produtores venderem mais energia nas horas de maior consumo através de um novo algoritmo de consenso chamado *Proof of Energy Market* (PoEM). Este mecanismo ainda é estendido para um outro chamado *Private Proof of Energy Market* (PPoEM) através da integração de privacidade diferencial de modo a fornecer a seleção do preço de forma privada. Em [54] é igualmente descrito um novo mecanismo de consenso chamado *blockchain network collaboration mechanism* (BNCM) que consiste numa adaptação do algoritmo já existente PBFT. Os resultados de uma simulação mostraram que este novo mecanismo é mais adequado para cenários VPP com um elevado número de DERs dada a sua performance e o seu curto tempo de cooperação.

Com a intenção de melhorar a escalabilidade do sistema, surge uma abordagem diferente onde são utilizadas múltiplas *blockchains*. Um conjunto de 3 artigos apresentam esta abordagem. Em [55] é utilizada esta estratégia como estrutura de controlo para gestão de uma VPP. Em [56] a estrutura de transações da VPP é integrada num contexto *main-side blockchain*. Foca-se no mecanismo de mercado interno da VPP onde vários agregadores participam através de licitações após preverem os seus resultados de oferta e procura. Em [57] é proposta uma arquitetura *dual-blockchain*. Dada a elevada quantidade de dados necessários para o processo de escalonamento VPP, e para evitar ataques maliciosos como a sua alteração ou manipulação, também é proposto um algoritmo de recriptação baseado em cifras. Finalmente de modo a garantir o bom comportamento dos participantes no mercado é apresentado um mecanismo de leilão duplo contínuo baseado em reputação.

Um dos grandes papéis das VPPs consiste na integração de energias renováveis e no efeito que estas têm na diminuição da poluição emitida durante a produção de energia. Os autores do artigo [58] apresentam um mecanismo de transações baseado em *blockchain* que considera estabilidade na correspondência entre a procura e oferta de energia e prioriza energias renováveis.

A possibilidade de existirem várias VPPs que se apoiam umas às outras de modo a aumentar a eficiência da rede elétrica surge como uma proposta interessante no artigo [59] onde foi desenvolvida uma plataforma que permite aos participantes da VPP rastrear os seus níveis de consumo e produção energética. A procura de energia é guardada por slot de tempo em transações e as VPPs com excesso de energia podem transferi-la para outra VPP que necessite.

Em [60] é definida a arquitetura de uma plataforma para gestão de redes elétricas inteligentes que pode ser aplicada a três grandes cenários de gestão: flexibilidade de agregação e resposta à procura de energia, transações de energia P2P e por último, gestão VPP orientada à comunidade. No artigo [61] é proposta uma plataforma *blockchain* para regulação e transações de energia numa VPP. Esta permite transações de eletricidade P2P e integração das fontes internas para participarem em serviços auxiliares. Em [62] combinam *blockchain* e Internet of Things (IoT) como estrutura de transações numa VPP. Recorrem a *fog computing* para efetuar o processamento e análise de dados dos DERs através de servidores *fog* espalhados pela rede *blockchain*. Em [63] é proposto um método para integrar uma VPP numa *blockchain* que permite guardar dados sobre a sua operação. Neste caso, a VPP, composta por um conjunto de DERS, tem como função responder a pedidos de serviço por parte da rede energética. Em [64] é proposta uma *framework blockchain* de transações de energia para a cooperação de vários agentes. O preço da eletricidade é calculado por hora, baseado no fluxo de energia na rede elétrica. Em [65] é apresentado um sistema de operação e transação numa VPP baseado em *blockchain*. Os autores apresentam ideias de como é que a tecnologia pode ser utilizada para autenticação dos DERs, proteção de dados partilhados, e num sistema baseado em classificação de créditos. Em [66] é apresentado um modelo para coordenação e controlo VPP onde a *blockchain* une os meios de controlo coordenados e o comportamento de conexão independente de energia distribuída através de mecanismos de incentivo.

4.4 Casos de Uso

Após a leitura de todos os documentos na íntegra, foram identificados os casos de uso da tecnologia *blockchain* em cenários VPP. Uma primeira observação permitiu concluir que os casos de uso se dividem em dois grandes temas, sendo estes, transações energéticas e gestão da VPP. Estes foram agrupados em tabelas que associam à referência de cada artigo, os benefícios trazidos pela sua solução, o mecanismo de consenso utilizado e a plataforma escolhida para o

desenvolvimento da rede *blockchain*. Apresenta-se de seguida a tabela de casos de uso para transações energéticas.

Tabela 2 - Transações Energéticas

Ref.	Benefícios	Tipo de Rede	Mecanismo de Consenso	Plataforma
[39]	Exclusão de intermediários Redução de taxas de transação Segurança nas transações	NA	PoW	NA
[59]	Permite aos participantes da VPP rastrear a sua produção e consumo energéticos Automatização das transações através de <i>smart contracts</i>	Permissioned	PoS	Hyperledger Fabric
[40]	Pagamento em tempo real Melhoria na eficiência do estabelecimento de contratos Aumento de lucros	NA	NA	NA
[47]	Prevenção de gastos duplos	Permissionless	PoS	Ethereum
[52]	Redução da poluição	Permissioned	NA	Hyperledger Fabric
[61]	Melhora robustez e confiabilidade do armazenamento dos dados <i>Smart contracts</i> garantem os deveres e obrigações em ambos os lados Redução de custos	Permissioned	NA	Hyperledger Fabric
[58]	Bom efeito na economia VPP consegue fazer o balanceamento de oferta e procura	Permissioned	PoW	Ethereum
[62]	Impossibilidade de falsificar histórico de transações	NA	NA	NA
[46]	Estabelecimento automático de transações	Permissioned	NA	Hyperledger Fabric
[34]	Redução de perdas de energia Redução de custo de transações	NA	PoS	Ethereum

Ref.	Benefícios	Tipo de Rede	Mecanismo de Consenso	Plataforma
[48]	Transações energéticas mais eficientes e económicas Transparência nas transações Segurança de dados e privacidade	Permissioned	NA	Ethereum
[64]	Confiança mútua Transparência nas transações Transações automáticas Rastreamento das transações em tempo real Promoção de produção de energia limpa	Permissioned	C+I (novo mecanismo)	Ethereum
[53]	Melhoria na confiança	Permissioned	PPoEM (novo mecanismo)	NA
[44]	Partilha de dados automática Melhoria na eficiência das transações através de <i>smart contract</i> Ajuda consumidores e produtores a determinar preço de eletricidade Troca fácil de fornecedor	NA	PoW + PoS	Ethereum
[45]	Melhoria na transparência Aumento da automação de transações energéticas Melhoria na segurança devido à natureza imutável da <i>blockchain</i> Aumento na participação dos consumidores Redução de despesa na eletricidade Aumento da penetração de DERs na rede	NA	PoW	Ethereum
[56]	Aumento da escalabilidade do sistema Melhoria nos benefícios económicos a nível geral	Permissioned	DPoA	Hyperledger Fabric
[54]	Descentralização Transparência Automação da execução de contratos Rastreabilidade	NA	BNCM (novo mecanismo)	Ethereum

A Tabela 3 sumariza os casos de uso identificados para gestão das VPPs.

Tabela 3 - Gestão VPP

Ref.	Caso de uso	Benefícios	Tipo de Rede	Mecanismo de Consenso	Plataforma
[35]	Sistema de Controlo VPP	Desempenho em tempo real Redução dos custos de transação Transparência da informação Benefícios económicos para a VPP	Permissioned	PBFT	NA
[51]	Computação distribuída do algoritmo de escalonamento	Redução da carga computacional da VPP Redução do custo de operação Redução de preço médio de energia Redução do tempo de cálculo do escalonamento ótimo	Permissioned	Improved PoW	NA
[60]	Gestão	Comunicação bidirecional Aumento de confiança Fornecimento de recursos de comunicação poderosos Permite o controlo eficiente de cargas eléctricas	NA	PoA	NA
[63]	Framework de consenso entre VPP e DERs	Transparência e confiança na interação entre os participantes	NA	NA	Ethereum
[43]	Construção de VPP	Redução do tempo de construção de uma VPP	Permissionless	PoA	Ethereum
[49]	Coordenação de DERs	Máquina de computação confiável que executa coordenação de DERs Remoção da necessidade de um coordenador central Rede de comunicação segura e robusta Ferramenta de pagamento eficiente para transações energéticas e prémios de serviço Melhoria na eficiência do sistema	NA	PoA	Ethereum

Ref.	Caso de uso	Benefícios	Tipo de Rede	Mecanismo de Consenso	Plataforma
[50]	Gestão	Remoção da necessidade de um coordenador central Rede de comunicação escura e robusta Moeda digital facilita pagamento de transações energéticas e prémios de serviço	NA	PoA	Ethereum
[65]	Operação e transação	Transparência Redução de investimentos para recolha e verificação de dados Aumento na credibilidade e precisão dos dados	Permissioned	NA	NA
[66]	Coordenação e controlo de interconexão de energia distribuída	Melhoria da eficiência computacional no cálculo do algoritmo de escalonamento	NA	NA	NA
[55]	Gestão	Assegura prioridade à utilização de energia limpa Redução da geração de energia de forma tradicional Quantidade de dados no sistema pode ser reduzida em 33%	NA	(novo mecanismo)	NA
[42]	Coordenação de transações	Permite donos da VPP determinar a melhor maneira de organizar as transações Produtores podem utilizar o modelo para maximizar ganhos	NA	NA	NA

A análise das tabelas leva a várias conclusões. É possível identificar vários benefícios que a aplicação da tecnologia *blockchain* traz ao funcionamento de uma VPP. Vários artigos apontam para o aumento da transparência que faz com que exista mais confiança entre os participantes. A utilização de *smart contracts* aprimora a eficiência através da automatização de transações e reduz a necessidade de intermediários e intervenção manual. É observada uma redução nos custos da operação da VPP assim como nos custos da compra de energia. Também é possível observar melhorias na escalabilidade e na segurança.

Relativamente ao tipo de *blockchain*, aproximadamente 47% dos artigos recorrem a *permissioned blockchain*, aproximadamente 7% utilizam *permissionless blockchain* e aproximadamente 46% dos artigos não indicam o tipo de *blockchain*. Estes valores sugerem que *permissioned blockchain* é a mais indicado para isto tipo de sistema. Esta escolha reside no

facto de a energia ser um meio muito sensível, sendo, portanto, importante controlar quem se pode juntar à rede [53].

Relativamente ao mecanismo de consenso surgem 4 alternativas que se destacam em comparação com os outros, sendo eles, PoW, PoS, PoA e PBFT. O PoW, apesar de ser adotado em 3 artigos, não são apresentadas razões suficientes para a seleção do mesmo, inclusive um artigo aponta para o problema de baixa eficiência que este apresenta [58]. O PoS surge como alternativa ao PoW devido ao elevado consumo de energia deste último [47][59]. Para o PoA, são apresentadas razões semelhantes ao PoS, onde este surge como alternativa ao PoW devido ao elevado consumo energético [49]. O curto tempo para a confirmação das transações também é uma característica a favor deste algoritmo [50]. Finalmente o PBFT surge devido à sua eficiência e por ser tolerante a falhas [35] é apontado como alternativa ao PoS uma vez que este tem a possibilidade de excesso de concentração, ou seja, a possibilidade um pequeno grupo de participantes ter mais influência do que os restantes durante o consenso [57].

Outra questão importante consiste na escolha da plataforma onde a *blockchain* é desenvolvida. Surgem duas opções na lista de artigos, sendo estas a plataforma *Ethereum* (43%) e a plataforma *Hyperledger Fabric* (17%). A principal razão para a escolha da plataforma *Ethereum* é a possibilidade de implementar *smart contracts*. Um fator importante é também o facto de ser uma plataforma madura, bem verificada e que é suportada pela maioria dos sistemas operativos e hardware [49][50]. As razões para a escolha da plataforma *Hyperledger Fabric* residem no facto de ser possível implementar *smart contracts* (chamados *chaincodes* na plataforma) e de esta providenciar APIs convenientes [52]. Esta também é uma plataforma que suporta um serviço para gerir os membros da rede, tem um design modular, o que permite adicionar o mecanismo de consenso e mecanismos de encriptação [59].

Capítulo 5

Requisitos e Arquitetura de Software

Este capítulo começa por fornecer uma visão geral do cenário no qual se insere a solução proposta assim como dos seus principais objetivos. Posteriormente aprofunda-se na visão detalhada dos requisitos funcionais e não funcionais que moldam a base do protótipo. Finalmente, o capítulo examina a arquitetura de software que sustenta o protótipo, oferecendo uma compreensão abrangente da sua estrutura.

5.1 Introdução e objetivos do protótipo

A Fig. 13 apresenta uma visão geral do ambiente no qual a solução proposta será integrada, a fim de fornecer um contexto relevante antes da descrição detalhada dos requisitos do protótipo.

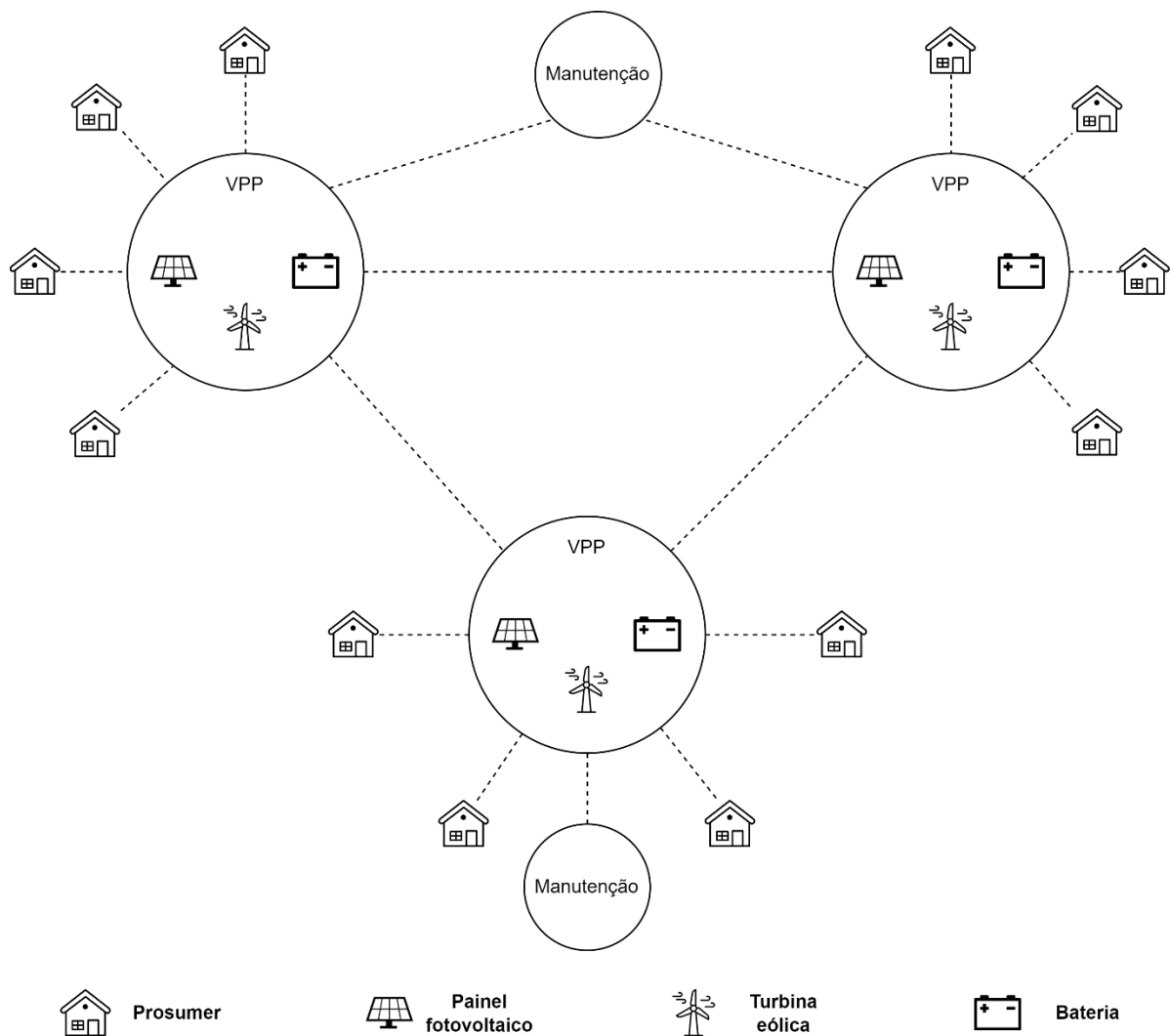


Fig. 13 - Cenário multi-VPP, Fonte: autor

O cenário ilustrado retrata um ambiente multi-VPP no qual estão envolvidas várias VPPs, *prosumers* e organizações responsáveis pela manutenção. Cada VPP tem a capacidade de agregar diferentes tecnologias de produção de energia renovável (ex. painéis fotovoltaicos, turbinas eólicas...), componentes de armazenamento de energia (ex. baterias) e *prosumers*. Dentro deste cenário, as VPPs encontram-se interligadas, podendo colaborar entre si através de transações de energia. Por exemplo, se uma VPP estiver a enfrentar uma queda na produção de energia devido a condições climáticas desfavoráveis, ela pode receber energia de uma VPP vizinha que esteja com uma elevada produção energética. Esta interação permite um melhor aproveitamento dos recursos e aumenta a confiabilidade do fornecimento de energia para os *prosumers*. Também é importante referir a presença de organizações de manutenção. Estas são responsáveis por garantir o funcionamento adequado e a manutenção regular dos ativos de cada VPP. Desempenham ainda um papel fundamental na monitorização, reparação e substituição de componentes quando necessário, tendo em vista maximizar a eficiência operacional das VPPs e garantir a disponibilidade de energia renovável.

Considerando o ambiente descrito, é proposta uma solução que facilita a análise da operação da VPP ao mesmo tempo que promove a cooperação entre as VPPs e as organizações responsáveis pela sua manutenção, facilitando o registo e o acompanhamento das atividades de manutenção em cada um dos componentes das VPPs.

Para tal, a solução integra-se com contadores inteligentes e sensores. Os contadores inteligentes permitem o registo preciso das transações de energia, garantido uma monitorização e contabilização fiável das trocas de energia entre diferentes VPPs e entre VPPs e *prosumers*. Os sensores permitem o registo de dados dos componentes (ex. produção de energia, alertas) permitindo a monitorização eficaz dos mesmos.

O objetivo principal da plataforma é auxiliar o trabalho dos operadores das VPPs e dos técnicos de manutenção. Os operadores das VPPs têm a possibilidade de utilizar a plataforma para analisar dados que permitam uma monitorização da operação da VPP podendo observar as transferências de energia, gerir componentes e acompanhar os processos de manutenção. Paralelamente, os técnicos de manutenção têm a possibilidade de monitorizar o estado dos componentes e realizar toda a gestão e monitorização da manutenção das VPPs.

A tecnologia *blockchain* é a base do protótipo. As suas propriedades permitem o cumprimento de protocolos e contratos entre as organizações, a atribuição de responsabilidades e garantem a integridade e transparência dos dados e das transações realizadas no protótipo. Através desta tecnologia, é possível estabelecer um ambiente imutável que proporcionará uma maior confiança entre os participantes envolvidos.

Em suma, o protótipo apresenta-se como uma solução que permite o apoio à gestão de manutenção e operação das VPPs, fornecendo possibilidades de colaboração entre as organizações.

5.2 Requisitos

Os requisitos servem para descrever o comportamento do sistema de software. Em primeiro lugar, a Fig. 14 apresenta o diagrama de casos de uso do sistema que possibilita uma melhor visualização dos atores do sistema e das suas funcionalidades principais.

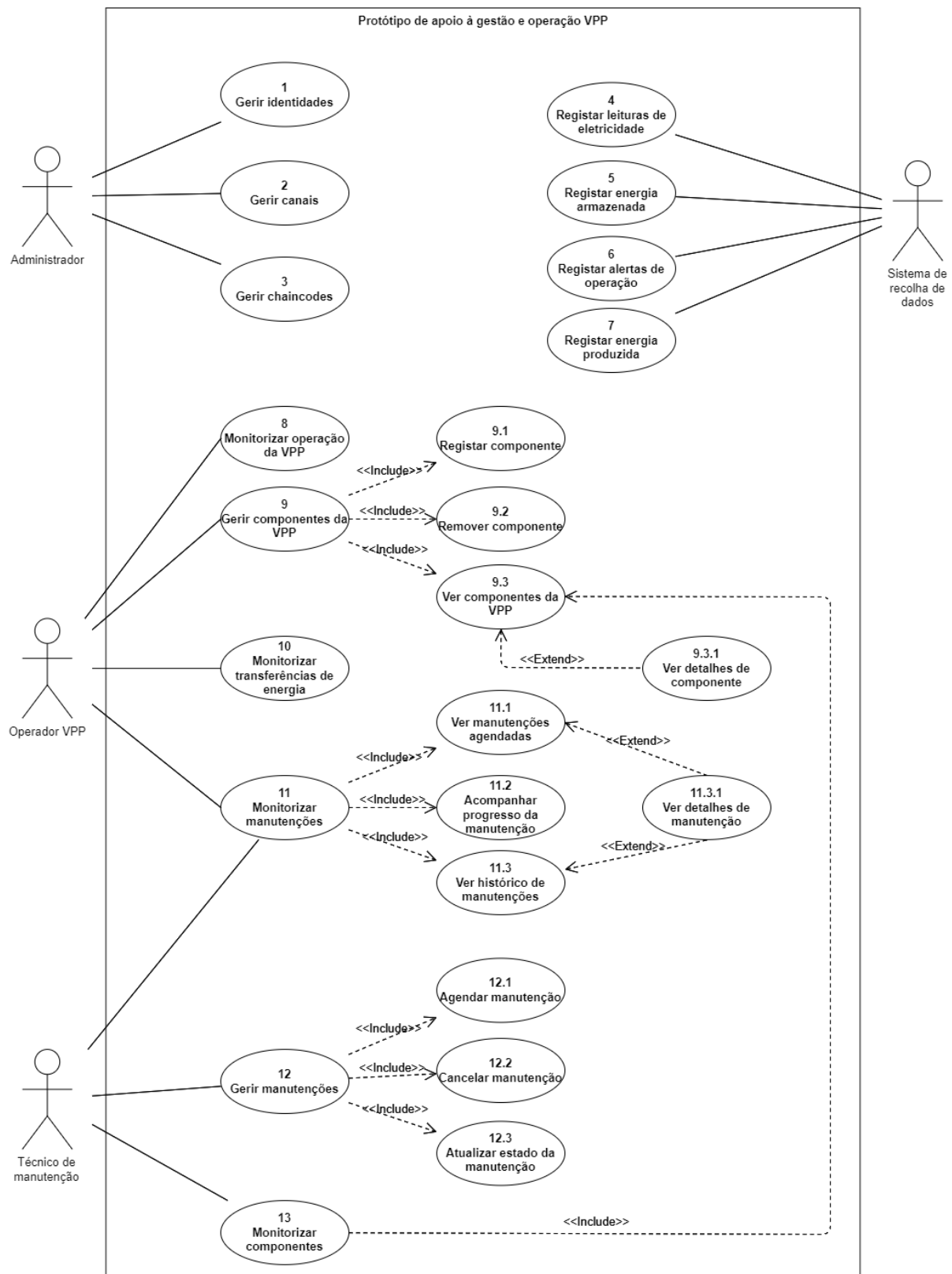


Fig. 14 - Casos de uso, Fonte: autor

No diagrama é possível observar que o sistema é composto por quatro atores. O administrador possui direitos de acesso privilegiado dentro do sistema, sendo este, responsável pela gestão geral do mesmo. O operador VPP é o utilizador responsável pela gestão da operação da VPP.

O técnico de manutenção é o utilizador que realiza as manutenções aos componentes da VPP. Finalmente, o sistema de recolha de dados é o responsável pela recolha, agregação e integração dos dados de contadores inteligentes e sensores dos componentes.

A Tabela 4 apresenta um exemplo de um caso de uso extraído do diagrama, oferecendo uma compreensão mais abrangente dos seus pormenores.

Tabela 4 - Caso de uso 9.1

Caso de uso 9.1 – Registrar componente	
Ator	Operador VPP
Descrição	Este caso de uso envolve o operador VPP pretender efetuar o registo de um componente.
Cenário de sucesso	<ol style="list-style-type: none"> 1. O sistema apresenta a página para registar um componente. 2. O utilizador insere os dados relacionados com o o registo de um componente: tipo de componente, fabricante, referência do produto, capacidade e coordenadas geográficas. 3. O utilizador submete o formulário. 4. O sistema efetua a validação dos dados submetidos. 5. O sistema faz o registo do componente no livro-razão garantindo a imutabilidade e transparência.
Extensões	<p>4a – Os dados são inválidos</p> <ol style="list-style-type: none"> 1. O utilizador corrige os dados que foram indicados como inválidos e submete o formulário. 2. O sistema efetua a validação dos dados submetidos. 3. O sistema faz o registo do componente no livro-razão garantindo a imutabilidade e transparência.

A tabela apresenta informação sobre o cenário de sucesso e um possível caminho alternativo esclarecendo assim a forma de como o sistema deve responder às interações do utilizador e apresentando um plano claro para a sua funcionalidade. Os restantes casos de uso estão disponíveis para referência no Apêndice A – Casos de uso.

Segue-se a descrição dos requisitos que definem de forma sucinta os comportamentos e as funcionalidades que o sistema deve suportar. É importante referir que, embora estes requisitos forneçam uma base para o desenvolvimento do sistema, não são exaustivos. Foram concebidos para orientar o desenvolvimento sem impor restrições muito rígidas.

5.2.1 Requisitos Funcionais

Os requisitos funcionais têm o objetivo de descrever o que o sistema deve fazer. Estes foram classificados de acordo com a sua importância. A priorização foi feita de acordo com o método de MoSCoW [67] que atribui os seguintes graus:

- **Must Have (M)** - Requisitos essenciais sem os quais o projeto não pode ser considerado realizado.
- **Should Have (S)** - Requisitos importantes que devem ser completados se possível.
- **Could Have (C)** - Requisitos que seria bom serem implementados, mas não são essenciais.

- **Won't Have (W)** - Requisitos que não vão ser considerados para o produto final.

A Tabela 5 apresenta a lista de requisitos funcionais que se correlacionam diretamente com os casos de uso.

Tabela 5 - Requisitos funcionais

ID	Requisito	Caso de uso	Prioridade
RF_1	O sistema deve permitir a criação de uma conta de utilizador por parte do administrador.	Caso de uso 1	C
RF_2	O sistema deve permitir a remoção de uma conta de utilizador por parte do administrador.	Caso de uso 1	C
RF_3	O sistema deve permitir a revogação a direitos de acesso a utilizadores ou <i>roles</i> na rede <i>blockchain</i> .	Caso de uso 1	C
RF_4	O sistema deve permitir a criação de novos canais na rede <i>blockchain</i> .	Caso de uso 2	C
RF_5	O sistema deve permitir que uma organização de junte a um canal.	Caso de uso 2	C
RF_6	O sistema deve permitir a instalação de um novo <i>chaincode</i> .	Caso de uso 3	C
RF_7	O sistema deve permitir a atualização de um <i>chaincode</i> .	Caso de uso 3	C
RF_8	O sistema deve permitir o registo das leituras dos contadores associados à VPP.	Caso de uso 4	M
RF_9	O sistema deve permitir o registo da energia armazenada nas baterias.	Caso de uso 5	M
RF_10	O sistema deve permitir o registo de mensagens de alerta (ex. performance, falha).	Caso de uso 6	M
RF_11	O sistema deve permitir o registo da energia produzida por parte dos componentes de produção.	Caso de uso 7	M
RF_12	O sistema deve apresentar dados relativos ao consumo de energia numa VPP.	Caso de uso 8	M
RF_13	O sistema deve apresentar dados relativos à produção de energia numa VPP, incluindo informação sobre as diferentes fontes de energia.	Caso de uso 8	M
RF_14	O sistema deve apresentar dados relativos ao armazenamento de energia numa VPP, incluindo informação sobre a quantidade de energia total armazenada e a sua capacidade.	Caso de uso 8	M

ID	Requisito	Caso de uso	Prioridade
RF_15	O sistema deve apresentar uma lista de alertas que tenham ocorrido em algum componente da VPP.	Caso de uso 8	M
RF_16	O sistema deve apresentar um formulário para inserir os dados do componente a registar.	Caso de uso 9.1	M
RF_17	O sistema deve validar a informação inserida relativa ao registo de um componente.	Caso de uso 9.1	M
RF_18	O sistema deve confirmar o registo de um componente.	Caso de uso 9.1	M
RF_19	O sistema deve permitir associar um ou mais ficheiros a um componente, como por exemplo o manual de instruções.	Caso de uso 9.1	C
RF_20	O sistema deve apresentar uma opção para remover um componente.	Caso de uso 9.2	M
RF_21	O sistema deve solicitar ao utilizador a confirmação do pedido de remoção de um componente.	Caso de uso 9.2	S
RF_22	O sistema deve confirmar a remoção de um componente.	Caso de uso 9.2	S
RF_23	O sistema deve apresentar a lista de componentes registados.	Caso de uso 9.3	M
RF_24	O sistema deve permitir ao utilizador a seleção de um componente da lista para uma vista detalhada sobre o mesmo.	Caso de uso 9.3	M
RF_25	O sistema deve apresentar dados relativos às transferências de energia realizadas, incluindo informação sobre as leituras dos contadores numa VPP.	Caso de uso 10	M
RF_26	O sistema deve fornecer uma página de fácil utilização para aceder à lista de manutenções agendadas.	Caso de uso 11.1	M
RF_27	O sistema deve apresentar a informação sobre todas as manutenções, incluindo detalhes específicos e componente associado.	Caso de uso 11.1	M
RF_28	O sistema deve apresentar uma lista de manutenções a decorrer permitindo uma fácil navegação e seleção.	Caso de uso 11.2	M
RF_29	O sistema deve apresentar uma maneira de visualizar as atualizações do estado de uma manutenção em tempo real.	Caso de uso 11.2	M
RF_30	O sistema deve apresentar uma lista de manutenções terminadas, incluindo detalhes específicos e componente associada.	Caso de uso 11.3	M

ID	Requisito	Caso de uso	Prioridade
RF_31	O sistema deve fornecer um formulário para inserir os dados de agendamento de uma manutenção.	Caso de uso 12.1	M
RF_32	O sistema deve permitir a associação de ficheiros a uma manutenção.	Caso de uso 12.1	C
RF_33	O sistema deve validar a informação inserida relativa ao agendamento de uma manutenção.	Caso de uso 12.1	M
RF_34	O sistema deve confirmar o agendamento da manutenção.	Caso de uso 12.1	S
RF_35	O sistema deve apresentar uma opção para cancelar uma manutenção.	Caso de uso 12.2	M
RF_36	O sistema deve solicitar ao utilizador a confirmação do pedido de cancelamento de uma manutenção.	Caso de uso 12.2	S
RF_37	O sistema deve fornecer uma opção para fornecer uma razão ou explicação para o cancelamento da manutenção.	Caso de uso 12.2	S
RF_38	O sistema deve apresentar uma opção para atualizar o estado da manutenção.	Caso de uso 12.3	M
RF_39	O sistema deve fornecer a opção de adicionar notas ou comentários à atualização do estado da manutenção.	Caso de uso 12.3	S
RF_40	O idioma do protótipo deve ser o inglês.	N/A	M

Esta lista de requisitos funcionais serve de orientação ao processo de desenvolvimento, assegurando que o produto final se alinha com os casos de uso apresentados.

5.2.2 Requisitos não funcionais

Os requisitos não funcionais, também conhecidos por atributos de qualidade, são descrições de como o sistema se deve comportar, ou de uma propriedade ou atributo do sistema. Os requisitos que se seguem foram definidos tendo em conta o contexto de desenvolvimento de um protótipo no âmbito de uma dissertação em que o tempo de desenvolvimento é limitado.

Performance

1. O protótipo deve suportar o acesso simultâneo de até 5 utilizadores sem exceder um tempo de resposta de 3 segundos para as principais operações.

Manutenibilidade

1. O código deve ser bem estruturado e seguir as melhores práticas e padrões para facilitar futuras modificações ou aprimoramentos.

Confiabilidade

1. O protótipo deve apresentar mecanismos de tratamento de erros, identificando e notificando adequadamente os utilizadores ou registando os erros encontrados.

Rastreabilidade

1. O protótipo deve permitir o rastreamento de todas as alterações realizadas pelos utilizadores.

No que diz respeito ao **desempenho**, e apesar da prioridade dada à funcionalidade e viabilidade, o protótipo deve ter a capacidade de suportar alguns utilizadores em simultâneo, mantendo tempos de resposta aceitáveis demonstrando a sua potencial escalabilidade. Relativamente à **manutenibilidade**, e considerando a possibilidade de transferência do código, a aposta em código bem estruturado, boas práticas de programação e documentação complementar, servem como uma base sólida para a compreensão e futura expansão do protótipo. A incorporação de mecanismos de tratamento de erros é importante para garantir que **confiabilidade** do sistema considerando que este se insere num contexto crítico. Por último, a **rastreabilidade** é essencial uma vez que acompanhamento transparente das ações dos utilizadores alinha o desenvolvimento do protótipo com o seu contexto operacional e a normas da indústria.

5.3 Arquitetura de Software

Para comunicar a arquitetura do sistema recorreu-se ao modelo de representação C4 [68]. Este serve para descrever arquiteturas de software e baseia-se na ideia de que um sistema poder ser representado recorrendo a quatro níveis de abstração: **contexto**, **container**, **componente** e **código**. À medida que se vai avançando pelos diferentes níveis, vai-se aumentando a especificidade e o detalhe.

5.3.1 Diagrama de Contexto

O **diagrama de contexto**, representado na Fig. 15, corresponde ao nível que apresenta menor detalhe do sistema. Permite a visualização geral do ambiente onde o sistema opera, a observação dos seus utilizadores e como este interage com outros sistemas.

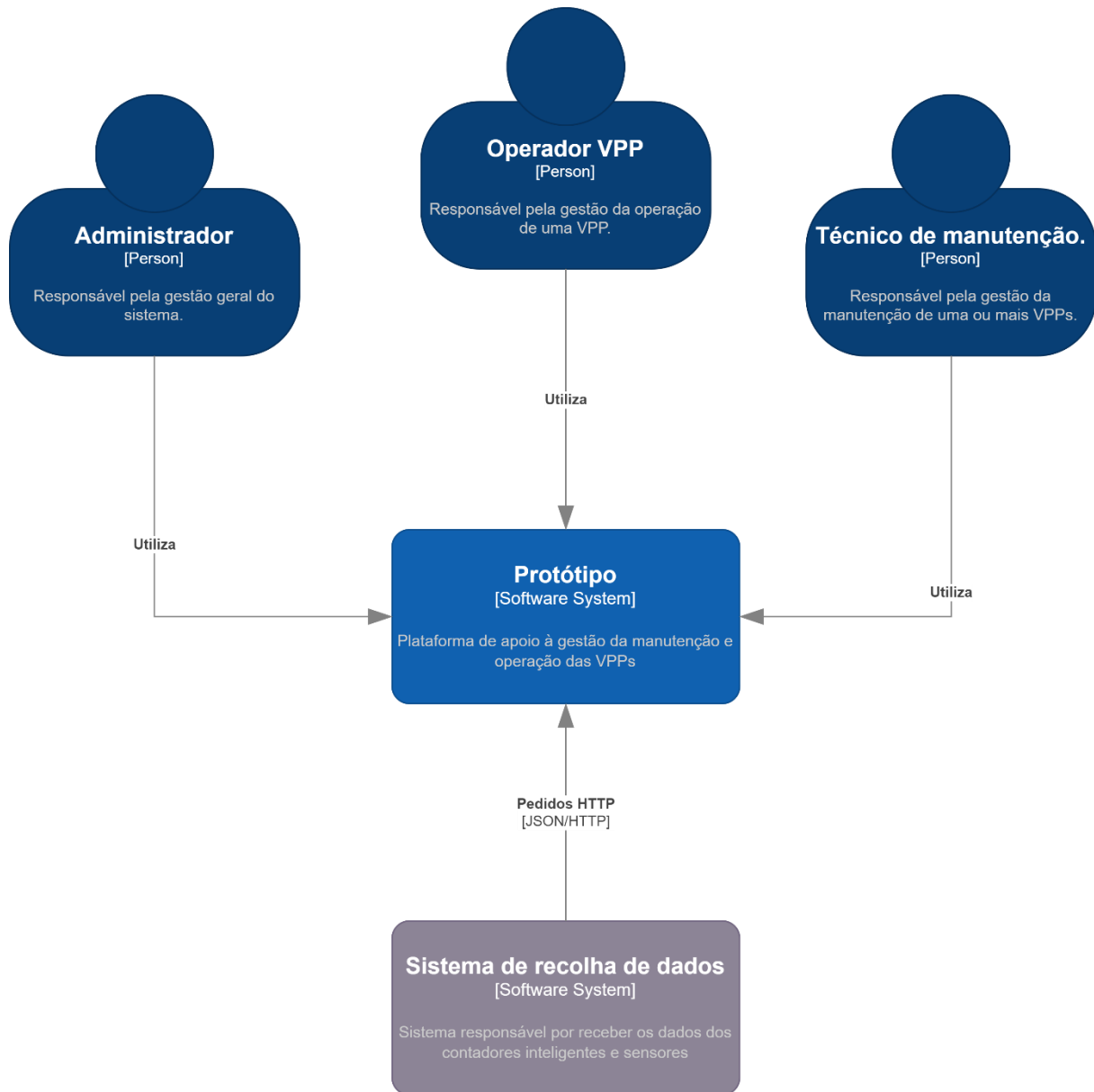


Fig. 15 - Diagrama de contexto, Fonte: autor

Este diagrama de contexto oferece uma visão de alto nível das interações que envolvem o protótipo. Ilustra a forma como os três utilizadores distintos e um sistema de recolha de dados externo se relacionam com o protótipo. As setas dos utilizadores indicam que estes interagem com o protótipo, enquanto a seta do sistema de recolha de dados significa a comunicação com o protótipo através de pedidos HTTP.

5.3.2 Diagrama de Container

O *container diagram* é representado na Fig. 16. Este corresponde a uma ampliação para o interior do sistema e mostra a sua arquitetura de alto nível. Através deste diagrama é possível observar os principais módulos em conjunto com as escolhas tecnológicas associadas a estes, e as principais comunicações que ocorrem no sistema.

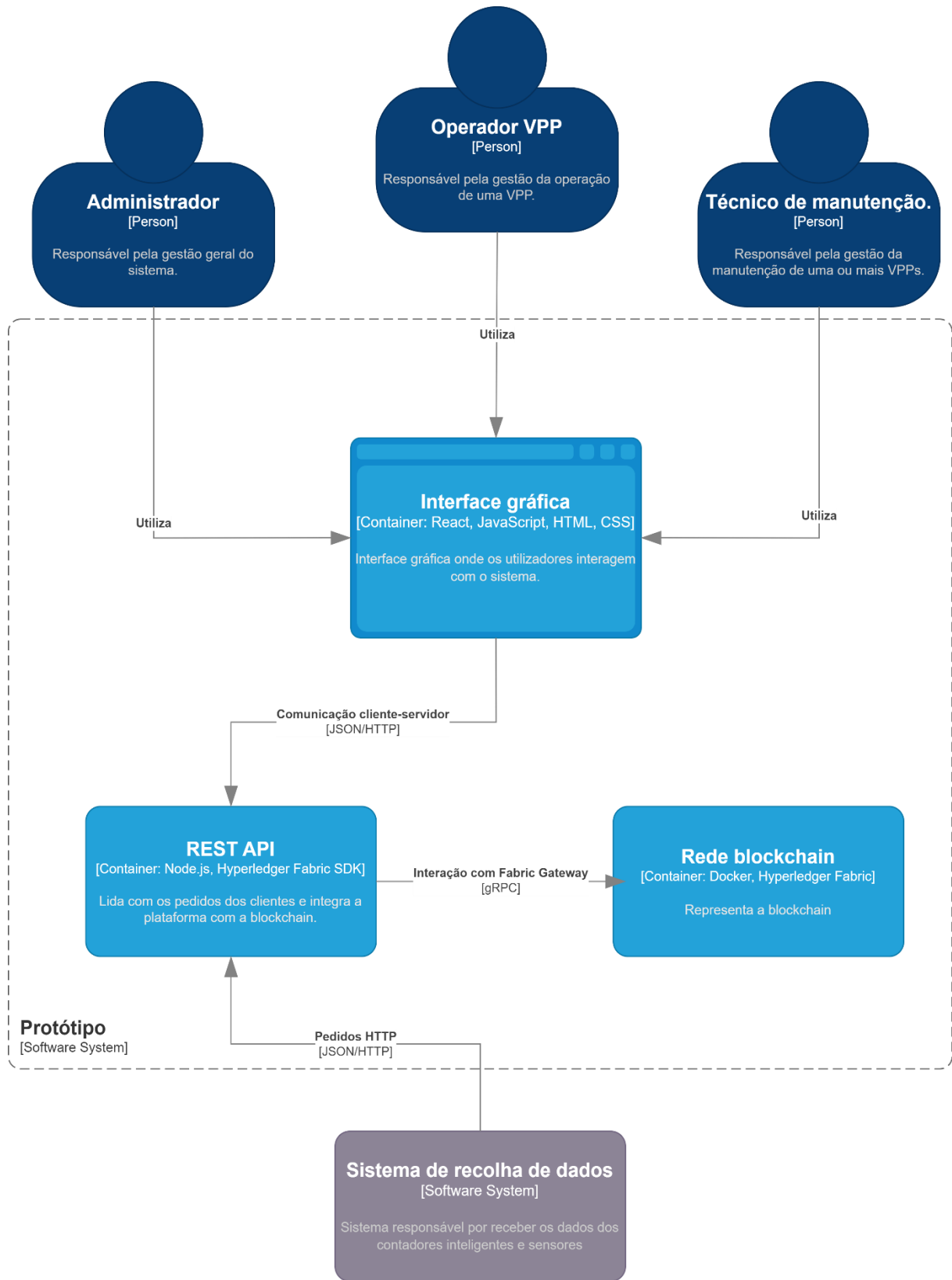


Fig. 16 - Container Diagram, Fonte: autor

O contentor **Interface gráfica** representa uma aplicação web que serve de ponto de entrada para as interações do utilizador com o sistema. Este contentor é composto pelas tecnologias HTML, CSS, JavaScript e React.

O contentor **REST API** é responsável pelo tratamento de pedidos HTTP. Recorre ao Hyperledger Fabric SDK para comunicar com a rede blockchain, podendo assim, interagir com os *chaincodes*, enviar transações e receber eventos originados da rede. A comunicação entre o servidor web e a rede *blockchain* representa a interação necessária para tratar os pedidos dos utilizadores que envolvem transações. O servidor processa os pedidos HTTP, que podem envolver a leitura ou escrita na *blockchain* através da rede.

A **Rede blockchain** corresponde à base do sistema. Dentro deste contentor, encontra-se a estrutura da base de dados distribuída, incluindo os nós da *blockchain*, o livro-razão, os *chaincodes* e os mecanismos de consenso.

A escolha de Hyperledger Fabric como plataforma para o desenvolvimento da rede *blockchain* é justificada, uma vez que no contexto do projeto, é necessário uma *blockchain* que ofereça um ambiente controlado onde apenas entidades de confiança possam ter acesso e contribuir, devido à sensibilidade dos dados. Por outro lado, entre as opções disponíveis para o desenvolvimento de uma *permissioned blockchain*, destacou-se a plataforma Hyperledger Fabric uma vez que foi a única plataforma deste tipo mencionada na literatura, o que sublinha ainda mais ser a escolha adequada para cumprir os requisitos específicos no contexto VPP.

Neste capítulo, foram aprofundados os requisitos e a arquitetura de software, estabelecendo uma base para a implementação bem-sucedida da solução proposta. O capítulo seguinte descreve a fase de desenvolvimento onde são explorados os detalhes sobre a implementação do protótipo.

Capítulo 6

Implementação

Este capítulo centra-se nos pormenores da implementação do protótipo e descreve o processo de desenvolvimento dividido em três partes essenciais para o mesmo. A primeira parte cobre o desenvolvimento suportado na *blockchain* Hyperledger Fabric. Explora as decisões tomadas relativamente à configuração da rede e à implementação dos *chaincodes* que servem de base à lógica de negócio. A segunda parte é dedicada ao desenvolvimento de uma REST API que permite a interação com a rede *blockchain*. Por último, a interface gráfica serve de porta de entrada para a utilização do protótipo. Em conjunto, estes três componentes proporcionam uma visão global do percurso de desenvolvimento que deu vida ao protótipo.

6.1 Blockchain Hyperledger Fabric

Esta secção descreve o processo de desenvolvimento relativo à *blockchain* Hyperledger Fabric. Este foi dividido em várias etapas que envolveram a tomada de diversas decisões. Inicialmente, são descritos os detalhes relativos à configuração da rede *blockchain*. Posteriormente, é efetuada uma análise sobre o processo de desenvolvimento dos *chaincodes*. Cada uma destas etapas constitui um passo fundamental para o desenvolvimento bem-sucedido do protótipo.

6.1.1 Configuração da rede

A implementação do protótipo começou pela configuração da rede *blockchain*. Inicialmente tentou-se a implementação de uma rede de produção, mas, atendendo a tratar-se de um protótipo e ao excessivo tempo estimado para esta tarefa, foi encontrada a solução alternativa de utilizar a rede de teste Hyperledger Fabric durante a fase de desenvolvimento do protótipo. Esta rede é pré-configurada e fácil de utilizar, o que eliminou todas as complicações relativas à configuração da rede de produção. Isto permitiu a concentração na implementação e validação das funcionalidades principais do protótipo, poupando muito tempo. Além disso, a rede de teste também serviu como uma ferramenta de aprendizagem, uma vez que ao longo do desenvolvimento foi proporcionando familiaridade com a estrutura Hyperledger Fabric e os seus componentes. Em última análise, esta decisão foi crucial para a simplificação e aceleração do processo de desenvolvimento.

Como parte do processo de configuração da rede de teste, foi necessário definir o tipo de base de dados. A Hyperledger Fabric fornece duas opções para isso: LevelDB ou CouchDB.

- **LevelDB** é uma base de dados chave-valor projetada para ser rápida e eficiente. Esta oferece simplicidade e utilização eficiente de recursos. Os dados são armazenados em formato binário, o que a torna mais eficiente para guardar grandes quantidade de dados [69].
- **CouchDB** é uma base de dados *open-source*, distribuída e orientada a documentos. Foi concebida para ser escalável, tolerante a falhas e fácil de utilizar. Os dados são armazenados em formato JSON, o que facilita a sua consulta e manipulação [70].

Após análise de ambas as bases de dados, a LevelDB foi escolhida pela sua simplicidade e eficiência. Esta escolha eliminou a necessidade de configurar, gerir e operar uma base de dados externa, como a CouchDB, reduzindo assim a complexidade geral do desenvolvimento.

Após a criação da rede Hyperledger Fabric, a fase seguinte do desenvolvimento envolveu a criação e a implementação dos *chaincodes*, que encapsulam a lógica de negócio do protótipo.

6.1.2 Desenvolvimento dos *chaincodes*

A criação dos *chaincodes* representa um passo essencial, uma vez que estes são responsáveis pela definição da lógica de negócio que governa as interações e transações dentro de rede *blockchain*. A seleção da linguagem de programação adequada para implementar tem impacto no processo de desenvolvimento e a plataforma Hyperledger Fabric oferece três opções: Go (Golang), Node.js ou Java. Após uma análise das opções disponíveis, foi decidido implementar os *chaincodes* em Go uma vez ser a linguagem recomendada pela plataforma e com mais suporte. A implementação da lógica de negócio envolveu a tradução dos requisitos do sistema em código executável.

O desenvolvimento dos *chaincodes* começou pela definição das estruturas de dados dos ativos (*assets*) que integram a VPP e dos seus respetivos ciclos de vida. No contexto Hyperledger Fabric, um ativo refere-se a uma representação digital de um item e pode representar uma vasta gama de objetos tangíveis ou intangíveis. A definição dos ciclos de vida para os ativos permitiu representar com precisão as possíveis transações entre os vários estados. Cada estado corresponde aos dados atuais do ativo. Este passo foi muito importante, pois garante a consistência dos ativos e das transações. No contexto do protótipo, foi concebida uma estrutura de dados que permite gerir eficazmente o fluxo de energia no sistema e assegurar o acompanhamento adequado das manutenções. De seguida, são apresentadas as descrições detalhadas dos ativos que representam a estrutura de dados guardados na *blockchain*, acompanhadas da explicação dos respetivos ciclos de vida:

Componente

O componente representa os vários DERs que se encontram instalados numa VPP. Esta estrutura é composta pelos seguintes dados:

- **Identificador único:** A cada componente é atribuído um identificador distinto, que facilita a identificação e referência inequívocas dentro do sistema.
- **Capacidade:** Cada componente tem uma capacidade de produção de energia no caso dos componentes de produção ou capacidade de armazenamento no caso de baterias.
- **Tipo de componente:** Este campo especifica a categoria ou tipo de componente, permitindo a distinção entre vários componentes, como painéis solares, turbinas eólicas, baterias, entre outros ativos relacionados com a energia.
- **Fabricante:** Captura a informação relativa ao fabricante ou produtor do componente, oferecendo informações sobre a origem do componente.
- **Localização:** A localização geográfica, registada sobre a forma de coordenadas precisas (latitude e longitude), é essencial para a localização exata do ativo.
- **Referência:** A referência corresponde ao número de modelo específico, útil para a obtenção de informações sobre as especificações do componente.
- **Estado:** Indica o estado operacional atual do componente, indicando se está ativo, inativo ou em manutenção.

- **Identificador do proprietário:** Aponta para a entidade ou indivíduo responsável pelo componente.

A Fig. 17 ilustra o ciclo de vida de um componente.

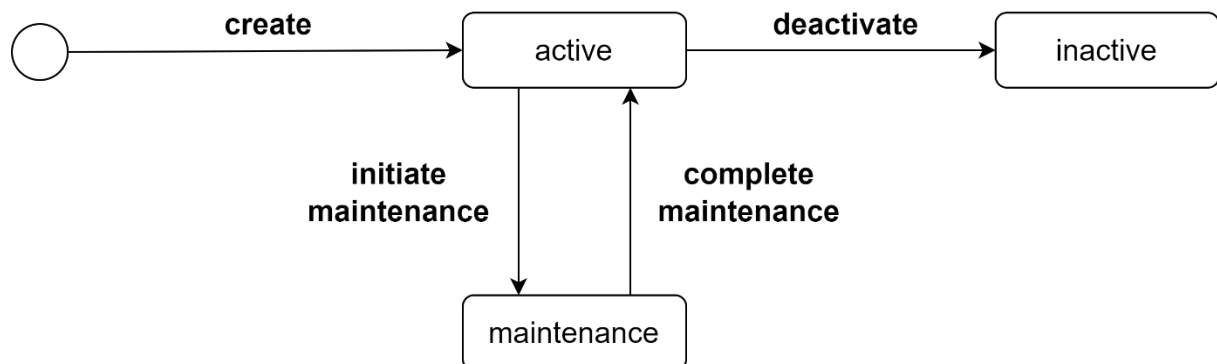


Fig. 17 - Ciclo de vida de um componente, Fonte: autor

O ciclo de vida de um componente começa com a **criação** do mesmo. Esta transação inicia o componente capturando e registando a informação relevante. Após a sua criação, o componente passa para o estado **ativo**. Este estado representa o componente totalmente operacional e a funcionar na VPP, estando disponível para atividade de produção ou armazenamento de energia. No estado **ativo**, o componente pode ser submetido a uma manutenção ou ser desativado. Quando a manutenção é necessária, o componente passa para o estado de **manutenção**. Esta transação é invocada quando o processo de manutenção é iniciado. O componente encontra-se em estado de manutenção até que as atividades estejam concluídas. Uma vez completas as atividades de manutenção, esta transação é executada para assinalar a conclusão da mesma. O estado do componente passa de novo para **ativo**, indicando que está novamente a funcionar de forma normal. Por vezes é necessário retirar um componente do funcionamento ativo. Esta transação é utilizada para desativar um componente, passando-o para o estado inativo. A desativação ocorre quando, por exemplo, um componente é removido. No estado inativo, o componente já não se encontra a contribuir ativamente para as operações da VPP. No entanto, é importante manter um registo da sua existência e o seu histórico dentro do sistema. O estado inativo serve como um marcador histórico, indicando que o componente já fez parte da VPP.

Manutenção

A estrutura de dados manutenção foi concebida para gerir informações essenciais relativas às atividades de manutenção associadas a cada componente. Esta estrutura é composta pelos seguintes dados:

- **Identificador único:** Este identificador é atribuído a cada manutenção para garantir que cada registo é distinto e identificável no sistema.
- **Identificador do componente:** Especifica o componente ao qual foi associada a manutenção.
- **Descrição:** Corresponde a uma descrição textual que fornece informações sobre a natureza e o objetivo dos trabalhos de manutenção. Pode incluir pormenores sobre tarefas a realizar, questões abordadas ou qualquer outra informação relevante.
- **Tipo de manutenção:** Indica se a manutenção é preventiva ou corretiva de maneira a ajudar a perceber o objetivo da manutenção.

- **Data de início:** Corresponde à data para o qual é agendado o início dos trabalhos de manutenção.
- **Data de fim:** Corresponde à data para o qual é previsto o final dos trabalhos de manutenção.
- **Estado:** Indica o estado atual da manutenção, permitindo saber se esta se encontra agendada, a decorrer, terminada ou se foi cancelada.

Em suma, este ativo permite o registo de informações cruciais relacionadas com a manutenção dos componentes, assegurando que os eventos sobre as manutenções possam ser rastreados e analisados de forma eficaz. A Fig. 18 ilustra o ciclo de vida de uma manutenção.

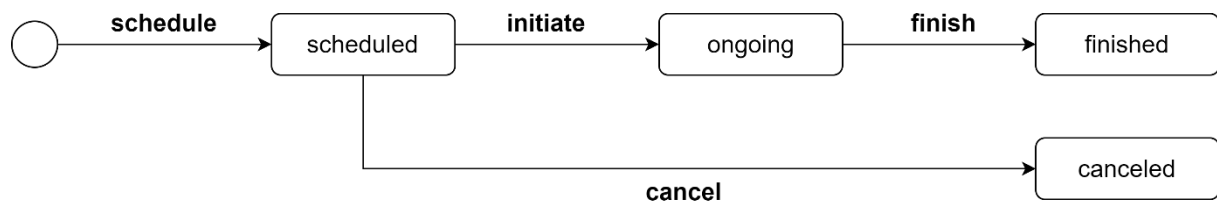


Fig. 18 - Ciclo de vida de uma manutenção, Fonte: autor

O ciclo de vida de uma manutenção começa com a transação **agendar** onde são capturados os dados relevantes a este fim e onde o seu estado inicial é definido como **agendada**. Quando a manutenção está **agendada** significa que esta se encontra planeada, mas ainda não está em andamento. A tarefa é programada para uma data e hora. Enquanto estiver neste estado a manutenção pode ser iniciada ou cancelada. A transação **iniciar** passa o estado da manutenção de **agendada** para **a decorrer**. O estado **a decorrer** representa que a tarefa de manutenção está a ser executada ativamente. A equipa de manutenção executa as atividades necessárias para responder às necessidades de manutenção do componente. Uma vez terminadas estas atividades, e executada a transação de **terminar**. Esta transação atualiza o estado da manutenção para **terminada** como forma de indicar que a manutenção foi concluída com sucesso. Se, por algum motivo, a tarefa de manutenção precisar de ser cancelada, é chamada a transação **cancelar**. Esta transação só é aplicável se a tarefa de manutenção estiver no estado **agendada** e atualiza o estado para **cancelada**, indicando que a manutenção foi cancelada.

Alerta

Esta estrutura representa os alertas associados a cada componente que são emitidos por sensores. A sua estrutura é composta pelos seguintes dados:

- **Identificador único:** Este identificador é atribuído a cada alerta para garantir a identificação e referência inequívoca dentro do sistema.
- **Identificador do componente:** Este identificador especifica o componente ao qual foi associado o alerta.
- **Tipo de alerta:** Indica se o alerta corresponde a um aviso ou a uma falha de maneira a perceber a gravidade do alerta emitido.
- **Descrição:** Corresponde a uma descrição textual que fornece informações sobre a natureza do alerta.
- **Data de emissão:** Data em que o alerta foi emitido.

A Fig. 19 ilustra o ciclo de vida de um alerta.

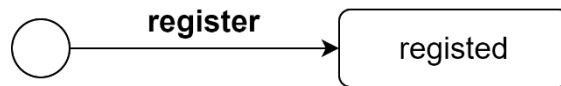


Fig. 19 - Ciclo de vida de um alerta, Fonte: autor

O ciclo de vida de um alerta começa com o seu registro. Esta transação é executada para registrar os dados do alerta capturando informações relevantes sobre o mesmo. O estado **registrado** significa que o alerta se encontra disponível no livro-razão.

Leitura do contador

Esta estrutura representa os dados de energia registrados pelos contadores inteligentes e inclui a seguinte informação:

- **Identificador único:** É atribuído a cada registro de leitura de forma a garantir a identificação e referência inequívoca dentro do sistema.
- **Consumo:** Representa o consumo de energia durante o intervalo de tempo da leitura.
- **Produção:** Representa a produção de energia durante o intervalo de tempo da leitura.
- **Procura máxima:** Representa o valor máximo de potência registrado no intervalo de tempo da leitura.
- **Energia importada:** Corresponde à energia recebida da rede.
- **Energia exportada:** Corresponde à energia exportada para a rede.
- **Proprietário do contador:** Estabelece uma relação entre o contador e o seu proprietário.

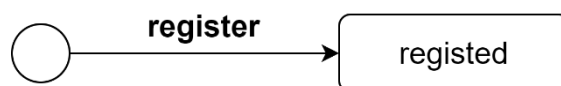


Fig. 20 - Ciclo de vida de uma contagem, Fonte: autor

O ciclo de vida de uma leitura de um contador de energia começa e termina com a transação **registrar**. Esta transação é executada para registrar dados energéticos de um determinado contador e captura informações relevantes, como a identificação do contador, o timestamp, e detalhes sobre a energia. O estado **registrado** indica que os dados de energia estão disponíveis para análise, monitorização, faturação ou quaisquer outros processos relevantes à gestão da VPP. Confirma que o evento relacionado com a energia foi processado e documentado com sucesso, fornecendo um registro fiável da atividade energética que ocorreu num determinado momento.

Após a definição da estrutura de dados dos ativos, foi dada especial atenção à estratégia de armazenamento dos mesmos. A base de dados selecionada, LevelDB, segue um modelo de dados chave-valor. Neste modelo, os dados são representados como uma coleção de pares chave-valor em que uma chave é um indentificador único que mapeia para um valor

correspondente. Para garantir a organização e a consulta eficiente dos dados, foi imperativo estabelecer uma estrutura de chaves bem definida.

Ao criar a estrutura de chaves, foram tidas em conta várias considerações importantes. Em primeiro lugar, garantir a exclusividade de cada chave era prioridade para evitar quaisquer potenciais conflitos em que dois ativos pudessem partilhar a mesma chave. Em segundo lugar, a estrutura de chaves foi concebida para ser de fácil utilização, facilitando a geração de chaves e a sua utilização para consultar o livro-razão.

A abordagem escolhida envolveu a utilização dos identificadores únicos dos ativos como chave. Ao adotar esta estratégia foi possível garantir um método fiável e simples de identificar os ativos individuais. Contudo, a mera utilização dos identificadores únicos como chaves apresentou desafios quando se pretendia consultar dados com base em relações ou associações, como por exemplo:

- Consultar todos os registos de manutenção associados a um componente
- Consultar todas as leituras de um contador associado a um cliente
- Consultar todas as leituras de um contador que ocorreram num intervalo de tempo

Estes são apenas alguns exemplos em que apenas com uma chave simples não é possível obter o resultado de uma forma eficiente. Para resolver esta limitação, a estrutura de chaves foi melhorada com a introdução de chaves compostas, que permitem a criação de listas. Estas listas agrupam ativos associando-os a uma chave composta. A chave composta identifica exclusivamente um estado e serve como um meio eficaz de consultar a lista. Ao utilizar chaves compostas, é possível organizar e indexar ativos relacionados de uma forma que melhora significativamente o desempenho da consulta de dados. Veja-se o seguinte exemplo:

Ao registar uma manutenção no livro-razão, o sistema segue uma estratégia de chaves que permite uma consulta eficiente dos registos de manutenção associados a um componente específico. São criados dois tipos de chaves durante o processo de registo de uma manutenção:

- **Chave simples:** Esta chave serve para identificar uma manutenção específica e é formada pelo seu identificador único.
- **Chave composta (índice):** Esta chave é utilizada para agrupar manutenções com base no componente associado. É formada pela concatenação do identificador do componente a que se pretende associar a manutenção e do identificador da manutenção, com o prefixo “componente~manutenção”. Ao contrário da chave simples, esta chave não está associada a nenhum valor (é atribuído um valor nulo) para evitar a duplicação das manutenções no livro-razão. A chave composta atua como índice que organiza todas as manutenções pelo componente correspondente.

Para efetuar a consulta de todas as manutenções associadas a um componente são realizados os seguintes passos:

1. Cria-se uma chave composta com o prefixo “componente~manutenção” e associa-se o identificador do componente que se pretende consultar as manutenções.
2. Utiliza-se a chave composta para obter a lista de todas as chaves compostas que correspondem ao mesmo padrão.
3. Para cada chave composta recuperada da lista, retira-se o identificador da manutenção através da divisão da mesma.
4. Utiliza-se o identificador da manutenção como chave para se obter os dados registados no livro-razão.

5. As manutenções são retiradas e anexadas a uma lista de resultados.

Ao utilizar esta estratégia de chaves, o sistema organiza eficientemente as manutenções por componente permitindo uma consulta rápida.

Após a explicação dos detalhes envolvidos no estabelecimento da rede *blockchain* e no desenvolvimento dos *chaincodes*, o foco passa para a explicação da REST API.

6.2 REST API

A REST API atua como uma ponte entre a interface gráfica do utilizador e a rede *blockchain*. Esta é responsável pelo processamento de pedidos HTTP efetuados pelos utilizadores e tem a capacidade de interagir com a rede *blockchain* através da submissão de transações ou a consulta de dados do livro-razão.

O primeiro passo no desenvolvimento da API consistiu na configuração de um novo projeto Node.js recorrendo ao comando *npm*. Foi estabelecida a estrutura de pastas necessárias e as dependências essenciais, como o Express e o Hyperledger Fabric SDK, foram instaladas.

Express [71] é uma *framework* para o Node.js que simplifica a criação de APIs através de um conjunto de funcionalidades. Entre estas funcionalidades, o Express fornece a possibilidade de encaminhamento de pedidos HTTP para os respetivos *endpoints* permitindo assim o tratamento dos mesmos. Fornece métodos e propriedade para aceder aos dados dos pedidos e enviar respostas aos clientes.

Hyperledger Fabric SDK [72] é uma dependência crucial que permite a interação com a rede Hyperledger Fabric através de um código simples e minimalista. Dois componentes essenciais desta são os módulos **fabric-network** e **fabric-ca-client**. O **fabric-network** [73] é responsável por abstrair a camada de comunicação entre a aplicação cliente e a rede *blockchain* permitindo a conexão à rede *blockchain*, a submissão de transações, a consulta de dados do livro-razão e a interação com os *chaincodes*. O **fabric-ca-client** [74] lida com a gestão de identidades, garante um controlo de acesso seguro e fornece mecanismos de autenticação para os participantes na rede *blockchain*.

Quando a API recebe um pedido HTTP, este é encaminhado para o respetivo *endpoint* iniciando assim o processo de interação com a rede *blockchain*. Antes da interação, o pedido é processado e os dados necessários são retirados do corpo do pedido.

Para que cada pedido seja executado, é criada uma conexão à rede *blockchain*. O processo de ligação à rede começa com a inicialização de um objeto da classe “Gateway” disponível no Hyperledger Fabric SDK. Este objeto representa a ligação a um ponto central de acesso à rede Hyperledger Fabric. Para criar este objeto, é necessário especificar detalhes de configuração importantes como o perfil de conexão e a identidade que são obtidos através dos seguintes passos:

1. **Criar perfil de conexão:** O perfil de conexão é um objeto JSON que especifica os parâmetros da rede Hyperledger Fabric.
2. **Retirar informação da carteira digital.** O processo de retirar informação da carteira digital no Hyperledger Fabric começa com a criação de um objeto carteira através da utilização da classe “Wallets” importada do Hyperledger Fabric SDK. Este objeto representa um ficheiro, que é um local seguro para armazenar informações de identidade, tais como os certificados e chaves privadas dos utilizadores. Uma vez criado, este objeto é utilizado para obter informações sobre qualquer utilizador que se encontre

registrado. Estas informações incluem o certificado de registo do utilizador, que o identifica na rede, e a sua chave privada, que é utilizada para assinar transações.

Uma vez criado o perfil de conexão, e retirada a identidade da carteira digital, pode-se estabelecer ligação com a rede *blockchain*. Este passo é realizado através do método “**connect()**” no objeto *gateway*. Assim que o gateway estiver ligado à rede, o pedido pode ser processado e a interação com a rede pode ser realizada.

6.3 Interface gráfica

A interface gráfica serve de ligação entre os utilizadores e as funcionalidades do software subjacente. No desenvolvimento do protótipo, foi utilizado o React que é uma biblioteca JavaScript conhecida por permitir a criação de interfaces dinâmicas e responsivas. De seguida serão exploradas as várias páginas e funcionalidades que foram concebidas. A Fig. 21 apresenta o dashboard, com as informações chave para a operação da VPP.

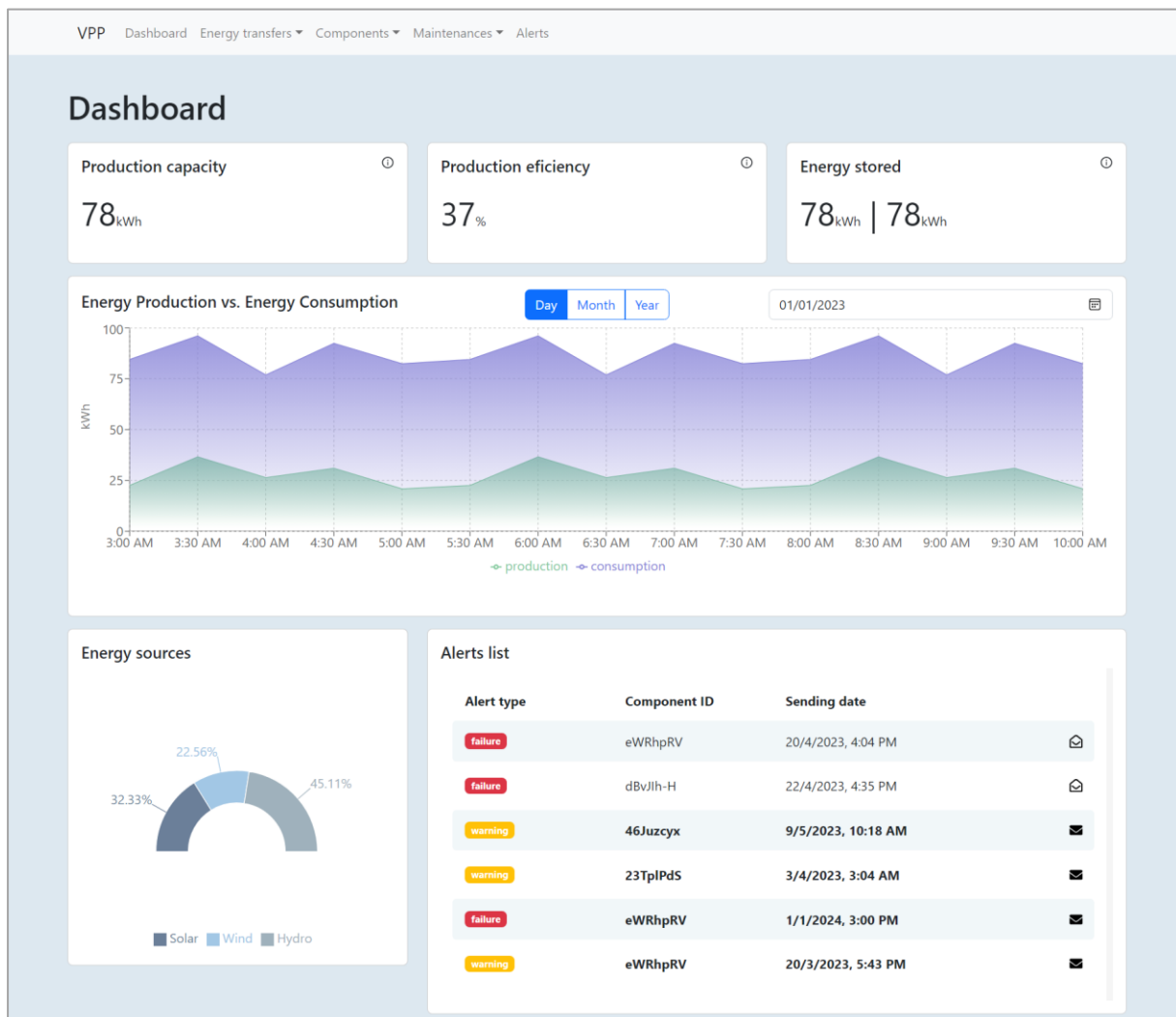


Fig. 21 - Dashboard, Fonte: autor

O *dashboard* oferece uma visão global do desempenho da VPP, fornecendo aos utilizadores informações cruciais para uma gestão e monitorização eficazes. Apresenta a capacidade de

produção combinada de todos os componentes de produção operacionais, juntamente com métricas de eficiência de produção. Esta métrica representa a capacidade da VPP para utilizar a sua capacidade instalada de forma otimizada e gerar a máxima produção de energia. A energia armazenada no sistema de armazenamento é outra métrica essencial apresentada no painel de controlo uma vez que permite aos utilizadores compreender as reservas de energia da VPP e a sua capacidade de fornecer energia durante os períodos de baixa produção. O *dashboard* inclui também gráficos que mostram a energia consumida e produzida, bem como a distribuição das fontes de energia. Finalmente, o *dashboard* fornece uma lista de alertas relacionados com potenciais problemas de manutenção que possam surgir no funcionamento da VPP. Em geral, a página permite aos utilizadores monitorizar eficazmente as operações da VPP e tomar decisões informadas que apoiam a operação sustentável e eficiente da VPP.

De seguida serão apresentadas as páginas associadas ao menu, na parte superior, iniciando-se pelas transferências de energia (Fig. 22).

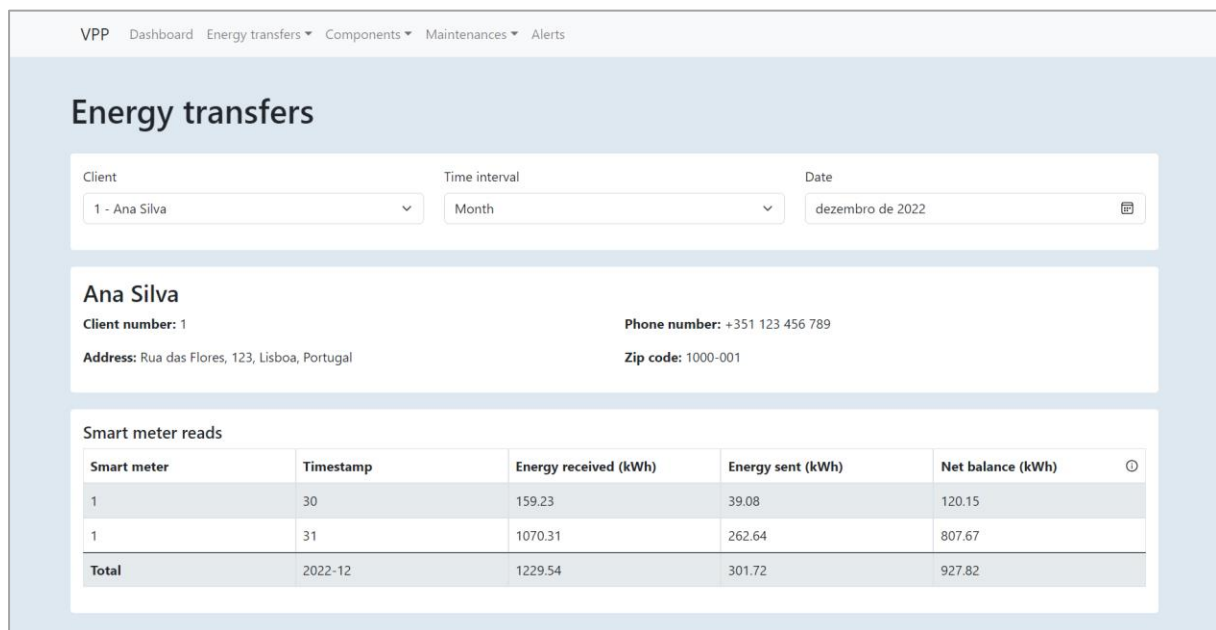


Fig. 22 – Transferências de energia, Fonte: autor

A página “transferências de energia” fornece os dados energéticos detalhados de cada cliente da VPP. Estes dados correspondem a leituras dos contadores inteligentes e são registados no sistema através de um pedido HTTP para o respetivo *endpoint* na API. Ao selecionar um cliente específico, os utilizadores podem ver a informação completa sobre as suas transferências de energia. A página apresenta uma tabela com as leituras recolhidas pelos contadores inteligentes do cliente em intervalos de tempo regulares. Estes dados mostram a energia que o cliente consumiu e enviou para a VPP. Esta informação melhora a visibilidade dos dados de energia que servem de apoio à gestão eficaz da energia na VPP.

No que diz respeito à gestão dos componentes da VPP, existem três páginas: uma página para registar um componente, outra para listar os componentes pertencentes à VPP e uma terceira para visualizar os detalhes de um componente. A Fig. 23 apresenta a página de registo de um componente.

VPP Dashboard Energy transfers Components Maintenances Alerts

Component registration

Please fill out the form with the relevant information to register the component.

Component type

Select the component type.

Manufacturer Product reference Capacity (kW)
Insert the product manufacturer. Insert the product reference. Insert the component capacity.

Geographic location
 Latitude Longitude
Insert the latitude in degrees. Insert the longitude in degrees.

[Register](#)

Fig. 23 - Registrar componente, Fonte: autor

A página “registrar componente” apresenta um formulário de fácil utilização que permite o registo rápido de novos componentes na VPP. Os utilizadores podem seleccionar o tipo de componente, o fabricante, a referência do produto, a capacidade de produção ou armazenamento e as coordenadas geográficas. O formulário valida a introdução dos dados para garantir a sua exatidão. Após a submissão, os utilizadores recebem uma mensagem de confirmação a indicar que o sucesso do registo. De seguida, a Fig. 24 apresenta a página que permite a visualização dos componentes registados na VPP.

VPP Dashboard Energy transfers Components Maintenances Alerts

Components

[Register component](#)

Active

10

Inactive

0

Under maintenace

0

Components list

Active
Inactive
Under maintenace

Component ID	Component type	Component status
1	solar panel	active
10	biomass Generator	active
2	wind turbine	active
3	battery energy storage system	active
4	hydroelectric Generator	active
5	biomass Generator	active
6	solar panel	active

Fig. 24 – Lista de componentes, Fonte: autor

A página “lista de componentes” fornece aos utilizadores uma visão geral abrangente de todos os componentes da VPP. A página lista todos os componentes e agrupa-os em diferentes categorias com base no seu estado. Os utilizadores podem filtrar facilmente os componentes para ver os ativos, inativos ou em manutenção. Na parte superior da página, existem três cartões que apresentam o número de componentes em cada estado. Se um utilizador selecionar um determinado componente, este é redirecionado para a respetiva página de detalhes, apresentada na Fig. 25.

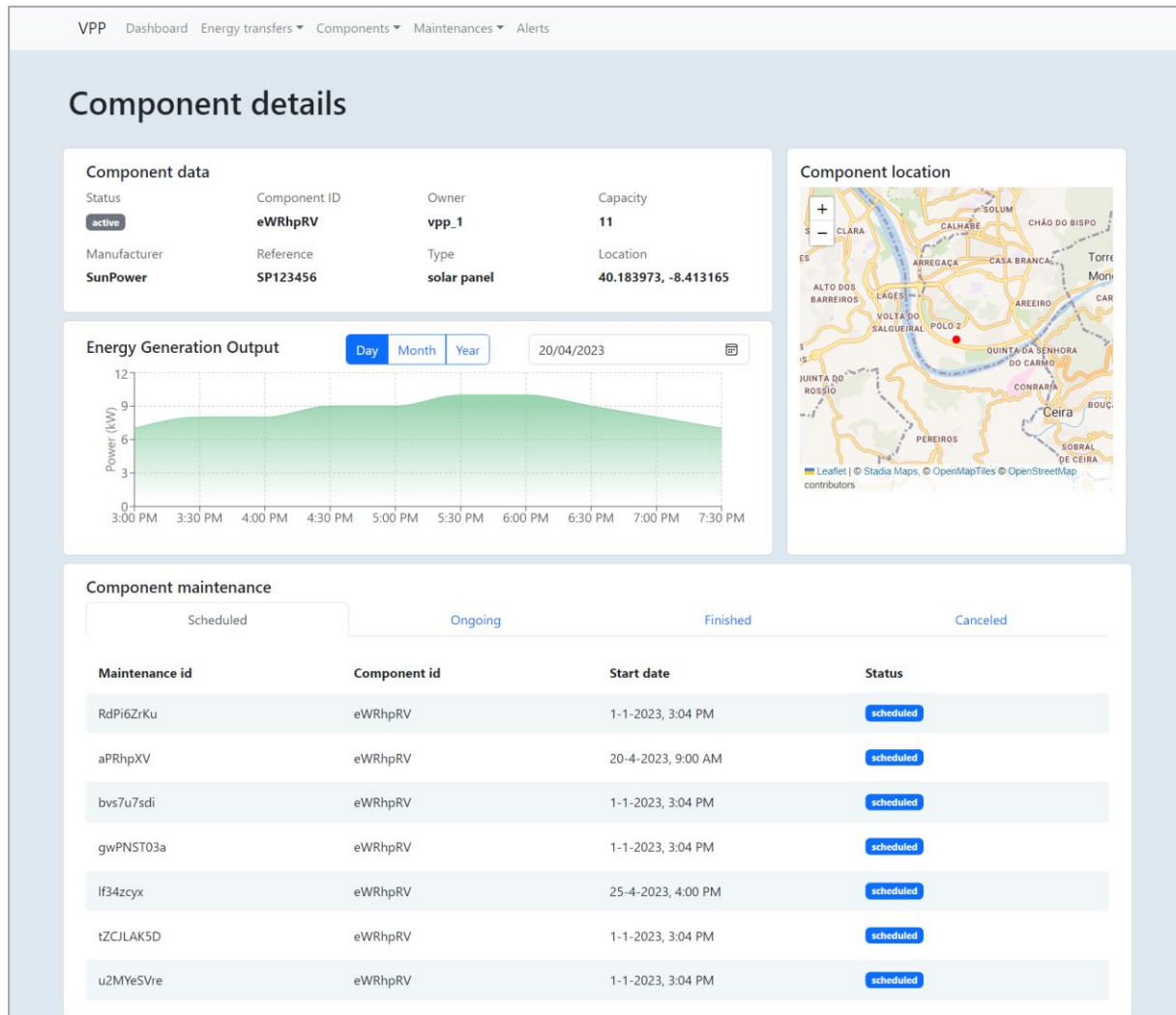
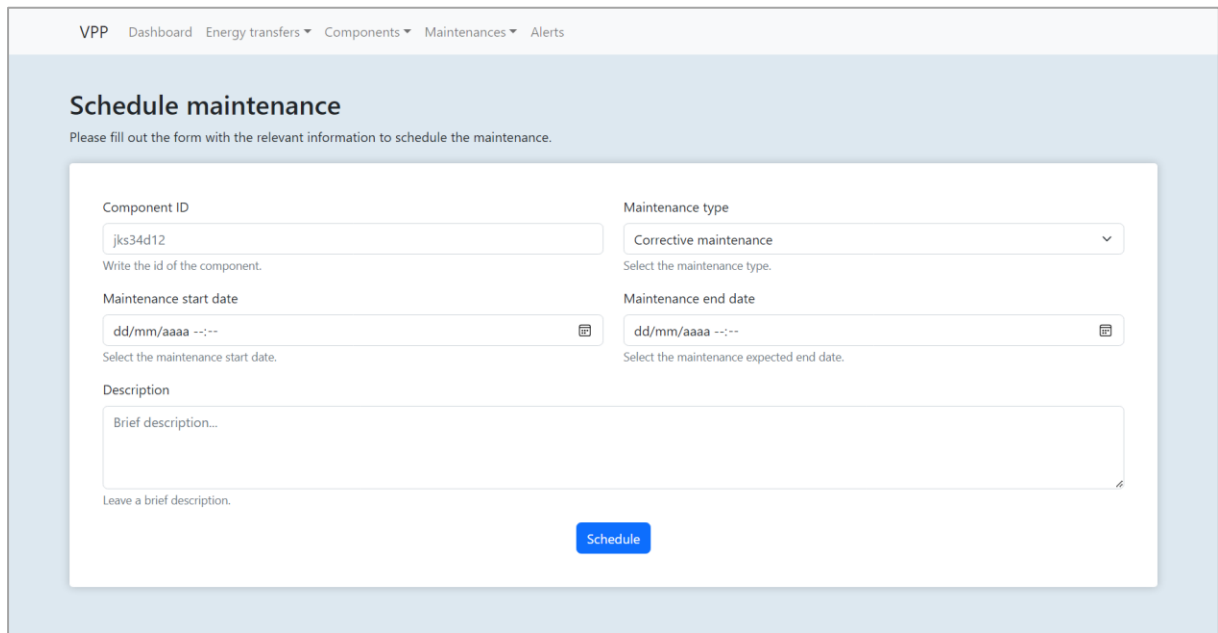


Fig. 25 - Detalhes de um componente, Fonte: autor

A página “detalhes de um componente” oferece aos utilizadores um resumo completo dos componentes. As informações do componente são apresentadas A visualização do mapa ajuda os utilizadores a localizar rapidamente o componente no VPP, aumentando a consciência espacial. A página também apresenta um gráfico que mostra a atividade do componente, o que ajuda a avaliar o seu desempenho. Para componentes de produção, o gráfico mostra a atividade de produção de energia enquanto para os componentes de armazenamento, mostra as variações do nível de armazenamento. A lista de manutenções relacionadas com o componente também é apresentada, permitindo aos utilizadores rever histórico de manutenção, identificar problemas recorrentes e planear tarefas de manutenção futuras de forma eficiente.

Relativamente à gestão das manutenções, a Fig. 26 mostra a página concebida para efetuar o agendamento de uma manutenção.



The screenshot shows a web interface for scheduling maintenance. At the top, there is a navigation bar with the following items: VPP, Dashboard, Energy transfers, Components, Maintenances, and Alerts. Below the navigation bar, the main heading is "Schedule maintenance". Underneath the heading, there is a sub-heading: "Please fill out the form with the relevant information to schedule the maintenance." The form itself is contained within a white box and includes the following fields:

- Component ID:** A text input field containing "jks34d12". Below it, the instruction "Write the id of the component." is displayed.
- Maintenance type:** A dropdown menu with "Corrective maintenance" selected. Below it, the instruction "Select the maintenance type." is displayed.
- Maintenance start date:** A date picker field showing "dd/mm/aaaa --:--". Below it, the instruction "Select the maintenance start date." is displayed.
- Maintenance end date:** A date picker field showing "dd/mm/aaaa --:--". Below it, the instruction "Select the maintenance expected end date." is displayed.
- Description:** A large text area with the placeholder "Brief description...". Below it, the instruction "Leave a brief description." is displayed.

At the bottom center of the form, there is a blue button labeled "Schedule".

Fig. 26 - Agendar manutenção, Fonte: autor

A página “agendar manutenção” apresenta um formulário de fácil utilização que permite aos utilizadores agendar eficazmente novas manutenções para os componentes da VPP. Os utilizadores podem seleccionar o componente, escolher o tipo de manutenção, introduzir datas de início e fim e fornecer notas. O formulário inclui verificação de validação para garantir a exatidão da introdução dos dados. Uma vez submetido, é apresentada uma mensagem que confirma o agendamento. O formulário simplificado contribui para melhorar a manutenção da VPP aumentando assim a sua eficiência operacional. A lista de manutenções pode ser visualizada na página ilustrada na Fig. 27.

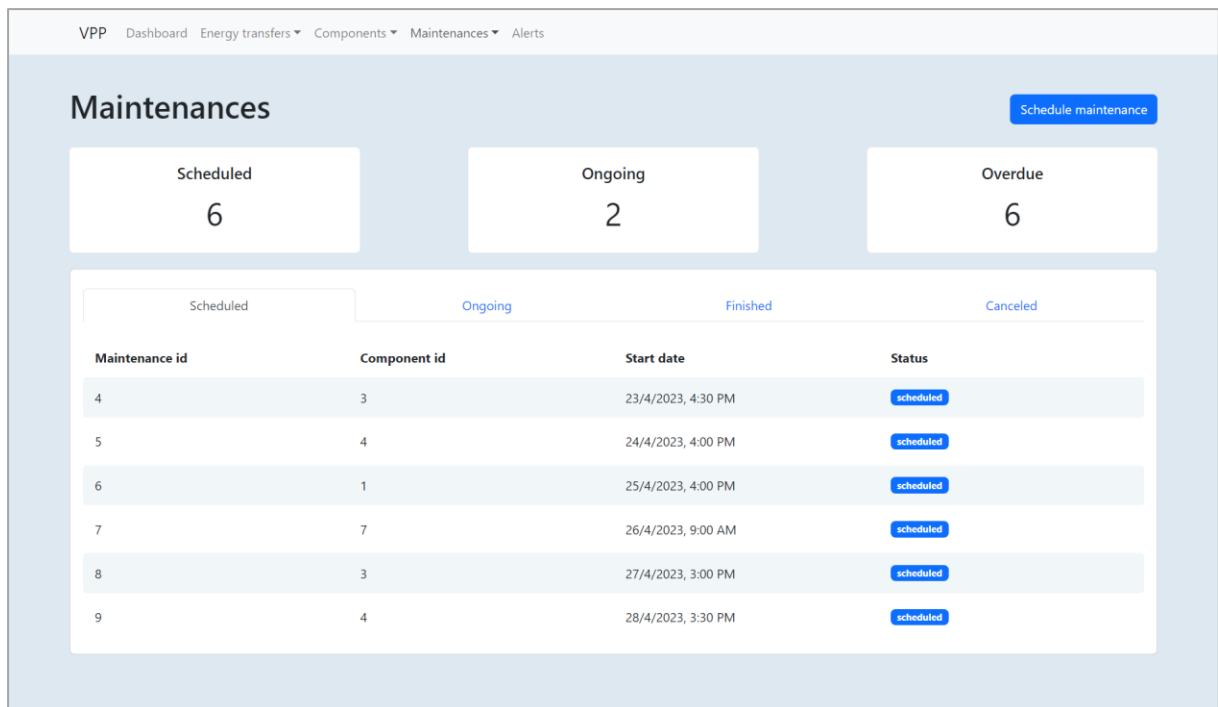


Fig. 27 - Lista de manutenções, Fonte: autor

A página “lista de manutenções” fornece aos utilizadores uma visão geral abrangente de todas as atividades de manutenção associadas aos componentes da VPP. Apresenta lista de manutenções, categorizadas em diferentes estados e oferece aos utilizadores filtros convenientes para visualizar manutenções agendas, a decorrer, terminadas e canceladas. No topo da página são apresentados 3 cartões que fornecem aos utilizadores uma visão rápida do número de tarefas de manutenção agendadas, a decorrer e que ultrapassaram a data de início programadas, mas ainda não começaram. Cada manutenção na lista é interativa, permitindo aos utilizadores selecionares manutenções específicas para aceder à respetiva página com aos seus detalhes e progresso (Fig. 28).

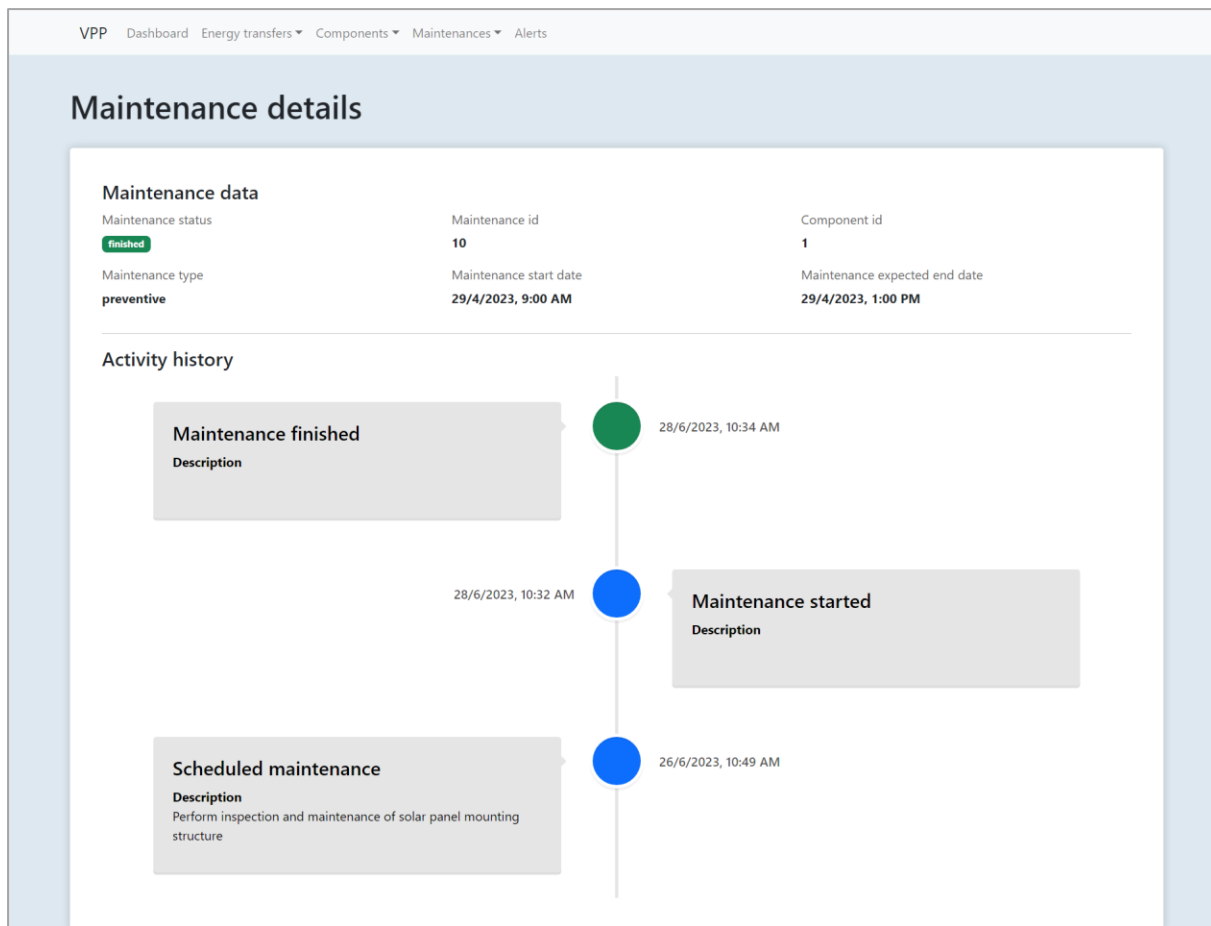


Fig. 28 - Detalhes de uma manutenção, Fonte: autor

A página “detalhes de uma manutenção” oferece aos utilizadores uma visão abrangente e em tempo real de uma tarefa de manutenção específica. Ao apresentar informações detalhadas e um linha temporal, os utilizadores podem acompanhar o progresso da manutenção, aumentando a transparência dos dados. As atualizações em tempo real e as notas de progresso garantem que todos os utilizadores estão informados sobre o estado atual da manutenção. Esta página contribui para a melhoria da gestão das manutenções.

Finalmente, a Fig. 29 ilustra a página onde se podem ver os detalhes de um alerta.

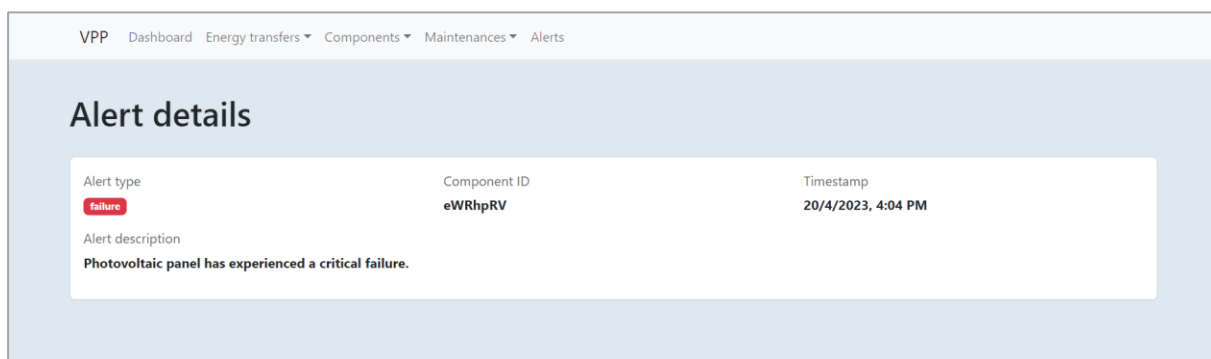


Fig. 29 - Detalhes de um alerta, Fonte: autor

A página com os detalhes de um alerta fornece aos utilizadores uma visão detalhada de alertas relacionados com os ativos da VPP. Esta página é acedida quando um utilizador clica num alerta específico da lista de alertas apresentados na página Dashboard ou na página Alertas. A página apresenta informações sobre o alerta em específico, incluindo o tipo, o componente associado, uma pequena descrição e um *timestamp*. Estas informações podem ser utilizadas para abordar de forma eficaz o alerta permitindo ao utilizador uma resposta que resolva eficazmente algum evento crítico que ocorra durante a operação da VPP.

Após a exploração do processo de desenvolvimento do protótipo, é altura de mudar o foco para a fase de testes. No próximo capítulo são apresentados os testes realizados ao protótipo após a sua implementação.

Capítulo 7 Testes

Este capítulo explora os procedimentos de teste realizados para validar a solução proposta. Começa por fornecer informações sobre o ambiente de teste, esclarecendo as condições em que o protótipo foi avaliado. Posteriormente, aprofunda os resultados dos testes unitários. Estes testes centram-se nos métodos implementados nos *chaincodes* para garantir que estes estão em conformidade com o previsto. De seguida, avança para os testes de integração, que servem de base para afirmar a integração da API com a rede *blockchain*. Finalmente, são destacados os testes de sistema que serviram para testar o protótipo como um todo, validando a implementação dos requisitos.

7.1 Configuração do ambiente de testes

Antes de analisarmos em detalhe os diferentes tipos de testes realizados, é importante compreender o ambiente de testes que foi criado para os realizar. A criação de um ambiente adequado desempenha um papel crucial na obtenção de resultados confiáveis e representativos. Para este fim, foi utilizada uma máquina virtual hospedada nos servidores do Departamento de Engenharia Informática da Universidade de Coimbra com o sistema operativo Ubuntu 20.04, 8 vCPUs, 16 GB de RAM e 150 GB de armazenamento.

Inicialmente, procedeu-se à criação da rede *blockchain*. Para isso, foi implementada uma rede de testes da plataforma Hyperledger Fabric. Esta escolha forneceu uma infraestrutura essencial para a realização dos testes.

Uma vez estabelecida a rede *blockchain*, foram instalados os *chaincodes* com a lógica de negócio do protótipo. Concluída a sua instalação, foi realizada a população da *blockchain* através da função “InitLedger()”, incorporada nos próprios *chaincodes*. Esta função desempenhou o papel vital de popular a *blockchain* com os ativos essenciais (ex.: manutenções, componentes), a fim de simular um cenário realista. A ausência de dados reais para testar o protótipo fez com que estes fossem cuidadosamente concebidos para representarem de forma abrangente os dados que o sistema pode cobrir.

Posteriormente, o servidor da REST API e a aplicação React foram colocadas em operação concluindo a configuração do ambiente de testes e deixando tudo pronto para a realização dos mesmos.

7.2 Testes unitários

Os testes unitários desempenham um papel fundamental no processo de desenvolvimento, permitindo uma avaliação minuciosa dos componentes individuais de um programa, como por exemplo métodos e funções. Esta prática visa isolar cada unidade de forma a verificar se esta funciona como é pretendido e se produz o resultado esperado [75].

Os casos de teste foram elaborados seguindo uma abordagem *black box* e com o propósito de abranger todos os métodos desenvolvidos nos *chaincodes*. A Tabela 6 apresenta um exemplo de um caso de teste.

Tabela 6 - Exemplo de teste unitário

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_1	RegisterComponent	Registro válido de componente	capacity=10, id="aPWd23sF", location="40.183973, -8.413165", manufacturer="SunPower", reference="SP123456", owner="vpp_1", componentType =1	nil

Cada caso de teste é composto por um identificador único, o método testado, uma breve descrição do teste, os dados de entrada utilizados no método, o resultado previsto e o resultado efetivo durante o teste. Para facilitar a análise, a coluna de resultados é codificada com cores. Os resultados de testes bem-sucedidos são assinalados a verde, enquanto os testes falhados são destacados a vermelho. No Apêndice B – Testes unitários são apresentadas duas tabelas com todos os casos de teste.

O processo de testagem desenrolou-se ao longo de várias iterações. Após a iteração inicial, tornou-se evidente que a maioria dos testes que não obtiveram sucesso estavam associados à validação dos parâmetros de entrada. Ao identificar esses problemas, foram implementadas as respectivas correções. Procedeu-se então a uma segunda ronda de testes que resultou na execução bem-sucedida de todos os testes.

Estes resultados ilustram como a análise dos testes unitários, pode conduzir a um sistema mais robusto e confiável. As correções efetuadas não só resolveram os problemas imediatos como contribuíram para elevar a qualidade geral do protótipo.

7.3 Testes de integração

Os testes de integração verificam como diferentes partes de um sistema funcionam em conjunto. Estes confirmam se as interações entre os diferentes componentes estão corretas e se os dados são transmitidos de maneira adequada.

Os testes foram concebidos para garantir que a *blockchain* e a REST API funcionam em conjunto de forma eficaz. Esta integração é fundamental para garantir a fluidez das transações de dados e da comunicação entre o cliente e a *blockchain*. Os casos de teste foram aplicados diretamente aos endpoints da REST API para garantir a sua forte integração com a tecnologia *blockchain* subjacente. A Tabela 7 ilustra um exemplo de um caso de teste de integração.

Tabela 7 - Exemplo de teste de integração

ID	Descrição	Input	Ações	Resultado esperado
ITC_1	Agendar manutenção com dados válidos	componente="eWRhpRV", type="1", description="description", startDate="2023-01-01T15:04", endDate="2023-01-01T17:04"	Enviar pedido POST para o endpoint '/maintenances' com os dados válidos no corpo do pedido.	HTTP 201 Created Mensagem de sucesso a indicar que a manutenção foi registada

Os casos de teste de integração foram desenvolvidos de forma semelhante aos testes unitários, mas apenas com a adição das ações necessárias para a realização do mesmo. No Apêndice C – Testes de Integração é apresentada uma tabela com todos os casos de teste deste tipo.

Durante o processo de testagem, foram identificados *endpoints* por implementar, para funcionalidades críticas do sistema, como o registo de dados de sensores e contadores inteligentes. Além disso, os testes também revelaram casos em que determinados parâmetros não estavam a ser adequadamente validados. Os problemas encontrados foram corrigidos fortalecendo assim o sistema contra erros.

7.4 Testes de sistema

Os testes de sistema desempenham um papel crucial em que os componentes integrados são avaliados para garantir que, coletivamente, produzem os resultados pretendidos. Estes testes têm o propósito de assegurar que o sistema vai ao encontro dos requisitos pré-definidos e que este é adequado para ser entregue aos utilizadores [75].

Inicialmente, foram desenvolvidos casos de teste de forma a replicar cenários do mundo real simulando as possíveis interações dos utilizadores e avaliando as respostas do sistema. Estes testes tentaram abranger alguns aspetos de validação dos dados e o comportamento da interface do utilizador. A Tabela 8 ilustra um exemplo de um caso de teste relacionado com os requisitos funcionais.

Tabela 8 - Exemplo de teste de sistema

ID	Descrição	Ações	Resultado esperado
STC_1	Apresentação de lista de alertas	1. Navegar até secção de alertas	Lista de alertas

Os resultados destes casos de teste revelaram algumas imperfeições que foram corrigidas permitindo um aperfeiçoamento do protótipo. No Apêndice D – Testes de Sistema é apresentada uma tabela com todos os casos de teste deste tipo.

Para além disso, com o intuito de analisar o desempenho do sistema foram efetuados dois testes de desempenho. Inicialmente, foi realizado um teste de carga, como descrito abaixo, para avaliar a resposta do protótipo em condições de utilização previstas. Posteriormente, foi executado um teste de stress para compreender o comportamento do protótipo sob uma carga

elevada, ultrapassando as expectativas normais. Para a execução destes testes foi utilizada a ferramenta *open source* k6 [76], concebida para avaliar o desempenho e a escalabilidade de aplicações Web e APIs.

A primeira etapa consistiu em realizar um teste de carga para avaliar o desempenho da API do protótipo quando submetida a uma carga de utilizadores simulada. O objetivo era verificar se o protótipo conseguia suportar o acesso simultâneo de até 5 utilizadores sem exceder o tempo de resposta de 3 segundos para as principais operações. Para este teste, foi desenvolvido um script k6 para simular o comportamento dos utilizadores que acedem a vários *endpoints* da API do protótipo. O script foi concebido para enviar pedidos em paralelo para os *endpoints*. O teste foi configurado para aumentar progressivamente até 5 utilizadores e teve uma duração total de 7 minutos. A Fig. 30 apresenta os resultados retornados.

```

scenarios: (100.00%) 1 scenario, 5 max VUs, 7m30s max duration (incl. graceful stop):
  * default: Up to 5 looping VUs for 7m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

✓ status is 200
✓ response time is less than 3s

checks.....: 100.00% ✓ 2576   ✗ 0
data_received.....: 5.1 MB  12 kB/s
data_sent.....: 120 kB  282 B/s
http_req_blocked.....: avg=7.1µs   min=0s     med=0s     max=3.99ms  p(90)=0s   p(95)=0s
http_req_connecting.....: avg=1.7µs   min=0s     med=0s     max=998.9µs p(90)=0s   p(95)=0s
http_req_duration.....: avg=398.17ms min=193.76ms med=365ms  max=948.89ms p(90)=625.88ms p(95)=687.1ms
  { expected_response:true }...: avg=398.17ms min=193.76ms med=365ms  max=948.89ms p(90)=625.88ms p(95)=687.1ms
http_req_failed.....: 0.00% ✓ 0   ✗ 1288
http_req_receiving.....: avg=134.44µs min=0s     med=0s     max=1.51ms  p(90)=715.51µs p(95)=971µs
http_req_sending.....: avg=8.57µs   min=0s     med=0s     max=541.69µs p(90)=0s   p(95)=82.45µs
http_req_tls_handshaking.....: avg=0s       min=0s     med=0s     max=0s      p(90)=0s   p(95)=0s
http_req_waiting.....: avg=398.03ms min=193.76ms med=364.88ms max=948.77ms p(90)=625.84ms p(95)=686.88ms
http_reqs.....: 1288  3.024581/s
iteration_duration.....: avg=5.74s   min=2.6s   med=5.75s  max=8.8s   p(90)=7.25s p(95)=7.54s
iterations.....: 322  0.756145/s
vus.....: 1   min=1   max=5
vus_max.....: 5   min=5   max=5

```

Fig. 30 – Resultados de teste de carga, Fonte: autor

O cenário simulou eficazmente o acesso simultâneo de utilizadores, gerando 1288 pedidos HTTP a uma taxa média de aproximadamente 3.02 pedidos por segundo. O tempo de resposta para as principais operações permaneceu dentro do aceitável, com um tempo médio de resposta de aproximadamente 389.17 milissegundos, um mínimo de 193.79 milissegundos e um máximo de 948.89 milissegundos. Estes valores confirmam que a que a API cumpre o requisito de responder dentro de 3 segundos para as operações principais.

Por fim, foi realizado um teste de stress com o objetivo de avaliar o desempenho da API do protótipo quando sujeita a uma carga superior ao normal, simulando condições que ultrapassam a sua capacidade prevista. Especificamente o teste procurou identificar como a API do protótipo lida com o aumento de acesso simultâneo dos utilizadores e se consegue manter um nível aceitável de resposta e estabilidade. O teste foi configurado para simular um aumento gradual até 20 utilizadores durante um minuto e mantendo-se com 20 utilizadores durante 5 minutos. A Fig. 31 apresenta os resultados retornados.

```

scenarios: (100.00%) 1 scenario, 20 max VUs, 6m30s max duration (incl. graceful stop):
  * default: Up to 20 looping VUs for 6m0s over 2 stages (gracefulRampDown: 30s, gracefulStop: 30s)

data_received.....: 7.3 MB 20 kB/s
data_sent.....: 172 kB 476 B/s
http_req_blocked.....: avg=9.46µs min=0s med=0s max=1.99ms p(90)=0s p(95)=0s
http_req_connecting.....: avg=6.7µs min=0s med=0s max=1.1ms p(90)=0s p(95)=0s
http_req_duration.....: avg=3.07s min=231.95ms med=3.26s max=5.12s p(90)=3.61s p(95)=3.72s
  { expected_response:true }...: avg=3.07s min=231.95ms med=3.26s max=5.12s p(90)=3.61s p(95)=3.72s
http_req_failed.....: 0.00% ✓ 0 X 1846
http_req_receiving.....: avg=128.32µs min=0s med=0s max=1.53ms p(90)=698.75µs p(95)=990.15µs
http_req_sending.....: avg=6.97µs min=0s med=0s max=576.09µs p(90)=0s p(95)=0s
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
http_req_waiting.....: avg=3.07s min=231.95ms med=3.26s max=5.12s p(90)=3.61s p(95)=3.72s
http_reqs.....: 1846 5.10848/s
iteration_duration.....: avg=3.58s min=745.59ms med=3.76s max=5.62s p(90)=4.12s p(95)=4.23s
iterations.....: 1846 5.10848/s

```

Fig. 31 - Resultados de teste de stress, Fonte: autor

O tempo médio de um pedido HTTP durante este teste foi de 3.07 segundos, sendo o tempo de resposta mínimo observado de 231.95 milissegundos e o tempo de resposta máximo de 5.12 segundos. Embora os tempos de resposta pareçam elevados em comparação com o cenário de utilização normal, estes mantêm-se dentro dos limites aceitáveis. Isto sugere que o protótipo é capaz de lidar com um aumento da carga sem uma degradação grave do seu desempenho. Os resultados deste teste indicam a resiliência do protótipo em condições exigentes.

Capítulo 8 Conclusão

Este documento descreve o trabalho realizado no âmbito da dissertação do Mestrado em Engenharia Informática com especialização em Engenharia de Software. O trabalho insere-se no contexto do projeto T4IVPP, liderado pela entidade VOLTALIA, que visa a investigação e o desenvolvimento de protótipos de soluções que viabilizem a operação de uma Central Elétrica Virtual.

No decorrer desta tese, foi realizada uma exploração dos conceitos fundamentais da *blockchain* e das VPPs. Com base no estudo realizado, foi conduzida uma revisão sistemática da literatura com o objetivo de analisar a utilização da tecnologia *blockchain* em VPPs.

A revisão sistemática revelou que a tecnologia *blockchain* tem o potencial para melhorar a eficiência e a transparência das operações nas VPPs. A literatura demonstrou que existe interesse na capacidade da tecnologia *blockchain* para facilitar as trocas de energia *peer-to-peer*, otimizar o despacho de energia e estabelecer confiança entre os produtores e consumidores. Porém, esta análise também permitiu confirmar a necessidade de ter um registo confiável da utilização dos componentes técnicos da VPP, sendo estes críticos para a eficiência na operação.

Após a revisão da literatura, foi iniciado o processo de desenvolvimento de um protótipo *blockchain* baseado na plataforma Hyperledger Fabric. Este protótipo foi planeado para servir de apoio à gestão da operação e da manutenção de VPPs. O desenvolvimento do protótipo envolveu três tarefas principais:

1. Implementação da lógica de negócio em *chaincodes*.
2. Desenvolvimento de uma REST API para permitir a interação com a rede *blockchain* através da submissão de transações.
3. Desenvolvimento de uma interface gráfica.

O protótipo cumpriu com os objetivos essenciais planeados para esta dissertação. Permite a gestão dos componentes das VPPs, a gestão das manutenções, a análise de dados dos contadores inteligentes que permitem uma visão sobre os fluxos energéticos e a análise de dados dos sensores dos componentes que permitem acompanhar o seu estado de funcionamento. Todas estas capacidades são sustentadas na tecnologia *blockchain* que garante a integridade e a transparência dos dados, proporcionando uma maior confiança entre as organizações.

Por outro lado, o protótipo, embora promissor, apresenta limitações. Uma limitação do protótipo é o facto de não ter sido possível apresentá-lo à empresa envolvida no projeto. Isto impediu que fosse avaliado por especialistas do setor com experiência e conhecimentos valiosos para avaliar as suas capacidades e funcionalidades. A falta de feedback limita a capacidade de demonstrar o seu verdadeiro valor no contexto de uma aplicação no mundo real. Outra limitação, está relacionada com a ausência de dados reais para testar o protótipo. Esta ausência torna o funcionamento do protótipo baseado em dados simulados, fazendo com que este possa não representar exatamente a operação real de uma VPP. No entanto, ambas as restrições estão associadas ao adiamento da data de início formal do projeto, existindo vantagens para o DEI em ter já este trabalho adiantado, podendo ser posteriormente testado com os especialistas e com utilização de dados reais das VPPs a construir.

Por fim, após a conclusão deste trabalho, é observado que ainda existe um caminho a percorrer para melhorar a viabilidade do projeto. Embora o protótipo funcione atualmente a um nível

básico, ainda há espaço para melhorias e para a expansão das suas capacidades. O futuro do seu desenvolvimento deve passar pelas seguintes tarefas:

1. **Avaliação por especialistas na operação da VPP, incluindo fornecedores:** Esta tarefa é fundamental para receber feedback de pessoas experientes na área, que podem fornecer informação valiosa sobre o contexto de operação das VPP e sobre os dados importantes para a gestão da sua operação e manutenção.
2. **Melhorar estrutura de dados:** Com base no feedback e nos conhecimentos obtidos na tarefa anterior, existe a necessidade de melhorar as estruturas de dados para que estes fiquem mais alinhados com os requisitos do mundo real.
3. **Integração com aparelhos IoT:** Realizar integração com os aparelhos IoT (ex.: contadores inteligentes e sensores), para monitorização e controlo em tempo real.
4. **Definir um algoritmo de otimização da manutenção:** A investigação de um algoritmo deste tipo pode ser uma tarefa interessante. O principal objetivo é maximizar a eficiência das operações da VPP através do agendamento das manutenções para o momento ideal. Esta abordagem pretende assegurar uma utilização ótima dos recursos energéticos e contribuir para a eficiência geral da VPP. A *blockchain* apresenta vantagens que podem ser úteis para o desenvolvimento de um algoritmo deste tipo que necessita de coordenação entre duas organizações.

Em suma, este trabalho apresenta uma exploração da relação entre *blockchain* e centrais elétricas virtuais, desde a revisão dos conceitos fundamentais até ao desenvolvimento de uma solução baseada em *blockchain*. Espera-se que seja uma contribuição para o avanço do panorama energético com *blockchain*, que se encontra atualmente em evolução.

Referências

- [1] “Electricity - Fuels & Technologies - IEA.” <https://www.iea.org/fuels-and-technologies/electricity> (accessed Jan. 14, 2023).
- [2] “EUR-Lex - 52019DC0640 - EN - EUR-Lex.” <https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1576150542719&uri=COM%3A2019%3A640%3AFIN> (accessed Jan. 14, 2023).
- [3] “Bloco 3 – Portugal Energia.” <https://www.portugalenergia.pt/setor-energetico/bloco-3/> (accessed Jan. 15, 2023).
- [4] F. H. Aghdam, M. S. Javadi, and J. P. S. Catalão, “Optimal stochastic operation of technical virtual power plants in reconfigurable distribution networks considering contingencies,” *International Journal of Electrical Power & Energy Systems*, vol. 147, p. 108799, May 2023, doi: 10.1016/J.IJEPES.2022.108799.
- [5] S. Haber and W. S. Stornetta, “How To Time-Stamp a Digital Document I,” *International Association for Cryptologic Research*, vol. 3, pp. 99–111, 1991.
- [6] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008, Accessed: Dec. 27, 2022. [Online]. Available: www.bitcoin.org
- [7] Z. Qu, Z. Zhang, and M. Zheng, “A quantum blockchain-enabled framework for secure private electronic medical records in Internet of Medical Things,” *Inf Sci (N Y)*, vol. 612, pp. 942–958, Oct. 2022, doi: 10.1016/J.INS.2022.09.028.
- [8] “Intro to Ethereum | ethereum.org.” <https://ethereum.org/en/developers/docs/intro-to-ethereum/#what-is-a-blockchain> (accessed Jan. 04, 2023).
- [9] H. Wang, H. Qin, M. Zhao, X. Wei, H. Shen, and W. Susilo, “Blockchain-based fair payment smart contract for public cloud storage auditing,” *Inf Sci (N Y)*, vol. 519, pp. 348–362, May 2020, doi: 10.1016/J.INS.2020.01.051.
- [10] A. Savelyev, “Copyright in the blockchain era: Promises and challenges,” *Computer Law and Security Review*, vol. 34, no. 3, pp. 550–561, Jun. 2018, doi: 10.1016/J.CLSR.2017.11.008.
- [11] W. Stallings *et al.*, *CRYPTOGRAPHY AND NETWORK SECURITY PRINCIPLES AND PRACTICE SEVENTH EDITION GLOBAL EDITION*. 2017. Accessed: Jan. 04, 2023. [Online]. Available: www.pearsonglobaleditions.com
- [12] H. Zhu, Y. Guo, and L. Zhang, “An improved convolution Merkle tree-based blockchain electronic medical record secure storage scheme,” *Journal of Information Security and Applications*, vol. 61, Sep. 2021, doi: 10.1016/J.JISA.2021.102952.
- [13] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” in *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, Institute of Electrical and Electronics Engineers Inc., Sep. 2017, pp. 557–564. doi: 10.1109/BigDataCongress.2017.85.

- [14] Y. C. Liang, "Blockchain for dynamic spectrum management," in *Signals and Communication Technology*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 121–146. doi: 10.1007/978-981-15-0776-2_5.
- [15] A. S. Rajasekaran, M. Azees, and F. Al-Turjman, "A comprehensive survey on blockchain technology," *Sustainable Energy Technologies and Assessments*, vol. 52, Aug. 2022, doi: 10.1016/J.SETA.2022.102039.
- [16] S. M. H. Bamakan, A. Motavali, and A. Babaei Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Syst Appl*, vol. 154, Sep. 2020, doi: 10.1016/J.ESWA.2020.113385.
- [17] H. Guo and X. Yu, "A survey on blockchain technology and its security," *Blockchain: Research and Applications*, vol. 3, no. 2, Jun. 2022, doi: 10.1016/J.BCRA.2022.100067.
- [18] S. Aggarwal, R. Chaudhary, G. S. Aujla, N. Kumar, K. K. R. Choo, and A. Y. Zomaya, "Blockchain for smart communities: Applications, challenges and opportunities," *Journal of Network and Computer Applications*, vol. 144, pp. 13–48, Oct. 2019, doi: 10.1016/J.JNCA.2019.06.018.
- [19] "Proof-of-stake (PoS) | ethereum.org." <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/#top> (accessed Dec. 28, 2022).
- [20] "Hyperledger Fabric – Hyperledger Foundation." <https://www.hyperledger.org/use/fabric> (accessed Jan. 15, 2023).
- [21] M. Castro, M. Research, and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 398–461, 2002.
- [22] "Ethereum Whitepaper | ethereum.org." <https://ethereum.org/en/whitepaper/> (accessed Dec. 27, 2022).
- [23] N. Szabo, "Smart Contracts." <https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html> (accessed Dec. 28, 2022).
- [24] "Smart contracts | ethereum.org." <https://ethereum.org/en/smart-contracts/> (accessed Dec. 28, 2022).
- [25] L. Moore and N. Sawhney, "Under the hood: The West Virginia Mobile Voting Pilot," 2019, Accessed: Dec. 27, 2022. [Online]. Available: <https://voatz.com>
- [26] "Blockchain technology in health care: A primer for surgeons | The Bulletin." <https://bulletin.facs.org/2017/12/blockchain-technology-in-health-care-a-primer-for-surgeons/> (accessed Dec. 27, 2022).
- [27] "Powerledger Energy Projects." <https://www.powerledger.io/> (accessed Dec. 27, 2022).
- [28] L. K. Fachhochschule *et al.*, "The Risks of the Blockchain A Review on Current Vulnerabilities and Attacks MAKEpatho-Machine Learning & Knowledge Extraction for Digital Pathology View project The Risks of the Blockchain A Review on Current Vulnerabilities and Attacks", doi: 10.22667/JISIS.2020.08.31.110.

- [29] D. Yang, C. Long, H. Xu, and S. Peng, "A Review on Scalability of Blockchain," *ACM International Conference Proceeding Series*, pp. 1–6, Mar. 2020, doi: 10.1145/3390566.3391665.
- [30] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything You Wanted to Know about the Blockchain: Its Promise, Components, Processes, and Problems," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, Jul. 2018, doi: 10.1109/MCE.2018.2816299.
- [31] D. Pudjianto, C. Ramsay, and G. Strbac, "Virtual power plant and system integration of distributed energy resources," *Renew. Power Gener*, no. 1, pp. 10–16, 2007, doi: 10.1049/iet-rpg:20060023.
- [32] G. Zhang, C. Jiang, and X. Wang, "Comprehensive review on structure and operation of virtual power plant in electrical system," *IET Generation, Transmission and Distribution*, vol. 13, no. 2. Institution of Engineering and Technology, pp. 145–156, Jan. 22, 2019. doi: 10.1049/iet-gtd.2018.5880.
- [33] H. M. Rouzbahani, H. Karimipour, and L. Lei, "A review on virtual power plant for energy management," *Sustainable Energy Technologies and Assessments*, vol. 47, Oct. 2021, doi: 10.1016/J.SETA.2021.101370.
- [34] R. Ma, C. Zhang, H. Zhou, G. Sun, W. Qian, and H. Zang, "Study on the Transaction Management Mode of Virtual Power Plants Based on Blockchain Technology."
- [35] Xi'an jiao tong da xue and Institute of Electrical and Electronics Engineers, "Research on Coordination Control System of Virtual Power Plant Based on Blockchain,"
- [36] B. Li, F. Yang, B. Qi, X. F. Bai, Y. Sun, and S. Chen, "Research on key technologies of P2P transaction in virtual power plant based on blockchain," *IET Smart Grid*, vol. 5, no. 4. John Wiley and Sons Inc, pp. 223–233, Aug. 01, 2022. doi: 10.1049/stg2.12064.
- [37] J. Webster and R. T. Watson, "ANALYZING THE PAST TO PREPARE FOR THE FUTURE: WRITING A LITERATURE REVIEW," 2002. [Online]. Available: <http://www.misq.org/misreview/announce.html>
- [38] X. ting Wu, J. bao Men, Y. bin Jiao, and T. Shen, "A New Blockchain Consensus Algorithm for Virtual Power Plants," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 89, pp. 1373–1386, 2022, doi: 10.1007/978-3-030-89698-0_141.
- [39] Y. J. Song, "Blockchain-based power trading process," *Journal of System and Management Sciences*, vol. 9, no. 3, pp. 78–91, 2019, doi: 10.21742/ijhit.2019.12.2.02.
- [40] Y. M. Kim, D. Jung, Y. Chang, and D. H. Choi, "Intelligent micro energy grid in 5G era: Platforms, business cases, testbeds, and next generation applications," *Electronics (Switzerland)*, vol. 8, no. 4, Apr. 2019, doi: 10.3390/electronics8040468.
- [41] A. Mnatsakanyan, H. Albeshr, A. Almarzooqi, C. Iraklis, and E. Bilbao, "Blockchain mediated virtual power plant: From concept to demonstration," *The Journal of Engineering*, vol. 2022, no. 7, pp. 732–738, Jul. 2022, doi: 10.1049/tje2.12158.
- [42] S. Mishra, C. J. Crasta, C. Bordin, and J. Mateo-Fornés, "Smart contract formation enabling energy-as-a-service in a virtual power plant," *Int J Energy Res*, vol. 46, no. 3, pp. 3272–3294, Mar. 2022, doi: 10.1002/er.7381.

- [43] T. Cioara, M. Antal, V. T. Mihailescu, C. D. Antal, I. M. Anghel, and D. Mitrea, "Blockchain-Based Decentralized Virtual Power Plants of Small Prosumers," *IEEE Access*, vol. 9, pp. 29490–29504, 2021, doi: 10.1109/ACCESS.2021.3059106.
- [44] J. Lu, S. Wu, H. Cheng, and Z. Xiang, "Smart contract for distributed energy trading in virtual power plants based on blockchain," *Comput Intell*, vol. 37, no. 3, pp. 1445–1455, Aug. 2021, doi: 10.1111/coin.12388.
- [45] M. Gough *et al.*, "Blockchain-Based Transactive Energy Framework for Connected Virtual Power Plants," *IEEE Trans Ind Appl*, vol. 58, no. 1, pp. 986–995, 2022, doi: 10.1109/TIA.2021.3131537.
- [46] Z. Wen, Y. Li, Y. Zheng, and Y. Yang, "Virtual power plant trading strategy based on block-chain to satisfy clean energy partiality," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration: Connecting the Grids Towards a Low-Carbon High-Efficiency Energy System, EI2 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 3004–3010. doi: 10.1109/EI250167.2020.9347243.
- [47] S. Seven, G. Yao, A. Soran, A. Onen, and S. M. Muyeen, "Peer-to-peer energy trading in virtual power plant based on blockchain smart contracts," *IEEE Access*, vol. 8, pp. 175713–175726, 2020, doi: 10.1109/ACCESS.2020.3026180.
- [48] X. Wang, J. Zhang, Z. Jia, A. Hu, D. Li, and X. Wu, "A new energy trading model for virtual power plants based on blockchain," in *Proceedings of 2021 IEEE International Conference on Emergency Science and Information Technology, ICESIT 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 93–99. doi: 10.1109/ICESIT53460.2021.9696896.
- [49] Q. Yang and H. Wang, "Exploring blockchain for the coordination of distributed energy resources," in *2021 55th Annual Conference on Information Sciences and Systems, CISS 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021. doi: 10.1109/CISS50987.2021.9400211.
- [50] Q. Yang, H. Wang, T. Wang, S. Zhang, X. Wu, and H. Wang, "Blockchain-based decentralized energy management platform for residential distributed energy resources in a virtual power plant," *Appl Energy*, vol. 294, Jul. 2021, doi: 10.1016/j.apenergy.2021.117026.
- [51] W. Shao, W. Xu, Z. Xu, B. Liu, and H. Zou, "A Grid Connection Mechanism of Large-scale Distributed Energy Resources based on Blockchain; A Grid Connection Mechanism of Large-scale Distributed Energy Resources based on Blockchain," 2019.
- [52] Z. Wen, Y. Zheng, and Y. Li, "Analysis of decentralized energy transactions based on smart contract," in *Proceedings of 2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence, ICIBA 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 819–824. doi: 10.1109/ICIBA50161.2020.9276990.
- [53] M. U. Hassan, M. H. Rehmani, and J. Chen, "VPT: Privacy Preserving Energy Trading and Block Mining Mechanism for Blockchain based Virtual Power Plants," Feb. 2021, [Online]. Available: <http://arxiv.org/abs/2102.01480>

- [54] D. Wang, Z. Wang, and X. Lian, "Research on Distributed Energy Consensus Mechanism Based on Blockchain in Virtual Power Plant," *Sensors*, vol. 22, no. 5, Mar. 2022, doi: 10.3390/s22051783.
- [55] Q. Xue, Z. Xue, H. Ma, Y. Sun, and X. Shi, "Design of Virtual Power Plant System Model under Master-Slave Multi-chain," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Jan. 2022. doi: 10.1088/1742-6596/2166/1/012043.
- [56] S. Yin, Q. Ai, J. Li, D. Li, and Q. Guo, "Trading mode design for a virtual power plant based on main-side consortium blockchains," *Appl Energy*, vol. 325, Nov. 2022, doi: 10.1016/j.apenergy.2022.119932.
- [57] X. Zhang, Z. Song, and A. J. Moshayedi, "Security scheduling and transaction mechanism of virtual power plants based on dual blockchains," *Journal of Cloud Computing*, vol. 11, no. 1, Dec. 2022, doi: 10.1186/s13677-021-00273-3.
- [58] D. Huang *et al.*, "Consortium blockchain-based decentralized energy trading mechanism for virtual power plant," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration: Connecting the Grids Towards a Low-Carbon High-Efficiency Energy System, EI2 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 3084–3089. doi: 10.1109/EI250167.2020.9346653.
- [59] Institute of Electrical and Electronics Engineers, "Hyperledger Fabric Smart Grid Communication Testbed on Raspberry PI ARM Architecture,"
- [60] T. Cioara, C. Pop, R. Zanc, I. Anghel, M. Antal, and I. Salomie, "Smart grid management using blockchain: Future scenarios and challenges," in *Proceedings - RoEduNet IEEE International Conference*, IEEE Computer Society, Dec. 2020. doi: 10.1109/RoEduNet51892.2020.9324874.
- [61] M. Fan, X. Tang, W. Pei, and L. Kong, "Peer-peer electricity transaction and integrated regulation of VPP based on blockchain," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration: Connecting the Grids Towards a Low-Carbon High-Efficiency Energy System, EI2 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 727–732. doi: 10.1109/EI250167.2020.9347041.
- [62] Y. J. Song and J. K. Lee, "A blockchain and internet of things based architecture design for energy transaction," *Journal of System and Management Sciences*, vol. 10, no. 2, pp. 122–140, 2020, doi: 10.33168/jsms.2020.0209.
- [63] A. Mnatsakanyan, H. Albeshr, A. al Marzooqi, and E. Bilbao, "Blockchain-integrated virtual power plant demonstration," in *2020 2nd International Conference on Smart Power and Internet Energy Systems, SPIES 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 172–175. doi: 10.1109/SPIES48661.2020.9242966.
- [64] X. Wang, P. Liu, and Z. Ji, "Trading platform for cooperation and sharing based on blockchain within multi-agent energy internet," *Global Energy Interconnection*, vol. 4, no. 4, pp. 384–393, Aug. 2021, doi: 10.1016/j.gloi.2021.09.009.
- [65] D. Li, Q. Guo, D. Bai, and W. Zhang, "Research and Implementation on the Operation and Transaction System Based on Blockchain Technology for Virtual Power Plant," in *Proceedings - 2022 International Conference on Blockchain Technology and Information Security, ICBCTIS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 165–170. doi: 10.1109/ICBCTIS55569.2022.00046.

- [66] Y. Ma, F. Tu, X. Zhou, and D. Xu, "Application of Block Chain Technology in Virtual Power Plant," in *2022 IEEE International Conference on Mechatronics and Automation, ICMA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 122–127. doi: 10.1109/ICMA54519.2022.9855926.
- [67] "Chapter 10: MoSCoW Prioritisation." <https://www.agilebusiness.org/dsdm-project-framework/moscow-prioritisation.html> (accessed Jan. 15, 2023).
- [68] "The C4 model for visualising software architecture." <https://c4model.com/> (accessed Jan. 10, 2023).
- [69] "GitHub - google/leveldb: LevelDB is a fast key-value storage library written at Google that provides an ordered mapping from string keys to string values." <https://github.com/google/leveldb> (accessed Jul. 20, 2023).
- [70] "Apache CouchDB." <https://couchdb.apache.org/> (accessed Jul. 20, 2023).
- [71] "Express - Node.js web application framework." <https://expressjs.com/> (accessed Jul. 18, 2023).
- [72] "Hyperledger Fabric SDK for Node.js | fabric-sdk-node." <https://hyperledger.github.io/fabric-sdk-node/> (accessed Jul. 18, 2023).
- [73] "fabric-network - npm." <https://www.npmjs.com/package/fabric-network> (accessed Jul. 18, 2023).
- [74] "fabric-ca-client - npm." <https://www.npmjs.com/package/fabric-ca-client> (accessed Jul. 18, 2023).
- [75] I. Sommerville *et al.*, "SOFTWARE ENGINEERING Ninth Edition," 2011.
- [76] "Load testing for engineering teams | Grafana k6." <https://k6.io/> (accessed Aug. 17, 2023).

Apêndices

Apêndice A – Casos de uso

Este apêndice apresenta dos casos de uso que servem como pontos de referência para os requisitos funcionais uma vez que oferecem uma visão detalhada das interações e funcionalidades do sistema. As tabelas descrevem o nome de cada caso de uso, os atores envolvidos, a descrição, o cenário de sucesso e alguns cenários alternativos.

Tabela A 1 - Caso de uso 4

Caso de uso 4 – Registrar leituras de eletricidade	
Ator	Sistema de recolha de dados
Descrição	O sistema de recolha de dados pretende registar as leituras dos contadores inteligentes.
Cenário de sucesso	O contador inteligente inicia o processo para registar leituras de eletricidade no intervalo de tempo agendado de 15 minutos. O contador inteligente envia as quantidades de energia consumida e enviada para a rede registadas nos últimos 15 minutos. O sistema de recolha de dados recebe os dados, agrega-os e envia-os para o servidor. As leituras são registadas na <i>blockchain</i> , garantindo imutabilidade das mesmas. O sistema de recolha de dados recebe o <i>acknowledgement</i> relativo ao sucesso do registo.
Extensões	2a – O contador não envia os dados devido a uma falha de internet O contador inteligente guarda localmente as leituras. O contador periodicamente, tenta reenviar as leituras guardadas localmente até a conexão ser restaurada.

Tabela A 2 - Caso de uso 5

Caso de uso 5 – Registrar energia armazenada	
Ator	Sistema de recolha de dados
Descrição	O sistema de recolha de dados pretende registar os dados de um componente de armazenamento de energia pertencente à VPP.
Cenário de sucesso	O sensor gera os dados com a informação sobre a energia armazenada. O sensor envia os dados para o sistema de recolha de dados. O sistema de recolha de dados recebe os dados, agrega-os e envia-os para o servidor. Os dados são registados no livro-razão garantindo a imutabilidade e transparência dos mesmos. O sistema de recolha de dados recebe o <i>acknowledgement</i> relativo ao sucesso do registo.

Extensões	3a – Os dados são inválidos O sistema regista um alerta relacionado com a invalidade dos dados apresentados pelo sensor.
------------------	--

Tabela A 3 - Caso de uso 6

Caso de uso 6 – Registar alertas de operação	
Ator	Sistema de recolha de dados
Descrição	O sistema de recolha de dados pretende registar alertas sobre condições específicas ou eventos relacionados com os componentes numa VPP.
Cenário de sucesso	Os sensores monitorizam os componentes para condições específicas ou eventos que requeiram o registo de um alerta. Quando uma condição ou evento é detetado, o sensor gera uma mensagem de alerta e especifica o tipo de alerta (performance, falha) O sistema de recolha de dados recebe os dados, agrega-os e envia-os para o servidor. Os dados são registados no livro-razão garantindo a imutabilidade e transparência dos mesmos. O sistema de recolha de dados recebe o <i>acknowledgemente</i> relativo ao sucesso do registo.

Tabela A 4 - Caso de uso 7

Caso de uso 7 – Registar energia produzida	
Ator	Sistema de recolha de dados
Descrição	O sistema de recolha de dados pretende registar os dados relativos à energia produzida por parte de um componente de produção de energia pertencente à VPP.
Cenário de sucesso	O sensor deteta a energia produzida e envia para o sistema de recolha de dados. O sistema de recolha de dados recebe os dados, agrega-os e envia-os para o servidor. Os dados são registados no livro-razão garantindo a imutabilidade e transparência dos mesmos. O sistema de recolha de dados recebe o <i>acknowledgemente</i> relativo ao sucesso do registo.
Extensões	3a – Os dados são inválidos O sistema regista um alerta relacionado com a invalidade dos dados apresentados pelo sensor.

Tabela A 5 - Caso de uso 8

Caso de uso 8 – Analisar dados da operação VPP	
Ator	Operador VPP
Descrição	Este caso de uso envolve o operador da VPP analisar os dados relevantes da operação da VPP para obter para obter uma melhor percepção do funcionamento da mesma e assim e tomar decisões informadas caso estas sejam necessárias.
Cenário de sucesso	O operador da VPP abre o <i>dashboard</i> . O sistema apresenta uma interface com informação relevante à análise por parte do operador. O operador VPP pode visualizar gráficos que ilustram o consumo e produção de energia e visualizar alertas de eventos críticos ou condições anormais na VPP. Isto inclui dados históricos, permitindo ao operador a identificação de padrões ou tendências.
Extensões	<p>2a – Não existem dados disponíveis</p> <p>A plataforma apresenta uma mensagem a indicar a indisponibilidade de dados. O operador VPP pode optar por refrescar a página ou selecionar outro cliente.</p> <p>4a- Existem alertas no dashboard</p> <p>O operador da VPP seleciona o alerta. O sistema apresenta uma interface com a informação relativa ao alerta. O operador VPP toma as medidas apropriadas, como investigar a causa ou notificar os responsáveis à resolução deste tipo de problemas.</p>

Tabela A 6 - Caso de uso 9.2

Caso de uso 9.2 – Remover componente	
Ator	Operador VPP
Descrição	O operador VPP pretende remover um componente da VPP.
Cenário de sucesso	<p>O sistema apresenta uma lista com todos os componentes registados na VPP. O utilizador seleciona o componente que pretende remover. O sistema apresenta os detalhes do componente, incluindo o tipo, a localização e outras informações relevantes. O utilizador verifica que o componente selecionado é o que pretende remover e seleciona a opção de remover. O sistema solicita a confirmação da remoção do componente. O utilizador confirma que pretende remover. O sistema remove o componente, atualiza a lista de componentes. O sistema apresenta uma mensagem ao utilizador a confirmar a remoção do componente.</p>

Tabela A 7 - Caso de uso 9.3

Caso de uso 9.3 – Ver componentes da VPP	
Ator	Operador VPP, Técnico de manutenção
Descrição	O operador VPP ou o técnico de manutenção pretendem ver os componentes registados numa VPP.
Cenário de sucesso	<p>O sistema apresenta a lista de componentes registados numa VPP.</p> <p>O utilizador selecciona um componente da lista para ver informação detalhada.</p> <p>O sistema apresenta os detalhes do componente, fornecendo informação sobre o componente e apresentando dados relacionados como manutenções e alertas.</p> <p>O utilizador observa os detalhes do componente e analisa os dados relativos ao mesmo.</p>

Tabela A 8 - Caso de uso 10

Caso de uso 10 – Monitorizar transferências de energia	
Ator	Operador VPP
Cenário de sucesso	<p>O operador VPP inicia o processo de visualizar transferências de energia.</p> <p>O sistema apresenta uma interface ao operador VPP exibindo as leituras registadas pelos contadores inteligentes.</p> <p>O operador VPP selecciona a o proprietário de um ou mais contadores inteligente desejado para visualizar as leituras de energia.</p> <p>O sistema devolve as leituras para o proprietário seleccionado.</p> <p>As leituras de energia são apresentadas ao operador VPP, fornecendo uma visualização da energia consumida e fornecida à VPP.</p>
Extensões	<p>4a – Não existem leituras disponíveis</p> <p>A plataforma apresenta uma mensagem a indicar que não existem leituras do contador para visualizar.</p> <p>O operador VPP pode optar por refrescar a página ou seleccionar outro cliente.</p> <p>4b- Falha ao devolver as leituras do contador</p> <p>O sistema gera uma mensagem de erro a indicar que existiu uma falha ao tentar retirar os dados da blockchain.</p> <p>O operador VPP pode tentar refrescar a página ou contactar o administrador para algum tipo de assistência.</p>

Tabela A 9 - Caso de uso 11.1

Caso de uso 11.1 – Ver manutenções agendadas	
Ator	Operador VPP, Técnico de manutenção
Descrição	O operador VPP e o técnico de manutenção querem aceder e ver as manutenções agendadas numa VPP.
Cenário de sucesso	O utilizador acede à página de manutenções. O sistema apresenta a lista de manutenções agendadas no sistema. O utilizador seleciona uma manutenção específica da lista para ver os detalhes. O sistema apresenta a informação detalhada da manutenção incluindo o tipo de manutenção, a data, a duração e o componente associado.
Extensões	2a – Não existem manutenções agendadas O sistema apresenta uma mensagem indicando que não existem manutenções agendadas.

Tabela A 10 - Caso de uso 11.2

Caso de uso 11.2 – Acompanhar progresso da manutenção	
Ator	Operador VPP, Técnico de manutenção
Descrição	O operador VPP ou o técnico de manutenção querem aceder e ver o progresso das manutenções.
Cenário de sucesso	O sistema apresenta a lista de manutenções a decorrer. O utilizador seleciona uma manutenção. O sistema apresenta as informações detalhadas da manutenção juntamente com notas associadas e atualizações. O utilizador pode ver os updates relativos ao progresso na manutenção.

Tabela A 11 - Caso de uso 11.3

Caso de uso 11.3 – Ver histórico de manutenções	
Ator	Operador VPP, Técnico de manutenção
Descrição	O operador VPP e o técnico de manutenção querem ver o histórico de manutenções numa VPP, incluindo detalhes associados.
Cenário de sucesso	O utilizador acede à página de manutenções. O sistema fornece a lista de manutenções terminadas. O utilizador observa a lista e seleciona uma manutenção desejada. O sistema apresenta informação detalhada sobre a manutenção.

Tabela A 12 - Caso de uso 12.1

Caso de uso 12.1 – Agendar manutenção	
Ator	Técnico de manutenção
Descrição	Este caso de uso envolve o técnico de manutenção querer agendar uma manutenção para um componente da VPP.
Cenário de sucesso	O sistema apresenta a página para agendar uma manutenção. O utilizador insere os dados relacionados com o agendamento da manutenção: id do componente, data da manutenção, duração, descrição e anexação de um ficheiro de manutenção. O utilizador submete o formulário. O sistema efetua a validação dos dados submetidos. O sistema faz o registo da manutenção no livro-razão garantindo a imutabilidade e transparência da mesma.
Extensões	4a – Os dados são inválidos O utilizador corrige os dados que foram indicados como inválidos e submete o formulário. O sistema efetua a validação dos dados submetidos. O sistema faz o registo da manutenção no livro-razão garantindo a imutabilidade e transparência da mesma.

Tabela A 13 - Caso de uso 12.2

Caso de uso 12.2 – Cancelar manutenção	
Ator	Técnico de manutenção
Descrição	O técnico de manutenção pretende cancelar uma manutenção agendada numa VPP.
Cenário de sucesso	<p>O sistema fornece a lista de manutenções agendadas.</p> <p>O utilizador seleciona a manutenção que pretende cancelar.</p> <p>O sistema apresenta os detalhes de uma manutenção.</p> <p>O utilizador verifica a informação para assegurar que corresponde à manutenção que pretende cancelar e seleciona o botão para cancelar.</p> <p>O sistema solicita ao utilizado a confirmação do cancelamento.</p> <p>O utilizador confirma o cancelamento da manutenção.</p> <p>O sistema atualizada o estado da manutenção para indicar o seu cancelamento e apresenta uma mensagem para confirmar o sucesso da tarefa.</p>

Tabela A 14 - Caso de uso 12.3

Caso de uso 12.3 – Atualizar estado da manutenção	
Ator	Técnico da manutenção
Descrição	O técnico de manutenção quer fazer a atualização do estado de uma manutenção numa VPP.
Cenário de sucesso	<p>O sistema apresenta a lista de manutenções.</p> <p>O utilizador seleciona a manutenção que pretende atualizar o estado.</p> <p>O sistema apresenta os detalhes da manutenção.</p> <p>O utilizador verifica a informação para assegurar que corresponde à manutenção que pretende atualizar e seleciona a opção pretendida.</p> <p>O sistema atualiza o estado da manutenção.</p>

Apêndice B – Testes unitários

Este apêndice apresenta os testes unitários realizados para avaliar os métodos implementados nos *chaincodes* do protótipo.

Tabela B 1 - Testes unitários aos métodos do *chaincode* "energy"

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_1	GetEnergyDataAsset	ID de leitura registrada	id="1"	asset, nil
UTC_2	GetEnergyDataAsset	ID de leitura não registrada	id="fake_id"	nil, erro
UTC_3	RegisterEnergyData	Registro válido	id="new_id", consumption=30.82, production=10.33, maxDemand=13, importedEnergy=24.52, exportedEnergy=7.81, owner="1", timestamp="2022-12-30T20:30"	nil
UTC_4	RegisterEnergyData	Registro inválido: id já existente	id="1", consumption=30.82, production=10.33, maxDemand=13, importedEnergy=24.52, exportedEnergy=7.81, owner="1", timestamp="2022-12-30T20:30"	erro
UTC_5	RegisterEnergyData	Registro inválido: formato de data inválido	id="1", consumption=30.82, production=10.33, maxDemand=13, importedEnergy=24.52, exportedEnergy=7.81, owner="1", timestamp="2022-1230T20:30"	erro
UTC_6	GetEnergyDataByYear	Retirar dados de um ano com registros	ownerID="1", year="2023"	Lista de assets, nil
UTC_7	GetEnergyDataByYear	Retirar dados de um ano sem registros	ownerID="1", year="2024"	Lista vazia, nil
UTC_8	GetEnergyDataByMonth	Retirar dados de um mês com registros	ownerID="1", year="2023", month="12"	Lista de assets, nil
UTC_9	GetEnergyDataByMonth	Retirar dados de um mês sem registros	ownerID="1", year="2023", month="11"	Lista vazia, nil

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_10	GetEnergyDataByDay	Retirar dados de um dia com registros	ownerID="1", year="2023", month="12", day="31"	Lista de assets, nil
UTC_11	GetEnergyDataByDay	Retirar dados de um dia sem registros	ownerID="1", year="2023", month="11", day="31"	Lista vazia, nil
UTC_12	GetOverallEnergyDataByYear	Retirar dados de um ano com registros	year="2023"	Lista de assets, nil
UTC_13	GetOverallEnergyDataByYear	Retirar dados de um ano sem registros	year="2024"	Lista vazia, nil
UTC_14	GetOverallEnergyDataByMonth	Retirar dados de um mês com registros	year="2023", month="12"	Lista de assets, nil
UTC_15	GetOverallEnergyDataByMonth	Retirar dados de um mês sem registros	year="2023", month="11"	Lista vazia, nil
UTC_16	GetOverallEnergyDataByDay	Retirar dados de um dia com registros	year="2023", month="12", day="31"	Lista de assets, nil
UTC_17	GetOverallEnergyDataByDay	Retirar dados de um dia sem registros	year="2023", month="11", day="31"	Lista vazia, nil

Tabela B 2 - Teste unitários aos métodos do *chaincode* "components"

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_18	RegisterComponent	Registro válido	capacity=10, id="aPWd23sF", location="40.183973, -8.413165", manufacturer="SunPower", reference="SP123456", owner="vpp_1", componentType =1	nil
UTC_19	RegisterComponent	Registro Inválido: Capacidade inválida	capacity=-10, id="aPWd23sF", location="40.183973, -8.413165", manufacturer="SunPower", reference="SP123456", owner="vpp_1", componentType =1	erro

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_20	RegisterComponent	Registro inválido: id já existente	capacity=10, id="eWRhpRV", location="40.183973, -8.413165", manufacturer="SunPower", reference="SP123456", owner="vpp_1", componentType =1	erro
UTC_21	RegisterComponent	Registro inválido: formato da localização inválido	capacity=10, id="aPWd23sF", location="40.183973 -8.413165", manufacturer="SunPower", reference="SP123456", owner="vpp_1", componentType =1	erro
UTC_22	RegisterComponent	Registro inválido: Tipo de componente inválido	capacity=10, id="aPWd23sF", location="40.183973, -8.413165", manufacturer="SunPower", reference="SP123456", owner="vpp_1", componentType =6	erro
UTC_23	GetComponent	ID de componente registrado	id="eWRhpRV"	asset
UTC_24	GetComponent	ID de componente não registrado	id="aPWd23sF"	erro
UTC_25	SetInactiveComponent	Tornar inativo componente registrado	id="eWRhpRV3"	nil
UTC_26	SetInactiveComponent	Tornar inativo componente não registrado	id="aPWd23sF"	erro
UTC_27	GetAllComponents	Retirar lista de componentes quando se encontram um ou mais componentes registrados		Lista de assets, nil
UTC_28	GetAllComponents	Retirar lista de componentes quando não se encontram componentes registrados		Lista vazia, nil

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_29	ScheduleMaintenance	Registo válido	id="2AZhDzs", component="eWRhpRV", description="Replace faulty solar panel in array 3", maintenanceType=1, startDate="2023-04-20T15:00", endDate="2023-05-20T15:00"	nil
UTC_30	ScheduleMaintenance	Registo inválido: id já existente	id="aPRhpXV", component="eWRhpRV", description="Replace faulty solar panel in array 3", maintenanceType=1, startDate="2023-04-20T15:00", endDate="2023-05-20T15:00"	erro
UTC_31	ScheduleMaintenance	Registo inválido: Componente não existente	id="2AZhDzs", component="aPWd23sF", description="Replace faulty solar panel in array 3", maintenanceType=1, startDate="2023-04-20T15:00", endDate="2023-05-20T15:00"	erro
UTC_32	ScheduleMaintenance	Registo inválido: Tipo de manutenção inválido	id="2AZhDzs", component="aPWd23sF", description="Replace faulty solar panel in array 3", maintenanceType=3, startDate="2023-04-20T15:00", endDate="2023-05-20T15:00"	erro
UTC_33	ScheduleMaintenance	Registo inválido: Formato de data errado	id="2AZhDzs", component="aPWd23sF", description="Replace faulty solar panel in array 3", maintenanceType=3, startDate="202304-20T15:00", endDate="2023-05-20T15:00"	erro
UTC_34	ScheduleMaintenance	Registo inválido: Data de fim inferior à data de início	id="2AZhDzs", component="aPWd23sF", description="Replace faulty solar panel in array 3", maintenanceType=3, startDate="2023-05-20T15:00", endDate="2023-04-20T15:00"	erro
UTC_35	GetMaintenance	ID de manutenção registada	id="aPRhpXV"	asset, nil

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_36	GetMaintenance	ID de manutenção não registada	id="test_id"	nil, nil
UTC_37	UpdateMaintenance	Atualizar o estado de uma manutenção que não se encontra registada	id="test_id", description="Alguma descrição", cancelMaintenance=False	nil
UTC_38	UpdateMaintenance	Passar do estado "scheduled" para o estado "ongoing"	id="id", description="Alguma descrição", cancelMaintenance=False	nil
UTC_39	UpdateMaintenance	Passar do estado "ongoing" para o estado "finished"	id="id", description="Alguma descrição", cancelMaintenance=False	nil
UTC_40	UpdateMaintenance	Cancelar uma manutenção que se encontra agendada	id="id", description="Alguma descrição", cancelMaintenance=True	nil
UTC_41	UpdateMaintenance	Cancelar uma manutenção que se encontra a decorrer	id="id", description="Alguma descrição", cancelMaintenance=True	erro
UTC_42	GetComponentMaintenance	Retirar a lista de manutenções de um componente	component="eWRhpRV"	Lista de assets, nil
UTC_43	GetComponentMaintenance	Retirar as manutenções de um componente que não se encontra registado	component="test_id"	nil, erro
UTC_44	GetComponentMaintenance	Retirar a lista de manutenções de um componente que não tem manutenções associadas	component="hwX6aOr7"	Lista vazia, nil
UTC_45	GetAllMaintenances	Retirar lista de todas as manutenções quando se encontram manutenções registadas		Lista de assets, nil

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_46	GetAllMaintenances	Retirar a lista de manutenções quando não se encontra nenhuma manutenção registada		Lista vazia, nil
UTC_47	GetMaintenanceHistory	Retirar o histórico de uma manutenção que não se encontra registada	id="test_id"	nil, erro
UTC_48	GetMaintenanceHistory	Retirar o histórico de uma manutenção que se encontra registada e no estado "scheduled"	id="eWRhpRV"	histórico, nil
UTC_49	GetMaintenanceHistory	Retirar o histórico de uma manutenção que se encontra registada e no estado "ongoing"	id="eWRhpRV"	histórico, nil
UTC_50	GetMaintenanceHistory	Retirar o histórico de uma manutenção que se encontra registada e no estado "finished"	id="eWRhpRV"	histórico, nil
UTC_51	GetMaintenanceHistory	Retirar o histórico de uma manutenção que se encontra registada e no estado "canceled"	id="eWRhpRV"	histórico, nil
UTC_52	RegisterAlert	Registo válido	id="asD43qX", componentID="eWRhpRV", alert_type=2, description="Alguma descrição", timestamp="2023-04-20T19:00"	nil
UTC_53	RegisterAlert	Registrar alerta com um id que já existe	id="fi4t7A2Z", componentID="ifd", alert_type="2", description="Alguma descrição", timestamp="2023-04-20T19:00"	erro

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_54	RegisterAlert	Associar alerta a um componente que não se encontra registado	id="asD43qX", componentID="test_id", alert_type=2, description="Alguma descrição", timestamp="2023-04-20T19:00"	erro
UTC_55	RegisterAlert	Tipo de alerta inválido	id="asD43qX", componentID="eWRhpRV", alert_type=3, description="Alguma descrição", timestamp="2023-04-20T19:00"	erro
UTC_56	RegisterAlert	Formato de data inválido	id="asD43qX", componentID="eWRhpRV", alert_type="2", description="Alguma descrição", timestamp="2023-0-20T19:00"	erro
UTC_57	GetAlert	Retirar alerta que se encontra registada	id="fi4t7A2Z"	asset, nil
UTC_58	GetAlert	Retirar alerta que não se encontra registada	id="fake_id"	nil, erro
UTC_59	GetAllAlerts	Retirar lista de todas as alertas quando se encontram alertas registadas		lista com assets, nil
UTC_60	GetAllAlerts	Retirar lista de todas as alertas quando não se encontram alertas registadas		lista vazia, nil
UTC_61	UpdateAlert	Atualizar alerta que não se encontra registado	id="fake_id"	nil, erro
UTC_62	UpdateAlert	Atualizar alerta que se encontra no estado "false" para "true"	id="fi4t7A2Z"	nil, nil
UTC_63	GetComponentAlerts	Retirar alertas de um componente que não se encontra registado	id="fake_id"	nil, erro

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_64	GetComponentAlerts	Retirar a lista de alertas de um componente que não tem alertas associadas	id="hwX6aOr7"	lista vazia, nil
UTC_65	GetComponentAlerts	Retirar a lista de alertas de um componente que tem alertas associadas	id="eWRhpRV"	lista de assets, nil
UTC_66	RegisterComponentSensorMetering	Registro válido	id="new_id", componentID="hwX6aOr7", production="10,2" soc="100", timestamp="2023-04-20T19:00"	nil
UTC_67	RegisterComponentSensorMetering	Registro inválido: id já existente	id="3eG7pQ4H", componentID="hwX6aOr7", production="10,2" soc="100", timestamp="2023-04-20T19:00"	erro
UTC_68	RegisterComponentSensorMetering	Registro inválido: componente não registado	id="new_id", componentID="fake_id", production="10,2" soc="100", timestamp="2023-04-20T19:00"	erro
UTC_69	RegisterComponentSensorMetering	Registro inválido: valor de produção inválido	id="new_id", componentID="hwX6aOr7", production="10,2" soc="100", timestamp="2023-04-20T19:00"	erro
UTC_70	RegisterComponentSensorMetering	Registro inválido: valor de soc inválido	id="new_id", componentID="hwX6aOr7", production="10,2" soc="100", timestamp="2023-04-20T19:00"	erro
UTC_71	RegisterComponentSensorMetering	Registro inválido: formato de data inválido	id="new_id", componentID="hwX6aOr7", production="10,2" soc="100", timestamp="2023-04-20T19:00"	erro
UTC_72	GetComponentSensorDataSet	Retirar registo de sensor válido	id="3eG7pQ4H"	asset, nil
UTC_73	GetComponentSensorMetering	Retirar registo de sensor que não existe	id="fake_id"	nil, erro

ID	Função	Descrição	Dados de entrada	Resultado esperado
UTC_74	GetComponentSensorDataByYear	Retirar registos de sensores válido	componentID="eWRhpRV", year="2023"	lista de assets, nil
UTC_75	GetComponentSensorDataByYear	Retirar registo de sensores de componente não registado	componentID=" fake_id ", year="2023"	lista vazia, nil
UTC_76	GetComponentSensorDataByMonth	Retirar registos de sensores válido	componentID="eWRhpRV", year="2023", month="12"	lista de assets, nil
UTC_77	GetComponentSensorDataByMonth	Retirar registo de sensores de componente não registado	componentID=" fake_id ", year="2023", month="12"	lista vazia, nil
UTC_78	GetComponentSensorDataByDay	Retirar registos de sensores válido	componentID="eWRhpRV", year="2023", month="12", day="31"	lista de assets, nil
UTC_79	GetComponentSensorDataByDay	Retirar registo de sensores de componente não registado	componentID="fake_id", year="2023", month="12", day="31"	lista vazia, nil

Apêndice C – Testes de Integração

Este apêndice apresenta os testes de integração realizados para avaliar a integração da REST API com a rede *blockchan*.

Tabela C 1 – Testes de integração

ID	Descrição	Input	Ações	Resultado esperado
ITC_1	Agendar manutenção com dados válidos	component="eWRhpRV", type=1, description="Descrição", startDate="2023-01-01T15:04", endDate="2023-01-01T17:04"	Enviar pedido POST para o endpoint '/maintenances' com os dados válidos no corpo do pedido	HTTP 201 Created com uma mensagem de sucesso a indicar que a manutenção foi registrada
ITC_2	Agendar manutenção com dados em falta	component="eWRhpRV", type=1, description="Descrição", startDate="2023-01-01T15:04"	Enviar pedido POST para o endpoint '/maintenances' com os dados incompletos no corpo do pedido	HTTP 400 Bad Request com uma mensagem de erro a indicar os dados em falta
ITC_3	Consulta de todos os registos de manutenção		Enviar pedido GET para o endpoint '/maintenances'	HTTP 200 OK com um array que contém todos os registos de manutenção
ITC_4	Consultar o histórico de uma manutenção		Enviar pedido GET para o endpoint '/maintenances/1rfw_d9Z' com o id de uma manutenção existente	HTTP 200 OK com um array que contém o histórico das alterações realizadas no ativo
ITC_5	Consultar histórico de uma manutenção sem alterações		Enviar pedido GET para o endpoint '/maintenances/1rfw_d9Z' com o id de uma manutenção que não tenha sofrido alterações de estado	HTTP 200 OK array com apenas o registo da manutenção associado ao estado de agendada.

ID	Descrição	Input	Ações	Resultado esperado
ITC_6	Atualizar manutenção para estado "ongoing"	description="Something", cancelMaintenance=false	Enviar pedido PUT para o endpoint '/maintenances/1rfw_d9Z' com o id de uma manutenção que esteja no estado "scheduled"	HTTP 200 OK com mensagem a indicar que a manutenção foi atualizada com sucesso.
ITC_7	Atualizar manutenção para estado "finished"	description="Something", cancelMaintenance=false	Enviar pedido PUT para o endpoint '/maintenances/1rfw_d9Z' com o id de uma manutenção que esteja no estado "ongoing"	HTTP 200 OK com mensagem a indicar que a manutenção foi atualizada com sucesso.
ITC_8	Cancelar manutenção	description="Something", cancelMaintenance=true	Enviar pedido PUT para o endpoint '/maintenances/1rfw_d9Z' com o id de uma manutenção que esteja no estado "scheduled"	HTTP 200 OK com mensagem a indicar que a manutenção foi atualizada com sucesso.
ITC_9	Registrar um componente com dados válidos	capacity=12, location="40.18, -8.43", manufacturer="test_manufacturer", reference="test_ref", type="1"	Enviar pedido POST para o endpoint '/components' com os dados válidos no corpo do pedido	HTTP 201 Created com uma mensagem de sucesso a indicar que o componente foi registrada
ITC_10	Registrar um componente com dados em falta	capacity=12, location="40.18, -8.43", manufacturer="test_manufacturer", reference="test_ref"	Enviar pedido POST para o endpoint '/components' com os dados incompletos no corpo do pedido	HTTP 400 Bad Request com uma mensagem de erro a indicar os dados em falta
ITC_11	Consulta de todos os componentes registrados no livro-razão		Enviar pedido GET para o endpoint '/components'	HTTP 200 OK com um array que contém todos os componentes registrados.
ITC_12	Consultar um componente que se encontra registrado no livro razão		Enviar pedido GET para o endpoint '/components/eWRh pRV'	HTTP 200 OK com os dados do componente com o id enviado no url.

ID	Descrição	Input	Ações	Resultado esperado
ITC_13	Consultar todas as manutenções de um componente		Enviar pedido GET para o endpoint '/components/eWRhpRV/maintenances'	HTTP 200 OK com um array que contém todos as manutenções associadas ao componente com o id.
ITC_14	Registrar dados de sensor de componente	componentID=eWRhpRV, production=10, soc=0, timestamp="2024-01-01T15:00"	Enviar pedido POST para o endpoint '/sensor-data'	HTTP 201 Created com uma mensagem de sucesso os dados do sensor foram registrados
ITC_15	Consultar dados dos sensores para o intervalo de um ano		Enviar pedido GET para o endpoint '/components/eWRhpRV/sensor-data?year=2023'	HTTP 200 OK com um array que contém todos os dados dos sensores associados ao componente com o id presente no url.
ITC_16	Consultar dados dos sensores para o intervalo de um mês		Enviar pedido GET para o endpoint '/components/eWRhpRV/sensor-data?year=2023&month=12'	HTTP 200 OK com um array que contém todos os dados dos sensores associados ao componente com o id presente no url.
ITC_17	Consultar dados dos sensores para o intervalo de um dia		Enviar pedido GET para o endpoint '/components/eWRhpRV/sensor-data?year=2023&month=12&day=31'	HTTP 200 OK com um array que contém todos os dados dos sensores associados ao componente com o id presente no url.
ITC_18	Consultar dados dos sensores com os parâmetros inválidos		Enviar pedido GET para o endpoint '/components/eWRhpRV/sensor-data?year=2023&month=13'	HTTP 400 Bad Request com uma mensagem de erro a indicar que a data é inválida
ITC_19	Registrar alerta	componentID=eWRhpRV, alert_type=1, description="description", timestamp="2023-01-01T15:00"	Enviar pedido POST para o endpoint '/alerts'	HTTP 201 Created com uma mensagem de sucesso o alerta foi registrado

ID	Descrição	Input	Ações	Resultado esperado
ITC_20	Consultar os registos de todos os alertas		Enviar pedido GET para o endpoint '/alerts'	HTTP 200 OK com um array que contém todos os alertas registados.
ITC_21	Consultar os registos de um alerta		Enviar pedido GET para o endpoint '/alerts/fi4t7A2Z'	HTTP 200 OK com um array que contém dados do alerta com o id presente no url.
ITC_22	Registrar leitura de um contador	consumption=28.17, production=5.52, maxDemand= 10, imported_energy=25.17, exported_energy=5.32, timestamp="2024-01-01T15:00"	Enviar pedido POST para o endpoint '/energy-metering'	HTTP 201 Created com uma mensagem de sucesso a leitura foi registada
ITC_23	Consultar leituras de um contador para o intervalo de um ano		Enviar pedido GET para o endpoint '/energy-metering/{id}?year=2023'	HTTP 200 OK com um array que contém todas as leituras registadas no intervalo de tempo.
ITC_24	Consultar leitura de um contador para o intervalo de um mês		Enviar pedido GET para o endpoint '/energy-metering/{id}?year=2023&month=12'	HTTP 200 OK com um array que contém todas as leituras registadas no intervalo de tempo.
ITC_25	Consultar leituras de um contador para o intervalo de um dia		Enviar pedido GET para o endpoint '/energy-metering/{id}?year=2023&month=12&day=31'	HTTP 200 OK com um array que contém todas as leituras registadas no intervalo de tempo.
ITC_26	Consultar leitura de um contador com parâmetros inválidos		Enviar pedido GET para o endpoint '/energy-metering/{id}?year=2023&month=13&day=31'	HTTP 400 Bad Request com uma mensagem de erro a indicar que a data é inválida
ITC_27	Consultar as leituras para um intervalo de um ano		Enviar pedido GET para o endpoint '/energy-metering?year=2023'	HTTP 200 OK com um array que contém todas as leituras registadas no intervalo de tempo.

ID	Descrição	Input	Ações	Resultado esperado
ITC_28	Consultar as leituras para o intervalo de um mês		Enviar pedido GET para o endpoint '/energy-metering?year=2023&month=12'	HTTP 200 OK com um array que contém todos as leituras registadas no intervalo de tempo.
ITC_29	Consultar as leituras para o intervalo de um dia		Enviar pedido GET para o endpoint '/energy-metering?year=2023&month=12&day=31'	HTTP 200 OK com um array que contém todos as leituras registadas no intervalo de tempo.
ITC_30	Consultar as leituras com parâmetros inválidos		Enviar pedido GET para o endpoint '/energy-metering?year=2023&month=13&day=31'	HTTP 400 Bad Request com uma mensagem de erro a indicar que a data é inválida

Apêndice D – Testes de Sistema

Este apêndice apresenta os testes de sistema realizados para avaliar se o sistema cumpre os requisitos funcionais.

Tabela D 1 - Testes de sistema

ID	Descrição	Ações	Resultado esperado
STC_1	Registrar leitura de contador inteligente	1. Enviar pedido POST para o endpoint '/energy-metering'	Registro da leitura do contador inteligente
STC_2	Registrar dados dos sensores dos componentes	1. Enviar pedido POST para o endpoint '/sensor-data'	Registro dos dados do sensor
STC_3	Registro de alertas	1. Enviar pedido POST para o endpoint '/alerts'	Registro do alerta
STC_4	Apresentação de dados de consumo de energia	1. Navegar até Dashboard 2. Observar gráfico com dados relativos ao consumo de energia	Dados de consumo de energia
STC_5	Apresentação de dados de produção de energia	1. Navegar até Dashboard 2. Observar gráfico com dados relativos à produção de energia	Dados de produção de energia
STC_6	Apresentação de dados relativos ao armazenamento de energia	1. Navegar até Dashboard 2. Observar dados sobre a capacidade de armazenamento e a energia armazenada	Dados de armazenamento de energia
STC_7	Apresentação de lista de alertas	1. Navegar até secção de alertas 2. Observar lista de alertas	Lista de alertas
STC_8	Apresentação de detalhes de alerta	1. Navegar até secção de alertas 2. Selecionar alerta 3. Observar detalhes sobre alerta selecionada	Detalhes do alerta selecionado
STC_9	Apresentação de formulário para registo de componente	1. Navegar até secção de registo de componente	Formulário para registo de componente

ID	Descrição	Ações	Resultado esperado	
STC_10	Validação de dados relativos ao registo do componente	<ol style="list-style-type: none"> 1. Navegar até secção de registo de componente 2. Preencher formulário com dados inválidos 	Mensagem de erro para dados inválidos	
STC_11	Confirmação de registo de componente	<ol style="list-style-type: none"> 1. Navegar até secção de registo de componente 2. Preencher formulário 3. Submeter formulário 	Confirmação de registo do componente	
STC_12	Associação de ficheiros a componente	Não desenvolvido	Não desenvolvido	
STC_13	Remoção de componente	<ol style="list-style-type: none"> 1. Navegar até secção de componentes 2. Selecionar componente 3. Selecionar opção de remoção 	Componente é removido com sucesso passando ao estado de inativo	
STC_14	Apresentação de lista de componentes	<ol style="list-style-type: none"> 1. Navegar até secção de componentes 	Lista de componentes	
STC_15	Apresentação de dados de um componente selecionado	<ol style="list-style-type: none"> 1. Navegar até secção de componentes 2. Selecionar componente 	Detalhes do componente	
STC_16	Apresentação de manutenções associadas a um componente	<ol style="list-style-type: none"> 1. Navegar até secção de componentes 2. Selecionar componente 	Lista de manutenções associadas	
STC_17	Apresentação de dados de funcionamento de um componente	<ol style="list-style-type: none"> 1. Navegar até secção de componentes 2. Selecionar componente 	Gráfico com dados energéticos	
STC_18	Apresentação das leituras de um contador inteligente	<ol style="list-style-type: none"> 1. Navegar até secção de dados energéticos 2. Selecionar cliente 3. Selecionar intervalo de tempo 	Lista de leituras de contador inteligente	
STC_19	Apresentação da lista de manutenções agendadas	<ol style="list-style-type: none"> 1. Navegar até secção de manutenções 	Lista de manutenções agendadas	
STC_20	Apresentação de detalhes de manutenção selecionada	<ol style="list-style-type: none"> 1. Navegar até secção de manutenções 2. Selecionar manutenção 	Detalhes da manutenção	

ID	Descrição	Ações	Resultado esperado
STC_21	Apresentação de lista de manutenções a decorrer	1. Navegar até secção de manutenções	Lista de manutenções a decorrer
STC_22	Apresentação de histórico de atualizações sobre uma manutenção	1. Navegar até secção de manutenções 2. Selecionar manutenção	Detalhes de atualizações realizadas à manutenção
STC_23	Apresentação de lista de manutenções terminadas	1. Navegar até secção de manutenções	Lista de manutenções terminadas
STC_24	Apresentação de formulário para agendamento de manutenção	1. Navegação até secção de agendamento de manutenção	Formulário para o agendamento de manutenção
STC_25	Associação de ficheiros a manutenção	Não desenvolvido	Não desenvolvido
STC_26	Validação de dados relativos ao agendamento de uma manutenção	1. Navegar até secção de agendamento de manutenção 2. Preencher formulário com dados inválidos	Mensagem de erro para dados inválidos
STC_27	Confirmação do agendamento de manutenção	1. Navegar até secção de agendamento de manutenção 2. Preencher formulário 3. Submeter formulário	Confirmação de agendamento de manutenção
STC_28	Cancelamento de uma manutenção	1. Navegar até secção de manutenções 2. Selecionar manutenção 3. Cancelar manutenção 4. Inserir razão de cancelamento	Confirmação de cancelamento de manutenção
STC_29	Atualização do estado de manutenção para a decorrer	1. Navegar até secção de manutenções 2. Selecionar manutenção 3. Iniciar manutenção	Confirmação de início de manutenção
STC_30	Atualização do estado de manutenção para terminada	1. Navegar até secção de manutenções 2. Selecionar manutenção 3. Terminar manutenção	Confirmação de término de manutenção