



UNIVERSIDADE ESTADUAL DO PIAUÍ
CENTRO DE TECNOLOGIA E URBANISMO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Luiz Gustavo Araújo Gomes

**HydroChain: Um software protótipo para emissão de
certificados de origem para hidrogênio verde em
*blockchain***

TERESINA

2025

Luiz Gustavo Araújo Gomes

**HydroChain: Um software protótipo para emissão de
certificados de origem para hidrogênio verde em
*blockchain***

Monografia de Trabalho de Conclusão de
Curso apresentado na Universidade Esta-
dual do Piauí – UESPI como parte dos re-
quisitos para conclusão do Curso de Bacha-
relado em Ciência da Computação.

Orientador: Prof. Dr. Alcemir Rodrigues Santos

TERESINA
2025

G633h Gomes, Luiz Gustavo Araujo.

HydroChain: Um software protótipo para emissão de certificados de origem para hidrogênio verde em blockchain / Luiz Gustavo Araujo Gomes. - Teresina, 2025.

57f.: il.

Monografia (Graduação) - Universidade Estadual do Piauí - UESPI, Centro de Tecnologia e Urbanismo, Bacharelado em Ciência da Computação, 2025.

"Orientador: Prof. Dr. Alcemir Rodrigues Santos".

1. Hidrogênio Verde. 2. Blockchain. 3. Certificação. 4. Hyperledger FireFly. 5. Ethereum. I. Santos, Alcemir Rodrigues . II. Título.

CDD 005.75

Ficha elaborada pelo Serviço de Catalogação da Biblioteca da UESPI
JOSÉ EDIMAR LOPES DE SOUSA JÚNIOR (Bibliotecário) CRB-3^a/1512

HydroChain: Um software protótipo para emissão de certificados de origem para hidrogênio verde em blockchain

Luiz Gustavo Araújo Gomes

Monografia de Trabalho de Conclusão de Curso apresentado na Universidade Estadual do Piauí – UESPI como parte dos requisitos para conclusão do Curso de Bacharelado em Ciência da Computação.

Prof. Dr. Alcemir Rodrigues Santos
Orientador

Nota da Banca Examinadora: 9.0

Banca Examinadora:

Prof. Dr. Alcemir Rodrigues Santos,
Presidente

Prof. Dr. Constantino Augusto Dias Neto.
Membro

Prof. Dr. Sérgio Barros de Sousa.
Membro

Este trabalho é dedicado à minha família.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me conceder o dom da vida e por estar presente em cada passo da minha caminhada, sustentando-me nos momentos difíceis e alegrando os dias de conquista.

Aos meus pais, minha mãe e meu pai, pelo amor, apoio incondicional e pela educação que me deram pilares fundamentais para que eu pudesse chegar até aqui.

Aos meus irmãos, por serem presença constante e companheiros nos momentos bons e nos desafiadores, sempre ao meu lado com carinho e cumplicidade.

Aos professores que cruzaram meu caminho ao longo da vida, desde o ensino básico até o superior, por compartilharem conhecimento, inspirarem crescimento e contribuírem profundamente com minha formação.

Aos colegas de trabalho, pela parceria, aprendizado mútuo e pelo ambiente de troca que tanto me fez evoluir como profissional e como pessoa.

Ao meu orientador, agradeço pela dedicação, paciência e orientação ao longo deste trabalho. Sua experiência, conselhos e disponibilidade foram fundamentais para o desenvolvimento deste trabalho. Sou grato pelo apoio constante e pela confiança depositada em mim durante todo o processo.

RESUMO

Frente aos desafios do aquecimento global e a busca por descarbonização impulsiona pelo Acordo de Paris, o Hidrogênio Verde, produzido via eletrólise com energias renováveis, destaca-se como alternativa promissora. No entanto, sua sustentabilidade e a confiança do mercado são dificultadas pela ausência de um processo nacional de certificação no Brasil e pela falta de padronização, apesar da relevância de esquemas como o CertifHy e do potencial da tecnologia blockchain como “máquina de confiança” para rastreabilidade e transparência. Para endereçar essa lacuna, este trabalho propõe o HydroChain, um sistema que emite certificados de origem de hidrogênio verde como NFTs utilizando *blockchain*, visando fomentar o mercado e oferecer a rastreabilidade digital da produção. A sistema desenvolvido aprimora a credibilidade e eficiência da certificação, servindo como um ponto de partida para discussões no Brasil.

Palavras-chaves: Hidrogênio Verde, Blockchain, Certificação, Hyperledger FireFly, Ethereum.

ABSTRACT

Faced with the challenges of global warming and the quest for decarbonization driven by the Paris Agreement, Green Hydrogen, produced via electrolysis with renewable energies, stands out as a promising alternative. However, its sustainability and market confidence are hampered by the absence of a national certification process in Brazil and the lack of standardization, despite the relevance of schemes such as CertifHy and the potential of blockchain technology as a “trust machine” for traceability and transparency. To address this gap, this paper proposes HydroChain, a system that issues green hydrogen certificates of origin such as NFTs using blockchain, with the aim of boosting the market and offering digital traceability of production. The system developed improves the credibility and efficiency of certification, serving as a starting point for discussions in Brazil.

Keywords: Green Hydrogen, Blockchain, Certification, Hyperledger FireFly, Ethereum.

LISTA DE ILUSTRAÇÕES

Figura 1 – Principais atores envolvidos no processo de certificação.	19
Figura 2 – Representação de uma <i>blockchain</i> .	21
Figura 3 – Etapas adotadas na execução deste TCC.	25
Figura 4 – Visão geral dos protótipos da aplicação.	26
Figura 5 – Kanban durante o desenvolvimento da aplicação.	28
Figura 6 – Arquitetura visual da proposta.	32
Figura 7 – Representação de uma aplicação com e sem o uso do FireFly.	33
Figura 8 – Retorno da resposta do FireFly.	35
Figura 9 – Interface Web do Swagger.	40
Figura 10 – Requisição realizada no Insomnia ao endpoint /solicitacao da API, utilizando o método POST.	40
Figura 11 – Tela de registro de titular de conta.	47
Figura 12 – Tela de registro de dispositivos de produção.	47
Figura 13 – Tela de registro de lotes de produção.	48
Figura 14 – Tela de Solicitação de certificado.	48
Figura 15 – Tela de verificação de autenticidade de certificado.	49
Figura 16 – Tela de aprovação de dispositivos de produção.	50
Figura 17 – Tela de visualização das solicitações em análise.	50
Figura 18 – Tela de avaliação final de solicitação.	51

LISTA DE ABREVIATURAS E SIGLAS

CO ₂	Dióxido de Carbono
H ₂	Hidrogênio
H ₂ V	Hidrogênio Verde
API	Application Programming Interface
CO	Monóxido de Carbono
GC	Gaseificação de Carvão
GEE	Gases de Efeito Estufa
LES	Laboratório de Engenharia de Software
NFTs	<i>tokens non-fungibles</i>
RVM	Reforma a Vapor de Metano
SDK	<i>Software Development Kit</i>
SQL	<i>Structured Query Language</i>
TCC	Trabalho de Conclusão de Curso

LISTA DE QUADROS

1	Funções e responsabilidades dos principais atores envolvidos no processo de certificação.	20
2	Heurísticas de Nielsen.	29
3	Descrição das rotas implementadas na API REST. Fonte: Autoria própria	36
4	Escala de severidade.	52
5	Análise heurística do sistema.	53

LISTA DE ALGORITMOS

1	Trecho de código responsável pelo registro de carteira de usuário na rede FireFly	41
2	Trecho de código responsável pelo emissão de um NFT.	43

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Justificativa	15
1.2	Objetivos	16
1.2.1	Objetivo geral	16
1.2.2	Objetivos específicos	16
1.3	Fora do Escopo	16
1.4	Organização do documento	16
2	REFERENCIAL TEÓRICO	18
2.1	Hidrogênio Verde	18
2.2	Certificação	18
2.3	Blockchain	20
2.4	Trabalhos Relacionados	22
3	METODOLOGIA	25
3.1	Levantamento de Requisitos	25
3.2	Modelagem: Construção de Protótipos de Alta Fidelidade	26
3.3	Modelagem: Seleção do Uso de <i>Blockchain</i>	27
3.4	Desenvolvimento e Implantação	27
3.5	Avaliação	28
4	HYDROCHAIN	30
4.1	Motivação	30
4.2	Requisitos	31
4.3	Arquitetura do Sistema	32
4.4	Back-End	33
4.4.1	Hyperledger FireFly	33
4.4.2	API	35
4.5	Front-End	46
4.5.1	Titular de Conta	46
4.5.2	Órgão Emissor	49
5	AVALIAÇÃO	52
6	CONSIDERAÇÕES FINAIS	54
6.1	Limitações	55

6.2	Trabalhos Futuros	55
	REFERÊNCIAS	56

1 INTRODUÇÃO

Devido aos problemas que o mundo enfrenta em relação ao aquecimento global, entidades governamentais uniram forças para buscar formas de mitigar tais problemas. Com isso, surge o Acordo de Paris (Mani et al., 2018), um tratado internacional assinado durante a 21^a Conferência das Partes da Convenção-Quadro das Nações Unidas sobre Mudança do Clima, em 2015. O principal objetivo do Acordo de Paris é manter o aumento da temperatura média global bem abaixo de 2°C em comparação com os níveis pré-industriais, buscando limitar o aquecimento a 1,5°C. Essa meta ambiciosa visa reduzir os impactos negativos das mudanças climáticas, incentivando a transição para fontes de energia renovável e uma economia de baixo carbono.

Com este cenário, na busca pela descarbonização, o Hidrogênio Verde (H_2V) surge como uma alternativa favorável que tem recebido atenção significativa nos últimos anos. Segundo Hassan et al. (Hassan et al., 2024), o conceito de H_2V está relacionado ao ato de gerar hidrogênio utilizando fontes de energias renováveis, como a eólica, e solar. Ao contrário do método tradicional de produção de hidrogênio, que é intensivo em carbono, o H_2V é considerado um método sustentável.

O método tradicional de produção de hidrogênio, dominado pela Reforma a Vapor de Metano (RVM) e pela Gaseificação de Carvão (GC), é altamente intensivo em carbono. Na RVM, o gás natural, rico em metano, reage com vapor d'água em altas temperaturas, produzindo Hidrogênio (H_2) e Dióxido de Carbono (CO_2). Já na GC, o carvão é aquecido em um ambiente com pouco oxigênio, liberando gases como H_2V , Monóxido de Carbono (CO) e CO_2 . Ambos os processos emitem grandes quantidades de CO_2 , agravando o problema das mudanças climáticas. Em contraste, o H_2V utiliza o processo de eletrólise da água, que, quando alimentado por fontes renováveis como a eólica e a solar, se torna uma alternativa sustentável e promissora. No entanto, atualmente, apenas uma pequena fração do hidrogênio é produzida via eletrólise, como observa Gale et al. (Gale et al., 2024).

1.1 Justificativa

Com o crescimento da discussão de sustentabilidade e produção de energias limpas, surge a necessidade de verificar os meios e os processos de produção daquela energia produzida com um esquema criterioso e padronizado. Uma forma de ajustar este cenário é emitindo certificações que comprovem todos os critérios de uma produção de energia limpa de emissões de CO_2 (Atteya et al., 2023).

A certificação é um mecanismo que comprova a origem e a sustentabilidade da produção do H₂ (Mould et al., 2022). Ela permite que os negociadores possam ter mais confiança na qualidade do H₂, e que órgãos de regulamentação do setor possam assegurar padrões de produção ambientalmente responsáveis e de acordo com legislações vigentes. Oferecer a rastreabilidade da produção do H₂ permite que os negociadores rastreiem toda a origem desde a produção. Para aumentar a confiabilidade dessa rastreabilidade, o uso da tecnologia *blockchain* ajuda neste contexto, por ser uma tecnologia robusta e confiável.

1.2 Objetivos

1.2.1 Objetivo geral

Desenvolver um sistema protótipo que realize o processo de solicitação e emissão de certificados de origem de hidrogênio verde.

1.2.2 Objetivos específicos

- Implementar o modelo de certificação de hidrogênio.
- Desenvolver funcionalidade de emissão de certificados.
- Integrar a tecnologia *blockchain* para assegurar a confiabilidade e a imutabilidade dos dados armazenados.

1.3 Fora do Escopo

Ponto 1: Criação de um modelo de certificação, visto que será utilizado um modelo já existente;

Ponto 2: Criar um ecossistema de blockchain próprio, uma vez que frameworks prontos serão utilizados para facilitar o desenvolvimento;

Ponto 3: Implementar o módulo de auditoria do sistema;

1.4 Organização do documento

Este trabalho é estruturado em capítulos. O primeiro capítulo é uma introdução desta pesquisa para apresentar o contexto e a justificativa da mesma. O Capítulo 2 apresenta o referencial teórico. O Capítulo 3 expõe a metodologia utilizada no desenvolvimento do trabalho. O Capítulo 4 apresenta os detalhes do que foi desenvolvido no

trabalho. O Capítulo 5 mostra os detalhes da avaliação utilizada neste trabalho, e por fim, o Capítulo 6 apresenta a considerações finais com base no que foi atingido com esta pesquisa.

2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados conceitos importantes desta pesquisa, como (i) hidrogênio verde, (ii) certificação, e (iii) *blockchain*.

2.1 Hidrogênio Verde

O Hidrogênio Verde(H_2V) é basicamente hidrogênio produzido por meio da eletrólise da água, utilizando eletricidade gerada a partir de fontes de energia renováveis, como solar e eólica. Consiste em um processo químico que utiliza uma corrente elétrica para dividir uma substância em seus componentes básicos. No caso da eletrólise da água, a água é decomposta em hidrogênio e oxigênio. O processo ocorre em um dispositivo chamado eletrólito, que contém dois eletrodos: o ânodo (positivo) e o cátodo (negativo). Quando uma corrente elétrica é aplicada, as moléculas de água se dissociam. Os íons de hidrogênio, que são positivamente carregados, são atraídos para o cátodo, onde se combinam para formar gás hidrogênio. Ao mesmo tempo, os íons de oxigênio, que são negativamente carregados, se movem em direção ao ânodo, onde se combinam para formar gás oxigênio. O resultado é a produção de hidrogênio e oxigênio em estado gasoso, a eletrólise da água é uma maneira limpa de gerar hidrogênio por não emitir Gases de Efeito Estufa (GEE), especialmente quando a eletricidade utilizada provém de fontes renováveis (Elshafei; Mansour, 2023).

O H_2V é considerado uma alternativa promissora para reduzir as emissões de carbono, ele pode ser utilizado em diversas aplicações, como o transporte, aquecimento e o armazenamento de energia. À medida que o mundo busca alternativas aos combustíveis fósseis, o hidrogênio verde se destaca como um componente crucial na construção de economias de baixo carbono e na mitigação das mudanças climáticas (Elshafei; Mansour, 2023).

2.2 Certificação

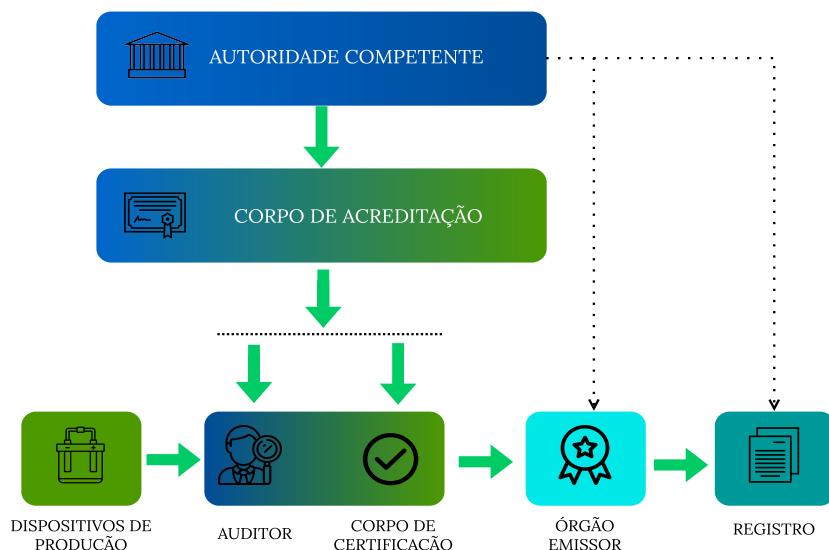
Os esquemas de certificação do H_2V são sistemas desenvolvidos para garantir que a produção desse combustível atenda a padrões específicos de sustentabilidade e redução de emissões de GEE. Esses esquemas registram informações detalhadas sobre a origem e o processo de produção, incluindo o tipo de energia utilizada e a quantidade de emissões associadas. Contudo, é importante ressaltar que a certificação do H_2V não visa rastrear o fluxo físico dos lotes de hidrogênio ao longo da cadeia de

suprimentos. Em vez disso, ela se concentra na “pegada de carbono” e nos padrões ambientais associados ao processo de produção em uma etapa específica (Mould et al., 2022).

Neste trabalho, o esquema de certificação da CertifHy (CertifHy, 2024) será utilizado como base para a implementação de um sistema de emissão de certificados para o H₂V por meio de software. O esquema foi escolhido por ser pioneiro e amplamente reconhecido na União Europeia, com critérios rigorosos de rastreabilidade e transparência. Na sequência, será detalhado o funcionamento do modelo da CertifHy, destacando seus principais critérios e requisitos.

O esquema é composto por atores que participam do processo, os **Auditores** garantem que os dispositivos de produção estão em conformidade com o esquema (corpo de certificação), eles pertencem a um **Organismo de Certificação** que é uma entidade que possui a acreditação necessária para realizar as atividades. O **Corpo de Acreditação** é responsável por assegurar a qualidade dos **Organismos de Certificação**, fortalecendo a confiança na avaliação de conformidade. Seu papel é garantir que esses organismos possuam a competência técnica necessária para desempenhar suas funções. A Responsabilidade de emitir os certificados pertence ao **Órgão Emissor**, que está sob controle direto da **Autoridade Competente**.

Figura 1 – Principais atores envolvidos no processo de certificação.



Fonte: Adaptado da Documentação (CertifHy, 2024).

É mostrado no Quadro 1 mais detalhes em relação aos *stakeholders* envolvidos no processo de certificação, incluindo a definição de cada um e suas principais tarefas e responsabilidades. Essas informações são essenciais para compreender a estrutura e o funcionamento do modelo de certificação proposto pela CertifHy.

Quadro 1 – Funções e responsabilidades dos principais atores envolvidos no processo de certificação.

Stakeholder	Definição	Tarefas
Autoridade Competente	Em relação ao exercício ou ao desempenho de qualquer função legislativa, governamental, regulamentar ou administrativa, o organismo devidamente autorizado pelas leis e regulamentos da UE a exercer ou desempenhar essa função.	Gerenciar Entidades.
Corpo de Acreditação	Entidade aceita pela autoridade competente para acessar e acreditar o corpo de certificação	Acreditação
Auditor	Pessoa designada por um organismo de certificação para avaliar a produção ou em relação à exigência do regime de garantia de origem	Avaliação da conformidade de origem garantida
Corpo de Certificação	Entidade autorizada a atuar como verificador ambiental ou organização de verificação ambiental e aprovada por um organismo de acreditação	Certificação de conformidade de origem garantida
Órgão Emissor	entidade responsável pelo registro de entidades, no caso de um dispositivo de produção, o registro é baseado em uma avaliação do auditor.adicionalmente, a entidade é responsável pela emissão de Garantia de Origem	Emissão de Garantia de Origem
Administrador de Registro	Entidade designada pela autoridade competente e/ou órgão emissor para operar e manter o registro	Registrar Transações

Fonte: Adaptado da Documentação (CertifHy, 2024).

2.3 Blockchain

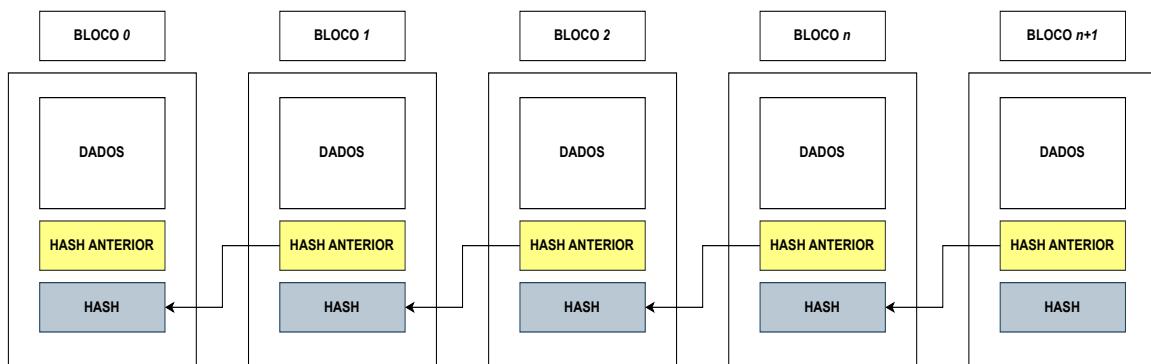
A tecnologia *blockchain* consiste em uma cadeia de blocos encadeados e criptografados conforme é mostrado na Figura 2, formando um livro-razão distribuído que armazena informações de maneira segura e resistente a alterações não autorizadas. Cada bloco na cadeia contém um conjunto de registros, um carimbo de tempo e uma referência ao bloco anterior, criando uma estrutura linear e sequencial. Essa arquitetura fundamental serve como a base tecnológica para a construção de redes *blockchain* (Martins et al.,).

Redes *blockchain*, por sua vez, utilizam a estrutura da tecnologia de cadeia de blocos para operar de forma descentralizada e distribuída. Em vez de depender de uma autoridade central, essas redes empregam protocolos de consenso para validar

e registrar novas informações. Esses protocolos asseguram que qualquer dado a ser adicionado ou modificado deve ser aprovado por todos os participantes ou por um número suficiente de nós da rede, aumentando a confiabilidade consideravelmente.

A descentralização e o uso de consenso tornam as redes *blockchain* altamente seguras e transparentes. Cada nó participante contribui para a integridade do sistema, eliminando pontos únicos de falha e fortalecendo a imutabilidade dos registros. Dependendo de sua configuração, as redes podem ser categorizadas como *permissionless* (sem permissão), abertas a qualquer participante, ou *permissioned* (com permissão), restritas a um grupo controlado de participantes (Yaga et al., 2019).

Figura 2 – Representação de uma *blockchain*.



Fonte: Autoria própria.

Permissionless: Este tipo de rede tem a característica de operar de forma pública, onde qualquer usuário consegue participação sem a necessidade de uma autorização prévia. faz uso de protocolos de consenso descentralizados, e tem maior transparência devido ao acesso público dos dados que estão presentes na rede, além do anonimato.

Permissioned: Redes *Permissioned* (com Permissão), podem ser caracterizadas como redes centralizadas, pois necessitam de uma autoridade para conceder a permissão aos usuários que desejam ingressar na rede, sendo um modelo interessante visando um ambiente privado ou corporativo. O acesso a leitura e quem pode ou não realizar transações na rede pode ser controlado, uma vez que apenas usuários com permissão podem ingressar na rede.

Embora as redes *blockchain* possam ser configuradas como *Permissioned* ou *Permissionless*, elas compartilham funcionalidades essenciais que suportam a execução de transações. Entre essas funcionalidades, destaca-se o uso de *Smart Contracts* (contratos inteligentes), que desempenham um papel crucial na automação e na segurança das operações realizadas na *blockchain*. Um *Smart Contracts* (contrato inteligente) é um programa digital executado em uma *blockchain* que permite a criação

de acordos e transações entre partes sem a necessidade de intermediador. As cláusulas previamente estabelecidas no código são executadas de forma automática, o que garante que o contrato seja cumprido conforme acordado. Assim, quando as condições estabelecidas são atendidas, o contrato inteligente executa as ações necessárias, como transferências de ativos ou registros de informações (Taherdoost, 2023).

Com a possibilidade de executar contratos personalizados, e criar aplicações baseadas em *blockchain* para vários contextos de negócio, surge o termo *Web3*, ou *Web 3.0*, que representa a nova dinâmica de aplicações web que funcionam de forma descentralizada. As aplicações descentralizadas garantem transparência, segurança e imutabilidade das informações. Essa nova abordagem também fomenta a criação de economias baseadas em *tokens*, onde os usuários podem ser recompensados por sua participação e contribuição para a rede, promovendo um ambiente colaborativo e incentivando a inovação. Com a *Web3*, a promessa é de um ecossistema digital mais inclusivo, onde a descentralização não apenas melhora a segurança e a privacidade, mas também empodera os usuários, permitindo que eles se tornem protagonistas em suas interações online (Martins et al.,).

2.4 Trabalhos Relacionados

A revisão bibliométrica de Chrysikopoulos et al. (Chrysikopoulos et al., 2024) apresenta o campo de pesquisa sobre certificados de origem, identifica tendências, publicações, instituições e áreas geográficas de maior produção acadêmica. Ela identifica que publicações sobre certificados de origem tiveram um crescimento relevante nos últimos anos, e as tendências de pesquisa incluem a eficácia dos certificados em incentivar a produção de energia renovável, o impacto da redução de emissões de carbono e a confiabilidade dos sistemas de certificação. Como desafios foram detectados a falta de um padrão internacional e dificuldade de integrar os certificados em sistemas energéticos variados. Ressalta-se uma necessidade de políticas mais uniformes para que seja possível garantir credibilidade e transparência em diferentes mercados. Sugere-se direções de pesquisa, uma delas é a inovação tecnológica e comércio de certificados verdes, é abordado como a *blockchain* tem um potencial em reduzir custos transacionais, além de fornecer uma garantia de origem e aumentar a transparência.

Na obra de Amenta, Sanseverino e Stagnaro (Amenta; Sanseverino; Stagnaro, 2021), os autores observam a relação entre a inovação digital, a regulação e a governança do setor elétrico, com atenção específica na implementação de *blockchain* em um contexto de sustentabilidade, ele discute como a *blockchain*, ao oferecer soluções descentralizadas e transparentes, pode mudar a governança de energia elétrica, facilitando o monitoramento de transações entre consumidores e fornecedores, especial-

mente em mercados de energia renovável. A pesquisa também discute as implicações regulatórias dessa transformação digital, ressaltando que para que a *blockchain* tenha sucesso na promoção de sistemas neste mercado, uma revisão das regulamentações existentes é necessária, para que sejam adaptadas para um modelo de governança mais flexível e inovador. Embora a *blockchain* tenha um potencial de promover sustentabilidade, é enfatizado que a implementação da mesma deve ser integrada de forma responsável ao marco regulatório existente para garantir alinhamento com objetivos ambientais e econômicos mais amplos, como a transição para energias renováveis.

As obras de Amenta, Sanseverino e Stagnaro (Amenta; Sanseverino; Stagnaro, 2021) e Chrysikopoulos et al. (Chrysikopoulos et al., 2024) se relacionam com este trabalho ao destacarem a importância de garantir transparência, rastreabilidade e conformidade com normas regulatórias nos processos de certificação. Estas características são fundamentais para assegurar a sustentabilidade e a confiança no mercado, especialmente em setores que dependem de certificações de origem, como o de energia renovável. A proposta deste trabalho se alinha a essas demandas, ao explorar a aplicação de tecnologias como a *blockchain* para aprimorar a credibilidade, a eficiência e a integridade dos sistemas de certificação, contribuindo para a transição energética e a promoção de práticas sustentáveis.

No artigo de De Filippi, Mannan e Reijers (De Filippi; Mannan; Reijers, 2020), é discutido como a tecnologia *blockchain* pode aumentar a confiança em sistemas descentralizados, porém enfatiza que isso não elimina a necessidade de confiança. Ao contrário de uma “tecnologia sem confiança”, ela é descrita como uma “máquina de confiança”, que utiliza criptografia, incentivos econômicos e regras matemáticas para gerar segurança nas transações e operações. Partindo desse pressuposto, seu funcionamento depende de uma rede complexa de atores que precisam ser considerados na governança. Apesar da ausência de intermediários para garantir a transparência nas interações, a confiança é redistribuída entre os participantes da rede, como programadores, validadores e mineradores. Esses agentes desempenham um papel essencial na operação e manutenção do sistema, evidenciando que a tecnologia não elimina completamente a necessidade de confiança, mas a reconfigura para elementos técnicos e sociais. O artigo também ressalta que a eficácia e a confiabilidade de sistemas descentralizados dependem de uma governança robusta, capaz de assegurar que todos os envolvidos atuem de maneira ética e transparente.

Esta análise dialoga diretamente com o uso dessa tecnologia na certificação de hidrogênio verde, onde a confiabilidade do sistema é crucial para garantir rastreabilidade, transparência e sustentabilidade na produção. Assim, a implementação de um modelo descentralizado de certificação exige não apenas a adoção de ferramentas seguras, mas também um modelo de governança que regule a atuação dos par-

ticipantes, assegurando o cumprimento dos critérios de conformidade estabelecidos, como os definidos pela CertifHy. Nesse contexto, a tecnologia pode funcionar como uma “máquina de confiança”, desde que os desafios relacionados à governança sejam adequadamente enfrentados, garantindo ampla aceitação e legitimidade por parte dos usuários.

3 METODOLOGIA

Este capítulo apresenta o detalhamento da metodologia utilizada para a construção deste Trabalho de Conclusão de Curso (TCC), cujo objetivo foi o desenvolvimento da aplicação HydroChain. A Figura 3 apresenta um esquema ilustrativo das etapas metodológicas que nortearam a execução do trabalho. O desenvolvimento foi conduzido em quatro etapas principais: levantamento de requisitos, modelagem, desenvolvimento e avaliação. Cada uma das etapas está descrita a seguir:

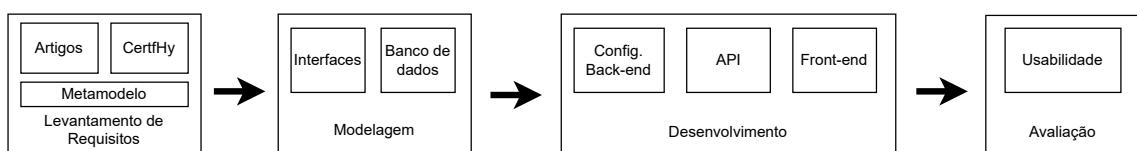
Levantamento de Requisitos: esta etapa consistiu na análise de trabalhos relacionados, no estudo de certificações de hidrogênio verde, como a CertifHy, e na compreensão do metamodelo proposto por trabalhos anteriores, o que permitiu estabelecer a base conceitual e técnica para a aplicação.

Modelagem: com os requisitos definidos, foram elaborados os primeiros protótipos da aplicação por meio de *mockups* de interface e estruturação inicial das funcionalidades, a fim de orientar o desenvolvimento posterior. Nesta etapa, também foi realizada a modelagem do banco de dados, definindo as entidades, atributos e relacionamentos necessários para o funcionamento do protótipo.

Desenvolvimento: nesta fase, a implementação foi dividida em subetapas: banco de dados, configuração da Application Programming Interface (API), codificação do back-end e front-end.

Avaliação: nesta etapa realizou-se uma avaliação qualitativa da aplicação com base nas heurísticas de usabilidade propostas por Nielsen (Nielsen, 1994), permitindo identificar eventuais oportunidades de melhoria.

Figura 3 – Etapas adotadas na execução deste TCC.



Fonte: Autoria própria

3.1 Levantamento de Requisitos

Nesta seção, serão explanados detalhes sobre o levantamento de requisitos que teve início com identificação dos requisitos necessários para a construção de um

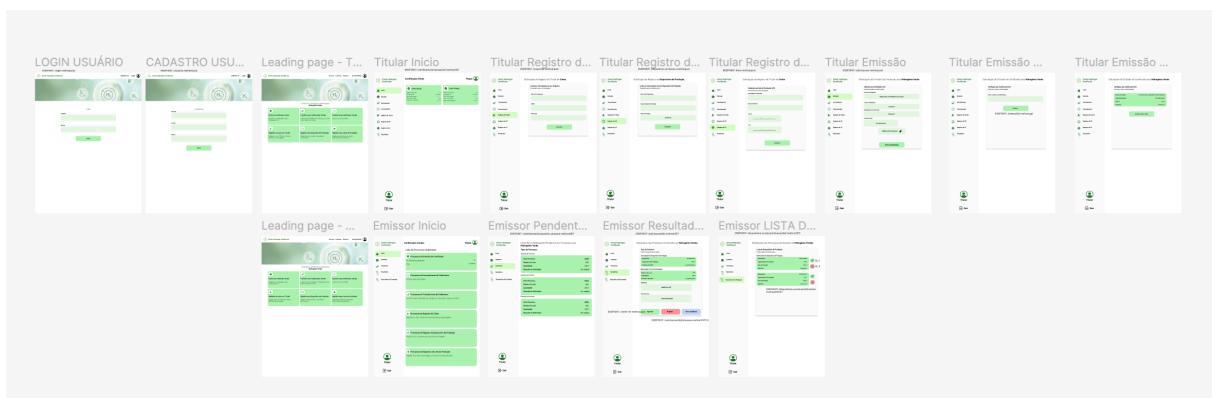
sistema de certificação da origem da produção de H₂V. Como atualmente não existe um processo nacional consolidado para essa finalidade, o levantamento de requisitos foi feito de maneira *ad-hoc* a partir de artigos científicos conhecidos de Chrysikopoulos et al. (Chrysikopoulos et al., 2024), Amenta, Sanseverino e Stagnaro (Amenta; Sanseverino; Stagnaro, 2021), De Filippi, Mannan e Reijers (De Filippi; Mannan; Reijers, 2020), o modelo de certificação proposto pela iniciativa europeia *CertifHy* (CertifHy, 2024), e de trabalhos anteriores conduzidos no grupo de pesquisa (Sampaio; Santos, 2024).

O trabalho de Sampaio e Santos (Sampaio; Santos, 2024) explora o modelo de certificação de uma lista de processos de certificação, dentre eles está o da CertifHy. Tanto a CertifHy, quanto o metamodelo construído por Sampaio e Santos (Sampaio; Santos, 2024) apontam vários atores no processo de certificação. Sampaio e Santos (Sampaio; Santos, 2024) identificou atores envolvidos no esquema, e com isso gerou um metamodelo para o processo de certificação de H₂V, partindo disso, os principais atores envolvidos no processo foram: Titular de Conta (produtor de H₂V), Auditores, Órgão Emissor, Autoridade Competente.

3.2 Modelagem: Construção de Protótipos de Alta Fidelidade

Após levantamento de requisitos concluído, foi possível iniciar a etapa de modelagem de protótipos para a aplicação. A modelagem dos protótipos foi feita utilizando a ferramenta *Figma*. A Figura 4 apresenta uma visão geral dos protótipos de alta fidelidade do HydroChain. Ao todo, foram modeladas 14 telas para a aplicação. As telas foram divididas de acordo com os dois atores levados em consideração. Três telas são comuns a ambos, relacionadas a recepção dos usuários e a autenticação dos mesmos no sistema. Cinco, referentes à atuação do “Órgão emissor” e seis referentes à atuação do “Titular de conta”.

Figura 4 – Visão geral dos protótipos da aplicação.



Fonte: Autoria própria.

3.3 Modelagem: Seleção do Uso de *Blockchain*

Aplicações baseadas em *blockchain*, apesar de transmitirem a ideia de descentralização total, nem sempre são adequadas para armazenar todos os dados diretamente na cadeia de blocos. Como argumentam Molina-Jimenez et al. (Molina-Jimenez et al., 2019), a depender do cenário, pode ser necessária uma abordagem híbrida, que combine o armazenamento de informações tanto *on-chain* quanto *off-chain*, significando que nem todos os dados são registrados na *blockchain*.

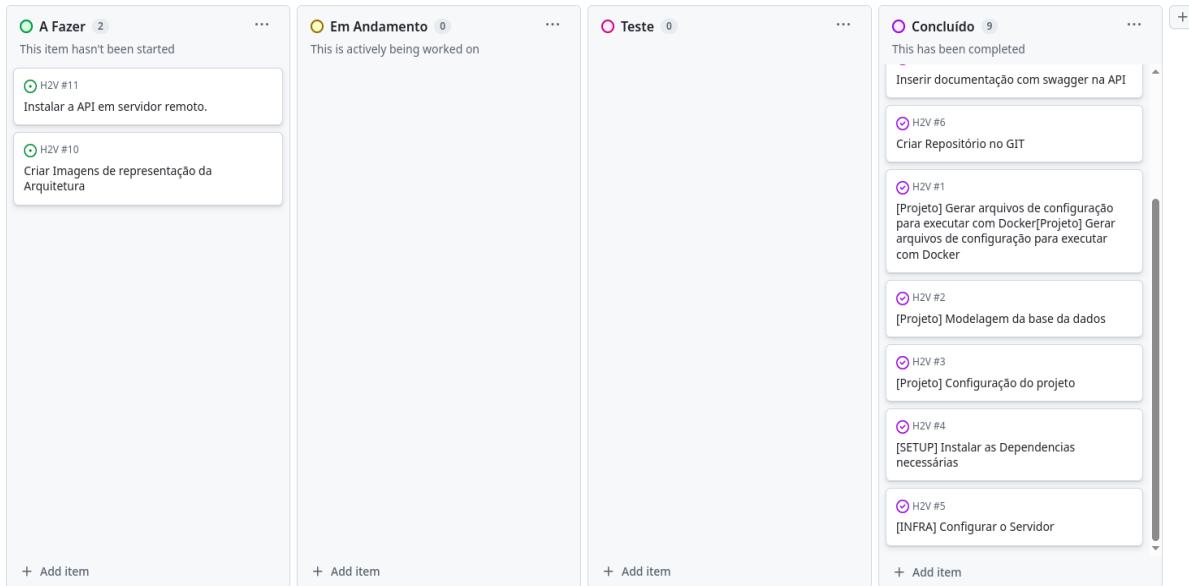
A princípio se pensou no uso de uma *blockchain* permissionada para o sistema, porém a complexidade no aprendizado com o tempo disponível tornou este caminho inviável, com essa limitação uma *blockchain* pública foi selecionada pela facilidade de integração. Tendo isso em vista, durante o processo de desenvolvimento do sistema, foi necessária a elaboração de uma modelagem para um banco de dados externo que funcionasse como mecanismo *off-chain*, auxiliando no desempenho e na escalabilidade futura da aplicação.

Após a modelagem da base de dados, realizou-se a definição das tecnologias que seriam utilizadas no desenvolvimento da API. A escolha das tecnologias utilizadas levou em consideração critérios práticos observados durante o desenvolvimento, como a facilidade de integração com *blockchain*, o suporte a tipos e validação de dados, além da familiaridade prévia com as ferramentas.

3.4 Desenvolvimento e Implantação

O trabalho foi construído utilizando-se a metodologia ágil Kanban (Anderson, 2011). O Kanban foi escolhido por ser uma técnica visual e flexível de controle do fluxo de trabalho, oferecendo algumas vantagens, como a melhoria da organização das atividades, a flexibilidade para adaptação e a promoção da melhoria contínua. O aumento da eficiência ocorre principalmente pela limitação do trabalho em progresso e pela priorização visual das tarefas, o que reduz o tempo de ciclo e evita a sobrecarga da equipe. Sua flexibilidade permite adicionar novas tarefas conforme a necessidade, sem a restrição de ciclos fixos, como ocorre em outras metodologias ágeis, tais como SCRUM (Schwaber; Beedle, 2001) e RUP (Kruchten, 2004), o que possibilita uma adaptação contínua às mudanças de prioridades. No contexto deste projeto, o Kanban foi utilizado para gerenciar as tarefas de desenvolvimento, desde o levantamento de requisitos até a implementação e os testes.

Figura 5 – Kanban durante o desenvolvimento da aplicação.



Fonte: Autoria própria.

A implantação foi realizada na infraestrutura do Laboratório de Engenharia de Software (LES) localizado no campus de Piripiri. Utilizou-se a tecnologia de virtualização baseada em containers, por meio da plataforma Linux Containers (Linux Containers, 2025). Um container foi disponibilizado especificamente para este trabalho, seguindo a prática já adotada pelo laboratório, que também fornece ambientes isolados para outros projetos desenvolvidos no LES, vale ressaltar que a implantação ainda não é um ambiente de produção para o funcionamento real do sistema.

3.5 Avaliação

O método de avaliação heurística foi utilizado como método de avaliação deste trabalho, a avaliação heurística é um método da IHC e foi criado para detectar problemas de usabilidade durante um processo de design iterativo. O método em pauta orienta os avaliadores a inspecionar de forma sistemática as interfaces a procura de problemas que prejudiquem a usabilidade do usuário e tem como base um conjunto de diretrizes de usabilidade que descrevem características desejáveis da interação e da interface (Barbosa et al., 2021). Nielsen (Nielsen, 1994) descreve um conjunto de heurísticas a serem usadas em seu método de avaliação, as heurísticas de Nielsen são mostradas no Quadro 2.

Quadro 2 – Heurísticas de Nielsen.

Identificador	Heurística	Descrição
H01	Visibilidade do estado do sistema	O sistema mantém o usuário sempre informado sobre o que está acontecendo no mesmo.
H02	Consistência de padrões	Evitar que o usuário tenha que pensar se ações ou situações diferentes significam a mesma coisa.
H03	Flexibilidade e eficiência de uso	Fornece opções que otimizam a experiência de usuário (atalhos, teclas de funções, abreviações).
H04	Estética e design minimalista	Evita o uso de informações irrelevantes.
H05	Compatibilidade do sistema e o mundo real	O sistema utiliza uma linguagem aos usuários, em vez de termos técnicos e específicos.
H06	Reconhecimento de erros	Utilizar linguagem simples para apresentar os erros e mostrar como contorná-los.
H07	Controle do usuário e liberdade	Oferece saída de emergência, permitindo que os usuários saiam facilmente de situações inesperadas.
H08	Reconhecimento e memorização	Fazer com que as ações e opções, sejam visíveis, o usuário não tem a obrigação de lembrar de tudo.
H09	Prevenção de erros	Prevenir a ocorrência de erros.
H10	Ajuda e documentação	Fornece informações que podem ser facilmente encontradas e orienta os usuários.

Fonte: Nielsen (Nielsen, 1994)

4 HYDROCHAIN

Este capítulo detalha o desenvolvimento do HydroChain. O HydroChain é um sistema que possibilita a rastreabilidade de certificados de origem do H₂V. O sistema desenvolvido faz uso de tecnologia *blockchain* para oferecer a rastreabilidade, transparência durante todo o ciclo de vida dos certificados. Essa rastreabilidade digital permite que os compradores verifiquem a autenticidade da origem do hidrogênio, bem como o processo percorrido desde a criação até a certificação como *hidrogênio verde* pelas autoridades competentes. Esta garantia pode incentivar o mercado e a competitividade do hidrogênio verde, uma vez que não apenas facilita a certificação de hidrogênio verde, mas também aumentar a confiança dos investidores no mercado.

O restante do capítulo descreve (i) a motivação para um sistema como o proposto, (ii) o levantamento de requisitos, (iii) a arquitetura definida para o sistema baseado em *blockchain*, (iv) o desenvolvimento do *back-end* e (v) do *front-end* do sistema proposto.

4.1 Motivação

Com a crescente atenção às fontes de energia renováveis, o mercado tem observado um aumento significativo na demanda por certificações que garantam a qualidade e a transparência das práticas de sustentabilidade. Esses certificados, conhecidos como “green certificates” ou certificados verdes, são instrumentos que comprovam que uma determinada quantidade de eletricidade foi gerada a partir de fontes renováveis. Eles permitem que as organizações demonstrem seu compromisso com a energia limpa e contribuam para a redução da pegada de carbono.

A importância dessas certificações se reflete na sua capacidade de promover a transparência no mercado de energia, possibilitando que os consumidores façam escolhas informadas sobre as fontes de energia que utilizam. Além disso, a evolução dos mecanismos de certificados verdes acompanhou o crescimento do setor de energias renováveis em diversas partes do mundo, desde a União Europeia até os Estados Unidos e países como China e Austrália, que implementaram seus próprios sistemas de certificação (Chrysikopoulos et al., 2024). Para o desenvolvimento da proposta, o processo de certificação da CertifHy (CertifHy, 2024) será utilizado. A escolha desse processo de certificação se justifica pela ausência de uma regulamentação nacional.

4.2 Requisitos

Durante a análise dos requisitos e *stakeholders*, identificou-se a necessidade de estruturar a solução em três(3) módulos principais de acordo com os três(3) dos quatro(4) principais atores apontados por Sampaio e Santos (Sampaio; Santos, 2024). Os módulos considerados fundamentais para o HydroChain são:

1. **Titular de Conta:** Representa os usuários produtores de H₂V, responsáveis por submeter pedidos de certificação de seus lotes.
2. **Órgão Emissor:** Responsável por analisar os dados recebidos, aprovar/rejeitar solicitações e emitir o certificado digital.
3. **Auditores:** Parte fundamental do processo de verificação, com a função de auditar tecnicamente os dados fornecidos antes da emissão dos certificados. Este módulo interage diretamente com o órgão emissor.

No entanto, foi necessário simplificar o processo para viabilizar a construção deste primeiro protótipo. Daqueles atores apontados por Sampaio e Santos (Sampaio; Santos, 2024), decidiu-se pela implementação de apenas dois (2) módulos referentes aos atores: O *Titular de conta*, e o *Órgão emissor*. Essa delimitação se deve a restrições de tempo e à complexidade envolvida, sendo este projeto um protótipo inicial que busca demonstrar a viabilidade da ideia e contribuir com os primeiros passos na discussão sobre certificação de hidrogênio verde.

Assim, considerando estes dois atores e considerando o estudo de Sampaio e Santos (Sampaio; Santos, 2024), enumerou-se os requisitos funcionais do sistema relacionados ao “*Titular de conta*”:

Cadastro de Solicitantes - O sistema deve permitir que empresas/produtores se cadastram como titular de conta para solicitar certificados de hidrogênio verde.

Autenticação de Usuários - O sistema deve oferecer autenticação segura para acesso ao painel de solicitação de certificados.

Solicitação de Certificação - O sistema deve permitir que usuários autenticados enviem uma solicitação de certificação.

Validação de Dados - O sistema deve validar os dados recebidos para garantir conformidade com critérios de certificação.

Registro do Certificado na *Blockchain* - O sistema deve registrar o certificado na *blockchain*, emitindo um token representando o certificado de H₂V.

Consulta Pública de Certificados - O sistema deve permitir a consulta pública de certificados emitidos, com base em identificadores.

Visualização de Certificado - O sistema deve gerar uma representação visual do certificado na interface de uma tela web.

Quanto às atividades do “Órgão emissor”, foram definidos os seguintes requisitos funcionais:

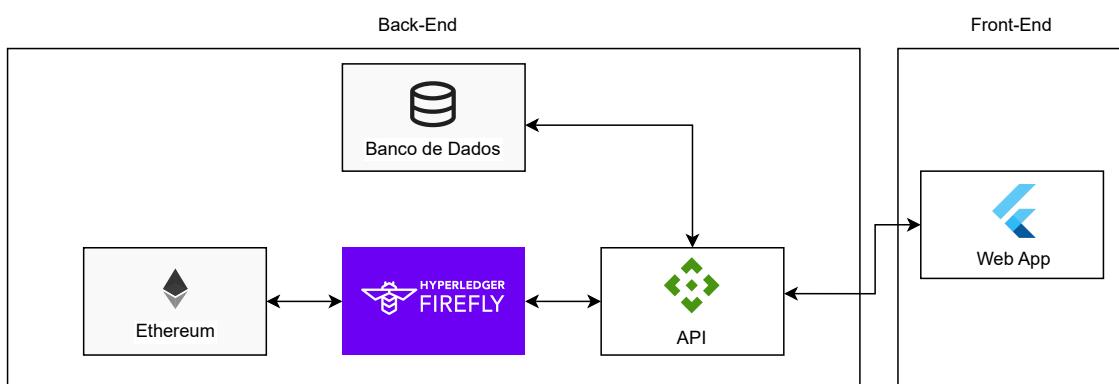
Aprovação Manual ou Semiautomática - O sistema deve permitir que um operador (ou autoridade certificadora) aprove ou rejeite uma solicitação com base nos dados recebidos.

Emissão de Certificado Digital - Uma vez aprovado, o sistema deve gerar um certificado digital para o lote de H₂V, com um identificador único, data de emissão e dados relevantes.

4.3 Arquitetura do Sistema

Nesta seção, será explanado a configuração utilizada para o desenvolvimento do protótipo, conforme é mostrado na Figura 6, o protótipo será composto por cinco componentes, que estão divididos em *front-end* com a aplicação web, que será responsável por fornecer uma interface ao usuário, e *back-end* com a API que servirá para configurar interfaces personalizadas de requisições e fornecer *endpoints* à aplicação web, o FireFly como framework para facilitar o uso de *blockchain*, a rede Ethereum e um banco de dados. O fluxo de funcionamento da arquitetura consiste na execução de ações pelo usuário na interface do *Web App*, que envia requisições à API. A API, por sua vez, interage com o banco de dados e com o *Hyperledger FireFly*, que atua como ponte de comunicação com a rede Ethereum.

Figura 6 – Arquitetura visual da proposta.



Fonte: Autoria própria.

4.4 Back-End

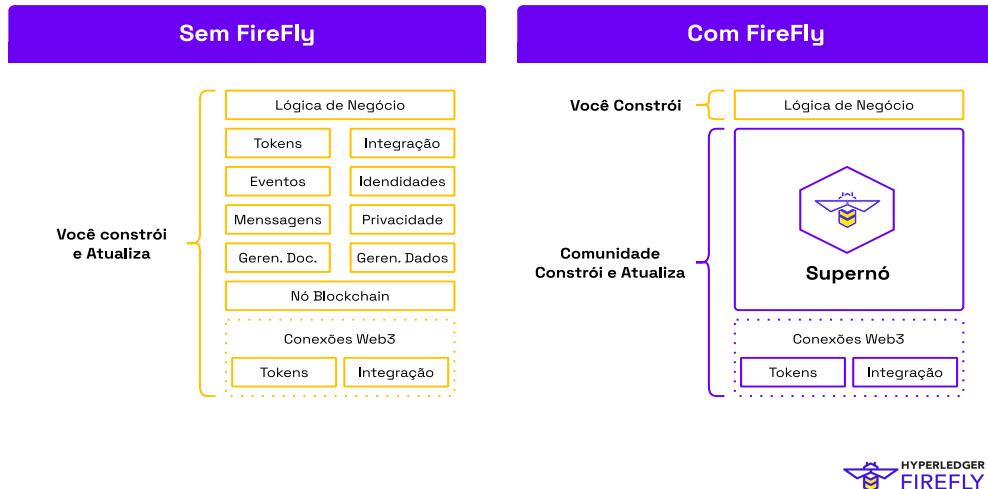
Esta seção apresenta os componentes da arquitetura desenvolvida para o protótipo em questão que foram apresentados na seção anterior.

4.4.1 Hyperledger FireFly

Esta ferramenta tem a finalidade de facilitar a configuração de conectores para uma aplicação se comunicar com uma rede *blockchain*. A mesma é uma plataforma de código aberto que facilita o desenvolvimento de aplicações corporativas baseadas em *blockchain*, oferecendo ferramentas para integração, orquestração e gerenciamento de transações em redes descentralizadas.

Na Figura 7, é notável a quantidade de microserviços que deveriam ser configurados para o funcionamento pleno de uma aplicação. com o FireFly, esse problema é sanado e o desenvolvedor irá se preocupar apenas com a lógica de negócios por trás do sistema que será desenvolvido. Dessa forma, a utilização deste *framework* torna-se indispensável para o desenvolvimento deste trabalho.

Figura 7 – Representação de uma aplicação com e sem o uso do FireFly.



Fonte: Adaptado da Documentação (Foundation, 2024).

A rede Ethereum foi utilizada no desenvolvimento deste trabalho devido à sua facilidade de configuração, extensa documentação e suporte a funcionalidades essenciais, como contratos inteligentes e emissão de *tokens* fungíveis e não fungíveis. Por se tratar de uma *blockchain* pública e não permissionada, conforme discutido no referencial teórico, a Ethereum permite que qualquer participante interaja livremente com a rede.

A rede foi configurada em uma máquina com sistema operacional Linux, que oferece um ambiente propício para a instalação de todas as dependências necessárias. Após a instalação dos pré-requisitos, como Docker, Git e cURL, uma instância da rede FireFly com Ethereum foi configurada para atuar como a rede *blockchain* da aplicação.

Configurando o FireFly: Para este passo, foi necessário a instalação do *framework*, para o mesmo também é necessário a utilização do docker, que foi instalado previamente. A instalação do FireFly consiste em realizar o download dos binários e extrair nos diretórios do sistema.

De acordo com a documentação do Foundation (Foundation, 2024) e Martins et al. (Martins et al.,) após o download dos binários do framework, deve-se executar o seguinte comando no terminal:

```
sudo tar -zxf ~/Downloads/firefly-cli_*.tar.gz -C /usr/local/bin ff && rm  
~/Downloads/firefly-cli_*.tar.gz
```

Em seguida, o desenvolvedor já terá acesso aos comandos do FireFly diretamente no terminal. Ao executar o comando ff, o seguinte resultado será exibido na Figura 8.

Para instanciar uma pilha do FireFly, foi utilizado o seguinte comando:

```
ff init ethereum h2v-stack 1 -d postgres
```

Nesse comando, ff é utilizado para executar o framework; init serve para iniciar uma nova pilha; ethereum especifica a blockchain que será utilizada; h2v-stack é o nome atribuído à pilha; o número 1 indica a quantidade de membros na rede (inicialmente criada com apenas um, para fins de desenvolvimento); e -d postgres define que o banco de dados utilizado será o PostgreSQL. O retorno da execução do comando informado deve se parecer com isso:

1. initializing new FireFly stack...
2. Stack 'h2v-stack' created!
3. To start your new stack run:
4. ff start h2v-stack
5. Your docker compose file for this stack can be found at:
6. /home/gustavo/.firefly/stacks/h2v-stack/docker-compose.yml

Conforme o log retornado, a pilha foi criada com sucesso, e para iniciar o serviço, basta executar o comando informado na linha 4, o FireFly vai se encarregar de chamar o docker para realizar o download das imagens necessária para a criação dos containers e inicialização dos serviços, o retorno deve ser algo parecido com o log a seguir:

```
ff start h2v-stack
done
Web UI for member '0': http://127.0.0.1:5000/ui
Swagger API UI for member '0': http://127.0.0.1:5000/api
Sandbox UI for member '0': http://127.0.0.1:5109
To see logs for your stack run:
ff logs h2v-stack
```

Figura 8 – Retorno da resposta do FireFly.

Fonte: Autoria própria.

4.4.2 API

A criação de uma API personalizada para comunicação e transporte de dados entre os módulos do sistema tem o objetivo de criar *endpoints* com interfaces específicas para alguns casos de uso, como requisições para registrar identidades na rede, emitir certificados, ou verificar a autenticidade de um certificado.

A API será responsável por centralizar e padronizar o acesso aos dados e serviços do sistema. Dentre os principais recursos que serão desenvolvidos, destacam-se: o gerenciamento de usuários e empresas, a associação de dispositivos de produção com as empresas, a emissão e consulta de certificados digitais representados por tokens, a criação de *tokens non-fungibles* (NFTs), além da integração com a *blockchain* por meio de chamadas ao serviço do Hyperledger FireFly. No Quadro 3, estão listados os *endpoints* desenvolvidos nesta etapa com o método, rota e descrição de cada um deles.

Quadro 3 – Descrição das rotas implementadas na API REST. Fonte: Autoria própria

Método	Endpoint
GET	/identidades-bchain Descrição: Rota para buscar todas as identidades na blockchain.
GET	/usuarios Descrição: Rota para listar todos os usuários.
POST	/usuarios Descrição: Rota para criar um novo usuário.
GET	/usuarios/{uuid} Descrição: Rota para buscar um usuário filtrando pelo uuid.
PATCH	/usuarios/{id}/registrar-carteira Descrição: Rota para registrar a carteira do usuário na blockchain.
POST	/login Descrição: Rota para realizar o login do usuário.
POST	/token-pool Descrição: Rota para registrar um token diretamente na blockchain.
POST	/mint-token Descrição: Rota para efetivar valor ao token.
GET	/tokens Descrição: Rota para listar os tokens existentes na rede.
POST	/emitir-nft Descrição: Rota para emitir um NFT.
GET	/tokens/{id} Descrição: Rota para buscar um token pelo seu id(hash).
GET	/certificados/empresa/{id} Descrição: Rota para buscar os tokens que pertencem a uma empresa.
POST	/tipos-energia Descrição: Rota para criar um tipo de energia.
DELETE	/tipos-energia Descrição: Rota para deletar um tipo de energia.
GET	/tipos-energia Descrição: Rota para listar os tipos de energia.
GET	/tipos-energia/{id} Descrição: Rota para buscar um tipo de energia pelo id.
POST	/dispositivos-producao Descrição: Rota para registrar um dispositivo de produção.
GET	/dispositivos-producao Descrição: Rota para listar todos os dispositivos de produção.

Quadro 3 - Continuação

Método	Endpoint
DELETE	/dispositivos-producao
	Descrição: Rota para deletar um dispositivo de produção.
GET	/dispositivos-producao/{id}
	Descrição: Rota para buscar um dispositivo de produção pelo id.
PATCH	/dispositivos-producao/{id}/validar
	Descrição: Rota para validar um dispositivo de produção.
GET	/dispositivos-producao/situacao/{id}
	Descrição: Rota para listar por situação.
GET	/dispositivos-producao/situacao/{id_situacao}/empresa/{id_empresa}
	Descrição: Rota para listar filtrando por empresa e situação.
GET	/dispositivos-producao/empresa/{id_empresa}
	Descrição: Rota para listar filtrando por empresa.
GET	/empresas
	Descrição: Rota para listar todas as empresas.
POST	/empresas
	Descrição: Rota para criar uma nova empresa.
POST	/tipos-empresa
	Descrição: Rota para registrar um tipo de empresa.
GET	/tipos-empresa
	Descrição: Rota para listar os tipos de empresa.
DELETE	/tipos-empresa
	Descrição: Rota para deletar um tipo de empresa.
GET	/tipos-empresa/{id}
	Descrição: Rota para buscar tipo de empresa pelo id.
POST	/empresas/{id}/usuarios
	Descrição: Rota para associar um usuário a uma empresa.
DELETE	/empresas/{id}/usuarios/{id_usuario}
	Descrição: Rota para remover a associação de um usuário.
GET	/etapas
	Descrição: Rota para listar as etapas.
GET	/etapas/{id}
	Descrição: Rota para buscar uma etapa pelo id.
POST	/etapas
	Descrição: Rota para criar uma nova etapa.
DELETE	/etapas
	Descrição: Rota para deletar uma etapa.

Quadro 3 - Continuação

Método	Endpoint
POST	/grupos Descrição: Rota para criar um novo grupo.
GET	/grupos Descrição: Rota para listar os grupos.
GET	/grupos/{id} Descrição: Rota para buscar um grupo específico pelo id.
POST	/grupos/{id}/grupos-usuarios Descrição: Rota para associar usuário a um grupo.
DELETE	/grupos/{id}/grupos-usuarios Descrição: Rota para excluir associação de usuário a grupo.
POST	/lotes Descrição: Rota para registrar um novo lote de produção.
GET	/lotes Descrição: Rota para listar todos os lotes de produção.
GET	/lotes/{id_empresa} Descrição: Rota para listar lotes filtrando por empresa.
POST	/solicitacoes Descrição: Rota para registrar nova solicitação.
GET	/solicitacoes Descrição: Rota para listar todas as solicitações.
GET	/solicitacoes/{id} Descrição: Rota para listar uma solicitação pelo id.
POST	/solicitacoes/{id}/andamentos Descrição: Rota para registrar andamento da solicitação.
POST	/solicitacoes/{id}/avaliacoes Descrição: Rota para registrar avaliação da solicitação.
GET	/solicitacoes/{id}/avaliacoes Descrição: Rota para listar avaliações da solicitação.
GET	/solicitacoes/situacao/{id_situacao} Descrição: Rota para listar solicitações por situação.
PATCH	/solicitacoes/{id}/situacao Descrição: Rota para atualizar situação da solicitação.

Com a pilha do FireFly, criada e configurada, tem-se um ambiente pronto para dí inicio ao desenvolvimento da API, ela foi desenvolvida com TypeScript utilizando o framework Fastify. A API serve para atender às necessidades de comunicação entre

o *front-end* e o *back-end*. A parte prática do desenvolvimento da API conta com uma estruturação do projeto, primeiramente instalando as ferramentas.

Como mencionado no capítulo 3.3, uma base de dados foi necessária para o sistema. Para o gerenciamento do versionamento do schema, foi utilizada a ferramenta Prisma.js. O Prisma.js atua como uma camada de abstração entre a aplicação e o banco de dados, permitindo o gerenciamento do *schema* por meio de uma interface declarativa. Ele traduz o código para *Structured Query Language* (SQL), verifica o estado atual do banco e gera os scripts necessários para aplicar as modificações desejadas, além de manter um controle de versões do *schema* de forma integrada ao projeto da API.

Com o banco de dados devidamente instanciado, foi realizada a instalação do *Software Development Kit* (SDK) do FireFly, disponibilizado oficialmente pelos desenvolvedores do framework. Esse SDK facilita a integração da aplicação com os serviços da rede blockchain, permitindo chamadas e transações de forma simplificada. Em seguida, o ambiente da API foi configurado utilizando TypeScript em conjunto com o Fastify, um framework web de alta performance voltado para desenvolvimento com Node.js. A API foi estruturada para realizar a comunicação entre o *front-end*, o banco de dados e os serviços da rede FireFly. A seguir, será apresentado um exemplo de implementação de um dos endpoints desenvolvidos, com o objetivo de ilustrar a estrutura do projeto e a forma como as requisições são tratadas. O controle de versionamento do código-fonte foi feito utilizando a ferramenta Git, garantindo o acompanhamento das alterações realizadas durante o desenvolvimento da aplicação e facilitando a colaboração e rastreabilidade do projeto.

Para facilitar a visualização, o teste e a documentação dos endpoints desenvolvidos na API, foi realizada a integração com o Swagger, uma ferramenta amplamente utilizada para descrever e consumir APIs REST de forma interativa. A instalação foi feita por meio do pacote fastify-swagger, compatível com o framework Fastify utilizado no projeto. O Swagger permite a geração automática da documentação com base nos esquemas definidos nos próprios endpoints, garantindo que a interface da documentação se mantenha atualizada à medida que a API evolui.

A configuração incluiu a definição do título da aplicação, versão da API e a rota de acesso à interface web de documentação, geralmente acessível através do caminho /docs ou semelhante que apresenta a interface mostrada na Figura 9. Essa interface permite que qualquer usuário visualize os *endpoints* disponíveis, verifique os parâmetros de entrada e saída, e realize testes diretamente pelo navegador, o que facilita a validação e depuração durante o desenvolvimento. Com essa integração, a API passou a contar com uma documentação inicial, que pode ser expandida conforme a evolução do projeto.

Figura 9 – Interface Web do Swagger.

The screenshot shows the Swagger UI interface for the API H2V. At the top, it displays the API title "API H2V 1.0.0 OAS 3.0" and the URL "http://localhost:3000/docs/json". Below this, there's a "Servers" dropdown set to "http://localhost:3000". The main area is titled "default" and contains several API endpoints under the "Usuários" category. The "/usuarios" endpoint is highlighted with a green background, indicating it is the current focus. Other endpoints shown include "/identidades-bchain", "/usuarios", "/usuario/{uuid}", "/usuarios/{id}/registrar-carteira", and "/login". There are also sections for "Tokens" and other categories like "default" and "Tokens".

Fonte: Autoria Própria.

Foram realizados testes manuais da API desenvolvida para validação das funcionalidades essenciais do protótipo. Os testes foram conduzidos com as ferramentas Swagger e Insomnia como é apresentado na Figura 10, simulando requisições aos *endpoints*, verificando as respostas e validando o fluxo do sistema. A escolha por testes manuais se deve ao caráter experimental e exploratório da aplicação, cujo objetivo principal é demonstrar a viabilidade do uso de *blockchain* (via Hyperledger FireFly) na certificação de processos sustentáveis. Como trabalho futuro, sugere-se a adoção de testes automatizados para maior cobertura e manutenibilidade da aplicação.

Figura 10 – Requisição realizada no Insomnia ao endpoint /solicitacao da API, utilizando o método POST.

The screenshot shows the Insomnia REST Client interface. On the left sidebar, there is a list of API endpoints categorized by method (POST, GET, PATCH, etc.) and path. The "solicitacao" endpoint under the "POST" category is selected. The main panel shows a "POST" request to "localhost:3000/solicitacoes". The "Params" tab is active, showing a "URL PREVIEW" of "http://localhost:3000/solicitacoes". The "QUERY PARAMETERS" section has a single entry: "name" with value "value". The "PATH PARAMETERS" section is empty. Above the preview, the status bar shows "200 OK", "10.9 ms", "113 B", and "11 Days Ago". The preview pane shows a JSON response with fields like "id", "id_usuario", "id_empresa", "id_lote", "id_dispositivo_producao", "situacao", and "anexo".

Fonte: Autoria própria.

Para demonstrar a aplicação prática do sistema proposto e evidenciar a integração com a tecnologia *blockchain*, serão apresentados os códigos de dois *endpoints* que estão listados no Quadro 3. O Algoritmo 1 apresenta a implementação da rota responsável por registrar a carteira de um usuário na rede *Ethereum*, por meio da integração com o FireFly. A funcionalidade é acessada via uma requisição POST à rota /usuarios/:id/registrar-carteira, e espera como entrada a chave da carteira do usuário (chave) e o identificador da organização a que ele será vinculado (parent). Após validar a existência do usuário no banco de dados, o sistema realiza uma chamada à API do FireFly para criar uma identidade na rede. Em seguida, os dados retornados pela API como uuid, did e o hash da carteira são armazenados no banco, finalizando o processo de registro e integração do usuário com a infraestrutura *block-chain*.

Algoritmo 1 – Trecho de código responsável pelo registro de carteira de usuário na rede FireFly

```
1 app.patch('/usuarios/:id/registrar-carteira', {
2   schema: {
3     tags: ['Usuários'],
4     summary: 'Registrar carteira de usuário',
5     description: 'Atualiza o usuário com chave da carteira e registra na
6       rede FireFly',
7     body: {
8       type: 'object',
9       required: ['chave', 'parent'],
10      properties: {
11        chave: { type: 'string', description: 'Hash da carteira' },
12        parent: { type: 'string', description: 'Id da org ao qual o
13          usuario vai fazer parte' },
14      },
15    },
16    params: {
17      type: 'object',
18      required: ['id'],
19      properties: { id: { type: 'integer' } },
20    },
21    handler: async (request, reply) => {
22      try {
23        const { id } = request.params;
24        const { chave, parent } = request.body;
25
26        const usuarioDb = await app.prisma.usuarios.findUnique({
27          where: { id },
28          select: { usuario: true }
29        });
30
31        const fireflyResponse = await app.flyfire.createUser({
32          parent,
33          name: `Carteira ${chave}`,
34          did: `did:eth:1:${chave}`,
35          publicKey: `0x${chave}`,
36        });
37
38        const { did, address, pk } = fireflyResponse;
39
40        await app.prisma.usuarios.update({
41          where: { id },
42          data: { carteiraDid: did, carteiraAddress: address, carteiraPk: pk }
43        });
44
45        reply.status(201).json(fireflyResponse);
46      } catch (error) {
47        reply.status(500).json({ error: 'Erro ao registrar a carteira' });
48      }
49    }
50  }
```

```
29
30     if (!usuarioDb) {
31         return reply.status(404).send({ message: 'Usuário não encontrado.' });
32     }
33
34     const params = {
35         name: usuarioDb.usuario,
36         description: 'Identidade gerada via API',
37         type: 'custom',
38         parent: parent,
39         key: chave
40     };
41
42     const fireflyResponse = await firefly.createIdentity(params);
43
44     const usuario = await app.prisma.usuarios.update({
45         where: { id },
46         data: {
47             hashCarteira: chave,
48             uuid: fireflyResponse.id,
49             did: fireflyResponse.did,
50         },
51     });
52
53     return reply.status(200).send({
54         message: 'Usuário atualizado e registrado no FireFly com sucesso.',
55         fireflyResponse,
56         usuario: {
57             id: usuario.id,
58             usuario: usuario.usuario,
59             email: usuario.email,
60             uuid: usuario.uuid,
61             did: usuario.did,
62             hashCarteira: usuario.hashCarteira,
63         },
64     });
65
66 } catch (err) {
67     console.error('Erro ao registrar carteira:', err);
68     return reply.status(500).send({ error: 'Erro interno ao registrar
69         carteira.' });
70 }
71});
```

O Algoritmo 2 apresenta a implementação do *endpoint* responsável pela emis-

são de um NFT, que representa digitalmente a certificação do hidrogênio verde. A funcionalidade é acessada por meio de uma requisição POST à rota /emitir-nft, que recebe como parâmetros o identificador da solicitação de avaliação, o identificador do usuário autenticado e o do órgão emissor. O sistema recupera os dados da avaliação no banco de dados, gera um *Token Pool* do tipo *non-fungible*(não-fungível), e em seguida realiza a emissão do NFT através de uma requisição à API do FireFly, vinculando o *token* à carteira digital do solicitante. O id do certificado gerado é então armazenado em uma tabela própria no banco de dados, associando as informações de emissão à avaliação e à empresa solicitante.

Algoritmo 2 – Trecho de código responsável pelo emissão de um NFT.

```

1 app.post('/emitir-nft', {
2   schema: {
3     tags: ['Tokens'],
4     summary: 'Emitir NFT',
5     description: 'Cria um Token Pool para NFTs e mint um NFT na
6       blockchain.',
7     body: {
8       type: 'object',
9       required: ['id_solicitacao', 'usuario_logado', 'orgao_emissor'],
10      properties: {
11        id_solicitacao: {
12          type: 'integer',
13          title: 'ID da solicitação'
14        },
15        usuario_logado: {
16          type: 'integer',
17          title: 'ID do usuário logado'
18        },
19        orgao_emissor: {
20          type: 'integer',
21          title: 'ID do órgão emissor'
22        }
23      },
24      response: {
25        200: {
26          type: 'object',
27          properties: {
28            poolId: { type: 'string' },
29            token: { type: 'object' },
30            recipient: { type: 'string' }
31          }
32        }
33      }
34    },

```

```
35  handler: async (request, reply) => {
36    try {
37      interface Body {
38        id_solicitacao: number;
39        usuario_logado: number;
40        orgao_emissor: number;
41      }
42
43      const {
44        id_solicitacao,
45        usuario_logado,
46        orgao_emissor
47      } = request.body as Body;
48
49      const hoje = new Date();
50      const dataGeracao =
51        `${hoje.getFullYear()}-
52        ${(hoje.getMonth() + 1).toString().padStart(2, '0')}-
53        ${hoje.getDate().toString().padStart(2, '0')}-
54        ${hoje.getHours().toString().padStart(2, '0')}-
55        ${hoje.getMinutes().toString().padStart(2, '0')}-
56        ${hoje.getSeconds().toString().padStart(2, '0')}`;
57
58      const nome = `nfth2v_${id_solicitacao}_${dataGeracao}`;
59      const simbolo = `H2V${id_solicitacao}`;
60
61      //Busca os metadados da solicitação com o Prisma
62      const avaliacao = await app.prisma.avaliacao.findFirst({
63        where: {
64          id_solicitacao
65        },
66        include: {
67          solicitacao: {
68            include: {
69              usuario: true
70            }
71          }
72        }
73      );
74
75      if (!avaliacao || !avaliacao.solicitacao?.usuario?.hashCarteira) {
76        return reply.status(404).send({
77          error: 'Avaliação ou carteira do usuário não
78            encontrada para essa solicitação.'
79        });
80      }
81    }
```

```
82     const para = avaliacao.solicitacao.usuario.hashCarteira;
83
84     const metadata = {
85         id_solicitacao: avaliacao.id_solicitacao,
86         data_avaliacao: avaliacao.data_avaliacao,
87         dispositivo_producao: avaliacao.dispositivo_producao,
88         tamanho_producao: avaliacao.tamanho_producao,
89         anexo_relatorio: avaliacao.anexo_relatorio
90     };
91
92     //Cria o Token Pool
93     const pool = await firefly.createTokenPool(
94     {
95         name: nome,
96         symbol: simbolo,
97         type: 'nonfungible',
98         config: { blockNumber: '0' }
99     },
100     { publish: true }
101 );
102
103     const response = await fetch(
104         'http://localhost:5000/api/v1/namespaces/default/tokens/mint',
105     {
106         method: 'POST',
107         headers: { 'Content-Type': 'application/json' },
108         body: JSON.stringify({
109             pool: pool.name,
110             to: para,
111             amount: '1',
112             data: [
113                 {
114                     value: JSON.stringify(metadata)
115                 } //ou metadata se já for uma string
116             ]
117         })
118     );
119
120     const nft = await response.json();
121
122     await app.prisma.certificados_emitidos.create({
123         data: {
124             id_avaliacao: avaliacao.id,
125             emitido_por: usuario_logado,
126             hash_certificado: pool.id,
127             id_empresa: avaliacao.solicitacao.id_empresa,
128             empresa_emissora: orgao_emissor
```

```
129         }
130     });
131
132     reply.send({
133         poolId: pool.id,
134         token: nft,
135         recipient: para
136     });
137
138 } catch (error) {
139     console.error("Erro ao emitir NFT:", error);
140     reply.status(500).send({ error });
141 }
142 }
143});
```

Esses exemplos comprovam a utilização efetiva da rede *blockchain* por meio do Hyperledger FireFly como camada de abstração e comunicação com a rede Ethereum.

4.5 Front-End

O *front-end* foi desenvolvido utilizando Flutter, visando criar uma interface intuitiva a princípio, que atenda às necessidades de certificação do H₂V. O Flutter foi escolhido por sua capacidade de desenvolver aplicações responsivas e de fácil manutenção, garantindo a integração eficiente com a API desenvolvida.

4.5.1 Titular de Conta

O módulo destinado ao ator “Titular de Conta” foi criado para atender às necessidades dos produtores de H₂V, permitindo que realizem o cadastro da empresa, registrem lotes de produção, dispositivos utilizados e acompanhem a certificação de seus produtos. Durante o desenvolvimento, buscou-se aliar usabilidade e completude funcional, considerando os requisitos identificados na etapa de levantamento de requisitos no Capítulo 4.2.

Ao longo da implementação, observou-se a necessidade de estruturar a jornada do usuário de forma intuitiva. Para isso, o fluxo foi organizado em etapas: primeiro, o cadastro do titular de conta e autenticação; em seguida, os registros de dispositivos e lotes de produção; por fim, a solicitação e acompanhamento dos certificados.

O formulário de cadastro de titular de conta na Figura 11, por exemplo, coleta informações essenciais da empresa produtora(nome, CNPJ, endereço), que são posteriormente associadas às solicitações de certificação. Esta abordagem permite rastre-

Figura 11 – Tela de registro de titular de conta.

Fonte: Autoria Própria.

abilidade e vinculação direta entre os lotes certificados e a entidade responsável por sua produção.

O mesmo se aplica ao registro de dispositivos de produção na Figura 12 e lotes de produção na Figura 13, cujas informações são armazenadas pelo sistema antes de permitir o envio da solicitação ao órgão certificador.

Figura 12 – Tela de registro de dispositivos de produção.

Fonte: Autoria Própria.

Figura 13 – Tela de registro de lotes de produção.

Fonte: Autoria Própria.

A tela de solicitação de certificados na Figura 14 foi projetada para integrar os dados cadastrados anteriormente, exigindo que o usuário selecione um lote e um dispositivo, além de anexar um documento técnico de comprovação. Essa etapa, embora aparentemente simples, demandou atenção quanto à verificação de integridade dos dados enviados, garantindo que os certificados emitidos estejam associados a registros válidos.

Figura 14 – Tela de Solicitação de certificado.

Fonte: Autoria Própria.

No tocante à transparência e confiabilidade do sistema, destaca-se a funcionalidade de verificação pública de autenticidade dos certificados emitidos que pode ser visualizada na Figura 15. Qualquer interessado pode verificar, por meio do hash de um certificado, se aquele registro corresponde a um dado armazenado na *blockchain*, assegurando a imutabilidade e confiabilidade do processo.

Figura 15 – Tela de verificação de autenticidade de certificado.



Fonte: Autoria Própria.

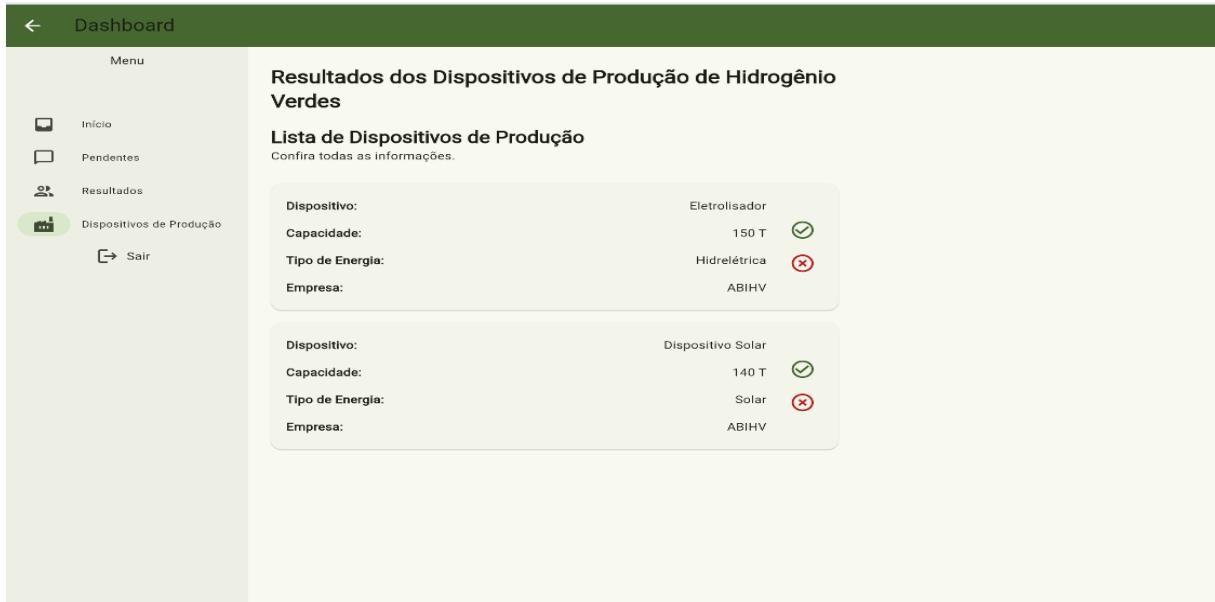
4.5.2 Órgão Emissor

A interface de usuário do módulo do referente ao ator “Órgão Emissor” é composta por telas que permitem a visualização de dispositivos de produção pendentes de aprovação, a tela de solicitações que ainda estão em análise e uma tela de avaliação da solicitação submetida pelo titular de conta.

É apresentado na Figura 16, a tela de aprovação de dispositivos de produção cadastrados previamente pelo titular de conta, os dispositivos cadastrados ficam em uma fila aguardando avaliação por parte do órgão emissor no sistema a fim de manter uma verificação nos dispositivos validados.

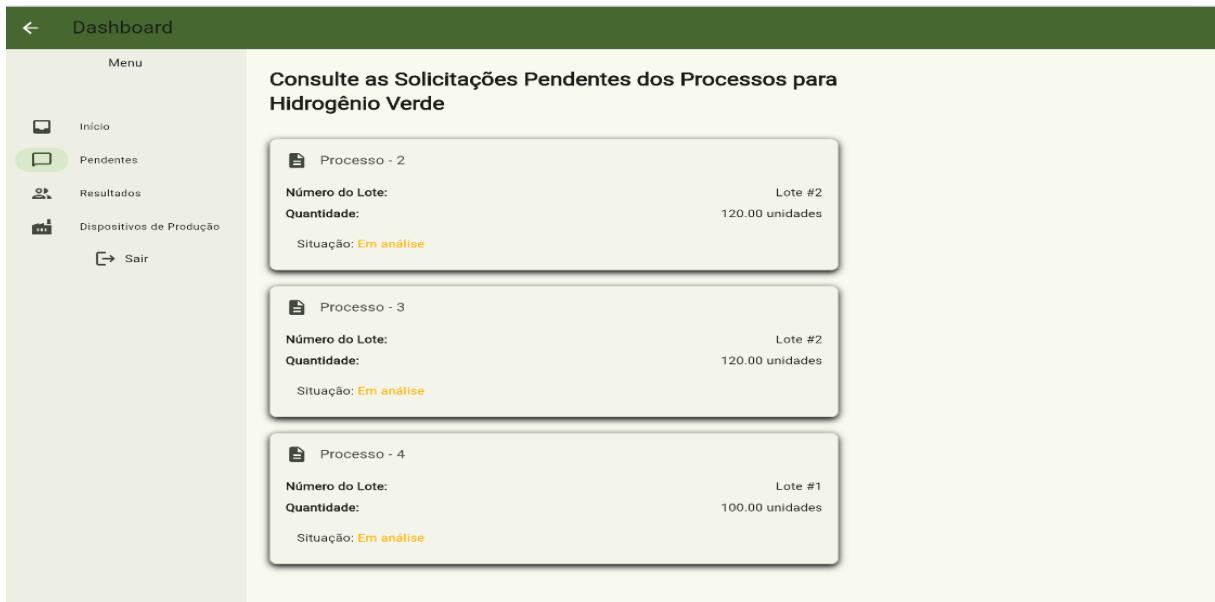
É apresentado na Figura 17 a interface que lista as solicitações que ainda estão em análise para que o órgão emissor tenha acesso a cada uma delas e consiga acessar a interface da Figura 18.

Figura 16 – Tela de aprovação de dispositivos de produção.



Fonte: Autoria Própria.

Figura 17 – Tela de visualização das solicitações em análise.



Fonte: Autoria Própria.

A Figura 18, mostra uma tela ao órgão emissor com informações relacionadas a uma solicitação específica, como detalhes do dispositivo e lote de produção associados a essa solicitação, também apresenta o documento anexado pelo titular de conta e um relatório de auditoria(seção para implementação futura). Com base nessas informações o usuário associado ao órgão emissor tomará a decisão de aprovar ou rejeitar a solicitação do titular de conta, sendo aprovada um alerta de confirmação é mostrado

informando o status para o usuário, caso seja rejeitada, um alerta também é mostrado.

Figura 18 – Tela de avaliação final de solicitação.

The screenshot shows a mobile application interface for 'Dashboard'. On the left, a vertical menu includes 'Início', 'Pendentes', 'Resultados' (which is highlighted in green), and 'Dispositivos de Produção'. Below these are 'Sair' and a back arrow. The main content area has a title 'Resultados dos Processos Certificados de Hidrogênio Verde'. It displays details for a 'Solicitação ID: 1' (Details confirmed) involving a 'Reator Solar H2-Green' device. The device's capacity is 10 units, it uses 'Solar' energy, and its status is 'Aprovado'. A production batch 'Lote de Produção ID: 1' is shown with a quantity of 100 kg and 5 kgCO₂eq emissions, occurring from 01/05/2025 03:00 to 21/05/2025 03:00. It includes a PDF audit report ('auditoria.pdf') and a document ('documento.pdf'). At the bottom are three buttons: 'Aprovar' (Approve), 'Rejeitar' (Reject), and 'Nova Auditoria' (New Audit).

Fonte: Autoria Própria.

5 AVALIAÇÃO

Como apresentado na metodologia, a avaliação foi conduzida por meio de heurísticas. O resultado esperado da avaliação é uma lista de problemas de usabilidade, nem sempre é viável resolver todos os problemas e devido a isso eles deveram ser priorizados de acordo com a escala de severidade mostrada na Tabela 4. As prioridades são melhor definidas com base em dados experimentais sobre o impacto dos problemas no desempenho do usuário, mas, às vezes, faz-se necessário contar apenas com intuições. Os resultados da avaliação heurística não apenas identificarão os problemas de usabilidade, mas também fornecerão contribuições para melhorias no design da interface (Nielsen, 1994).

Quadro 4 – Escala de severidade.

Identificador	Severidade	Descrição
S01	Problema cosmético ou superficial	Problema normalmente estético, corrigido quando se tem tempo sobrando.
S02	Problema de usabilidade pequeno	Problema que pode ser contornado pelo usuário e possui baixa severidade para ser solucionado.
S03	Problema de usabilidade grande	Problema com impacto considerado alto na funcionalidade do sistema e por isso deve ser solucionado rápido.
S04	Problema de usabilidade crítico	Problema crítico que deve ser solucionado antes do lançamento do sistema.

Fonte: Nielsen (Nielsen, 1994)

Com base nos problemas de usabilidade identificados, que são apresentados no Quadro 5, algumas melhorias são fortemente recomendadas para aprimorar a experiência do usuário no sistema HydroChain. Primeiramente, é necessário ajustar o contraste do *label* do menu superior em toda a interface, garantindo acessibilidade visual conforme diretrizes de design minimalista (H04). Elementos como botões e caixas de lista devem seguir um padrão visual consistente (H02), utilizando o mesmo estilo de bordas, sombras e alinhamentos, o que evita confusão e reforça a previsibilidade do sistema. Na tela de listagem de Dispositivos de produção ao “Órgão Emissor” conforme é mostrado na Figura 16, a realocação e o agrupamento claro dos botões relacionados às informações específicas são essenciais para melhorar o reconhecimento

Quadro 5 – Análise heurística do sistema.

Problema	Violado	Severidade	Descrição
Contraste do <i>label</i> no layout geral do tema.	H04	S02	O <i>label</i> está com um texto na cor escura em um fundo que também tem cor escura.
Botão “sair” do layout geral do tema em posição fora do padrão	H02	S01	O botão está centralizado, em um contexto onde os elementos próximos dele estão alinhados à esquerda.
Caixas de lista fora do padrão.	H02	S01	As caixas estão ferindo o padrão, sendo algumas com sombra e outras sem sombra.
Botões da caixas na tela de lista de DPs mal posicionados.	H08	S02	Não está claro ao que os botões estão relacionados, se é à caixa inteira ou à informação específica.
Ausência de Flexibilidade	H03	S01	A aplicação não fornece atalhos ou formas de eficiência de uso.
Ausência de mecanismos para prevenção de erros.	H09	S04	Funcionalidades têm que ser executadas em sequência e não há bloqueios para que a sequência correta seja garantida.

Fonte: Autoria própria.

de elementos (H08). A ausência de flexibilidade (H03) também deve ser tratada com a introdução de atalhos ou opções rápidas que otimizem a navegação para usuários experientes. Por fim, é prioritário implementar mecanismos para prevenir erros (H09), como a validação automática de etapas obrigatórias e alertas contextuais que impeçam a sequência incorreta de ações.

Embora as melhorias descritas sejam fortemente recomendadas, limitações de tempo impediram sua implementação na presente etapa do desenvolvimento. Ainda assim, os problemas foram mapeados e organizados conforme suas respectivas heurísticas e severidades, o que possibilita uma priorização mais clara em eventuais ciclos de refinamento. A sistematização dessas observações oferece uma base para futuras decisões de design, com foco na consistência, acessibilidade e eficiência do Hydro-Chain.

6 CONSIDERAÇÕES FINAIS

A realização deste Trabalho de Conclusão de Curso teve como objetivo geral o desenvolvimento do protótipo HydroChain, voltado à certificação de hidrogênio verde de forma digital com *blockchain*. Para atingir esse objetivo, foram conduzidas atividades estruturadas conforme os objetivos específicos do projeto. Como resultado, obteve-se uma aplicação funcional que representa um passo inicial em direção à digitalização do processo de certificação de hidrogênio verde, contribuindo com uma proposta de solução tecnológica para um tema emergente no cenário energético e ambiental.

No que se refere aos objetivos específicos apresentados no Capítulo 1.2.2, dos três estabelecidos, dois foram plenamente alcançados e um foi parcialmente atendido. O sistema desenvolvido contempla a integração com a tecnologia *blockchain* e permite a emissão de certificados conforme a finalidade proposta. No entanto, a implementação completa do modelo de certificação não foi possível, uma vez que a ausência do módulo de auditoria compromete diretamente esse aspecto.

As figuras apresentadas no Capítulo 4 demonstram que as interfaces foram efetivamente implementadas e que os requisitos funcionais definidos foram alcançados. Em especial, as Figuras 14 e 15 evidenciam o atendimento a dois dos principais requisitos do sistema: a possibilidade de realizar solicitações de certificação e a verificação da autenticidade dos certificados emitidos. O objetivo específico de emitir um certificado digital foi atingido como é possível verificar na figura 18, no momento em que a solicitação é aprovada, o NFT é gerado e entregue à carteira do “*Titular de conta*”.

Mais do que atender aos requisitos levantados, a implementação contribui para validar o modelo proposto. A utilização da tecnologia *blockchain* por meio do FireFly permitiu que os certificados emitidos fossem dotados de autenticidade, rastreabilidade e resistência à falsificação. Esses atributos estão presentes em cada etapa de interação do titular de conta com o sistema, reforçando o potencial da solução como uma alternativa viável para processos digitais de certificação sustentáveis.

Diante do atual cenário energético relacionado ao hidrogênio verde (H_2V) no Brasil, este trabalho configura-se como um ponto de partida relevante para discussões sobre a certificação da origem e sustentabilidade dessa fonte energética. A iniciativa aqui apresentada abre caminho para avanços e investigações futuras, que poderão ser determinantes para a consolidação de um mercado de H_2V no país, mais transparente e tecnologicamente robusto.

6.1 Limitações

Embora tenha-se utilizado *frameworks* como o FireFly (Foundation, 2024) para implementar a tecnologia *blockchain* no protótipo, a escassez de documentação traduzida e a complexidade de aplicação dessas ferramentas em contextos específicos resultaram em uma curva de aprendizado mais lenta do que o esperado. Além disso, durante a etapa de codificação, surgiram limitações técnicas relacionadas a *bugs* e comportamentos inesperados das ferramentas empregadas, cuja resolução foi dificultada pela falta de fontes de consulta e exemplos práticos disponíveis na comunidade.

Outra limitação encontrada foi a necessidade de utilizar uma rede pública, neste caso, a Ethereum. Inicialmente, considerou-se a adoção de uma rede *blockchain* permissionada, no entanto, devido à maior complexidade envolvida em sua configuração e operação, a curva de aprendizado exigida para integrá-la ao HydroChain se mostrou incompatível com o tempo disponível para o desenvolvimento deste trabalho.

6.2 Trabalhos Futuros

Partindo do fato de que o escopo geral não foi implementado neste trabalho, ficam algumas sugestões de trabalhos futuros para darem continuidade ao tema. As sugestões de trabalhos futuros estão enumeradas a seguir:

1. Implementação do módulo de auditoria, permitindo o registro e a verificação de eventos importantes do ciclo de certificação, como alterações de dados, aprovações e validações.
2. Realização de testes de usabilidade com usuários reais (*stakeholders* do setor energético ou certificadores), a fim de validar a interface e obter *feedback* mais robusto para melhorias.
3. Aprimoramento do desempenho do protótipo por meio da realização de testes automatizados de carga, visando avaliar a escalabilidade e a estabilidade da aplicação em diferentes condições de uso.
4. Realização de um estudo para a implementação de um modelo de negociação dos certificados emitidos, com base em conceitos de *criptoativos*, permitindo a *tokenização* e a comercialização desses certificados em plataformas digitais.
5. Implementar o HydroChain utilizando uma rede permissionada.

REFERÊNCIAS

- AMENTA, C.; SANSEVERINO, E. R.; STAGNARO, C. Regulating blockchain for sustainability? the critical relationship between digital innovation, regulation, and electricity governance. *Energy Research Social Science*, v. 76, p. 102060, 2021. ISSN 2214-6296. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214629621001535>>. Citado 3 vezes nas páginas 22, 23 e 26.
- ANDERSON, D. J. Kanban: mudança evolucionária de sucesso para seu negócio de tecnologia. Sequim, WA: Blue Hole Press, 2011. Citado na página 27.
- ATTEYA, A. I. et al. A comprehensive review on the potential of green hydrogen in empowering the low-carbon economy: development status, ongoing trends and key challenges. *Green energy and environmental technology*, InTechOpen, v. 2, 2023. Citado na página 15.
- BARBOSA, S. D. J. et al. Interação humano-computador e experiência do usuário. Auto publicação, 2021. Citado na página 28.
- CERTIFHY. CertifHy: Europe's Leading Guarantee of Origin Scheme for Hydrogen. 2024. <<https://www.certifhy.eu>>. Accessed: 2024-10-08. Citado 4 vezes nas páginas 19, 20, 26 e 30.
- CHRYSIKOPOULOS, S. K. et al. Green certificates research: Bibliometric assessment of current state and future directions. *Sustainability*, v. 16, n. 3, 2024. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/16/3/1129>>. Citado 4 vezes nas páginas 22, 23, 26 e 30.
- De Filippi, P.; MANNAN, M.; REIJERS, W. Blockchain as a confidence machine: The problem of trust challenges of governance. *Technology in Society*, v. 62, p. 101284, 2020. ISSN 0160-791X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0160791X20303067>>. Citado 2 vezes nas páginas 23 e 26.
- ELSHAFEI, A. M.; MANSOUR, R. Green hydrogen as a potential solution for reducing carbon emissions: a review. *Journal of Energy Research and Reviews*, v. 13, n. 2, p. 1–10, 2023. Citado na página 18.
- FOUNDATION, H. Hyperledger FireFly Documentation. 2024. Accessed: 2024-10-09. Disponível em: <<https://hyperledger.github.io/firefly>>. Citado 3 vezes nas páginas 33, 34 e 55.
- GALE, F. et al. Renewable hydrogen standards, certifications, and labels: A state-of-the-art review from a sustainability systems governance perspective. *International Journal of Hydrogen Energy*, Elsevier, v. 59, p. 654–667, 2024. Citado na página 15.
- HASSAN, Q. et al. Green hydrogen: A pathway to a sustainable energy future. *International Journal of Hydrogen Energy*, Elsevier, v. 50, p. 310–333, 2024. Citado na página 15.

- KRUCHTEN, P. The Rational Unified Process: An Introduction. 3rd. ed. Boston: Addison-Wesley, 2004. ISBN 978-0-321-19770-2. Citado na página 27.
- Linux Containers. Linux Containers – LXC/LXD. 2025. <<https://linuxcontainers.org/>>. Acesso em: 11 de junho de 2025. Citado na página 28.
- MANI, M. et al. Paris climate agreement and the global economy: winners and losers. Washington, DC: The World Bank, 2018. Citado na página 15.
- MARTINS, M. d. R. de F. et al. Construindo aplicações baseadas em blockchain com o hyperledger firefly. Citado 3 vezes nas páginas 20, 22 e 34.
- MOLINA-JIMENEZ, C. et al. On and off-blockchain enforcement of smart contracts. In: MENCAGLI, G. et al. (Ed.). Euro-Par 2018: Parallel Processing Workshops. Cham: Springer International Publishing, 2019. p. 342–354. ISBN 978-3-030-10549-5. Citado na página 27.
- MOULD, K. et al. A comparative analysis of biogas and hydrogen, and the impact of the certificates and blockchain new paradigms. International Journal of Hydrogen Energy, Elsevier, v. 47, n. 93, p. 39303–39318, 2022. Citado 2 vezes nas páginas 16 e 19.
- NIELSEN, J. Usability engineering. San Francisco, CA: Morgan Kaufmann, 1994. Citado 4 vezes nas páginas 25, 28, 29 e 52.
- SAMPAIO, R. R.; SANTOS, A. R. Certificação de Hidrogênio Verde: um metamodelo e uma especificação de requisitos. 2024. Documento técnico / Artigo do Laboratório de Engenharia de Software - UESPI. Citado 2 vezes nas páginas 26 e 31.
- SCHWABER, K.; BEEDLE, M. Agile software development with Scrum. Upper Saddle River, NJ: Prentice Hall PTR, 2001. Citado na página 27.
- TAHERDOOST, H. Smart contracts in blockchain technology: A critical review. Information, v. 14, n. 2, 2023. ISSN 2078-2489. Disponível em: <<https://www.mdpi.com/2078-2489/14/2/117>>. Citado na página 22.
- YAGA, D. et al. Blockchain technology overview. arXiv preprint arXiv:1906.11078, 2019. Citado na página 21.