**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**THIAGO RIBEIRO BAIZI DIAS FURTADO**

# PRESCRIÇÃO MÉDICA ELETRÔNICA COM VERIFICAÇÃO DESCENTRALIZADA UTILIZANDO BLOCKCHAIN

**PATO BRANCO**

**2023**

**THIAGO RIBEIRO BAIZI DIAS FURTADO**

# PRESCRIÇÃO MÉDICA ELETRÔNICA COM VERIFICAÇÃO DESCENTRALIZADA UTILIZANDO BLOCKCHAIN

# ELECTRONIC MEDICAL PRESCRIPTION WITH VERIFICATION DECENTRALIZED USING A BLOCKCHAIN

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia da Computação do Curso de Bacharelado em Engenharia da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Me. Adriano Serckumecka

**PATO BRANCO**

**2023**

**THIAGO RIBEIRO BAIZI DIAS FURTADO**

# PRESCRIÇÃO MÉDICA ELETRÔNICA COM VERIFICAÇÃO DESCENTRALIZADA UTILIZANDO BLOCKCHAIN

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia da Computação do Curso de Bacharelado em Engenharia da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 30/dezembro/2023

---

Adriano Serckumecka
Mestre
Universidade Tecnológica Federal do Paraná

---

Fabio Favarim
Doutor
Universidade Tecnológica Federal do Paraná

---

Luis Cassiano Goularte Rista
Mestre
Universidade Tecnológica Federal do Paraná

**PATO BRANCO**

**2023**

Gostaria de dedicar este trabalho à Deus e minha família, pois graças aos esforços de todos, tive sustento e base para a realização deste trabalho nesta difícil etapa de minha vida acadêmica.

## AGRADECIMENTOS

Primeiramente agradecer à Deus, por me ajudar a ultrapassar todos os obstáculos ao longo do árduo caminho. À minha mãe, Elizabeth Ribeiro Baizi Furtado, meu pai, Silvio Dias Furtado, e minha irmã, Karine Ribeiro Baizi Dias Furtado, por me confortar com palavras, gestos e ações nos mais difíceis momentos deste ano, trazendo paz e conforto enquanto escrevendo e desenvolvendo o projeto. Sem seus ensinamentos e palavras de carinho não conseguiria ter forças para enfrentar tantas lutas diárias. Gostaria de agradecer ao meu orientador, Adriano Serckumecka, e meus colegas de turma, Igo Brasil e Edinei Martinello, pelo suporte neste complexo tema. Independente da diferença de horário, sempre estiveram aptos à buscar soluções aos meus questionamentos, oferecendo suporte sempre que possível. Agradecer à banca examinadora, composta pelos professores Dr. Fabio Favarim e Me. Luis Cassiano Goularte Rista, pelos comentários e dicas realizados no momento de avaliação, pois foram de suma importância para a entrega da proposta.

“O sucesso é a soma de pequenos esforços
repetidos dia após dia.” Robert Collier

## RESUMO

Este trabalho realiza uma análise das prescrições médicas, examinando sua evolução histórica e atual utilização. É explorado o potencial para serem integradas novas tecnologias, como *Blockchain* e *Smart Contract*, para aprimorar esses sistemas. A pesquisa complementa no desenvolvimento de um sistema completo, compreendendo uma interface *web* e conectividade com uma rede *Blockchain* descentralizada para a inserção de prescrições médicas eletrônicas. Este trabalho realiza um estudo dedicado à tecnologia *Blockchain* e seus *frameworks*, uma estrutura que serve de suporte para a construção do código, visando uma compreensão mais ampla e aprofundada da eficácia dessas tecnologias no contexto específico das prescrições médicas. A conclusão aborda a versão final do sistema, destacando suas vantagens e desvantagens de forma crítica. O estudo oferece uma percepção sobre as transformações tecnológicas no contexto das prescrições médicas, destacando as oportunidades e desafios associados à implementação de soluções inovadoras nesse campo.



**Palavras-chave:** *blockchain*; *smart contract*; prescries micas.

**ABSTRACT**

This work conducts a comprehensive analysis of medical prescriptions, examining their historical evolution and current usage. The potential integration of new technologies, such as Blockchain and Smart Contracts, to enhance these systems is explored. The research culminates in the development of a complete system, comprising a web interface and connectivity with a decentralized Blockchain network for the insertion of electronic medical prescriptions. The study delves into a dedicated examination of Blockchain technology and its frameworks, a structure that provides support for code construction, aiming for a broader and deeper understanding of the effectiveness of these technologies in the specific context of medical pres- criptions. The conclusion addresses the final version of the system, critically highlighting its advantages and disadvantages. The study provides insight into technological transformations in the context of medical prescriptions, emphasizing the opportunities and challenges associated with implementing innovative solutions in this field.

**Keywords:** *blockchain*; *smart contract*; online prescriptions.

## LISTA DE FIGURAS

## LISTA DE ABREVIATURAS E SIGLAS

### Abreviaturas

| | |
|---|---|
| ANVISA | Agência Nacional de Vigilância Sanitária |
| CELEPAR | Companhia de Tecnologia da Informação e Comunicação do Paraná |
| ITI | Instituto Nacional de Tecnologia da Informação |
| ICP-Brasil | Infraestrutura de Chaves Públicas Brasileira |
| GSUS | Sistema de Gestão da Assistência de Saúde do SUS |
| CFF | Conselho Federal de Farmácia |
| CFM | Conselho Federal de Medicina |
| CPF | Cadastro de Pessoa Física |
| *PoW* | *Proof-of-Work* |
| *PoS* | *Proof-of-Stake* |

### Siglas

| | |
|---|---|
| PDF | *Portable Document Format* |
| HTML | Linguagem de Marcação de Hipertexto |
| JPG | *Joint Photographic Experts Group* |
| JS | linguagem de programação *Javascript* |
| CPF | Comprovante de Pessoa Física |
| HTTP | *Hypertext Transfer Protocol* |

**SUMÁRIO**

# 1 INTRODUÇÃO

A constante evolução do mundo digital tem assumido um papel significante para a população em seu dia-a-dia, tanto na atuação profissional quanto pessoal. Tarefas antes complexas de serem executadas, tornaram-se facilitadas com a colaboração do avanço tecnológico.

A medicina tem evoluído muito com o passar dos anos, trazendo consigo medicamentos novos, diversos tipos de tratamentos e, mais recentemente, as consultas *online*. A telessaúde e consultas remotas, permitem acesso eficiente aos cuidados da saúde, e oferecem melhor coordenação de cuidados e resultados de tratamento (Ahmad *et al.*, 2021).

Além disso, com o agravante da pandemia de COVID-19, órgãos como o Conselho Federal de Medicina e o Conselho Federal de Farmácia, trabalharam juntos para uma expansão das consultas *online*, permitindo que médicos prescrevessem receitas médicas, com assinaturas digitais, válidas em todo o território nacional, de acordo com leis e normas impostas, gerando segurança no momento da compra de remédios.

Entretanto, as prescrições digitais (nomeadamente prescrições em formato de arquivo eletrônico como JPG, PDF e HTML) estão em estágios iniciais quanto à sua utilização neste formato (He *et al.*, 2019), ocorrendo casos onde farmácias demandem que pacientes imprimam a receita médica, de forma à prevenir os vários tipos de fraude que podem ocorrer no sistema atual. Dados estão propensos a serem comprometidos, uma vez que são passíveis de ataques cibernéticos, além de adulterações em prescrições médicas enviadas por e-mail. Existem atualmente projetos e ferramentas que realizam essa verificação e validação *online* da receita médica, previnindo possíveis falhas humanas e possíveis fraudes relacionadas a criação de receitas falsas ou duplicações de receitas reais, como é o caso do sistema atual realizado pelo Conselho Federal de Medicina em sei *site* oficial.

Outra forma da realização dessa verificação e validação, utilizando ferramentas e serviços diferentes do que temos nos dias atuais, seria a inclusão de uma rede *Blockchain*. Essa arquitetura ainda é pouco utilizada no ambiente médico, mas vem se mostrando de grande utilidade em várias aplicações devido ao seu alto grau de segurança, rastreabilidade e confiabilidade quanto à proteção de dados.

Alinhada com os *Smart Contracts*, a *Blockchain* vem sendo utilizada em colaboração com diversas frentes, como mundo corporativo, bancos, entre outras. Essa nova tecnologia traz mais segurança tanto para o médico que prescreve a receita, quanto ao farmacêutico que a validará quando o cliente efetuar a compra do medicamento. Por possuir em seu sistema uma forma de validação mais eficaz, descentralizada e *online*, diminui a necessidade de receitas físicas. Trata-se, também, de um sistema que preza pela segurança dos dados em seu formato *ledger* público, uma espécie de livro-razão, que armazena seus dados e permite sua visualização, porém com a segurança de que os dados uma vez inclusos, não poderão ser alterados.

Realizando a unificação de um sistema de prescrições *online* com a *Blockchain* e os *Smart Contracts*, um novo sistema, seguindo os moldes atuais, foi criado. A arquitetura proposta

possui uma interface por meio de um *website*, similar com o modelo utilizado atualmente. Sua principal diferença se dá nos níveis de segurança que uma rede *Blockchain* pode oferecer, além da confiabilidade e fácil rastreabilidade dos dados, após estarem inseridos em sua rede.

## 1.1 OBJETIVOS

Os objetivos propostos dividem-se em objetivo geral, que representam os objetivos finais da proposta, e objetivos específicos, que explicita os objetivos pontuais para o qual a seguinte proposta pretende atingir ao longo do desenvolvimento do trabalho, até sua versão final.

### 1.1.1 OBJETIVO GERAL

Esta proposta tem como finalidade propor um sistema que permita ao médico prescrever uma receita digital ao paciente, de forma verificável e descentralizada, com o uso de uma *Blockchain* e *Smart Contracts*, tornando possível ao paciente utilizá-la nas farmácias sem a necessidade de impressão ou envio para o farmacêutico, impedindo também a reutilização das prescrições, evitando fraudes.

### 1.1.2 OBJETIVOS ESPECÍFICOS

- Análise de requisitos e modelagem da arquitetura para um sistema de prescrição *online*.
- Implementação do sistema de prescrição utilizando a *Blockchain* escolhida e *Smart Contracts* para verificação descentralizada.
- Validação do sistema através da realização de testes de funcionamento da rede e utilização pelo usuário.

## 1.2 JUSTIFICATIVA

O termo *Blockchain*, mencionado inicialmente por Nakamoto em 2008, e implementado em 2009, pode ser definida como uma cadeia de blocos conectados e verificados, em que cada bloco contém dados de transações e endereços criptografados, juntamente com data e hora. Estas informações ficam armazenados em uma *ledger*, um livro razão, que armazena os blocos de forma segura em uma espécie de lista encadeada. Nestas listas encadeadas, dentro de cada bloco há sempre uma parte de código que remete ao bloco anterior e ao próximo, garantindo maior segurança no armazenamento dos dados (Pranto *et al.*, 2021; Zheng *et al.*, 2018). A *Blockchain* é focada na descentralização, o que garante a múltiplos sistemas uma cópia

da cadeia de blocos, e integrada a várias tecnologias, como assinaturas digitais, baseadas em criptografias, e mecanismos de consenso distribuído. Isso eleva a eficiência e a segurança, uma vez que seus dados gravados não podem ser alterados (Zheng *et al.*, 2018).

Com a ajuda da *Blockchain*, transações descartam intermediários a partir de *Smart Contracts*, um contrato digital programado que acionará automaticamente o evento uma vez que os requisitos, previamente programados, sejam verificados (Pranto *et al.*, 2021; Jani, 2020). A possibilidade da inclusão desta tecnologia no mundo da medicina evita que um enorme mercado de prescrições ilegais permaneça ativo na sociedade, como foi o caso da farmacêutica CVS, acusada de forjar mais de 500 receitas médicas entre 2011 e 2014 nos EUA (Justice U.S. Attorney's Office, 2016).

Nesta proposta foi realizado um estudo e implementação de uma prescrição médica digitalizada e salva numa *Blockchain*, na qual o paciente a recebe em seu e-mail um *QR Code*, uma senha pessoal, juntamente do *link* para o *website* a qual sua prescrição está disponível para visualização. Ao dirigir-se à farmácia, o farmacêutico conseguirá realizar a validação prescrição a partir da leitura do *QR Code* e da senha pessoal. Realizada a validação de suas credenciais e da prescrição, o farmacêutico é automaticamente direcionado à página da prescrição, permitindo sua edição. Uma vez que a prescrição contida dentro do *Smart Contract* for totalmente vendida, o mesmo alterará seu status interno para "fechado", impossibilitando que o paciente e o farmacêutico a utilizem novamente.

## 1.3 ESTRUTURA DO TRABALHO

A estrutura do trabalho a seguir pontuará os tópicos do tema. No Capítulo 2 são tratados o sistema de prescrição médica, caracterizando o histórico até sua inserção no mundo digital, bem como os órgãos responsáveis e a forma como funciona atualmente as prescrições eletrônicas.

Em seguida são apresentados conceitos de *Blockchain* e *Smart Contracts*, explicando os principais fatores em que ambos colaboram para a evolução das receitas eletrônicas atuais, mas também destacando o lado da segurança, uma vez que ataques cibernéticos podem ocorrer, independente da tecnologia.

No Capítulo 3 são apresentados os materiais utilizados, e os métodos em que a proposta se baseou para a realização da monografia, descorrendo sobre a arquitetura projetada para o novo sistema implementado.

Na sequência, no Capítulo 4, é descorrida a implementação do sistema, demonstração através de testes realizados e seus resultados obtidos. Por fim, a escrita da conclusão, bem como a opinião do autor da monografia a respeito do projeto desenvolvido, com suas vantagens e desvantagens da utilização do mesmo.

# 2 REFERENCIAL TEÓRICO

Neste capítulo serão descorridos os principais tema relacionados ao projeto desenvolvido. Primeiramente foi explicado os conceitos relacionados à prescrição médica eletrônica e seu funcionamento nos dias atuais. Em sequência é discorrido sobre os temas *Blockchain* e *Smart Contract*, destacando seus components, funcionalidade e como inserí-los no contexto das prescrições médicas eletrônicas.

## 2.1 PRESCRIÇÕES MÉDICAS

Atualmente, um sistema de prescrição médica física depende apenas de carimbo, número do CRM (Conselho Regional de Medicina), nome do médico e a descrição do medicamento. É uma prática há muito utilizada em todo o mundo, mas que trouxe problemas ao longos dos anos. Independentemente do crescimento do mundo digital, em alguns locais ainda é necessária a apresentação de prescrições físicas para comprar remédios controlados. Mídias digitais ainda não são aceitas, sendo necessário o envio dessas por email para as instituições após o pedido de compra (He *et al.*, 2019).

Segundo um estudo realizado pela empresa britânica KLAS, são estimados que 68% dos erros relacionados à medicação ocorrem pela incompreensão da grafia médica no receituário. Além disso, outras estatísticas mostram que 39% dos erros médicos associados aos fármacos ocorrem no momento da prescrição (VEJA, 2019). Como forma de diminuir tais erros, foi proposto em meados de 2007, quando realizados levantamentos de requisitos pela CELEPAR ao o sistema eletrônico GSUS (CARDOSO, 2013), a prescrição eletrônica no Brasil.

Porém, somente em março de 2020 a Agência Nacional de Vigilância Sanitária (ANVISA) manifestou-se favorável à utilização de assinaturas digitais em receituários, devido a pandemia da Covid-19 (CFF, 2020b). Com o atendimento online, a prescrição de medicamentos e a solicitação de exames foram forçados a acompanhar esta mudança repentina do sistema. Pouco tempo após a aprovação pela ANVISA, em 2021, o Conselho Federal de Medicina (CFM) apresentou uma nova plataforma de prescrição eletrônica, criada em parceria com o Conselho Federal de Farmácia (CFF) e o Instituto Nacional de Tecnologia e Informação (ITI) para favorecer uma conexão mais ágil, segura e efetiva entre médicos, pacientes e farmacêuticos (CFF, 2020a).

### 2.1.1 PRESCRIÇÃO ELETRÔNICA

A prescrição eletrônica é a versão digital da prescrição médica tradicional, sendo um dos documentos mais importantes da prática médica, juntamente com o prontuário. Devem conter dados do prescritor, bem como clínica ou hospital, além dos contatos de identificação.

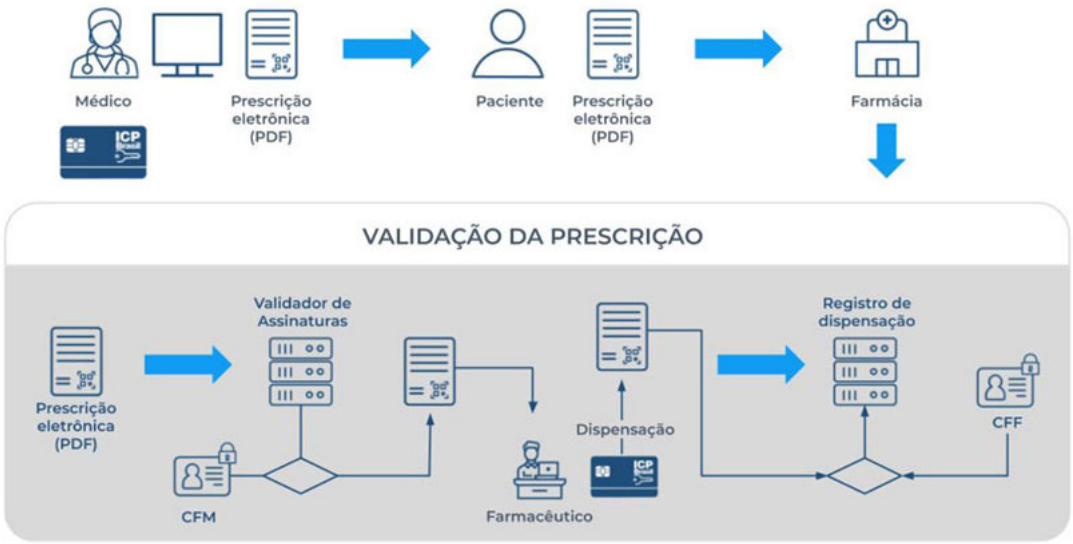
Já com relação ao paciente, devem conter nome completo, CPF ou outro documento, além das informações sobre os medicamentos que o mesmo necessitar. Por fim, todos os documentos digitais devem conter a assinatura digital.

A assinatura digital é o principal validador da prescrição, pois garante que as informações existentes no documento sejam válidas, sendo aceitas pelos demais profissionais que o paciente necessitar para atender às suas demandas de saúde. Todas as receitas digitais, para serem válidas, deverão estar assinadas digitalmente, usando a Infraestrutura de Chaves Públicas Brasileiras (ICP-Brasil) (CFF, 2020c).

A Infraestrutura de Chaves Públicas Brasileiras, a ICP-Brasil, é a identidade eletrônica de pessoa física ou jurídica em ambiente virtual. Um documento eletrônico identifica e gera a assinatura digital, garantindo autoria, integridade e autenticidade ao documento, uma vez que a tecnologia garante que o conteúdo não foi alterado ao momento que for assinado. Caso ocorra tentativa, o documento apresentará mensagem de erro (CFM, 2022). O documento é gerado e assinado por um órgão confiável, chamado de Autoridade Certificadora. Essa deve ser uma entidade que faz parte da infraestrutura do governo e que tenha uma classificação do nível de segurança. Desta forma, a autenticidade e validade do documento serão preservados.

Na Figura 1 é demonstrado o atual sistema eletrônico, que é gerado em PDF e enviado para o paciente. Após validações da assinatura do farmacêutico, garantido pela segurança do ICP-Brasil, a prescrição é enviada para o paciente, que por sua vez, envia para a farmácia. O documento enviado passa por uma série de validações até que seja liberada a compra.

**Figura 1 – Sistema de prescrição médica eletrônica atual**




**Fonte: CFM - Prescrição Eletrônica.**

Os benefícios da mudança do meio físico para o digital influencia não somente os médicos, mas os farmacêuticos e pacientes, sendo eles:

- Maior praticidade para os médicos ao emitir e acessar os documentos;

- Proteção dos dados e acesso restrito;
- Um documento de acesso nacional, que reduz o custo de impressão;
- Validação da prescrição mais rápida por parte dos farmacêuticos no Instituto Nacional de Tecnologia da Informação (ITI);
- Os pacientes recebem os documentos dos seus aparelhos eletrônicos, não precisando mais de papel.

Visando melhorias na parte de segurança e do sistema, surge uma tecnologia muito utilizada em várias vertentes do mundo tecnológico, tais como agricultura, realização de pagamentos, criptomoedas, entre outros, e que pode trazer um enorme benefício para a saúde, chamada de *Blockchain*.

## 2.2 *BLOCKCHAIN* E SUA ARQUITETURA

A *Blockchain* é uma espécie de *Ledger*, ou livro-razão em português, em que suas transações são armazenadas em blocos, crescendo de tamanho conforme novos blocos de transações são incluídos. As ideias centrais de se ter uma rede de computadores com uma cadeia de informações no formato de livro eletrônico surgiram entre o final dos anos 80 e início dos anos 90. Porém, seus conceitos foram combinados e publicados somente em 2009, por Satoshi Nakamoto, com a *Bitcoin* (Narayanan *et al.*, 2021).

A proposta da *Blockchain* refere-se a uma cadeia de blocos que armazena um grupo de informações sobre o bloco anterior e do bloco seguinte, tornando um bloco na cadeia inalterável e irremovível, pois isso mudaria a cada bloco subsequente (Alharby; Moorsel, 2017). Uma analogia para o que seriam os blocos da *Blockchain*, seria de um “cofre de banco” em que sua fachada é de vidro. Permite que todos vejam seu conteúdo, mas não podem acessá-lo. Apenas a pessoa que possui a chave exclusiva daquela caixa pode acessar, mas somente acessar. O indivíduo não pode alterá-lo ou violá-lo.

Sobre sua arquitetura, vale pontuar que a *Blockchain* possui diferentes tipos, como demonstrado na Figura 2 que diferem em termos de permissões de leitura/gravação. São elas:

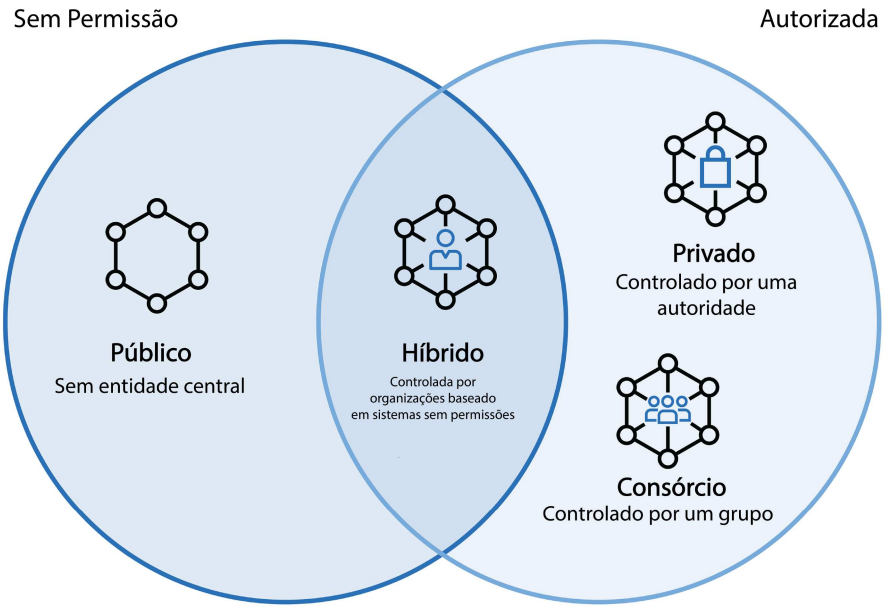
1. *Blockchains* públicos. São particularmente úteis quando nenhuma entidade central está disponível para verificar uma transação e é necessária uma descentralização completa. Eles também podem ser auditados por qualquer pessoa, e cada nó tem tanto poder de transmissão quanto qualquer outro. Antes de uma transação ser considerada válida, ela deve ser autorizada por cada um de seus nós constituintes por meio do processo de consenso da cadeia. Desde que cada nó cumpra as estipulações específicas do protocolo, suas transações podem ser validadas e, assim, adicionadas à cadeia (Narayanan *et al.*, 2021; Gatteschi *et al.*, 2018).

2. *Blockchains* privadas possuem usuários que autorizam ou restringem o acesso de leitura e gravação de informações. Por precisar de acesso, é necessário um convite que deve ser validado pelo iniciador da rede ou por um conjunto de regras implementadas inicialmente. E a partir do momento que tenha ingressado, terão um papel importante na manutenção da *Blockchain* de maneira descentralizada. Elas oferecem vantagens, como custo de validação mais baixos e tempo de validação mais curtos, risco reduzido e maior privacidade (Narayanan *et al.*, 2021; Gatteschi *et al.*, 2018).

3. *Blockchains* Híbridas são uma mistura das duas, podendo deixar alguns dados abertos e transparentes, mas com acessos restrito apenas a quem tiver permissão de operá-los. Ela é controlada por uma corporação, combinando permissões e permitindo organizações de construir sistemas privados, baseados em permissão, em paralelo com sistemas públicos sem permissões (CriptoFácil, 2021; Team, 2022).

4. *Blockchains* de consórcio é um meio termo entre cadeias pública e privada. Ele é parcialmente descentralizado, pois as o controle das redes é conduzido por um consórcio associado. Este grupo, por sua vez, pode limitar quem ingressa na rede, quem é removido, codifica as diretrizes da rede. Além disso, diferentemente das outras arquiteturas, apenas um conjunto de nós é responsável por validar os blocos na *Blockchain* do consórcio. (Zheng *et al.*, 2017; Yaga *et al.*, 2019).

**Figura 2 – Diferentes tipos de arquitetura da que uma rede *Blockchain* possui**




**Fonte: Traduzido de Foley & Lardner LLP.**

## 2.3 COMPONENTES DE UMA BLOCKCHAIN

Segundo Yaga *et al.* (2019), a *Blockchain* é composta de *Ledgers*, os blocos e como são encadeados, o endereço, sua função *hash* critográfica, as transações e a criptografia de chave assimétrica. A seguir são descritos cada um dos seus componentes.

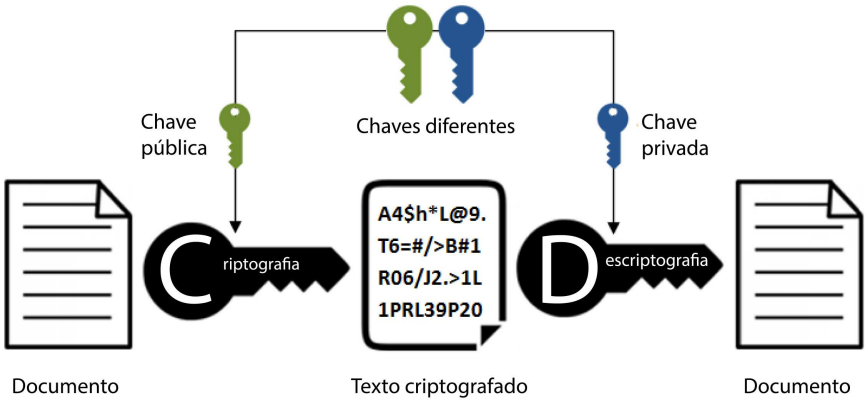
### 2.3.1 CRIPTOGRAFIA

1. **Criptografia Assimétrica**

   A criptografia assimétrica, também conhecida como criptografia de chave pública, consiste em um sistema matemático complexo para criar um par de chaves, chamadas chaves públicas e privadas. É um componente importante na tecnologia da *Blockchain*, pois seu uso garante a segurança do sistema para o salvamento de dados.

   Exemplificado na Figura 3, um documento que utiliza a criptografia assimétrica irá gerar um par de chaves, uma privada e outra pública. A chave privada é exclusiva do usuário que criptografou o conteúdo, podendo descriptografar quando quiser e de uma maneira segura. Já a chave pública estará disponível publicamente e poderá ser transmitida pela Internet, permitindo que enviem informações criptografadas, mas que só serão acessíveis utilizando a chave privada para descriptografar.

**Figura 3 – Criptografia assimétrica com a utilização de par de chaves**




**Fonte: Traduzido de (SSL, 2018).**

Existem algoritmos que realizam a criptografia e das chaves, sendo algumas opções:

- RSA, nomeado em homenagem aos seus três inventores (Rivest, Shamir e Adleman), é o algoritmo assimétrico mais amplamente utilizado, fornecendo um meio de gerar diretamente pares de chaves públicas/privadas exclusivas (Rhodes, 2020).

- ECC, *Elliptic Curve Cryptography*, ou Criptografia de curva elíptica, é uma alternativa ao RSA, pois cria chaves criptográficas mais rápidas, menores e mais eficientes por meio das propriedades da equação da curva elíptica (Brush *et al.*, 2021).

2. **Hashing**

   *Hashing* é um método de criptografia na qual transforma os dados de entrada em saídas relativamente únicas. São algoritmos que incorporam uma variável x (x = arquivo, mensagens, transações, etc) em um dado de valor único com tamanho fixo. Por ser uma função incremental, são unidirecionais e dificilmente se permite a recuperação do valor original do dado, satisfazendo suas propriedades de resistência a colisão e ocultação (Yaga *et al.*, 2019; Greve *et al.*, 2018).

   A resistência a colisão se dá por funções *hashs* com dados diferentes, mas que acabam possuindo uma função *hash* igual f(x) = f(y). Embora seja computacionalmente inviável encontrar duas mensagens que possam gerar uma colisão, isso não significa que as duas mensagens de colisão não existam (Ma *et al.*, 2020).

   Uma utilidade do *Hashing* é a fácil verificação da integridade de um documento. Para não precisar realizar uma comparação do conteúdo de um documento recebido ou enviado, basta comparar o valor do *hash* de ambos os arquivos. Se o valor for o mesmo, a transferência do arquivo é idêntica a cópia.

   Existem tipos diferentes de algoritmos, sendo os mais comuns MD5, SHA-2 e SHA-3. O MD5 é uma função *hash* que codifica uma sequência de informações em uma impressão 128 *bits*, mas por ser mais antigo, têm se tornado vulnerável quanto às colisões. Já SHA-2 consiste em funções de hash com diferentes valores, que vão de 224 até 512 *bits* (Chung, 2022). Por fim, SHA-3, lançado em 2012, que complementa suas versões anteriores (SHA-1 e SHA-2), sendo projetada em funções para fornecer propriedades especiais, como resistência a ataques de colisão (Dworkin, 2015).

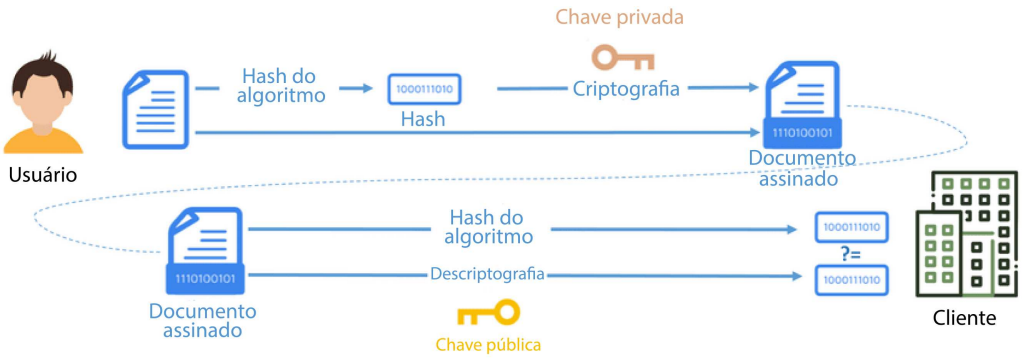
3. **Assinatura Digital**

   A assinatura digital é uma técnica de chaves assimétricas para garantir autenticidade, não repúdio (o usuário não pode negar que enviou a mensagem) e a integridade (garante que a mensagem não seja alterada durante a transmissão). Essas assinaturas são feitas por meio de criptografia de chaves assimétricas, a qual há uma chave privada e outra pública (Yaga *et al.*, 2019).

   Comumente utilizada a partir da criptografia assimétrica, a assinatura digital também possui métodos semelhantes quanto ao seu algoritmo. Além do RSA, o algoritmo DSA, Algoritmo de Assinaturas Digitais, é outro exemplo utilizado para assinaturas. Mais rápido para descriptografia, porém mais lento para criptografar que o primeiro, o algoritmo DSA é mais adequado para assinaturas digitais que o RSA (Jena, 2022).

O processo descrito na Figura 4 demonstra os passos do acesso e leitura de um documento assinado digitalmente. Quando um usuário quer assinar digitalmente um documento, o mesmo passa por etapas de *hash*, resultando no mesmo documento originário, mas com duas chaves criptografadas. O usuário, então, envia o documento criptografado e a chave pública resultante do *hash*. Uma vez que o cliente possui o documento criptografado e a chave pública, são feitas verificações se ambos possuem o mesmo valor de *hash*. em o possuindo, o cliente pode ter acesso ao documento(Greve *et al.*, 2018; Zheng *et al.*, 2017).

**Figura 4 – Representação de um documento sendo assinado digitalmente utilizando uma rede *Blockchain***




**Fonte: Traduzido de (Dhieb *et al.*, 2020), p. 4..**

### 2.3.2 *LEDGER*

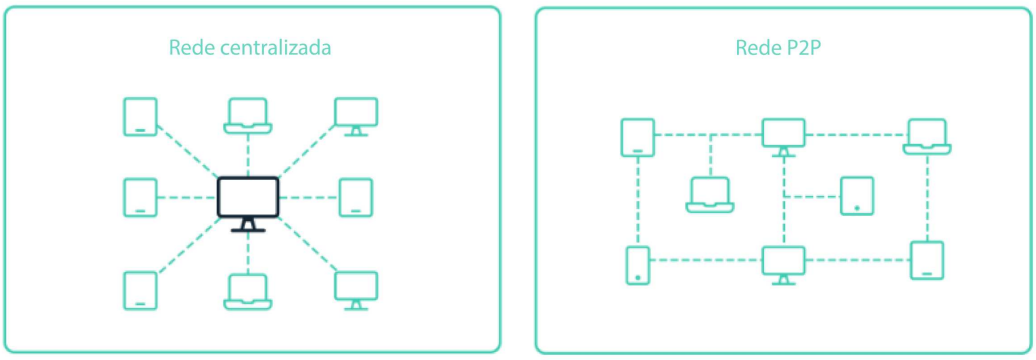
O *ledger* é um tipo de livro razão que registra diversas transações ou documentos de seus usuários, porém no formato digital. A diferença é que suas informações, uma vez registradas, não podem ser apagadas. Esses livros com propriedade centralizada podem ser implementados de maneira centralizada ou distribuída, podendo também armazenar todo tipo de informação, não apenas transações monetárias (Yaga *et al.*, 2019).

A utilização de *ledgers* distribuídos em tecnologias como a *Blockchain*, que possui uma rede *Peer-to-peer*, uma rede sem uma entidade central, permite que todos os pares conectados validem o registro, não somente um ponto único, como mostrado na Figura 5. Dessa forma, a perda de algum participante da rede não impede a comunicação dos dados armazenados em um *ledger* (Moreland, 2022; Yaga *et al.*, 2019).

### 2.3.3 BLOCOS

Cada bloco na *Blockchain* é composto por um dado, um *hash*, o *hash* anterior a ele e um metadado (timestamp, número do bloco). Como mostrado na Figura 6, o dado em um bloco pode ser dos mais variados tipos, desde um texto simples até uma lista de transações. Além do *hash*, do bloco pai, há também o *hash* da raíz da árvore *Merkle*, ou seja, uma estrutura

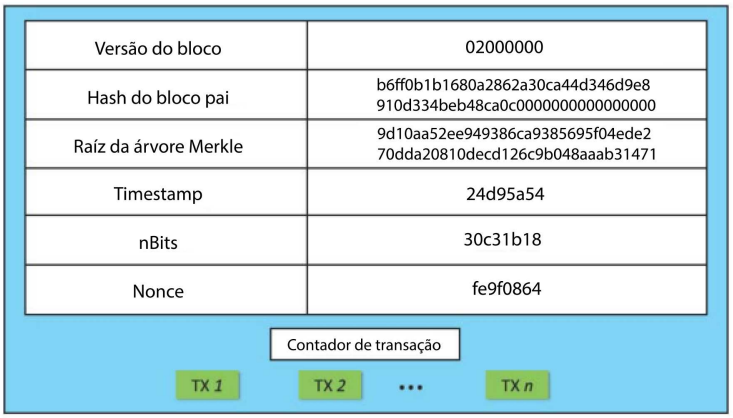
**Figura 5 – Diferença entre rede centralizadas e distribuídas, uma rede *Peer-to-peer***




**Fonte: Traduzido de (Moreland, 2022).**

matemática de dados composta por *hashs* de vários blocos de dados que resumem todas as transações em um bloco. Há, também, o *Nonce*, um número arbitrátrio que só pode ser utilizado uma vez, por isso recebe esse nome (N = *Number* (Número) e *Once* = Uma vez, em inglês) e serve para garantir que as comunicações antigas não possam ser reutilizados em ataques de repetição. Eles podem também ser úteis como vetores de inicialização e em função hash criptográfica. O número máximo de transações que um bloco pode conter depende do tamanho do bloco e do tamanho de cada transação (Zheng *et al.*, 2018).

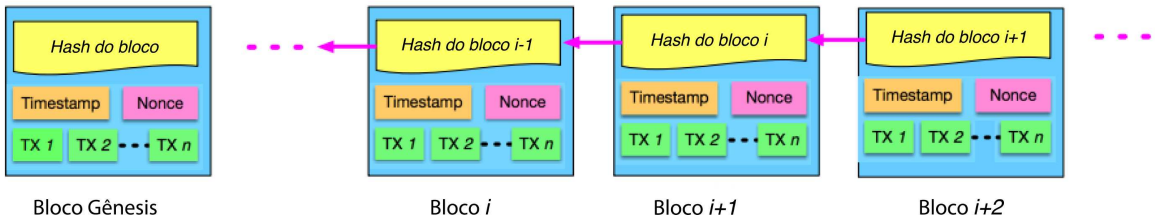
**Figura 6 – Bloco de informações da *Blockchain***




**Fonte: Traduzido de (Zheng *et al.*, 2018).**

Para o bloco inicial, este recebe o nome de bloco gênesis, sempre com valor 0. A Figura 7, adaptada do Zheng *et al.* (2018) exemplifica o funcionamento de uma rede de blocos na *Blockchain*. Cada bloco possui o *hash* do bloco antecessor, sendo uma das principais vantagens da *Blockchain*, a segurança.

**Figura 7 – Blocos conectados através de seu *hash*, componto uma rede *Blockchain***




**Fonte: Traduzido de (Zheng *et al.*, 2018).**

### 2.3.4 MECANISMO DE CONSENSO

O mecanismo de consenso na *Blockchain* é uma forma de validar uma transação que foi inserida na rede. Caso a transação satisfaça as condições, é adicionada ao bloco. Caso não seja validada, a transação é descartada. Em redes *Blockchains* públicas há sistemas de recompensas para efetivar tal validação de uma transação, utilizando criptomoedas como pagamento.

Existem, atualmente, diferentes tipos de mecanismos de consenso, dentre eles:

1. O *Proof-of-Work* é um mecanismo o qual o usuário publica um próximo bloco a partir do momento que resolver um quebra-cabeça computacional. A dificuldade do quebra-cabeça é ajustada a cada 2016 blocos anexados, pois assim a velocidade média para adição de cada block será de 1 a cada 10 minutos (Nguyen; Kim, 2018).

   Ao publicar nós, pequenas alterações são feitas em seus cabeçalhos, tentando sempre realizar um resumo do *hash* que atenda ao requisito. Isso acontece porque, a cada modificação, é gerado o *hash* do cabeçalho anterior, resultando em um processo computacional exigente, cuja intensidade está diretamente relacionada à frequência de publicação dos blocos (Yaga *et al.*, 2019; Nguyen; Kim, 2018).

   De acordo com Ferdous *et al.* (2020), o *PoW* serve para dois propósitos críticos, de ser um mecanismo de dissuasão contra ataque Sybil (criação de múltiplas identidades visando lucro na mineração de dados), e usado como entrada para a função de obter o consenso distribuído necessário quando ocorre bifurcação em uma *Blockchain*.

2. O *Proof-of-State* consegue neutralizar as limitações de qualquer algoritmo *PoW*, o *PoS* baseia em *stakes*. *Stakes* são um tipo de moeda, ou criptomoeda, a qual o modelo se baseia, recompensando o usuário que investiu mais tempo no sistema para validar as transações (Nguyen; Kim, 2018).

   O modelo recompensa o minerador que possui maior quantidade de *stakes* e participações. No entanto, o minerador é escolhido com base no estado do bloco a cada 60 minutos e, de forma randômica. Para a escolha randômica, a rede analisa os usuários com participação, e escolhe com base na proporção em relação a quantidade de *stakes* que o usuário possui (Nguyen; Kim, 2018; Ferdous *et al.*, 2020).

## 2.4 *SMART CONTRACTS*

Atualmente, transações entre partes dentro de qualquer sistema geralmente são conduzidas a partir de um mediador, ou de forma centralizada. Porém, altas taxas de transações ou falhas em segurança promoveram o surgimento de tecnologias capazes de amenizar essas falhas. Essa tecnologia recebeu o nome de *Smart Contracts*, ou em português, contratos inteligentes.

O termo *Smart Contract*, originalmente sendo referido para qualquer tipo e contrato automatizado, vem sendo ligado regularmente com tecnologias como a *Blockchain*. Sua ideia básica é a inclusão de cláusulas contratuais em algoritmos executados de forma síncrona em vários nós de um *ledger* distribuído (Zou *et al.*, 2019).

### 2.4.1 CICLO DE UM SMART CONTRACT

Conforme destacado por Zheng *et al.* (2020), o ciclo de um *Smart Contract* compreende quatro fases: criação, inserção, execução e conclusão. A seguir, serão detalhadas essas etapas, proporcionando uma complementação à Figura 8, a qual ilustra o ciclo de vida de um *Smart Contract*.

Na primeira etapa, de criação, são exemplificadas as obrigações, proibições que o contrato deve possuir. Similarmente ao desenvolvimento de aplicações web, um *Smart Contract* passará por processos de *design*, implementação e validações. Muitas vezes são necessárias várias rodadas de validações até o código final (Zheng *et al.*, 2020).

A etapa de inserção é a qual um *Smart Contract*, já validado, é inserido em uma rede *Blockchain*. Uma vez que é implementado, o *Smart Contract* não pode ser alterado (Alharby; Moorsel, 2017).

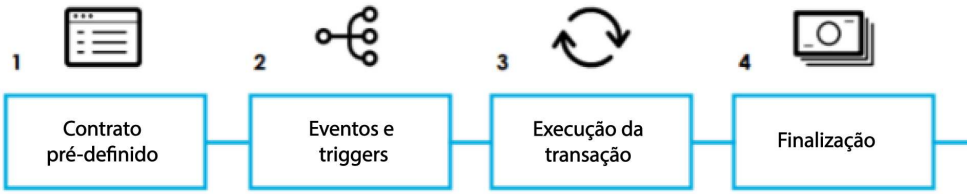
A primeira rede *Blockchain* que forneceu uma plataforma para a criação de *Smart Contracts* foi a Ethereum, em 2013, a partir da linguagem Solidity. Após a criação na linguagem, ele é compilado em bytecode e implantado na *Blockchain* (Wu, 2019).

Na execução, uma vez que a transação é enviada para o endereço do contrato, ela será executada por todos os nós, para chegarem a um consenso sobre sua saída. A *Blockchain* é reiniciada para apoiar a tomada de decisão das partes interessadas e gerar novos blocos quando necessário (Alharby; Moorsel, 2017; Peng *et al.*, 2020). Após a execução, um *Smart Contract* pode, com base na transação, ler/gravar em seu armazenamento privado, armazenar dinheiro em seu saldo, enviar/receber mensagens ou dinheiro ou até mesmo criar novos contratos (Alharby; Moorsel, 2017).

Um *Smart Contract* passa por fases após a sua criação e período de execução. Descrito na Figura 8, após a criação e implementação do mesmo a partir de um acordo pré-definido, quando um evento ocorrer na *Blockchain*, ele irá ativar *triggers* deste *Smart Contract Triggers* são recursos utilizados na programação que realizam ativação automática de eventos. Quando

um evento na *Blockchain* ocorre, o algortimo começará a verificar se todas as cláusulas pré-definidas no *Smart Contract* estão de acordo. Uma vez que a verificação ocorre e todos os eventos são checados e aprovados, o contrato é executado e a transação acontece. A última etapa, de finalização, para eventos digitais, ocorre de forma automática após a ativação e validação da etapa anterior.

**Figura 8 – Fases de um *Smart Contract***




**Fonte: Traduzido de *YOUTEAM*, 2022.**

### 2.4.2 SEGURANÇA

Assim como qualquer outro sistema, a *Blockchain* e *Smart Contracts* possuem falhas em segurança, mesmo que em menor quantidade quando se comparada com outros meios, sejam elas físicas ou digitais. Por lidar com uma grande quantia em dinheiro, ativos digitais, estoques ou dados, a segurança de um *Smart Contract* é de enorme importância, pois qualquer problema mínimo pode resultar em uma grande perda.

De acordo com a pesquisa realizada por Zou *et al.* (2019), 75% das respostas concordam que há um maior requisito de segurança para *Smart Contracts* do que *softwares* tradicionais. Na mesma pesquisa, 71,6% garantem dificuldades em promover a segurança dos *Smart Contracts* durante o desenvolvimento.

Rouhani e Deters (2019) lista possíveis falhas na segurança, voltadas para o sistema em que os *Smart Contracts* são inseridos, sendo descritas abaixo, com suas definições. Além disso, a falta de prática, de ferramentas para verificar a correção do código, falhas na compilação e o acesso público ao código são causas que programadores alertam sempre que realizam sua implementação (Zou *et al.*, 2019).

- Dependência de *Timestamp* se dá pelos carimbos de data/hora do bloco, definidos pelos mineradores com base no horario do sistema local e, portanto, podem ser manipulados.
- Exceções Incompletas ocorrem quando um contrato chama outro contrato. Se ocorrer alguma exceção, ele termina e retorna como falso, mas não notifica o contato chamador.

- Vulnerabilidade de Reentrância é quando um contrato chama outro, a execução espera até que ele termine. Isso gera abertura para usuários explorarem o estado intermediário do contrator.

- Dependência na ordem de transação ocorre quando a ordem de execução das transações depende dos mineradores, e os clientes não têm qualquer controle sobre eles.

Existem ferramentas, como OYENTE, além do *SMARTINSPECT*, que realizam uma série de testes, analisam e utilizam de técnicas de decompilação para prevenir os ataques mencionados por Rouhani e Deters (2019).

## 2.5 *FRAMEWORKS BLOCKCHAIN*

Em geral, um *framework* é uma estrutura real ou conceitual que serve de suporte para a construção de algo. Aplicando isso à *Blockchain*, existem estruturas para explorar o desenvolvimento e melhorias relacionadas à *Blockchain*. Dentre as mais conhecidas, existe a *Ethereum* [1]. Ela é uma plataforma que surgiu em 2015, introduzindo um recurso revolucionário à época, os *Smart Contracts*. Além disso, permitiu criar aplicativos descentralizados, expandindo-se rapidamente para campos além do setor financeiro. Em contrapartida, as empresas exigiam características de desempenho que as tecnologias de *Blockchains* sem permissões, como a *Ethereum*, não conseguiam fornecer. Para atender essa demanda, surgiu a *Hyperledger* [2].

A *Hyperledger*, criada pela *Linux Foundations*, surge como uma tecnologia *Blockchain* voltada para indústrias e empresas. Diferentemente das outras *Blockchains*, a *Hyperledger* não foi criada com propósito de servir para transações de criptomoedas. Inicialmente, foi proposto como uma ferramenta para fornecer soluções mais seguras para sistemas financeiros, e cadeia de suprimentos, porém, também indicada para programas relacionados ao sistema de saúde. Isso se deve por oferecer um sistema de controle de acesso único e granular, em que é necessário garantir que apenas pessoas autorizadas, como médicos, pacientes e farmacêuticos tenham acesso às informações sensíveis que vão ser incluídas.

De acordo com Vale (2020), as características da *Hyperledger* são:

- Extensibilidade, não restringindo o número de novas aplicações dentro da *Hyperledger*, ampliando a cobertura de soluções para os problemas.

- Modularidade, na qual há um grupo que produza tecnologias para sustentar o projeto. Com isso, um *design* de solução modular a aplicabilidade não fica restrita ao código para qual foi desenvolvida, mas para todas as outras soluções.

- Interoperabilidade, garantindo que *softwares* e *hardwares* diferentes sejam capaz de executar aplicações *Hyperledger*.

[1] https://ethereum.org/en/
[2] https://www.hyperledger.org/

A *Blockchain Hyperledger* é uma organização de código aberto que hospeda diversos projetos relacionados à tecnologia *Blockchain*. Cada projeto dentro da Hyperledger tem um foco específico e atende a diferentes necessidades e casos de uso. Dentre eles, descritos pela Foundation (2015), temos:

- *Hyperledger Fabric*, uma plataforma modular destinada ao desenvolvimento de soluções de *Blockchain* empresariais, caracterizada por sua arquitetura modular, suporte a *Smart Contracts* e escalabilidade em ambientes permissivos.
- *Hyperledger Sawtooth*, uma plataforma projetada para o desenvolvimento, implementação e operação de redes blockchain, distinguindo-se pela utilização de um modelo de consenso modular, permitindo diferentes algoritmos de consenso em partes distintas da rede.
- *Hyperledger Indy*, uma solução especializada em identidade descentralizada, oferecendo meios seguros e privados para criar, armazenar e transmitir identidades digitais.
- *Hyperledger Iroha*, projetado para aplicações nas áreas de finanças e gerenciamento de ativos, o Iroha adota um modelo de permissões baseado em contas, proporcionando simplicidade em casos de uso que demandam essa característica.
- *Hyperledger Besu*, uma implemmentação de código aberto do *Ethereum* compatível com a especificação *Hyperledger*, oferecendo suporte a *Smart Contracts* e facilitando a integração com outros projetos da *Hyperledger*.

Dentre as opções de projetos descritas, a que melhor adequa ao projeto desenvolvido foi a *Hyperledger Fabric*. Outro ponto que favorece sua utilização se dá por seus mecanismos de consenso serem mais flexíveis, permitindo sua customização de acordo com suas necessidades e restrições, resultando em um ambiente mais confiável para informações sensíveis. Por fim, outra funcionalidade se da por sua comunidade funcionar em formato de código aberto, na qual programadores possuem liberdade para desenvolver protocolos e funcionalidades diferentes, para serem utilizadas pelos mais diversos tipos de usuários.

# 3 MATERIAIS E MÉTODOS

O capítulo a seguir descreve quais materiais foram utilizados, tanto para o desenvolvimento da interface, quanto para o desenvolvimento da conexão com a *Blockchain*. Com os materiais listados, foi demonstrado o método utilizado para a implementação e as arquiteturas resultantes do sistema modelado.

## 3.1 MATERIAIS

Para o desenvolvimento da proposta, foram necessários um *laptop* de uso pessoal, um ambiente de desenvolvimento, materiais que forneçam suporte para a criação do código, conexão com o servidor, obtenção de dados via *API*, *Application Programming Interface*, ou em português, Interface de Programação de Aplicativos, podendo ser definida como conjunto de padrões de programação para o acesso a um aplicativo de *software* baseado na Web, e um ambiente de testes que comporte a integração para criação, gerenciamento e depuração de containers, que serão explicados mais à frente.

O ambiente de desenvolvimento do código utilizado foi *Visual Studio Code* [1], abreviadamente por *VS Code*. Este ambiente oferece suporte para uma alta variedade de linguagens de programação, dentre as utilizadas nesse projeto, *Javascript*, pela inclusão de extensões para adicionar funcionalidades específicas, um suporte para a integração de contêineres Docker e seu alto desempenho.

O código foi dividido em frentes, seja o *front-end* codificado em linguagem *Vue JS*[2] e o *back-end* em *Node JS* [3].

- O *front-end* é a representação da interface do sistema, a apresentação de um site para a obtenção e apresentação dos dados da receita digital. Foi escolhida a utilização do *Vue JS* por ser um *framework* progressivo, ou seja, uma estrutura que serve de base para a construção de aplicações *web* de finalidade específica na qual pode-se utilizar em pequenos pedaços do projeto e incorporar gradualmente conforme necessário. Além disso, o *Vue JS* oferece um sistema reativo, na qual os dados e a interface estão sempre sincronizados, simplificando a manipulação de dados e atualizações.

- O *back-end* é a parte do código que conecta a Internet, gerencia as conexões dos usuários e alimenta a aplicação *web*. A utilização da plataforma *Node JS* para o desenvolvimento se dá pela permissão de utilizar *JavaScript* tanto no *front-end* quanto no *back-end*, criando um desenvolvimento mais simples na comunicação de dados en-

[1] https://code.visualstudio.com/
[2] https://vuejs.org/
[3] https://nodejs.org/en

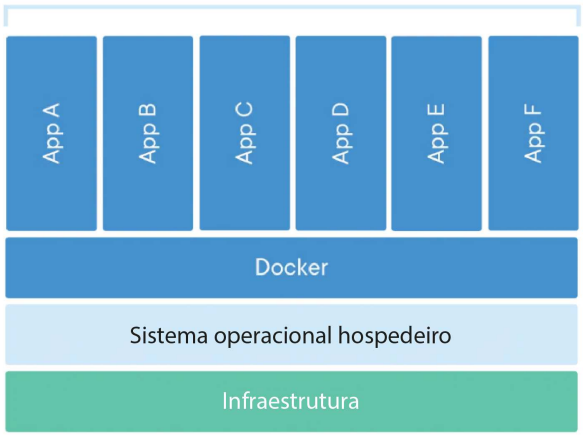
tre cliente e servidor. A plataforma também oferece uma flexibilidade para criações de *APIs*.

Para a criação da *Blockchain Hyperledger*, foram necessárias extensões no ambiente *VS Code*, além de um *software* que empacotasse todos os códigos e dependências para que o aplicativo fosse executado de forma rápida e confiável. Para isso foi utilizado o Docker.

Docker [4] é uma tecnologia de virtualização em formato de contêineres, uma forma de máquina virtual que permite que o desenvolvedor construa aplicativos dentro desses contêineres e os agrupe em imagens (Anderson, 2015). Essas imagens dentro dos contêineres possuem o aplicativo desenvolvido e todo seu ambiente de execução, facilitando o compartilhamento com outras pessoas, pois seu ambiente de execução possuirá a personalização perfeita para tal aplicativo (Docker, 2020).

Como demonstrado na Figura 9, um contêiner é uma unidade padrão de *software* que empacota o código para que seja executado de forma rápida e confiável em outro ambiente de computação. Uma imagem de contêiner Docker é um pacote de software leve, independente e executável que inclui tudo o que é necessário para executar um aplicativo: seu código, o tempo de execução, as ferramentas do sistema, as bibliotecas e configurações desse sistema. Em sequência, cada usuário, chamado por "App", irá ser executada com as mesma configurações.

**Figura 9 – Representação do sistema Docker e seus contêineres**




**Fonte: Website oficial Docker.**

A vantagem de utilizar o Docker para o sistema proposto se dá pela conexão com a *Blockchain* ser feita através de um contêiner. Uma vez que todo o ambiente possui as configurações ideiais, é possível que seja realizado o teste para esta conexão e seguir à etapa seguinte, da manipulação e edição das funções dos *Smart Contracts*.

[4] https://www.docker.com/

## 3.2 MÉTODOS

Inicialmente foi realizado um estudo bibliográfico sobre o tema, com o intuito de levantar os requisitos necessários para sistemas de prescrição médica, e sobre os trabalhos existentes relacionados à esta proposta. Após estudos e pesquisas realizadas, explicadas no Capítulo 2, foi feita uma análise do sistema de prescrições atual, na qual o Conselho Federal de Medicina utiliza para o envio das prescrições médicas online e seu sistema de validação. Foi verificado que o sistema de verificação se dá por um banco de dados próprio, contendo as informações dos médicos e farmacêuticos. Uma vez aprovado, o medicamento prescrito poderia ser vendido para o paciente.

Concluída a verificação do sistema atual, foi feito um estudo sobre o conceito de *Blockchain* e sua arquitetura, funcionalidades e código. Uma vez que os conceitos foram entendidos, fêz-se uma pesquisa sobre *Smart Contracts*, suas funcionalidades e como seria a melhor forma de organizar a proposta dentro dos temas procurados.

Uma vez que os conceitos foram entendidos, foi realizado um estudo dentre as principais redes *Blockchain* e seus *Frameworks*, com intuito de definir qual seria o mais indicado para a utilização no sistema proposto. A escolha para utilização foi o *framework Blockchain Hyperledger*, pois é uma plataforma que oferece solução mais segura para *Blockchains* sem permissões, possuindo também mecanismos de consenso que se adequariam melhor à proposta desenvolvida na área médica.

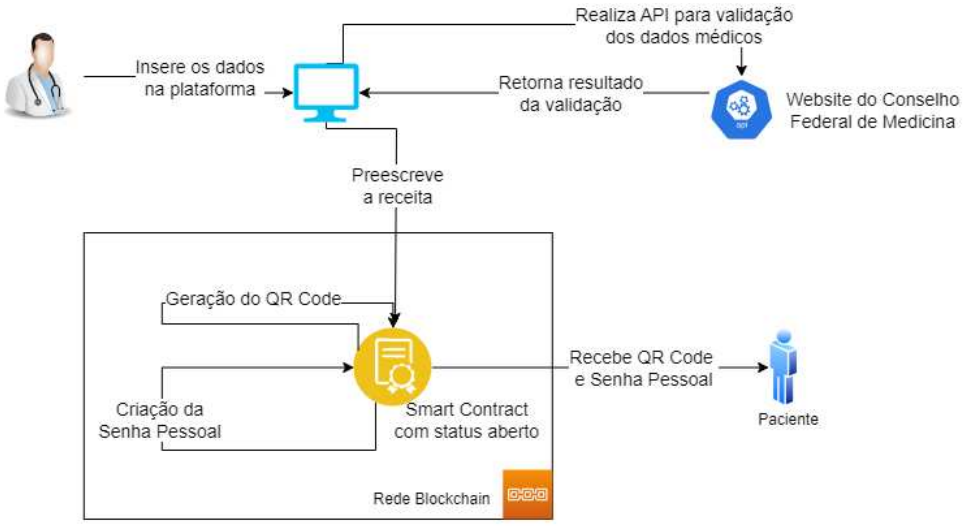
Com o sistema atual, representado pela Figura 1, ficou-se evidente que a CFM não utiliza de redes descentralizadas, como a *Blockchain*, em sua implementação. Com isso, para implementar a mesma, foi necessário uma conexão com o sistema atual, utilizado para validação dos médicos e farmacêuticos, de modo que não sejam permitidos nomes ou documentos falsos para a criação da prescrição.

Uma vez que foi compreendida a funcionalidade que o sistema deveria ter, foi desenvolvida a arquitetura do sistema. Por se tratar de um sistema no qual há três atores, o médico, o cliente e o farmacêutico, foram desenvolvidas três módulos separados, mas que compõem o sistema como um todo.

Como demonstrado na Figura 10, a arquitetura definida para a relação entre o médico e o paciente foi desenvolvida por meio da uma plataforma, na qual o médico inclui suas credenciais e o sistema a validará por meio de uma *API* utilizando o *website* oficial do Conselho Federal de Medicina. Uma vez que o sistema retorna um valor válido, o médico poderá seguir para a prescrição da receita.

Com a receita sendo feita, o sistema fará a criação do *Smart Contract* com todos os dados da receita, CPF do paciente e dados do médico e um *status* como "aberto". O sistema, a partir do *status*, permitirá sua leitura e edição somente enquanto a quantidade de remédios for maior que zero. Uma vez que todos os remédios forem validados como vendidos para o paciente, o *Smart Contract* será finalizado.

**Figura 10 – Arquitetura do sistema Médico - Paciente**




**Fonte: De autoria própria..**

Após o médico incluir os dados da receita, o sistema irá gerar uma senha de segurança para o paciente. Essa senha, que será incluída dentro do *Smart Contract* será previamente criptografada por motivos de segurança. Enquanto isso, no *e-mail* do paciente, o mesmo receberá a senha de acesso, juntamente do *QR Code*, que permitirá à ele e ao farmacêutico acesso a leitura de seu conteúdo.

A arquitetura em sequência, demonstrada na Figura 11 explica como o paciente fará a visualização do sistema. Por se tratar de uma interface sem um banco de dados tradicional, sem um login ou usuário, o paciente precisará apenas apresentar seu CPF, o *QR Code* e a senha pessoal fornecida pelo médico para que o sistema busque o conteúdo dentro do *Smart Contract*.

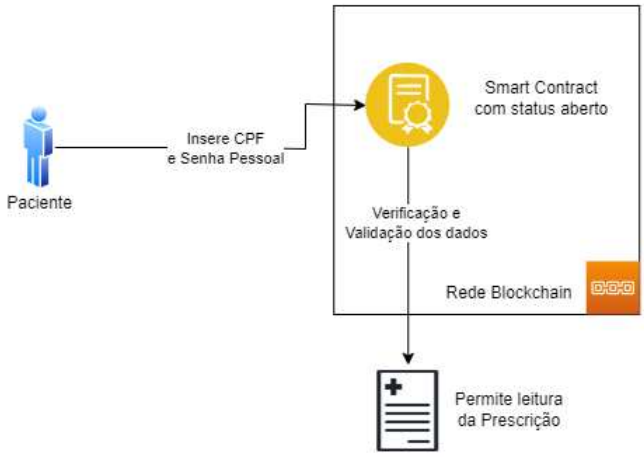
O sistema possui um validador, impossibilitando que usuários que não sejam farmacêuticos de editar as informações dentro do contrato. Isso garante que apenas pessoas qualificadas editem e dêem baixa no sistema quando um remédio for retirado pelo paciente.

Por fim, a terceira arquitetura, na Figura 12 explica a interação entre o paciente e o farmacêutico. A similaridade do sistema entre o médico e o farmacêutico se dá pelas plataformas de validação, na qual ambas permitem que os dados sejam verificados por meio de uma *API*.

Uma vez que os dados foram verificados, para realizar a visualização o farmacêutico incluirá o mesmo *QR Code* e senha pessoal que o paciente possui, porém com a diferença que o farmacêutico terá a possibilidade de alterar o valor das quantidades de remédios dentro da receita. O *Smart Contract* fará uma verificação se todos os remédios receberam baixas dentro de seu conteúdo e, uma vez que não exista mais remédio com uma quantidade maior que zero, o *status* do contrato será alterado para "Fechado"e o contrato será finalizado, com o novo bloco incluído na *Blockchain*.

Uma vez que o bloco receber o status de "Fechado", os valores da prescrição médica dentro do *Smart Contract* não estarão mais visíveis para leitura, tanto para o paciente quanto

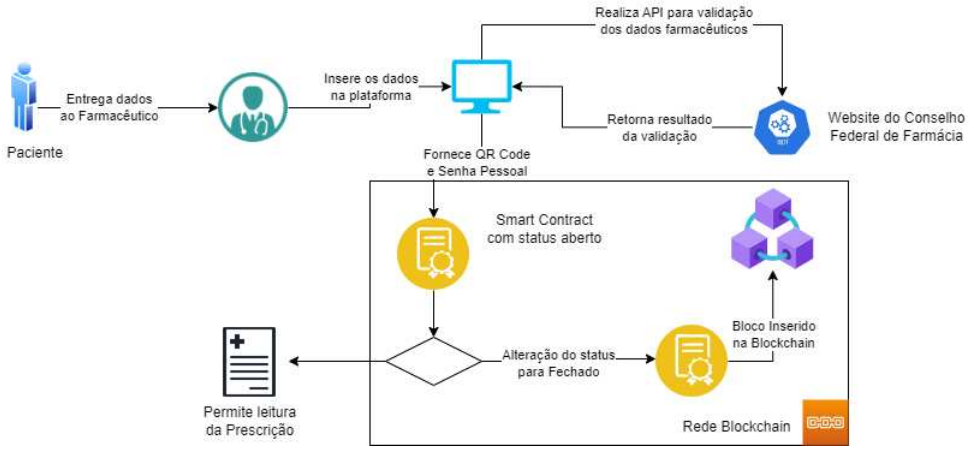
**Figura 11 – Arquitetura do sistema Paciente**




**Fonte: Autoria própria.**

para o farmacêutico. Este sistema foi utilizado para que não sejam permitidas leituras e vendas de fármacos já utilizados em *Smart Contracts* finalizados.

**Figura 12 – Arquitetura do sistema Paciente - Farmacêutico**




**Fonte: Autoria própria.**

### 3.2.1 A *API*

Para permitir o acesso tanto do médico, quanto do farmacêutico, foi necessária uma validação em seus respectivos *websites*. Por se tratar de uma validação de veracidade de dados inseridos na interface *web* por parte dos médicos e farmacêuticos que utilizarem a plataforma, foi necessária a criação de uma *API*, uma interface de programação de aplicação.

Para cada um dos atores, médicos e farmacêuticos, foram utilizados métodos diferentes em cada uma de suas plataformas de verificação. A validação dos dados médicos foi realizada no site da CFM, Conselho Federal de Medicina, onde possuem uma área específica de

busca por médicos. Demonstrado na Figura 13, os dados necessários para a realização da *API* deveriam ser o "Nome", "CRM", "UF", "Município", "Tipo de Inscrição"e "Situação". Como o intuito é validar a existência dos dados fornecidos, não era necessária a inclusão de "Especialidade"e "Área de atuação"pois a generalização em "Todos"como resposta já nos fornece a mesma validação.

**Figura 13 – Sistema de Busca por Médicos utilizada pelo site da CFM**


**Fonte: Disponível em: https://portal.cfm.org.br/busca-medicos/.**

Foi escolhido a utilização do *Puppeteer*[5] para a realização dessa verificação. *Puppeteer* é uma biblioteca pertencente ao *Node JS* capaz de realizar a automatização do envio de formulários, testes de UI, Interface do Usuário, entrada de dados no teclado, entre outras aplicações. Esta biblioteca incluirá os dados obtidos do médico e fará a automatização da validação dos dados, incluindo-os nos campos da Figura 13.

Tudo isso é capaz pelo fato desta biblioteca usar do conjunto de ferramentas para programadores *Web* incorporadas diretamente no navegador Google Chrome. Estas ferramentas permitem inspecionar o código do site em questão, permitindo que sejam realizadas inserção de valores em seus campos. Como saída de resultado, o site retorna os dados do médico procurado, juntamente com sua foto de registro. Ao finalizar a *API*, é retornado um valor booleano, uma resposta do sistema (*true* se a validação for verdadeira e *false* se a validação for negativa).

Para a realização da *API* do farmacêutico, também foi utilizado o *Puppeteer*, mas sem ser a necessidade das ferramentas do navegador, citadas previamente. O site escolhido para a validação do farmacêutico que precisar visualizar e editar o *Smart Contract* foi o site oficial do CFF, Conselho Federal de Farmácia. Da mesma maneira que o site da CFM possuia uma área de procura por médicos, o site da CFF possui a mesma área voltada aos farmacêuticos.

Como mostrado na Figura 14, os dados necessários para a *API* são "Nome/CRF", "Regional", que seria o UF do farmacêutico, "Cidade"e "Categoria", que pode ser farmacêutico ou técnico. Para a seguinte proposta, como pessoas podem digitar seus nomes com abreviações ou incompletos, foi solicitada a validação a partir do CRF.

Analizando o site pôde-se verificar que a validação também seria capaz de ser efetuada a partir do *hyperlink* do site. Explicitado na Figura 15 na qual foi realizado apenas um teste,

[5] https://pptr.dev/

**Figura 14 – Sistema de Busca por Farmacêuticos utilizado pelo site da CFF**


**Fonte: Disponível em: https://site.cff.org.br/farmaceutico.**

percebeu-se que no próprio *link* do site seria possível incluir os dados do farmacêutico, sem a necessidade de uma automatização de formulários, como feito para o site dos médicos. Isso é realizado devido a diferença dos métodos de envio dos parâmetros definidos no protocolo *HTTP*, Protocolo de Transferência de Hipertexto, em português.

O método *GET* transmite os parâmetros diretamente pela *url*, uma vez que esses parâmetros não são considerados confidenciais e podem ser acessados facilmente. Por outro lado, o método *POST* é empregado quando os parâmetros necessitam ser enviados no corpo da requisição, evitando qualquer exposição na *url*. Isso proporciona um controle mais robusto, especialmente ao lidar com dados sensíveis. Para isso, bastaria realizar a troca dos valores com os dados fornecidos na interface *web*.

Para os testes, melhor detalhados no Capítulo 4, selecionou-se "Pato Branco"para cidade, "Farmacêutico"para categoria e "Teste"de nome. Seguindo a ideia da *API* mencionada acima, ao finalizar, é retornado um valor booleano, podendo ser retornado verdadeiro ou falso. Para ambas *APIs* foram necessários testes com dados reais. Porém, para preservar identidades, não serão mostradas suas respectivas saídas na proposta.

**Figura 15 – *Hyperlink* para realização da *API* no site da CFF**


**Fonte: Disponível em: https://site.cff.org.br/farmaceutico.**

# 4 IMPLEMENTAÇÃO E RESULTADOS

Neste capítulo são apresentados os resultados do projeto desenvolvido. Por se tratar de um desenvolvimento *Full-Stack*, um desenvolvimento que consiste nna implementação do *front-end* e do *back-end*, os resultados foram separados para que primeiro sejam visualizadas as interfaces desenvolvidas e, por fim, os códigos que permitem que a inclusão dos *Smart Contracts* sejam criados e modificados. Após a implementação, foram documentados os resultados obtidos do sistema resultante.

## 4.1 APRESENTAÇÃO DA INTERFACE

O objetivo em relação à interface era criar um *site* simples e intuitivo, como demonstrado na Figura 16. A tela inicial possui 2 botões, cada um com sua respectiva frente. Ao clicar em qualquer um dos botões, a pessoa será direcionada à página de verificação e, consequentemente, para suas respectivas funcionalidades.

**Figura 16 – Tela de Menu do sistema resultante**


**Fonte: Autoria própria..**

Ao clicar no botão "Médico", a pessoa será direcionada à próxima página, de verificação através da *API*, na Figura17. Nesta página, algumas das opções que foram selecionadas para a validação do *site*, demonstrado na Figura 13 não foram incluídas. Isso se deu por algumas opções pré-estabelecidas serem tomadas. "Tipo de Inscrição", "Situação", "Especialidade"e "Área de atuação"foram campos em que os dados foram prontamente selecionados no *back-end*, de forma que apenas médicos ativos pudessem preescrever receitas médicas, como pode ser visualizado na Figura 17. As opções definidas para a realização da *API*, que terá seu código explicado mais adiante, podem ser visualizadas na Tabela 1.

**Figura 17 – Formulário de validação do médico**

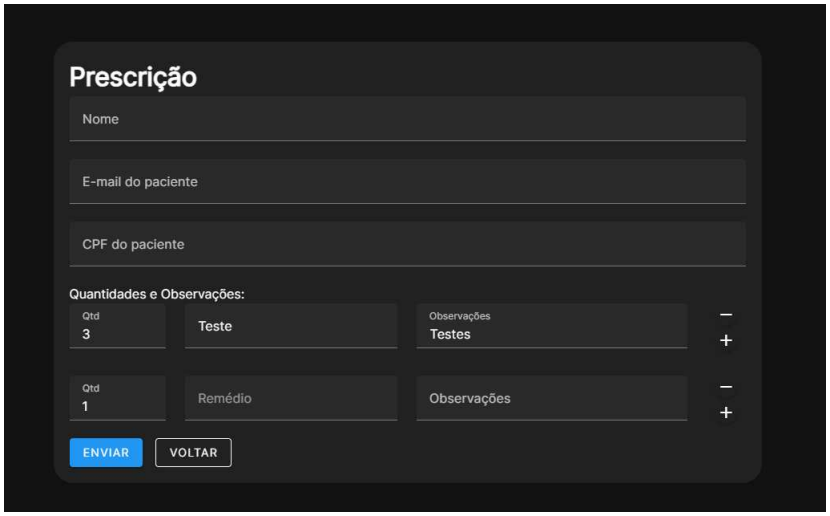

**Fonte: Autoria própria..**

| Variável | Valor de entrada |
|---|---|
| Inscrição | Principal |
| Tipo de Situação | Ativo |
| Situação | Todos os Ativos |
| Especialidade | Todas |
| Área de Atuação | Todas |

**Tabela 1 – Definições padrão para validação de médicos**

Uma vez que o médico fez sua validação, será direcionado para a página de criação da prescrição, demonstrado pela Figura 18. Nesta página o médico irá incluir os dados necessários do paciente que serão inseridos no contrato, além do e-mail do mesmo para que seja enviado por email o *link* do contrato, juntamente de seu *QR Code*, para que seja possível sua visualização por um *smartphone* e pelo farmacêutico, junto também de sua chave de acesso. Esse modelo do e-mail pode ser visualizado na Figura 19.

**Figura 18 – Formulário de prescrição da receita**




**Fonte: Autoria própria..**

**Figura 19 – Modelo do e-mail enviado para o paciente**


**Fonte: Autoria própria..**

Ao receber o email, o paciente poderá realizar a visualização da prescrição através do *link* ou do *QR Code*. Na página direcionada, o paciente necessita de seu CPF e sua chave enviada por e-mail, para que seja realizada a validação no sistema em busca do *Smart Contract*. Feita a validação, o paciente tem permissão apenas para visualização de sua prescrição, demonstrada na Figura 20.

**Figura 20 – Visualização da prescrição médica gerada**


**Fonte: Autoria própria..**

Quando o paciente se dirige à farmácia para comprar seus remédios, o mesmo possuirá 2 opções para que o farmacêutico visualize sua prescrição. O farmacêutico poderá utilizar seu leitor de *QR Code*, por meio de equipamentos da farmácia ou telefone celular, ou receber o

mesmo *QR Code* do paciente e baixá-lo em seu computador. Para medidas de segurança, se o farmacêutico escolher por utilizar o telefone celular e escanear o código do e-mail do paciente, terá as mesmas permissões que o paciente tem, de apenas leituras, pois não terá passado pela validação de suas credenciais.

Caso o farmacêutico for utilizar a prescrição para a venda dos remédios, ele deverá receber o *QR Code* do paciente e, utilizando a interface da Figura 16, clicar em "Farmacêutico"para ser direcionado à página de verificação de seus dados, demonstrado na Figura 21.

**Figura 21 – Menu do Farmacêutico**


**Fonte: Autoria própria..**

Seguindo a mesma estrutura da validação do médico, um dos campos explicitados na Figura 14 foi retirado por uma configuração pré-estabelecida. Para o campo "Categoria", a opção "Farmacêutico"foi selecionada.

Uma vez realizada a validação, o farmacêutico será direcionado para a página de inclusão do *link* do *QR Code* baixado em seu computador. Desta forma, quando o mesmo realizar a inclusão, será direcionado à mesma interface de visualização que o paciente possui, porém com a diferença que poderá ao lado do campo "Quantidade", definido na Figura 20, será possível apenas diminuir a quantidade de cada remédio listado na prescrição.

Quando a quantidade vendida de um remédio na prescrição chegar a zero, o mesmo continuará na prescrição, porém apenas para leitura. Isso se deve pelas observações que cada remédio possui. O sistema ainda deve permitir que o paciente possa tirar suas dúvidas quanto à dosagem e outras recomendações. Porém, uma vez que todos os remédios da prescrição estiverem com o campo "Quantidade"zerado, a prescrição não terá mais sua visualização permitida, pois seu *status* terá sido alterado.

## 4.2 APRESENTAÇÃO DO *BACK-END*

Para a realização da *API*, como alguns campos foram pré-estabelecidos para o médico e farmacêutico, foi preciso compreender qual o valor das opções que o *site* havia atribuído para as alternativas e utilizar de seus respectivos nomes para que o programa *Puppeteer* realizasse a validação.

Utilizando do conjunto de ferramentas para programadores *web* incorporadas diretamente no navegador Google Chrome, os campos foram encontrados e, a partir do código abaixo, os campos foram preenchidos.

```
await page.select("#inscricao", "P");
await page.select("#tipoSituacao", "A");
await page.waitForTimeout(1000);
await page.select("#situacao", "A");

await page.click(
  "#buscaForm > div > div.row.my-4 >
                    div.col-md-2.text-right > button"
);
```

Para a *API* do farmacêutico, a mesma ideia foi seguida. No código abaixo, com a função *fetchPharmacists*. Dentro do escopo da *url* do *site* foi pré-incluída a opção "farmacêutico", sendo vista em "&categoria=farmaceutico&". Para as opções que serão inseridas na interface, são separados os campos por "${varivel}"ao qual recebem os valores digitados pelo farmacêutico.

```
export async function fetchPharmacists(pharmacist) {
    const { uf, city, crf } = pharmacist;
    const BASE_URL = `https://site.cff.org.br/farmaceutico`;
    const PARAMS = `?uf=${uf.toUpperCase()}&
    cidade=${city.toUpperCase()}&
    categoria=farmaceutico&
    nome=${crf}&search=1`;

async function execute() {
    const browser = await initBrowser();
    const page = await createPage(browser, BASE_URL + PARAMS);
    let hasPharmacists = false;
```

Para a conexão da *Blockchain*, foi utilizado todo o esqueleto do código do site oficial da *Hyperledger Fabric*. O código foi importado do *GitHub*[1] e sua conexão foi iniciada através do Docker.

O Docker foi fundamental para padronizar os requisitos da rede e de sua organização para que o código rodasse sem problemas. Uma vez que toda a conexão com a *Blockchain* foi realizada, foi necessário codificar o *Smart Contract* e suas principais funções para o sistema proposto. Para a criação, edição e visualização, foram necessárias 3 funções bases para o desenvolvimento do contrato.

A função abaixo *CreatePrescription* é utilizada para a realização da criação da prescrição. Essa função recebe os parâmetros de *id, cpf, nome, medicações e a senha* enviada por e-mail para que sejam incluídos dentro do *Smart Contract*. Além desses parâmetros, o *status* é criado, pois será o validador necessário para a visualização da prescrição. Os dados são alocados nas variáveis e enviados em formato *JSON*, para que o *Smart Contract* seja criado.

*JSON*, Notação de Objeto *JavaScript*, em português, é um formato leve de troca de dados. Uma forma simples para leitura e escrita do ponto de vista do programador, mas também para o computador de gerar e analisar os dados formados. É uma estrutura única, como um bloco quem contém informações múltiplas sobre algum tema. Para a Figura descrita, os dados são enviados em formato *JSON* pela biblioteca *Stringify*, pois converte os dados de *JavaScript* para *JSON*.

```
async CreatePrescription{
    const prescription = {
      ID: id,
      Cpf: cpf,
      Name: name,
      Medications: medications,
      SecretKey: secretKey,
      Status: "OPEN",
      DoctorName: doctorName,
      DoctorCrm: doctorCrm,
    };
}
await savePrivateData(ctx, id);
const prescriptionBuffer =
            Buffer.from(JSON.stringify(prescription));
ctx.stub.setEvent("CreatePrescription", prescriptionBuffer);
return ctx.stub.putState(id, prescriptionBuffer);
```

[1] https://github.com/hyperledger/fabric-samples/tree/main/asset-transfer-events

Uma vez que a criação for enviada, o código retornará uma mensagem caso houver sucesso ou erro no momento da criação do *Smart Contract* com a prescrição.

Outra função, dentre as 3 principais para o projeto, é a de visualização dos *Smart Contracts*. Nela há uma junção dos códigos de *back-end* e *front-end*, uma vez que os dados são puxados da *Blockchain* e expostos numa página *web*. A Figura **??** exemplifica os dois pontos do código ao qual a leitura do *Smart Contract* ocorre.

Na primeira parte, no *back-end*, a função de leitura é chamada pelo sistema e, a partir do id da prescrição, retornará ao *front-end* os dados em formato *JSON*. O segundo corte de código, da função *ReadPrescription*, diz respeito ao *front-end*. Neste pedaço, por meio da utilização da linguagem *JavaScript* é possível fazer a junção e visualização dos dados. Os dados são puxados pelo sistema e alocados em uma estrutura tabular, sendo possível visualizar pela Figura 20.

```
    async ReadPrescription(ctx, id) {
  const prescription = await readState(ctx, id);
  await addPrivateData(ctx, prescription.ID, prescription);
  return JSON.stringify(prescription);
}
```

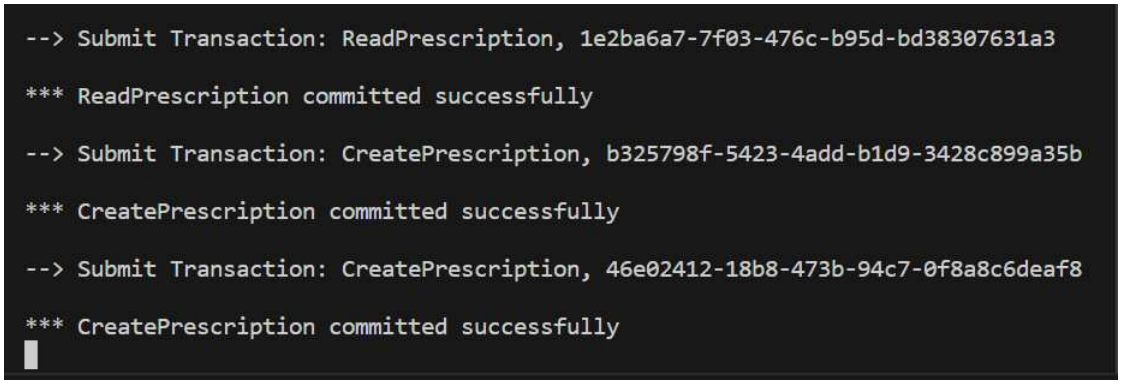
Por fim, a terceira função base do projeto é a de atualização da receita, demonstrada pelo nome *UpdatePrescription*. Seguindo a mesma linha da leitura, a atualização também é combinada com o *front-end*, uma vez que ao ser atualizada pelo farmacêutico, a função é invocada para que o campo "Quantidade"seja alterado. O *status* também é enviado como um parâmetro da função, pois em sua verificação, se a quantidade total for igual a zero, o *status* é alterado para "fechado"e a prescrição não poderá mais ser visualizada.

```
  async UpdatePrescription(ctx, id, medications, status) {
  const prescription = await readState(ctx, id);
  prescription.Medications = medications;
  prescription.Status = status;
  const prescriptionBuffer =
                Buffer.from(JSON.stringify(prescription));
  await savePrivateData(ctx, id);
  ctx.stub.setEvent("UpdatePrescription", prescriptionBuffer);
  return ctx.stub.putState(id, prescriptionBuffer);
}
```

## 4.3 RESULTADOS

Com a interface finalizada, e o *back-end* codificado, foi realizada a inserção de múltiplas prescrições e testado seu funcionamento na rede *Blockchain*. No terminal *Linux* do computador foram colocadas *flags*, sinais de alerta, para verificar se todas as funções dentro da *chaincode*, onde se localiza o *Smart Contract*, estavam funcionando sem erros. Como saída, está a Figura 22 e seus eventos realizados.

**Figura 22 – Leitura das funções no terminal *Linux***




**Fonte: Autoria própria..**

A partir da leitura dos eventos, foi realizada uma análise na ferramenta Docker para rastrear se os mesmos eventos estavam sendo inseridos na *Blockchain*. Para a realização da criação de um novo *Smart Contract* com a prescrição, a Figura 23. Os eventos listados pela Figura mencionam todos os passos realizados para a inclusão na *Blockchain*.

**Figura 23 – Criação do evento na *Blockchain***


**Fonte: Autoria própria..**

A primeira informação mostrada, da esquerda para a direita, é o metadado de tempo. A informação detalhada é que o evento todo, desde sua criação até a finalização durou 9 milissegundos. Há uma informação referente ao tipo de conexão da transação, na qual um aplicativo cliente pode chamar métodos diretamente em um aplicativo servidor de uma outra máquina, como se ela fosse um objeto local, facilitando a criação de aplicativos e serviços distribuídos.

Na sequência estão a submissão do evento, a recepção do bloco e a validação da transação. Por fim, a transação e o bloco são confirmados para sua criação da prescrição.

A partir do evento de criação, seguiu para a visualização da prescrição dentro do bloco. Os dados, criptografados, podem ser visualizados na Figura 24, na qual inicialmente se é possível visualizar o certificado da organização a qual a *Blockchain* está conectada. Esta organização é um dos primeiros níveis de segurança que a *Blockchain* possui, pois é necessário inserir sua chave privada para realizar a conexão com a rede. Em sequência está a prescrição criptografada, com seu id, não criptografado, na primeira linha, seguido dos dados da prescrição criptografados utilizando a criptografia SHA256.

**Figura 24 – Bloco contendo as informações na *Blockchain***


**Fonte: Autoria própria..**

O último evento verificado foi a de visualização da prescrição, que possui uma leve diferença relacionado ao primeiro evento, o de criação. A principal diferença é que o evento é chamado 2 vezes pelo sistema. Os dois eventos são realizados com os mesmos passos, o qual realizam a conexão gRPC, lêem a *chaincode*, realizam a submissão da ordem do evento e a validação do bloco. Todos os eventos sumarizam primeiramente o endossamento da transação e depois seu envio para a rede *Blockchain*.

Para a visualização de sua funcionalidade referente à descentralização, foi inicializada uma segunda organização, independente da primeira organização demonstrada na Figura 26. Inicializada na primeira conexão com o Docker, foi verificada, a partir da Figura 25, que a segunda organização criada também é computada os mesmos eventos de criação do bloco, no mesmo tempo em que foi criado na organização 1. Desta forma, o sistema mostrou-se ser des-

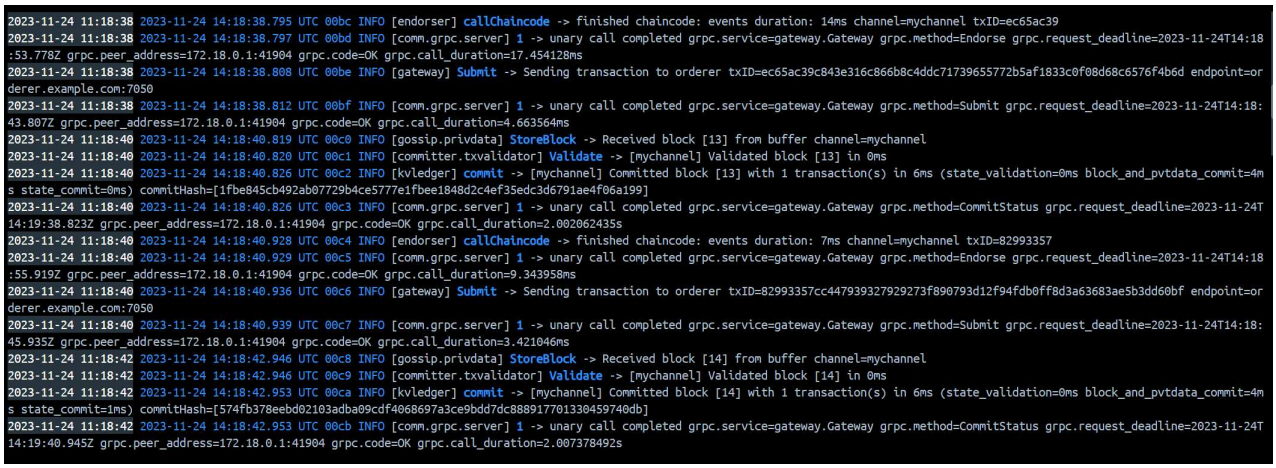
centralizado, uma vez que seus eventos ocorrem de forma síncrona em múltiplas organizações conectadas à *Blockchain*.

**Figura 25 – Visualização do evento na organização 2, comprovando seu evento foir ealizado de forma descentralizada**


**Fonte: Autoria própria..**

Porém, não somente as funções descritas são únicas no código em sua versão final. Tanto o *front-end* quando o *back-end* podem ser visualizados no *link* do *GitHub*: *https://github.com/Thiago-Furtado/tcc-prescricao-inteligente*

**Figura 26 – Visualização do bloco no Docker**




**Fonte: Autoria própria..**

# 5 CONCLUSÃO

O projeto de inclusão de prescrições médicas *online* em uma rede descentralizada *Blockchain* surgiu como uma alternativa para o sistema atual. Um sistema de prescrições *online*, que até meados de 2020 não se era muito utilizado pelas farmácias, atualmente vem crescendo substancialmente diante da digitalização do dos processos a nível mundial. Com isso, o projeto apresentado teve como foco a inclusão do sistema de prescrições médicas atual em uma rede descentralizada *Blockchain*, um formato diferente do que há atualmente devido à seus métodos de armazenamento e segurança.

A funcionalidade da rede proposta foi um grande desafio, visto que se trata de uma tecnologia ainda pouco explorada e documentada, quando comparada à outras mais difundidas. Além da funcionalidade, compreender corretamente como utilizá-la e como configurá-la, seguindo as medidas de segurança que a proposta deveria ter, foi desafiador.

É possível reconhecer que alguns aspectos da proposta podem e devem ser ajustados, caso seja um método a ser implementado no futuro. Por se tratar de um desenvolvimento *full stack*, na qual há um desenvolvimento que vai desde a criação da rede, em seu *back-end*, até a interface, *front-end*, foi preciso utilizar de uma rede *Blockchain* base. Em seu *site* oficial, a *Hyperledger Fabric* inclui *links* no *GitHub* com redes *Blockchain* configuradas. Foi feito o *upload* de uma rede e configurada para a utilização do projeto.

Também na interface, melhorias à nível de segurança *web* são possíveis e necessárias. Isso se dá pelo fato da validação do farmacêutico de editar a prescrição. Para o sistema atual, existe uma verificação simples, mas que pode ser melhorada em projetos futuros. Uma proposta de melhoria seria utilizar *Tokens JWT*, similar aos utilizados pelo *Facebook*. Neste tipo de segurança, a senha passada pelo farmacêutico ou paciente seria comparada com a senha criptografada no banco de dados e, uma vez válida, retornaria um *token* com um período de expiração. Com o *token*, seria possível acessar informações protegidas, como seria o caso da prescrição.

Porém, em termos de vantagem, uma rede descentralizada *Blockchain* pode ser uma excelente alternativa para o futuro, uma vez que pode-se ter um controle a partir da prescrição original, não uma cópia, como se é feito no sistema atual. Além da possibilidade de combiná-las com outras tecnologias, incluir funcionalidades como assinaturas digitais e chaves simétricas, que iriam favorecer os níveis de segurança do sistema.

O projeto proposto alcançou o objetivo inicial, realizando a criação e desenvolvimento de uma aplicação *web* em conjunto de uma rede descentralizada *Blockchain*, permitindo a visualização de um *Smart Contract* com dados de uma prescrição *online*. Num trabalho futuro, as questões já citadas de segurança poderiam ser melhoradas, além de outras funcionalidades estruturais e estéticas.

# REFERÊNCIAS


AHMAD, R. W. *et al.* The role of blockchain technology in telehealth and telemedicine. **International journal of medical informatics**, Elsevier, v. 148, p. 104399, 2021.

ALHARBY, M.; MOORSEL, A. V. Blockchain-based smart contracts: A systematic mapping study. **arXiv preprint arXiv:1710.06372**, 2017.

ANDERSON, C. Docker [software engineering]. **IEEE Software**, v. 32, n. 3, p. 102–c3, 2015.

BRUSH *et al.* **asymmetric cryptography (public key cryptography)**. 2021. Disponível em: https://www.techtarget.com/searchsecurity/definition/asymmetric-cryptography.

CARDOSO, A. M. Implantação de prescrição eletrônica a fim de otimizar a dispensação de medicamentos. **Revista Brasileira de Farmácia Hospitalar e Serviços de Saúde**, v. 4, n. 4, 2013.

CFF. **Nova plataforma de prescrição facilita a conexão entre farmacêuticos, médicos e pacientes**. 2020. Disponível em: https://www.cff.org.br/noticia.php?id=6579.

CFF. **Prescrição eletrônica de medicamentos sujeitos a controle especial**. 2020. Disponível em: https://www.cff.org.br/noticia.php?id=5674.

CFF. **Prescrição eletrônica de medicamentos sujeitos a controle especial**. 2020. Disponível em: https://www.cff.org.br/noticia.php?id=5674.

CFM. **Prescrição Eletrônica**. 2022. Disponível em: https://sistemas.cfm.org.br/prescricaoeletronica/.

CHUNG, C. **Introduction to Hashing and its uses**. 2022. Disponível em: https://www.2brightsparks.com/resources/articles/introduction-to-hashing-and-its-uses.html.

CRIPTOFÁCIL. **Blockchain pública, privada e híbrida: entenda as diferenças entre elas**. 2021. Disponível em: https://www.criptofacil.com/blockchain-publica-privada-e-hibrida-entenda-as-diferencas-entre-elas/.

DHIEB, N. *et al.* A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement. **IEEE Access**, IEEE, v. 8, p. 58546–58558, 2020.

DOCKER, I. Docker. **Inea].[Junio de 2017]. Disponible en: https://www. docker. com/what-docker**, 2020.

DWORKIN, M. J. Sha-3 standard: Permutation-based hash and extendable-output functions. Morris J. Dworkin, 2015.

FERDOUS, M. S. *et al.* Blockchain consensus algorithms: A survey. **arXiv preprint arXiv:2001.07091**, 2020.

FOUNDATION, H. **Hyperledger Foundation**. 2015. Disponível em: https://www.hyperledger.org/projects.

GATTESCHI, V. *et al.* Blockchain and smart contracts for insurance: Is the technology mature enough? **Future internet**, MDPI, v. 10, n. 2, p. 20, 2018.

GREVE, F. G. *et al.* Blockchain e a revolução do consenso sob demanda. **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos**, 2018.

HE, M. *et al.* Blockmeds: A blockchain-based online prescription system with privacy protection. *In*: SPRINGER. **International Conference on Service-Oriented Computing**. [*S.l.*], 2019. p. 299–303.

JANI, S. Smart contracts: Building blocks for digital transformation. **Indira Gandhi National Open University**, 2020.

JENA, B. K. **Digital Signature Algorithm (DSA) in Cryptography: How It Works and Advantages**. 2022. Disponível em: https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm.

JUSTICE U.S. ATTORNEY'S OFFICE, D. o. M. Department of. Cvs to pay 3.5 million to resolve allegations that pharmacists filled fake prescriptions. **United States Department of Justice**, 2016.

MA, Y. *et al.* A survey of blockchain technology on security, privacy, and trust in crowdsourcing services. **World Wide Web**, Springer, v. 23, n. 1, p. 393–419, 2020.

MORELAND, K. **What is Blockchain?** 2022. Disponível em: https://www.ledger.com/academy/blockchain/what-is-blockchain.

NARAYANAN, A. *et al.* Bitcoin and cryptocurrency technologies. **Curso Elaborado Pela**, 2021.

NGUYEN, G.-T.; KIM, K. A survey about consensus algorithms used in blockchain. **Journal of Information processing systems**, Korea Information Processing Society, v. 14, n. 1, p. 101–128, 2018.

PENG, G. *et al.* Equipment life cycle management based on private blockchain and smart contract. *In*: **Engineering Assets and Public Infrastructures in the Age of Digitalization**. [*S.l.*]: Springer, 2020. p. 339–348.

PRANTO, T. H. *et al.* Blockchain and smart contract for iot enabled smart agriculture. **PeerJ Computer Science**, PeerJ Inc., v. 7, p. e407, 2021.

RHODES, D. **Asymmetric Encryption: An Introduction To Asymmetric Cryptography**. 2020. Disponível em: https://komodoplatform.com/en/academy/asymmetric-encryption/.

ROUHANI, S.; DETERS, R. Security, performance, and applications of smart contracts: A systematic survey. **IEEE Access**, IEEE, v. 7, p. 50759–50779, 2019.

SSL. **Symmetric vs. Asymmetric Encryption – What are differences?** 2018. Disponível em: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences?ref=hackernoon.com.

TEAM, Z. **Guide to Hybrid Blockchain, Benefits and Use Cases**. 2022. Disponível em: https://www.zeeve.io/blog/guide-to-hybrid-blockchain-benefits-and-use-cases/.

VALE, S. **Entenda como a Hyperledger está desenvolvendo o ambiente de blockchain para as empresas**. 2020. Disponível em: https://www.voitto.com.br/blog/artigo/hyperledger-a-solucao-blockchain-para-empresas.

VEJA. **Prescrição médica eletrônica: como isso pode melhorar minha saúde?** 2019.

WU, K. An empirical study of blockchain-based decentralized applications. **arXiv preprint arXiv:1902.04969**, 2019.

YAGA, D. *et al.* Blockchain technology overview. **arXiv preprint arXiv:1906.11078**, 2019.

ZHENG, Z. *et al.* An overview of blockchain technology: Architecture, consensus, and future trends. *In*: IEEE. **2017 IEEE international congress on big data (BigData congress)**. [*S.l.*], 2017. p. 557–564.

ZHENG, Z. *et al.* Blockchain challenges and opportunities: A survey. **International journal of web and grid services**, Inderscience Publishers (IEL), v. 14, n. 4, p. 352–375, 2018.

ZHENG, Z. *et al.* An overview on smart contracts: Challenges, advances and platforms. **Future Generation Computer Systems**, Elsevier, v. 105, p. 475–491, 2020.

ZOU, W. *et al.* Smart contract development: Challenges and opportunities. **IEEE Transactions on Software Engineering**, IEEE, v. 47, n. 10, p. 2084–2106, 2019.