**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS**
**UNIDADE ACADÊMICA DE GRADUAÇÃO**
**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**LEONARDO REGINATO**

# CRIAÇÃO DE UM *FRAMEWORK* DE ACOMPANHAMENTO E CONTROLE DE DADOS DE PACIENTES DO SISTEMA ÚNICO DE SAÚDE (SUS) UTILIZANDO A TÉCNICA BLOCKCHAIN.

**São Leopoldo**
**2021**

LEONARDO REGINATO

# CRIAÇÃO DE UM FRAMEWORK DE ACOMPANHAMENTO E CONTROLE DE DADOS DE PACIENTES DO SISTEMA ÚNICO DE SAÚDE (SUS) UTILIZANDO A TÉCNICA BLOCKCHAIN.

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação, pelo Curso de Engenharia da Computação da Universidade do Vale do Rio dos Sinos - UNISINOS

Orientador(a): Prof(a). Dra. Ana Paula Mallmann

São Leopoldo

2021

## AGRADECIMENTOS

Este trabalho é dedicado aos meus pais Simone e Jair, pois este trabalho só foi possível graças aos seus eternos incentivos educacionais, além do amor, exemplo e apoio que me trouxeram até este momento. Minha irmã Larissa que sempre foi motivo de orgulho para mim e me inspirou a ser uma pessoa melhor.

Gostaria de agradecer também minha noiva Débora que sempre esteve ao meu lado, me ajudando de forma direta e indireta na realização deste trabalho, além de ser minha melhor amiga e parceira.

Não posso deixar de mencionar e agradecer minha orientadora professora Ana, que aceitou me orientar durante o desenvolvimento deste trabalho. Agradeço pelos seus incentivos, paciência e ajuda constante.

## RESUMO

Atualmente o Sistema Único de Saúde (SUS) do Brasil é referência mundial em saúde pública e conta com diversos programas que levam qualidade de vida e bem estar à população do país. Contudo, apesar de estar na vanguarda neste tipo de sistema, o SUS apresenta algumas possibilidades de melhoria em relação à sua gestão de dados e informações, como por exemplo a capacidade de identificação de histórico de consultas e gastos médicos de pacientes usuários do sistema. Por isso, este trabalho visa elaborar uma estratégia de controle de dados dos pacientes através de um *framework* específico e utilizando a técnica *blockchain*. Como o objeto principal deste fluxo de informações são os dados gerados durante os atendimentos aos usuários do sistema, tem-se como centro de manipulação das informações esse arquivo criado. Através de um fluxo proposto, desenvolveu-se um programa de simulação utilizando a linguagem de programação *javascript*, para que as etapas do processo pudessem ser analisadas e posteriormente testadas. Como uma rede de dados *blockchain* necessita de uma grande segurança das informações que ali participarão, métodos de criação de chaves de acesso foram utilizados para esta parte de autenticação de usuários (ou nodos) na rede particular do paciente. Com o programa criado foi possível realizar diversos testes para validar o funcionamento e a segurança da rede proposta e na adição de novos dados nos blocos que a constituem. A partir disso pode-se elencar que uma rede privada com acesso restrito utilizando a técnica *blockchain* pode aumentar a segurança das informações dos usuários do sistema público de saúde, além de ser possível possuir um histórico médico confiável e que não seja passível de alterações ou adulterações das informações, levando a uma maior confiabilidade da população e ainda um aumento de performance da gestão pública. Os resultados mostraram que a segurança é de fundamental importância no novo processo, pois a privacidade dos dados deve ser levada em consideração. Com a adição de uma nova estrutura de programa no fluxo já existente no SUS, foi possível garantir que a informação esteja descentralizada e que não seja possível manipulá-la ou edita-la após estar adicionada ao bloco e a rede.



**Palavras-chave:** Blockchain. Segurança. DATASUS. Sistema Único de Saúde. SUS.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE SIGLAS

| | |
|---|---|
| APAC | Autorização de Procedimento de Alta Complexidade |
| AIH | Autorização de Internação Hospitalar |
| BDN | Banco de Dados Nacional |
| BPA | Boletim de Produção Ambulatorial |
| CGSI | Coordenação Geral de Sistemas de Informação |
| CID | Classificação Internacional de Doenças |
| CIH | Comunicação de Internação Hospitalar |
| CNES | Cadastro Nacional de Estabelecimentos de Saúde |
| FPO | Ficha de Programação Orçamentária |
| GIH | Guia de Internação Hospitalar |
| GM/MS | Ministério da Saúde Gabinete do Ministro |
| ICP-Brasil | Brasil Infraestrutura de Chaves Públicas Brasileira |
| OPM | Órtese, prótese e meios auxiliares de locomoção |
| RAAS | Registro das Ações Ambulatoriais de Saúde |
| RAS-PSI | Registro das Ações de Saúde Psicossocial |
| SIA | Sistema de Informação Ambulatorial |
| SIAB | Sistema de Informação da Atenção Básica |
| SIAB | Sistema de Informações de Atenção Básica |
| SIGTAP | Sistema de Gerenciamento da Tabela de Procedimentos, Medicamentos e OPM do SUS |
| SIH | Sistema de Informação Hospitalar |
| SIHD | Sistema de Informação Hospitalar Descentralizado |
| SIM | Sistema de Informações sobre Mortalidade |
| SINAN | Sistema de Informações de Agravos de Notificações |
| SINASC | Sistema de Informações sobre Nascidos Vivos |
| SISAIH | Sistema Gerador do Movimento das Unidades Hospitalares |
| SISAIH01 | Programa de Apoio a Entrada de Dados das Autorizações de Internações Hospitalares |
| SUS | Sistema Único de Saúde |

**SUMÁRIO**

# 1 INTRODUÇÃO

O presente trabalho aborda o estudo de utilização de um método de controle e gerenciamento de informações utilizando as tecnologias atuais para aplicações em dados de pacientes do Sistema Único de Saúde (SUS). Este estudo delimita-se à técnica *blockchain* e como ela pode ser empregada para melhorar o acesso a informação e controlar os dados dos pacientes do Sistema Único de Saúde no Brasil.

O objetivo de aplicar este tipo de tecnologia visa aumentar a efetividade no controle das doenças, internações e conhecer o histórico da saúde do paciente de forma mais precisa. Para diagnosticar com maiores chances de sucesso, pretende-se auxiliar também a evitar reações indesejáveis, efeitos colaterais e possíveis diagnósticos tardios, além de diminuir custos através do corte de gastos desnecessários e em excessos por parte do sistema de saúde.

De forma a complementar o objetivo geral, definiu-se objetivos específicos para colaborar no direcionamento e foco do trabalho.

a) Definir o método de controle e rastreamento do *blockchain*;
b) Avaliar a aplicação na área da saúde e suas relações;
c) Relacionar os dados de informações simples com informações específicas.

Os pacientes que são atendidos nas unidades devem ter um atendimento de qualidade e um correto acompanhamento com um histórico de fácil acesso e que seja confiável, independente do profissional prestador do serviço e da unidade de atendimento.

O trabalho inicia por apresentar ao leitor algumas informações importantes sobre os objetos de estudo, sendo eles a técnica *blockchain* e o Sistema Único de Saúde do Brasil. Após estas definições, é iniciada a metodologia proposta como ideia para ser agregada ao já existente fluxo de informações do sistemas de saúde. Por fim, são realizados testes com a ideia proposta e é feita uma análise dos dados encontrados.

# 2 FUNDAMENTAÇÃO TEÓRICA

Para que a metodologia deste trabalho possa ser aplicada, devem ser introduzidos conceitos sobre a técnica *blockchain*, controle de dados pessoais e exames na área da saúde e como funciona o SUS. Após entender o funcionamento e fluxo de informações em uma unidade de atendimento de saúde, juntamente com a tecnologia escolhida para implementação, um novo *framework* e fluxo de dados pode ser criado.

## 2.1 O que é *blockchain*

O autor Bashir (2018) diz que a técnica *blockchain* foi primeiramente apresentada a sociedade em meados de 2008 com o surgimento da moeda eletrônica *bitcoin*. Ela teria um impacto não somente no setor financeiro, mas também em muitos outros como governo, mídia e arte.

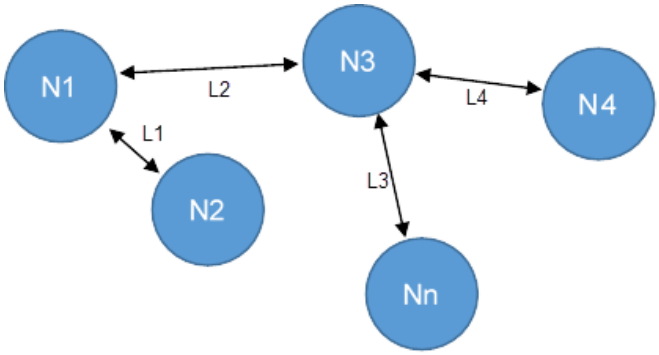
Segundo Gupta (2017), o *blockchain* é um sistema de registro distribuído e compartilhado que facilita o processo de registrar transações e rastrear bens em uma rede de negócios. Um bem pode ser tangível, como um carro, uma casa e etc., ou intangível como por exemplo uma patente ou uma marca. Virtualmente tudo pode ser rastreado e trocado dentro de uma rede de *blockchain*.

### 2.1.1 Sistemas distribuídos

Os sistemas distribuídos são de suma importância para entender o conceito da tecnologia *blockchain*, pois ele é um sistema distribuído em sua concepção. É um registro centralizado ou descentralizado. Porém, foi construído para que fosse trabalhado de forma descentralizada. Neste caso, dois ou mais nodos trabalham juntos de forma coordenada para que seja possível atingir um determinado resultado em conjunto. Os sistemas distribuídos são construídos para que o usuário final veja o processo parecendo ver somente um único processo lógico.

Um nodo, segundo Bashir (2018), pode ser definido como um único representante em um sistema distribuído. Todos os nodos podem receber e enviar mensagens uns para os outros. Eles ainda possuem memória e um processador. Um exemplo de um pequeno sistema distribuído de 5 nodos é mostrado na Figura 1.

Figura 1 - Exemplo de um pequeno sistema distribuído




Fonte: Elaborado pelo autor

Ainda segundo Bashir (2018), os sistemas distribuídos são tão desafiadores de construir que uma hipótese conhecida como teorema CAP (consistência, disponibilidade e tolerância à partição, do inglês *consistency, availability and partition tolerance*) provou-se válida, a qual traz que sistemas distribuídos não podem ter mais de 3 objetivos de saídas simultaneamente. Ou seja, os sistemas distribuídos buscam com que consistência, disponibilidade e partições sejam utilizadas de forma igualitária pelo sistema.

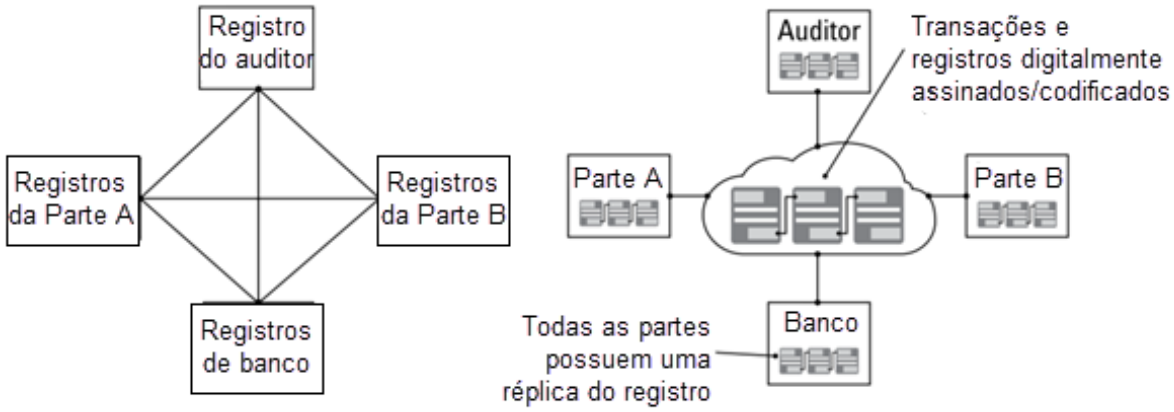
### 2.1.2 Redes de negócios antes x depois do *blockchain*

Nos métodos tradicionais de registros de transações e rastreamento de bens, os participantes de uma rede possuem seus próprios controles de registros. Estes métodos são muitas vezes caros por justamente envolver terceiros que cobram pelos serviços de intermediação. É ineficiente e custoso de tempo pois claramente tem-se um gasto elevado até as partes estarem de acordo e também por ter que atualizar todos seus registros de dados.

Com o *blockchain* a arquitetura dessa rede provém a todos os participantes a habilidade de compartilhar um mesmo registro. Este registro é atualizado, através da replicação *peer-to-peer*, toda vez que uma transação ocorre na rede.

*Peer-to-peer* significa que não há um controle central na rede, mas todos os participantes conversam entre si diretamente. Os registros são sincronizados ao longo de toda a rede assim que um dado é transferido de um novo para outro. A Figura 2 mostra as diferentes arquiteturas, sem e com *blockchain* implementado.

Figura 2 - Redes de negócios antes e depois do blockchain




Fonte: Gupta (2017)

As transações e registros que ocorrem ao longo da rede são assinadas e codificadas digitalmente, ao passo que todas as partes envolvidas recebem uma cópia deste registro. Já no modelo tradicional, não existe esse rastreamento de informações e dados e cada parte controla seus próprios registros.

### 2.1.2 Aplicação na área da saúde

Pode-se nomear o chamado *blockchain* 3.0 quando aplicado em áreas que não as financeiras, surgindo para nomear estas aplicações da tecnologia distribuída. Podemos usar como exemplo a aplicação no ciclo de pesquisa científica, como análises, experimentos e publicações. As aplicações podem ser implementadas como as chamadas “sem permissão”, onde cada usuário pode participar, ou a “permitida”, que abrange somente pesquisadores ou instituições autorizadas na rede.

Para analisar com precisão o uso da tecnologia nas aplicações biomédicas e na área da saúde, KUO et al. (2017) dizem que deve-se comparar o *blockchain* com os sistemas de gerenciamento de banco de dados tradicionais, como por exemplo sistemas baseados na linguagem SQL (Linguagem em Fila Estruturada, do inglês *Structured Query Language*) como o banco de dados da empresa *Oracle*® e sistemas que utilizam a linguagem *NoSQL* como por exemplo *Apache Cassandra*.

Quando comparados os sistemas de banco de dados tradicionais com o *blockchain*, o primeiro benefício chave é o gerenciamento descentralizado. Enquanto os primeiros são logicamente gerenciados de forma centralizada, o segundo já é uma tecnologia *peer-to-peer*. Por exemplo, cada nodo na rede pode exercer sua função

independentemente de outro enquanto segue os protocolos cabíveis. Ainda segundo os autores KUO et al. (2017), o *blockchain* é adequado para aplicações onde as partes interessadas (hospitais, pacientes e provedores de serviços) desejam colaborar entre si sem ceder o controle a um gerenciamento central como intermediário.

Outro ponto de grande utilidade para a área é o caminho de auditoria imutável. Ou seja, enquanto os bancos de dados clássicos permitem as funções de CRIAR, LER, ATUALIZAR e DELETAR, o *blockchain* somente suporta as funções CRIAR E LER. Embora seja possível mas ainda assim difícil, modificar os registros de dados.

Dentre as aplicações na área da saúde, a mais em alta é relacionada à adoção do *blockchain* como infraestrutura para a troca de informações de saúde, do inglês *Health Information Exchange (HIE)*. Estas informações podem ser então categorizadas baseadas em seus objetivos principais. São elas:

- Melhora no gerenciamento de registros médicos;
- Processo de seguros aprimorado;
- Pesquisa biomédica/clinica acelerada.

Portanto, como pode ser visto nas informações acima, a aplicação do *blockchain* na área da saúde resulta em diversos benefícios, tanto para instituições de pesquisa quanto hospitais públicos e privados. Os próximos capítulos explicam o que é o sistema público de saúde no Brasil e quais as tecnologias envolvidas.

## 2.2 O Sistema Único de Saúde (SUS)

Formalmente o SUS foi instituído na Constituição Federal de 1988, mas sua origem é bem mais antiga e data do século XX com suas origens na crise do modelo médico assistencial privatista. Segundo CONASS (2006), as diversas mudanças políticas e econômicas nas décadas de 70 e 80 determinaram o esgotamento do modelo, levando ao surgimento de sujeitos que propunham modelos alternativos de atenção à saúde pública no país.

O SUS é formado pelas ações e serviços sob domínio da gestão pública. É atuante em todo o território nacional e organizada em redes regionalizadas e hierarquizadas. De acordo com CONASS (2011), a constituição brasileira estabelece que a saúde é um dever do estado. Entende-se por estado não somente no âmbito federativo, mas também cabível a União, os estados, o Distrito Federal e todos os municípios.

Tomando por base a Lei n. 8.080/90 (BRASIL, 1990) que determina em seu artigo 9° que cada esfera do governo tem seu dever de exercer a direção do SUS de acordo com cada órgão:

I. no âmbito da União, pelo Ministério da Saúde;
II. no âmbito dos estados e do Distrito Federal, pela respectiva Secretaria de Saúde ou órgão equivalente;
III. no âmbito dos municípios, pela respectiva Secretaria de Saúde ou órgão equivalente.

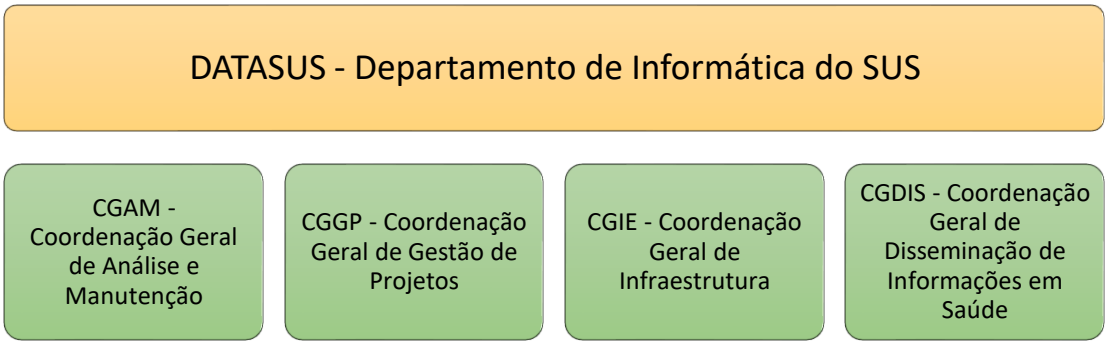
No Brasil tem-se como diretiva o acesso universal à saúde, neste caso através do SUS, significando que todos os brasileiros podem alcançar uma ação ou serviço de saúde de que necessitem sem qualquer barreia, seja legal, econômica, física ou cultural. Outro ponto importante é que a classe social indifere neste caso, ou seja, pobres ou ricos, homens ou mulheres, velhos ou crianças, todos devem ter acesso à saúde no país.

### 2.2.1 Departamento de Informática do SUS (DATASUS)

O DATASUS surgiu em 1991 e iniciou exercendo a função de controle e do processamento das contas relacionadas à saúde. Após estruturado, o DATASUS tornou-se responsável por prover aos órgãos do SUS os sistemas de informação e de suporte a informática.

Segundo o CONASS (2011), nos dias atuais, o departamento, através de seus sistemas, auxilia diretamente o Ministério da Saúde, bem como provém sistemas de software para as secretarias estaduais e municipais. Por esse motivo, ele está presente em todas as regiões do Brasil através das chamadas Regionais. A estrutura na qual o DATASUS pertence pode ser verificada na Figura 3.

Figura 3 - A estrutura na qual o DATASUS está inserido




Fonte: Elaborado pelo autor

As informações e dados tratados pelo DATASUS ficam armazenadas em duas salas-cofre localizadas nos estados do Rio de Janeiro e Brasília, onde estão os servidores do departamento. Estes servidores são os que hospedam a maior parte dos sistemas do Ministério da Saúde.

Hoje existem diversos sistemas criados pelo DATASUS, que vão desde sistemas hospitalares, ambulatoriais e epidemiológicos até sistemas de gestão, financeiros e estruturantes. Dentre todos vale destacar o sistema e-SUS Hospitalar, que tem como objetivo o controle completo das mais diversas camadas do SUS para facilitar o controle e organização do trabalho dos profissionais da saúde.

### 2.2.2 e-SUS Hospitalar

O Ministério da Saúde possui o sistema e-SUS Hospitalar, que realiza a gestão, controle dos processos e integra os setores do hospital. O foco vai desde a admissão do paciente até sua alta. Ele foi projetado para que seja realizado um atendimento positivo e de qualidade, além de auxiliar na melhora do desempenho da gestão hospitalar.

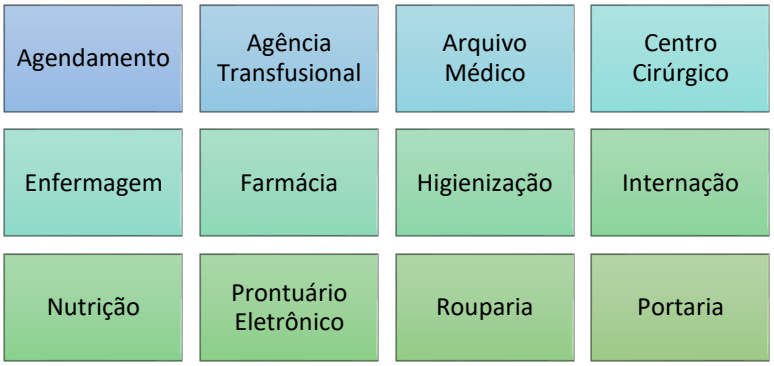
O sistema é construído em tecnologia *web* em linguagem .NET, que utiliza um banco de dados *Oracle/SQL Server* e os processos e fluxos de informações são feitos através de um *workflow designer*. O e-SUS Hospitalar é composto por dois principais blocos: o HIS (*Hospital Information System*) e PEP (Prontuário Eletrônico de Paciente).

- HIS: É o sistema central de informações dos dados de pacientes, como admissão, alocação de leito, medicações administradas e de integração de quadro clínico e/ou terapia.

- PEP: É um banco de dados ou repositório onde pode-se encontrar todas as informações clínicas e de saúde de toda a vida de um indivíduo.

O Sistema possui uma completa estrutura e abrange as diversas áreas do Hospital, como pode ser visto na Figura 4. De acordo com Perche (2020) o e-SUS Hospitalar é um sistema completo que oferece uma ferramenta para geração de relatórios, formulários diversos baseados em metadados, painel de indicadores sobre a gestão e sistemas de laboratório e ERP (do inglês *Enterprise Resource Planning* que traduzido significa Planejamento dos Recursos Empresariais).

Figura 4 - Áreas nas quais o sistema e-SUS Hospitalar pode ser utilizado




Fonte: Elaborado pelo autor

Atualmente o sistema não contempla as tecnologias PACS e RIS. Da sigla em inglês PACS - *Picture Archiving and Communication System* (Sistema de Comunicação e Arquivo de Imagem) que captura, exibe e distribui imagens médicas e RIS - *Radiology Information System* (Sistema de Informação de Radiologia) usado para automatizar o fluxo de trabalho de uma clínica radiológica.

### 2.2.3 Segurança de dados em saúde

Na área da saúde deve-se garantir a privacidade e segurança das informações aos usuários. Especificamente "*para o sistema de saúde, a quebra de confiança entre indivíduos provoca a queda na confiabilidade do próprio sistema.*" (CONASS, 2011). Atualmente está em vigor o "Manual de Certificação para Sistemas de Registro Eletrônico em Saúde", disponível no portal da Sociedade Brasileira de Informática em Saúde – Sbis. Podem-se destacar várias questões deste manual, ressaltando as seguintes:

- Podem ser utilizados sistemas informatizados para a guarda e o manuseio de prontuários de pacientes e para a troca de informação identificada em saúde, eliminando a obrigatoriedade do registro em papel, desde que esses sistemas atendam integralmente aos requisitos do nível de garantia de segurança 2 – NGS2, estabelecidos no Manual de Certificação para Sistemas de Registro Eletrônico em Saúde;
- O nível de garantia de segurança 2 – NGS2 exige o uso de assinatura digital, devendo utilizar certificado digital (padrão ICP-Brasil (Infraestrutura de Chaves Públicas Brasileira)).

Estas informações evidenciam que é possível realizar a informatização de informações na área da saúde, porém com diversos cuidados e seguindo diretrizes estabelecidas pelas áreas cabíveis. O cuidado do DATASUS com este tópico é apresentado no capítulo a seguir.

### 2.2.4 Gerenciamento de dados públicos em saúde

Diariamente o SUS registra e salva dados de saúde gerados por todo o país em dois sistemas distintos: o Sistema de Informações Hospitalares (SIH) e o Sistema de Informação Ambulatorial (SIA). No primeiro sistema são armazenados os dados que fazem referência às internações realizadas pelo SUS, sendo coletado via instrumento chamado Autorização de Internação Hospitalar (AIH).

O SIH nasceu em 1981 substituindo em 1982 o sistema GIH (Guia de Internação Hospitalar) e centralizando as informações do novo documento de AIH. O tratamento das AIH’s continuou centralizado até abril de 2006 onde foi descentralizado para os gestores de Secretaria de Saúde.

O escopo do AIH, do sistema SIH, é realizar o registro de todos os atendimentos realizados a partir das internações hospitalares financiadas pelo SUS. Sendo assim, os gestores podem gerar relatórios para realização do pagamento dos estabelecimentos de saúde. A centralização das informações geradas via AIH acontecia pelo Ministério da Saúde estritamente via DATASUS. Porém, no ano de 2004 através da Portaria GM/MS nº. 821/04, instaurou-se o desenvolvimento de um sistema descentralizado para corroborar com o chamado princípio da autonomia e da gestão local no SUS. No mês de dezembro de 2004 o Sistema de Informações Hospitalares Descentralizadas (SIHD ou SIHD1) foi finalizado e iniciaram-se suas

implantações nos municípios e estados brasileiros. O SIHD1 foi utilizado até dezembro de 2007.

O Sistema de Informações Hospitalares Descentralizadas 2 (SIHD2) realiza o agrupamento de todas as informações das redes nos âmbitos estaduais e municipais e envia para o nível Federal. O SIHD2 substituiu a sua primeira versão a partir do ano de 2008, ocorrendo pela nova versão do Manual do SIH para facilitar o processo de implantação da Tabela Unificada de Procedimentos, Medicamentos, Órteses e Próteses e Materiais Especiais do SUS.

Os principais objetivos do SIHD2 são realizar uma melhor obtenção dos registros de atendimentos aos usuários internados nos estabelecimentos de saúde do SUS e orientar as gestões dos estados e municípios quanto as novas informações contidas na Tabela Unificada de Procedimentos, Medicamentos, Órteses e Próteses e Materiais Especiais do SUS.

Já o SIA é o responsável por armazenar os dados de todos os tipos de procedimentos ambulatoriais feitos no SUS, tendo como fontes: o Boletim de Produção Ambulatorial (BPA), a Autorização de Procedimentos de Alta Complexidade (APAC) e o Registro de Ações Ambulatoriais em Saúde (RAAS).

O RAAS é um sistema desenvolvido pelo DATASUS cujo objetivo é informatizar as entradas de dados referentes às ações ambulatoriais de saúde. Atualmente ele se divide em duas áreas: o Registro das Ações de Saúde da Atenção Domiciliar (RAS-AD) e o Registro das Ações de Saúde Psicossocial (RAS-PSI). Este é um sistema multiusuário, ou seja, podem existir vários usuários cadastrados, e inclusive operando-o simultaneamente em rede. A instalação do sistema começa pelo gerenciador de banco de dados FIREBIRD e então o download e instalação do RAAS. O software possui três abas principais para coleta de dados: [P] Identificação do Paciente; [A] Identificação do Atendimento; [R] Dados das Ações Realizadas.

O sistema APAC registra dados individuais, identificando o usuário do SUS atendido, assim como todas as ações que foram realizadas durante o seu tratamento. Também registra os dados de sua situação de saúde através da Classificação Internacional de Doenças (CID). A instalação do APAC é realizada através do APAC Magnético sob o já instalado sistema SIA.

O BPA é um instrumento de registro com origem no SIA de forma individualizada. Ele mantém o registro das ações ambulatoriais de forma consolidada. Assim como o RAAS, a instalação do sistema se inicia pelo gerenciador de banco de

dados FIREBIRD e então o download e instalação do BPA. Como o FIREBIRD possivelmente já está instalado na máquina, não é necessário realizar este passo novamente. Um usuário MESTRE (instalador do sistema) pode na primeira instalação criar novos usuários que utilizarão o aplicativo. Após informar os dados do novo usuário, deve-se escolher as funcionalidades que estarão disponíveis para este. É possível então realizar a personalização dos acessos a partes ou funcionalidades especificas do sistema, por exemplo: registro de atendimentos, emissão de relatórios, importação e exportação de arquivos. A interface do BPA contém um conjunto de funcionalidades como: registrar ações, emitir relatórios, trocar arquivos, consultar e atualizar tabelas utilizadas pelo sistema, e fazer manutenção da base de dados.

Para Cerqueira et al. (2019), existem outras fontes de dados que são utilizadas mediante entradas de novas informações no SUS. Conforme (Cerqueira et al. 2019) "*cabe ressaltar que alguns sistemas possuem características comuns e que podem vir a gerar estatísticas complementares e comparativas entre eles.*". A Tabela 1 mostra um resumo dos principais sistemas de saúde nacionais.

Tabela 1 - Principais sistemas nacionais de informação em saúde

| Sigla | Sistema de informação em saúde | Ano de início | Documento | *Link de acesso* |
|---|---|---|---|---|
| **SIM** | Sistema de Informações sobre Mortalidade | 1975 | Declaração de óbito | <http://www2.datasus.gov.br/DATASUS/index.php?area=0901&item=1&acao=31&pad=31655> |
| **SINASC** | Sistema de Informações sobre Nascidos Vivos | 1990 | Declaração de nascidos vivos | <http://www2.datasus.gov.br/DATASUS/index.php?area=0901&item=1&acao=28&pad=31655> |
| **SIH** | Sistema de Informações Hospitalares | 1991 | AIH | <http://www2.datasus.gov.br/datasus/index.php?area=0901&item=1&acao=25> |
| **SINAN** | Sistema de Informações de Agravos de Notificações | 1993 | Ficha individual de notificação | <http://dtr2004.saude.gov.br/sinanweb/> |
| **SIA** | Sistema de Informações Ambulatoriais | 1994 | BPA | <http://www2.datasus.gov.br/DATASUS/index.php?area=0901&item=1&acao=22&pad=31655> |
| **SIAB** | Sistema de Informações de Atenção Básica | 1998 | Formulários de cadastro e seguimento das famílias atendidas por equipes de saúde da família e agentes comunitários de saúde | <http://www2.datasus.gov.br/SIAB/index.php?area=01> |

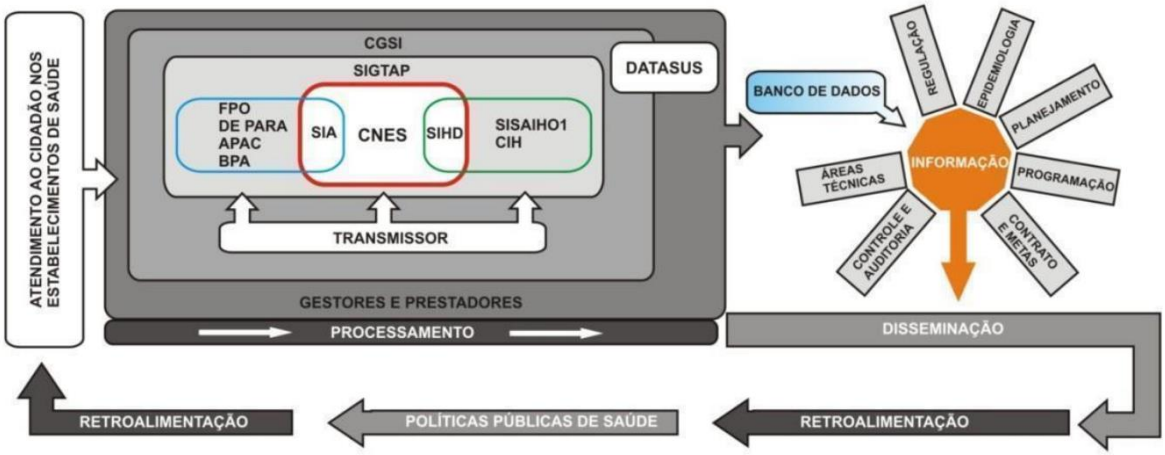
Fonte: Cerqueira et al. (2019)

Essas bases possuem todas as produções de dados relacionados à saúde do SUS. Por isso, está sendo amplamente utilizada para diversos fins, como por exemplo avaliação do sistema de saúde, gastos do sistema, ofertas e demandas de serviços dentre outros indicadores e relatórios.

Todos os sistemas mencionados trabalham de forma conjunta com outras partes do processo de atendimento de um paciente ou indivíduo em uma unidade do sistema público de saúde. A Figura 5 mostra o fluxo e disseminação da informação depois do processo de obtenção.

Na figura apresentada, entende-se por transmissor como sendo um aplicativo que possibilita o envio dos arquivos de banco de dados dos sistemas (CIH, SIA, SIAB, SIHD), visando alimentar o Banco de Dados Nacional (BDN) desses Sistemas de Informação. Ou seja, é a ferramenta utilizada para realizar a cópia em bancos de dados locais ou para enviá-las ao BDN.

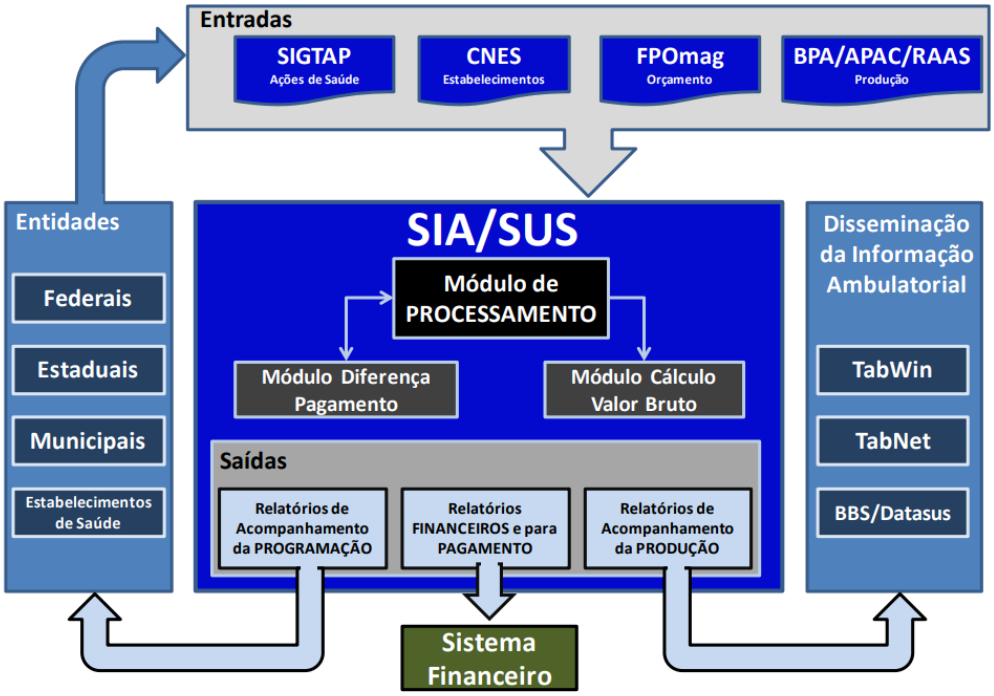
Figura 5 - Visão esquemática do processamento da informação do atendimento no SUS




Fonte – Ministério da Saúde (2017)

Especificamente para o sistema SIA, existe uma semelhança de fluxo de informações com o padrão do SUS. A Figura 6 mostra o fluxo das informações provenientes dos sistemas de entrada e o compartilhamento destas com o SIA.

Figura 6 - Fluxograma das informações e processos internos com o módulo SIA




Fonte - Ministério da Saúde (2010)

Cada sistema de entrada, como o BPA e RAAS possuem um banco de dados específico que é instalado nas máquinas que operam o sistema. Este bando de dados é o *Firebird* ®. O aprofundamento nesta aplicação não é explicado neste trabalho pois o foco não é o tratamento dos dados a nível de banco de dados.

Contudo, a partir da relação entre o fluxo de entrada e saída das informações e dados dos pacientes em um atendimento em uma unidade básica de saúde, pode-se propor uma nova ideia com a ligação entre uma rede *blockchain* confiável privada e este fluxo. Sendo assim, o próximo capítulo explica de forma detalhada a implementação desta ideia. A próxima seção inicia com uma explicação das definições básicas para a criação da rede *blockchain* do paciente e quais as informações necessárias que ela deve possuir.

# 3 METODOLOGIA

Neste trabalho foi realizado um estudo de criação de um sistema de rastreio e de controle de informações e dados utilizando a tecnologia *blockchain*. Um sistema de criptografia atua na rede para ser possível guardar e compartilhar informações sem o uso de intermediadores e de forma facilitada e, principalmente, garantir a segurança e a proteção dos dados envolvidos.

A primeira parte foi definir as bases para que a elaboração do *framework* fosse possível, dentre elas a base de dados e informações já existentes no sistema público de saúde e então como aplicar uma rede distribuída e compartilhar os possíveis registros. Com essas informações prontas, passou-se para a segunda etapa do trabalho referente à implementação do *framework* proposto, através de um programa criado na linguagem de programação *javascript*.

Após a criação do código e a implementação da ideia inicial, foram realizados testes para validar a ideia proposta. Com isso, algumas informações puderam ser comprovadas e explicadas ao longo dos testes, levando a um melhor entendimento da ideia. Os próximos itens do capítulo mostram detalhadamente todo o trabalho desenvolvido.

## 3.1 Definições da rede e do sistema

Foi definida a utilização de uma rede *blockchain* privada para este *framework*, pois assim obtém-se um ambiente controlado por somente alguns usuários de uma organização específica, que neste caso é o sistema público e o paciente. Outra etapa estabelecida foi a de tornar a rede permitida (*Permissioned Blockchain*). Neste tipo, não há a necessidade de se criar uma Prova-de-Trabalho (do inglês *Proof-of-Work*), pois os nodos na rede são confiáveis e não precisam de prova para que um novo bloco seja adicionado na rede.

Os nodos nessa rede são confiáveis e eles se conhecem. Geralmente não representam somente usuários, mas instituições inteiras. Por isso, a privacidade é um ponto de grande importância na rede *blockchain* permitida. Embora todas as organizações ou usuários sejam parte desta mesma rede, somente parte deles deveriam ter o poder de ver as transações que foram realizadas. Assim, a privacidade

é importante para que seja possível controlar quem pode ver uma informação transacional específica.

Ainda neste tipo de rede, é possível construir transações com muito mais eficiência do que no tipo de rede *blockchain* sem permissão (*Permissionless Blockchain*). Para isso pode-se utilizar os chamados Contratos Inteligentes (Smart Contracts), que são essencialmente códigos executados quando verificada e garantida que toda a informação necessária está correta.

Os Contratos Inteligentes podem ser criados a partir de simples linhas de código como “se/quando... então...” que são adicionadas ao *blockchain*. Assim, uma série de computadores na rede podem executar ou receber uma ação pré-determinada, como notificações, liberação de valores para um terceiro ou liberar um tíquete. Após alguma transação ser executada e completada somente então a rede é atualizada e não pode ser modificada.

Como já explicado, em uma aplicação *blockchain* permitida não há a necessidade de uma prova de trabalho, ao mesmo passo que a mineração também não é necessária. Portanto, neste estudo não foi utilizada a mineração tampouco a recompensa para usuários na rede.

## 3.2 Definição do *framework* criado entre os sistemas do SUS e a rede *blockchain*

A partir das informações definidas do ambiente e das premissas básicas da rede *blockchain*, o fluxo de informações e das relações entre cada nodo da rede puderam ser propostas e construídas através de um *framework*. Para este estudo levou-se em consideração a proposta de criação e uso de uma API e um SDK para intermediação das informações entre os nodos da rede e para a relação de segurança da mesma.

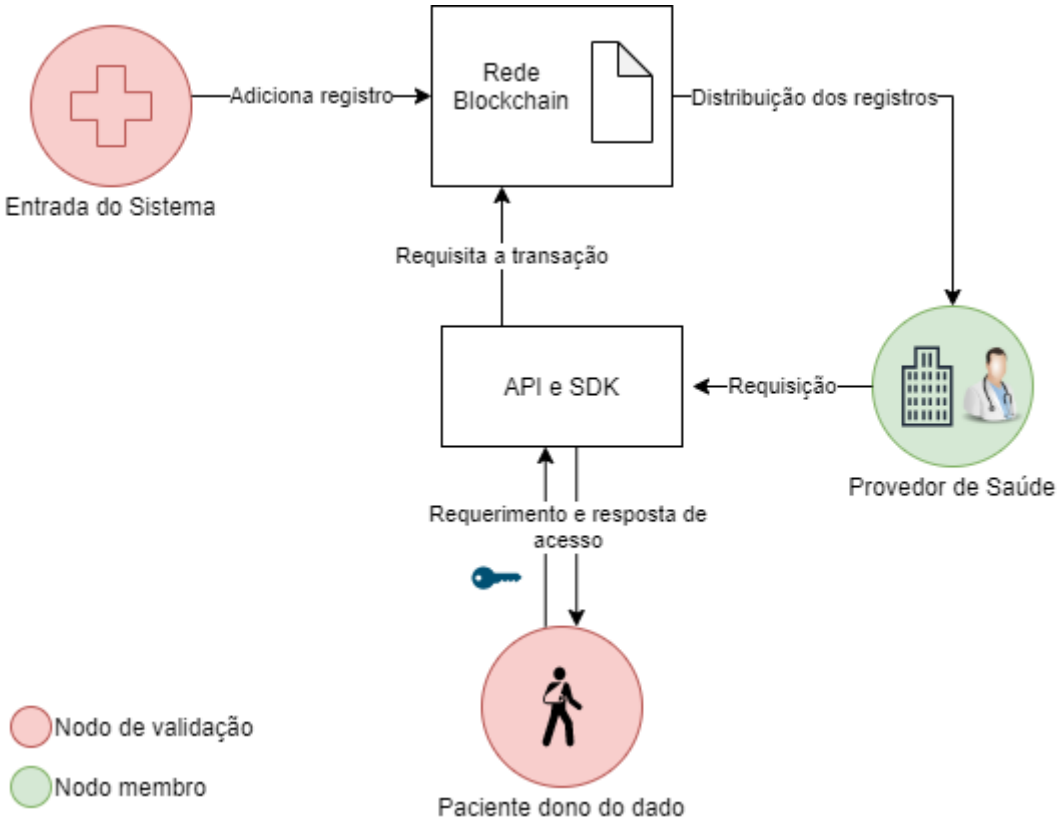
Como a rede é privada, presume-se que os nodos são confiáveis. Entretanto os nodos membros da rede somente possuirão uma cópia dos registros caso sejam autorizados. O próximo capítulo explica melhor essa relação.

### 3.2.1 Visão geral do *framework* proposto

Quando o paciente é atendido em alguma unidade de pronto atendimento, hospital ou outro estabelecimento do sistema público de saúde, é feita a entrada de

um novo registro informando que houve interação do paciente com o local, como por exemplo, um RAAS informando que foi realizado um procedimento de eletrocardiograma em um indivíduo. A Figura 7 mostra que este novo registro é vinculado, via o nodo do sistema em utilização, à rede *blockchain*. A informação na rede pode ser acessada por outros provedores de saúde, porém só é permitido o acesso se estes possuírem a chave privada do paciente dono do dado.

Figura 7 - Identificação dos nodos participantes da rede




Fonte: Elaborado pelo autor

Existem dois tipos de nodos criados: nodo de validação e nodo membro. O nodo de validação pode iniciar ou receber transações mas principalmente validá-las, enquanto que o nodo membro pode apenas iniciar ou receber transações. Cada tipo foi definido para que as questões de segurança pudessem ser levadas em consideração durante todo o fluxo. Neste caso, o sistema de entrada e os próprios pacientes são os detentores das permissões.

As requisições de acesso (permissão) podem ser realizadas por qualquer nodo confiável da rede, por exemplo um novo médico tentando obter os dados passados do seu paciente. Como o novo médico (nodo) não possui acesso por não possuir a

chave privada do paciente, ele deve realizar uma requisição. Esta requisição será feita através de uma API criada juntamente com um SDK do sistema.

Pode-se atribuir também outra característica a um novo nodo no sistema: provedor de saúde de um sistema público. Ou seja, caso um médico ou profissional da saúde pública em atendimento ao paciente necessite atualizar ou acrescentar nova informação ou um novo registro, este automaticamente torna-se um nodo de validação. Desta forma pode-se iniciar o ciclo do registro no início do processo. Esta ação é possível pois este prestador estará com posse da chave de acesso do paciente, pois faz parte do ciclo público das informações de saúde do paciente.

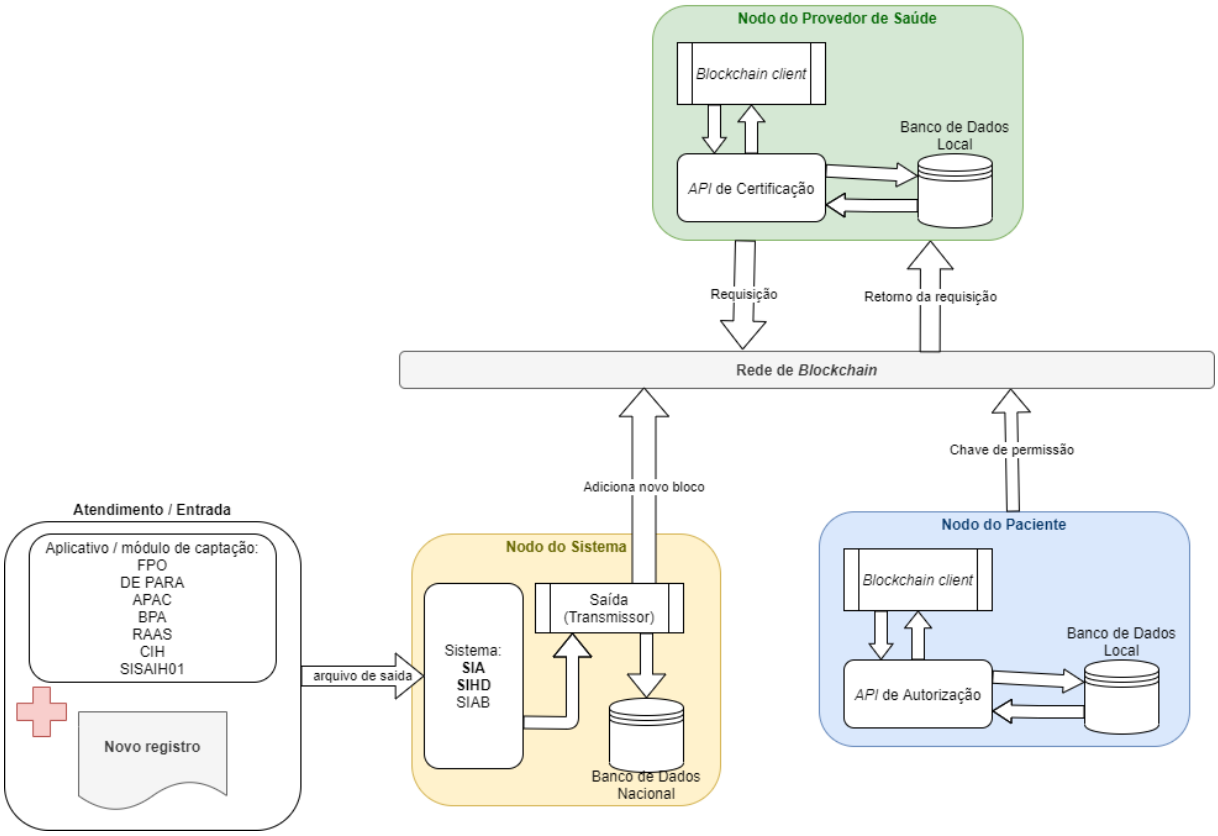
O dono do dado (paciente) pode realizar a escolha de liberar a chave ou não. Essa comunicação ainda está acontecendo via API. No momento em que é liberado o acesso, a API requisita a transação na rede *blockchain* através de uma autenticação de assinatura digital e então ocorre a distribuição do registro.

### 3.2.2 *Framework* proposto entre os sistemas de saúde e nodos da rede *blockchain*

A relação entre todos os nodos da rede é realizada através das APIs e SDKs propostos anteriormente. Contudo, cada nodo pré-definido na rede contém características próprias que farão com que as relações sejam configuradas e encaminhadas de forma correta.

Identifica-se como objeto de interesse neste estudo o registro gerado a partir de um atendimento e/ou entrada realizados por algum aplicativo ou módulo de captação em alguma unidade de saúde pública. Este tem como destino um arquivo de saída gerado pelos programas existentes nestas unidades e operados pelos profissionais competentes. O arquivo é então utilizado como dado e objeto alvo do sistema inteiro, contemplando também todos os nodos da rede. A Figura 8 ilustra o *framework* proposto neste trabalho, evidenciando a relação entre todos os nodos da rede.

Figura 8 - Orquestração do *framework* proposto: entrada de registro de um paciente




Fonte - Elaborado pelo autor

O nodo do sistema é a parte do processo responsável por realizar a conexão entre o dado (arquivo de saída) proveniente do aplicativo de captação. O sistema realiza a importação do arquivo seguindo o processo normal já existente. Quando o arquivo é enviado para o sistema transmissor, ele realiza o processo de compartilhamento da informação no banco de dados nacional, e em adição a isso, inicia o processo de criação de um novo bloco na rede *blockchain*. Portanto, existe essa integração entre uma nova arquitetura de *blockchain* à infraestrutura pública existente.

O design apresenta somente um novo componente: Saída do Transmissor com conectividade a API de integração com a rede *blockchain*. Como foi assumido que muitos nodos já possuem um gerenciamento seguro com o paciente, como no caso dos médicos prestadores de serviço, esta etapa do nodo do sistema pode ser replicada neste processo de adição de uma nova informação.

O nodo do paciente é um nodo importante no processo do framework proposto, pois é ele que possui a chave de permissão (chave segura) de acesso à rede

*blockchain*. Temos três componentes: *Blockchain client*, API de Autorização e Banco de Dados local. Cada componente é responsável por realizar um processo dentro do chamado nodo.

O Cliente *Blockchain* (do inglês *Blockchain client*) implementa toda a funcionalidade exigida para integrar e participar da rede *blockchain*. Usa diversas tarefas, como a conexão com a rede ponto-a-ponto (do inglês *peer-to-peer* abreviado P2P), codificação e envio de transações.

A API de autorização realiza a função de criar o código necessário para comunicação com o cliente *blockchain* no que tange a função de compartilhar e implementar a parte de segurança e autenticação utilizando a chave segura do usuário. Desta forma o *hash* é construído composto por essas informações e salvo no Contrato Inteligente Seguro. Caso o programa (rede) identifique que não possui autorização, este acesso é negado.

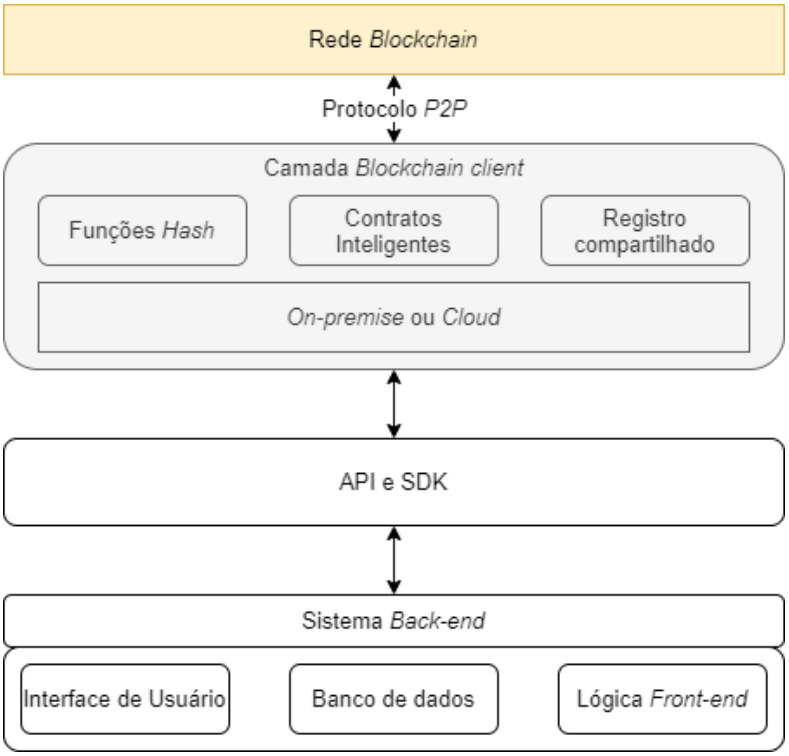
O Banco de Dados local neste caso funciona como um armazenamento de cache dos dados médicos do paciente/usuário. Caso algum dado esteja corrompido no banco de dados ou faltando, pode ser recuperado a qualquer momento da rede seguindo os protocolos de segurança. Este banco de dados local pode ser utilizado também como base para a aplicação SDK e Interface do Usuário desta.

O Nodo do Provedor de Saúde neste *framework* contém a mesma característica e componentes do nodo do paciente, retirando a parte de autenticação e autorização das chaves de acesso. Porém, a API necessária neste caso é a de certificação, onde realizará a requisição de acesso e obterá uma resposta da rede.

Uma arquitetura foi proposta para os nodos que não necessariamente sejam os principais, como do paciente e provedores de saúde por exemplo. A Figura 9 mostra os componentes desta arquitetura mais detalhadamente. Pode-se verificar que um sistema de API mais SDK formam a ponte entre a camada do cliente *blockchain* e o sistema *back-end* e *front-end* da aplicação.

A camada de cliente *blockchain* possui como sub-componentes funções *hash* de criação, os contratos inteligentes e os registros compartilhados na rede. A base desse sistema pode ser hospedada tanto localmente (do inglês *on-premise*) quanto na nuvem (em inglês *cloud)*. Esta camada também é responsável pela conexão à rede *P2P* e seu protocolo, assegurando que todos os estados das tarefas estejam corretos e de compartilhar os registros na rede com os nodos participantes.

Figura 9 - Arquitetura de um nodo secundário na rede




Fonte: Elaborado pelo autor

Para este estudo assumiu-se que a rede ponto-a-ponto (*P2P*) possui no mínimo dois nodos que possuem cópias dos arquivos, agindo tanto como servidor e cliente para outros nodos. Neste caso, se uma requisição de outro nodo for enviada para acessar arquivos deste primeiro nodo, esta pode ser feita pois a rede não possui uma centralização que detém todos os arquivos. Como dito, o foco deste trabalho é utilizar nodos que estejam no centro da rede *P2P*, tendendo a estarem sempre ligados para que a comunicação entre os nodos aconteça de forma correta e completa.

## 3.3 Determinação da *blockchain* do *framework*

Para esta *blockchain* foi utilizada a *hash* como sendo a assinatura digital de cada bloco e também uma *hash* para cada transação de cada bloco. Para este trabalho entende-se como transação o arquivo de saída gerado pelo sistema de saúde contendo as informações/dados dos atendimentos e do paciente.

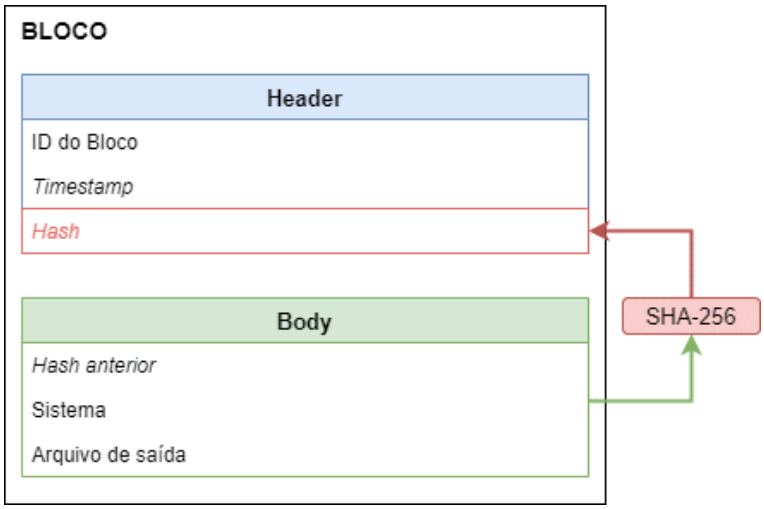
### 3.3.1 Definição da *hash*

A *hash* sempre vai ser construída através dos dados presentes no corpo (do inglês *body*) do bloco. O código, através de várias funções, transforma qualquer entrada em uma saída com tamanho fixo. No caso deste estudo, foi utilizado um algoritmo de *hashing* chamado SHA-256 (Algoritmo de *Hashing* Seguro, do inglês *Secure Hashing Algorithm (SHA) 256*) que nos retorna uma *string* com 256 bits fixos.

### 3.3.2 Definição do Bloco

Para a construção do bloco da rede houve a necessidade de levar em consideração alguns pontos importantes, como quais tipos de dados (informações) deveriam ser armazenados na rede, assim como a importância destas informações. Portanto, baseado no modelo já existente de arquivos de saída dos programas públicos de saúde, definiu-se pela utilização destes dados no corpo do bloco. A Figura 10 mostra o bloco proposto e seus componentes, assim como o código *hash* sendo criado via função *SHA-256*.

Figura 10 - Organização do bloco e seus componentes




Fonte - Elaborado pelo autor

Cada bloco é constituído por duas partes principais: o cabeçalho (do inglês *header*) e o corpo (*body*). Neste estudo foram definidas como componentes do cabeçalho as seguintes informações:

- ID do Bloco;
- Carimbo de data / hora (do inglês *timestamp*);

- *Hash.*

Para o corpo os seguintes dados foram adicionados:

- *Hash* anterior;
- Sistema;
- Arquivo de saída.

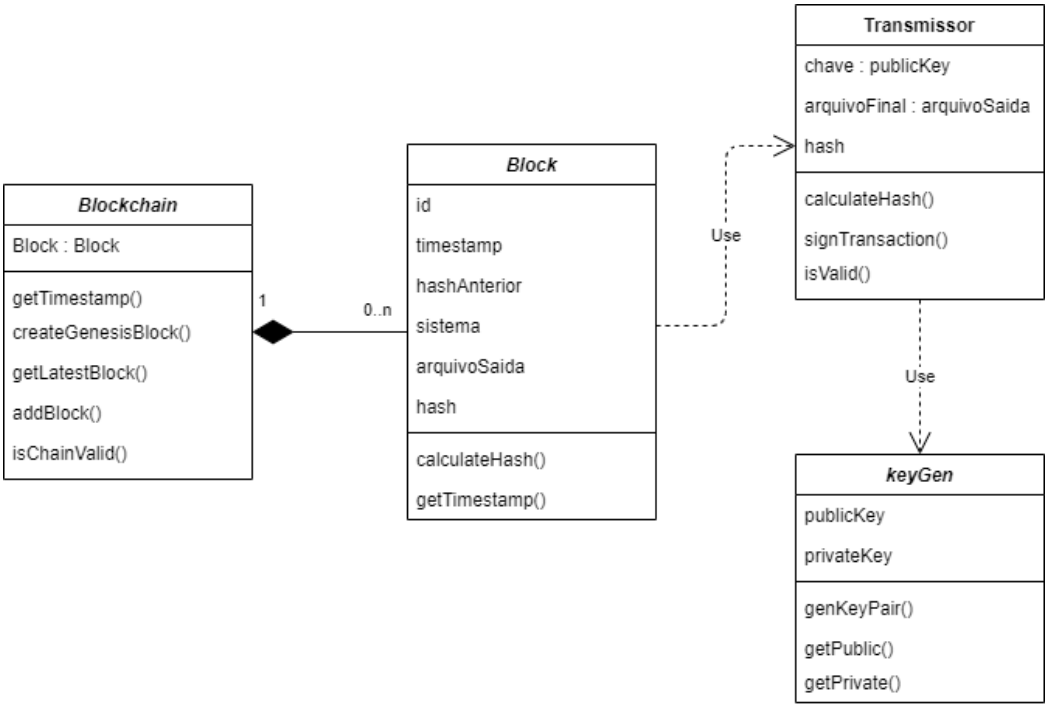
O ID do Bloco nada mais é do que uma identificação para o bloco construído e o *timestamp* marca o momento exato (data, hora, minutos e segundos) da criação do bloco. A *Hash* anterior faz referência ao bloco diretamente relacionado com o atual e a *Hash* atual do bloco. O nome da identificação do sistema de saúde de entrada é armazenado na variável Sistema. O Arquivo de saída compõe os dados gerados pelo sistema de entrada que levam às informações do estabelecimento de saúde, dados particulares do paciente e as ações tomadas durante o atendimento. Cada sistema de entrada gera um diferente arquivo de saída dependendo do ambiente em que o paciente se encontra.

## 3.4 Implementação do código do *framework*

Neste trabalho utilizou-se a linguagem de programação *javascript* para a criação do *blockchain* fictício juntamente com a tecnologia de código aberto *node.js*. Algumas outras bibliotecas e funções foram utilizadas durante o desenvolvimento do código principalmente para geração das chaves de acesso.

Tendo como base o diagrama apresentado anteriormente pela Figura 7 foram criadas quatro classes principais no programa para que todas as etapas do fluxo de dados pudessem ser implementadas. A Figura 11 exibe o diagrama UML de todas as classes do código.

Figura 11 - Diagrama de classes UML do código criado




Fonte - Elaborado pelo autor

A classe principal (do inglês *main*) não foi adicionada ao diagrama pois ela é somente responsável por iniciar o programa e não parte da modelagem da aplicação. A relação entre as classes *Blockchain* e *Block* são de composição onde *Block* tem que pertencer a *Blockchain*. Já para as demais classes a relação é de implementação ou uso.

### 3.4.1 Classe *blockchain*

Essa é a classe responsável por construir a corrente de blocos (*blockchain*) no programa. No construtor desta classe foi criada uma *array* para a criação do primeiro bloco, chamado de gênesis. Este primeiro bloco é único pois não aponta para nenhum outro bloco e, portanto, não possui uma ligação com o *hash* do bloco anterior.

Cinco métodos foram criados para esta classe:

- *getTimestamp():* método criado para gerar o carimbo de data / hora do momento da criação do *blockchain*
- *createGenesisBlock():* método para criação do bloco gênesis (primeiro da rede)
- *getLatestBlock():* método que retorna o último bloco na rede

- *addBlock(newBlock):* método que adiciona um novo bloco na rede, informando os parâmetros que são utilizados na classe *block*
- *isChainValid():* método criado para verificar se a rede é válida através de testes entre o *hash* do bloco atual e do bloco anterior

Os métodos acima são chamados durante a execução do programa através do código principal. A próxima seção explica a classe do bloco da rede.

### 3.4.2 Classe *block* (bloco)

Esta classe realiza a criação de um novo bloco que posteriormente é adicionado a rede *blockchain* existente. O construtor desta classe realiza o cálculo da *hash* do bloco e também do *timestamp* do bloco além de inicializar outras variáveis. Para cada bloco é dado um valor único chamado de ID, que em outras situações pode ser utilizado para identificar cada bloco.

Uma importante parte que constitui o bloco é o chamado arquivo de saída. Esta variável faz alusão ao arquivo criado pelo sistema de entrada no ambiente clínico do sistema de saúde e contém as informações de saúde e atendimento do paciente, como explicado no item 3.3.2.

Existem apenas dois métodos nesta classe, sendo eles *getTimestamp()* e *calculateHash()*. O método *calculateHash()* foi criado para calcular a *hash* utilizando as informações do *hash* do bloco anterior, nome do sistema de entrada e o arquivo de saída. Desta forma, todas estas informações são utilizadas para criar a assinatura do bloco e, caso algumas delas seja alterada, a assinatura também mudará e, portanto, tornar-se-á inválida na rede.

### 3.4.3 Classe transmissor

Esta classe simula o papel do transmissor no fluxo do *framework* proposto, como mostrado anteriormente na Figura 8. Neste caso, o transmissor além de salvar os dados no banco de dados local e enviá-los ao banco de dados nacional, ele cria, juntamente com o arquivo de saída, um novo bloco na rede *blockchain* do paciente.

Neste trabalho a classe transmissor também pode ser definida como o programa a ser utilizado para obter acesso à rede no caso de um provedor de saúde

(nodo provedor) onde este podem também adicionar um novo bloco na rede, caso possua a chave pública de acesso do paciente.

Como é disposto de exemplo na Figura 7 ocorre uma conexão dos nodos com uma API e SDK de autenticação através de chaves de acesso. Essa chave é que permite a adição de um novo bloco na rede ou a requisição de acesso a esta rede do paciente. O próximo item explica como são criadas essas chaves no código.

A classe transmissor possui um construtor que também realiza o cálculo de uma *hash* para este arquivo de transmissão, que é adicionado ao bloco que é adicionado a *blockchain*. Ou seja, o programa do *framework* realiza a criação de duas assinaturas digitais para que desta forma haja mais segurança nos dados da rede. Além disso, mais dois métodos foram criados para auxiliar no processo de transmissão da informação:

- *signTransaction(signingKey)*: método que realiza a assinatura da transação, que neste caso é o arquivo de saída, com o objeto chave de assinatura (par de chaves *Elliptic*) que contém uma chave privada. Caso a chave de acesso esteja errada, o sistema gera um erro em tela para o usuário. Do contrário é realizada assinatura do arquivo utilizando a *hash* desta transação.
- *isValid()*: este é um método criado para verificar se a transação (arquivo) é válida (se não foi adulterada) e utiliza o objeto chave como chave pública.

Os métodos acima são chamados durante a execução do programa através do código principal. A próxima seção explica a classe de geração das chaves públicas utilizadas nos testes do programa.

### 3.4.4 Classe *keyGen* (geradora de chave)

Para direcionar a questão das autorizações de acesso à rede *blockchain*, que no modelo proposto utiliza-se APIs e SDKs, foi criada uma nova classe geradora de par de chaves. Este par é composto por uma chave pública e uma privada.

A chave pública (do inglês *publicKey*) é utilizada como sendo a chave pública do paciente, que é disponibilizada por ele de acordo com suas intenções e confianças nos nodos da rede. Já a chave privada (do inglês *privateKey*) é a informação utilizada para assinar os arquivos de saída e realizar a verificação ou autenticação desta

informação no bloco da rede antes de ser adicionado ao *blockchain* do paciente. Este mesmo sistema é muito utilizado em criptomoedas como o *Bitcoin* e *Ethereum*.

O código desta classe gera um par de chaves através da biblioteca *elliptic* e exibe os valores do par em tela. Para fins de testes e utilização das chaves no programa, as chaves geradas neste código são utilizadas posteriormente no programa principal sendo elas:

- *privateKey* = minhaChave (chave do usuário/nodo)
- *publicKey* = chavePublicaPaciente (chave pública do paciente)

Utilizando as informações acima é possível realizar a ligação entre as informações das chaves para uso na rede. A próxima seção explica a classe principal do programa.

### 3.4.5 Classe *main* (principal)

A classe principal do programa é responsável pela criação da *blockchain* do usuário, que no exemplo utilizado neste trabalho é o paciente do SUS, criando uma nova instância da classe *blockchain* explicada anteriormente. Nesta classe é realizado o tratamento de autenticação das chaves de acesso simulando API de segurança, e também a adição de um novo bloco na rede criada utilizando o arquivo de saída.

Como mencionado anteriormente, os objetos minhaChave e chavePublicaPaciente são utilizados neste programa para que exista uma segurança das informações na rede do paciente. Ou seja, o usuário do programa somente terá acesso às funções se possuir uma chave privada compatível com a chave pública do paciente.

Após as definições de chaves e autenticação, o programa simula a criação de um arquivo de saída, que será explicado no item 3.4.7, para que estes dados sejam então embrulhados em um bloco que será adicionado a rede. Ao mesmo tempo é realizada uma assinatura da transação (arquivo) através da classe transmissor.

### 3.4.6 Bibliotecas utilizadas

Duas bibliotecas foram fundamentais durante o desenvolvimento do programa de simulação do *framework*. Ambas são de domínio público e possuem fácil acesso via rede. São elas:

- *Elliptic*

  Essa biblioteca realiza a criação de uma curva elíptica de criptografia em *javascript*. Esta é uma técnica amplamente utilizada em questões de segurança de tecnologia como conexões HTTP e transferência de dados entre centro de dados. Para este trabalho definiu-se a utilização desta técnica pois é a que mais possui aplicações na área de *blockchain*.

- *Crypto-js/sha-256*

  A biblioteca *Crypto-js* é uma coleção de algoritmos de criptografia seguros e padrões implementados em *javascript*. São rápidos, consistentes e possuem uma interface de utilização relativamente simples. Foi escolhido para se utilizado principalmente devido a sua possibilidade de criação do *Hash* para o código.

  A função *sha-256*, que é utilizada para o processo de *hashing* no programa, é acessada através da biblioteca *Crypto-js* e faz parte do conjunto de funções *sha-2*. Esse conjunto foi projetado pela Agência de Segurança Nacional dos EUA e significa *secure hash algorithm* (algoritmo de *hash* seguro).

Estas bibliotecas foram escolhidas para serem usadas no programa pois possuem ampla documentação e são de fácil utilização na linguagem *javascript*. Por esse motivo se adequam ao objetivo final deste trabalho e do programa.

### 3.4.7 Arquivo de saída

O arquivo de saída no programa foi simulado utilizando uma estrutura de dados no formato JSON escrita diretamente no código do programa principal. Foi definida a utilização deste formato para fins de testes e por ser de fácil manipulação. A Figura 12 exibe a definição do forma do arquivo JSON com suas informações baseadas em um arquivo de saída exemplo. O arquivo JSON foi definido em duas partes: cabeçalho (*header*) e corpo (*body*).

- Cabeçalho:

  No cabeçalho constam as informações gerais específicas do sistema que está sendo utilizado para a aquisição dos dados no momento do atendimento ao paciente. Como por exemplo o CNES (Cadastro Nacional de

Estabelecimentos de Saúde), o CNS (Cartão Nacional de Saúde) do profissional que está realizando o atendimento e outras informações.

- Corpo:

  No corpo do arquivo foram divididas mais duas partes: usuário e procedimento. A parte relativa ao usuário contém as informações particulares do paciente em atendimento. Já a seção de procedimento possui os dados e informações relativas ao(s) procedimento(s) realizado(s) durante o atendimento.

Figura 12 - Trecho de código extraído do programa principal contendo a estrutura de dados JSON do arquivo final

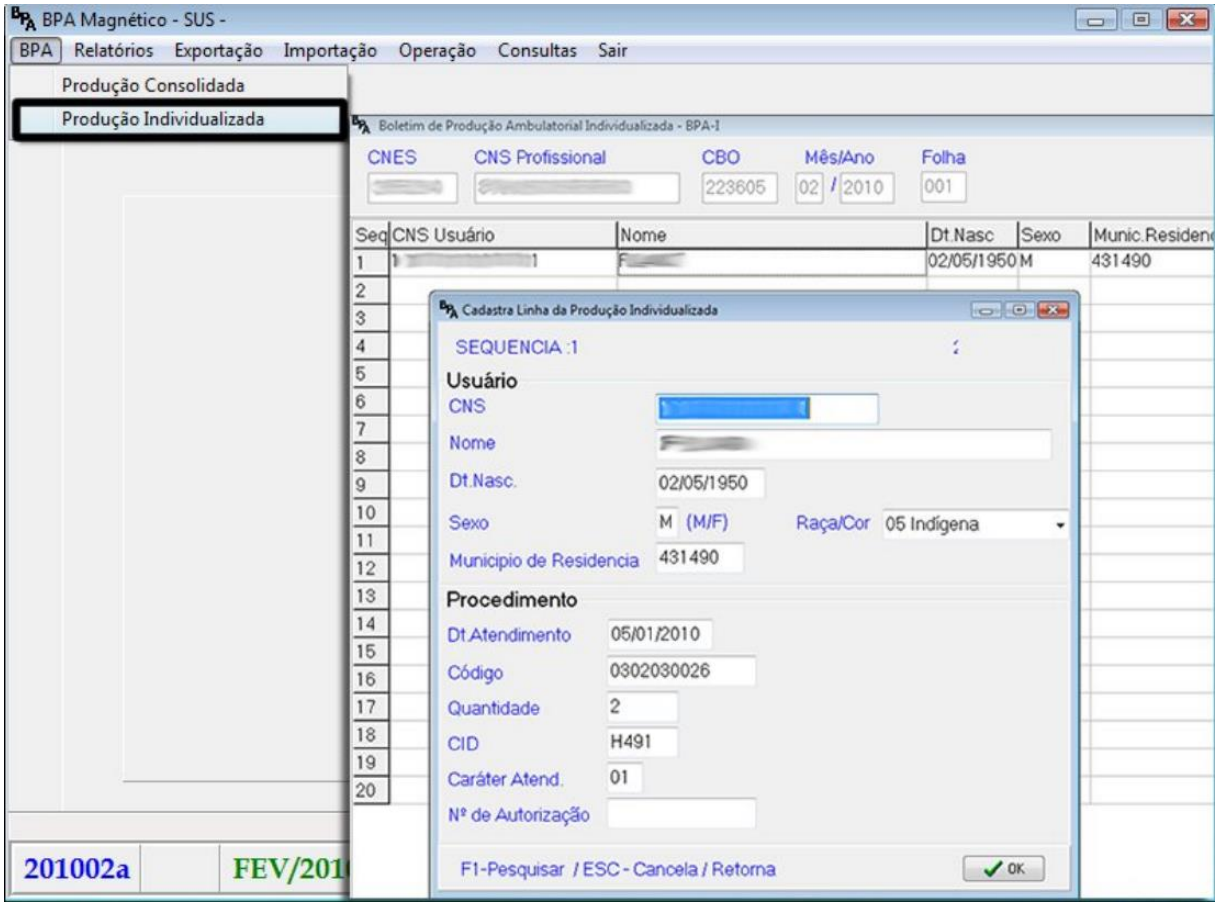
```
var arquivoFinal = {
    header: {
        CNES:'12345',
        CNSProfissional:'12354.789',
        CBO:'223605',
        mesANO:'06/2021',
        folha:'001'
    },
    body: {
        usuario: {
            CNS:'123456789',
            Nome:'Leonardo Reginato',
            DtNasc:'26/06/1993',
            Sexo:'M',
            RacaCor:'05 Branca',
            MunicipioDeResidencia:'431490'
        },
        procedimento: {
            DtAtendimento:'20/05/2021',
            Codigo:'0302030026',
            Quantidade:'2',
            CID:'H491',
            CaraterAtend:'01',
            NumAutorizacao:' '
        }
    },
};
```

Fonte - Elaborado pelo autor

Essas informações e estruturas montadas em formato JSON para simular o arquivo de saída foram obtidas e utilizadas através do exemplo da Figura 12. A figura mostra a tela do programa BPA-magnéticos do SUS e a visão individualizada de um atendimento que ocorreu em uma unidade de serviço a saúde.

A criação do arquivo de saída para simulação no programa criado utiliza como base o programa de Boletim de Produção Ambulatorial que contém os dados dos procedimentos realizados no paciente em atendimento. Com base no processo proposto no *framework*, estas são as informações que são guardadas no bloco e então adicionadas ao *blockchain* do paciente.

Figura 13 - Telas de registro dos BPA Individualizado




Fonte - Ministério da Saúde, 2012.

Este exemplo foi utilizado para fins de simulação com objetivo de realizar um procedimento o mais fiel possível a realidade dos atendimentos e arquivos de saída. Do mesmo modo que é possível utilizar diferentes arquivos gerados por diferentes sistemas dependendo do tipo de atendimento realizado.

## 3.5 Testes com o *blockchain* desenvolvido

Para validar o programa criado e a segurança das informações, como proposto no fluxograma principal do trabalho na Figura 7, foram criados alguns testes para

manipular os dados e informações do programa e posteriormente analisá-las. Para realizar os testes foram criadas novas instruções de códigos na classe principal (*main*) do programa. Após execução dos comandos, os resultados são exibidos em tela no próprio terminal da IDE de criação de código utilizada chamada *Visual Studio Code* ®.

Cada teste tem uma particularidade que é explicada em seguida. Contudo, vale ressaltar que para a completa validação, a simulação com a rede *blockchain* foi realizada utilizando-se dois blocos criados na rede, bloco 1 e bloco 2, cada qual com seus dados e informações provenientes de sistemas de saúde de entrada diferentes.

### 3.5.1 Gerando uma rede *blockchain* e novos blocos

Após o programa estar finalizado, é possível realizar a criação de uma nova rede *blockchain*, que neste caso é simulada para um paciente específico, sendo a rede particular deste usuário do serviço público. Este processo de criação é realizado através do código mostrado na Figura 14.

Figura 14 - Código que cria a nova rede blockchain

```
// cria uma nova instancia da classe blockchain
let leonardoChain = new Blockchain();
```

Fonte - Elaborado pelo autor

Após a criação dessa nova instância, é realizado o cálculo da *hash* do arquivo a ser transmitido e então a assinatura desta transação. Após isso, dois novos blocos são adicionados a rede criada anteriormente. Esse processo de criação e adição de bloco pode ocorrer então a qualquer momento desejado. As Figuras 15 e 16 mostram as informações da rede e do exemplo criado em formato JSON no terminal da IDE de programação.

Figura 15 – Primeira parte do arquivo de saída exibindo a rede *blockchain*, os blocos e seus conteúdos

```
{
    "chain": [
        {
            "id": 0,
            "timestamp": 1623520571846,
            "hashAnterior": "0",
            "sistema": "sistema",
            "arquivoSaida": "Este é o bloco Gênesis",
            "hash": "bd0e6ea0ffe91e09afbb3de154f23aa59ca16bab841a203cc72b3a329df1e1e1"
        },
        {
            "id": 1,
            "timestamp": 1623520571900,
            "hashAnterior": "bd0e6ea0ffe91e09afbb3de154f23aa59ca16bab841a203cc72b3a329df1e1e1",
            "sistema": "BPA-mag",
            "arquivoSaida": {
                "chave": "043281c1b7b93ffaf9410f45159bd63b080d7d6293604ac60a7cb7bb42e4926418bfadfd71c87f3f78c617b1e1c253ae846b4e2de91f600d87f65c7a33e785ec6f",
                "arquivoFinal": {
                    "header": {
                        "CNES": "12345",
                        "CNSProfissional": "12354.789",
                        "CBO": "223605",
                        "mesANO": "06/2021",
                        "folha": "001"
                    },
                    "body": {
                        "usuario": {
                            "CNS": "123456789",
                            "Nome": "Leonardo Reginato",
                            "DtNasc": "26/06/1993",
                            "Sexo": "M",
                            "RacaCor": "05 Branca",
                            "MunicipioDeResidencia": "431490"
                        },
                        "procedimento": {
                            "DtAtendimento": "20/05/2021",
                            "Codigo": "0302030026",
                            "Quantidade": "2",
                            "CID": "H491",
                            "CaraterAtend": "01",
                            "NumAutorizacao": " "
                        }
                    }
                },
                "hash": "df68576a5f5ec875fa47c6f6511ac3d91a70d77401a0ab47bfc96adb259c5395",
                "signature": "3045022008848f05319b3ec59121b921db4b2e65134d999a4d7373b4cf888633e9f8abff022100d83a3999cd3b33679cd97253c92089c33a0f1604c39622dad875370f91645ae3"
            },
            "hash": "7682d087c27a31b0484b215cbcbcffd3fd0e2b309200a66be235ccaa37d1835d"
        },
```


Fonte - Elaborado pelo autor

Figura 16 - Segunda parte do arquivo de saída exibindo a rede *blockchain*, os blocos e seus conteúdos


```
        {
            "id": 2,
            "timestamp": 1623520571903,
            "hashAnterior": "7682d087c27a31b0484b215cbcbcffd3fd0e2b309200a66be235ccaa37d1835d",
            "sistema": "APAC",
            "arquivoSaida": {
                "chave": "043281c1b7b93ffaf9410f45159bd63b080d7d6293604ac60a7cb7bb42e4926418bfadfd71c87f3f78c617b1e1c253ae846b4e2de91f600d87f65c7a33e785ec6f",
                "arquivoFinal": {
                    "header": {
                        "CNES": "67891",
                        "CNSProfissional": "67891.123",
                        "CBO": "334752",
                        "mesANO": "07/2021",
                        "folha": "001"
                    },
                    "body": {
                        "usuario": {
                            "CNS": "123456789",
                            "Nome": "John Doe",
                            "DtNasc": "04/10/1989",
                            "Sexo": "M",
                            "RacaCor": "05 Branca",
                            "MunicipioDeResidencia": "457125"
                        },
                        "procedimento": {
                            "DtAtendimento": "10/06/2021",
                            "Codigo": "0304610026",
                            "Quantidade": "1",
                            "CID": "J457",
                            "CaraterAtend": "01",
                            "NumAutorizacao": "001"
                        }
                    }
                },
                "hash": "df68576a5f5ec875fa47c6f6511ac3d91a70d77401a0ab47bfc96adb259c5395",
                "signature": "3045022008848f05319b3ec59121b921db4b2e65134d999a4d7373b4cf888633e9f8abff022100d83a3999cd3b33679cd97253c92089c33a0f1604c39622dad875370f91645ae3"
            },
            "hash": "fa2c59793dc92cdce9ee616dc376651f5bb3461b64f86d489e85aa9f0f8630f7"
        }
    ]
}
```


Fonte 1 - Elaborado pelo autor

O primeiro bloco da rede é bloco gênesis. Após ele vem encadeados os demais blocos adicionados, contendo todas as informações que foram adicionadas no processo, que neste caso do trabalho é um arquivo de saída de um atendimento ao paciente.

### 3.5.2 Adulteração de dados

Os primeiros testes são referentes a tentativas de alteração de dados nos blocos da rede. Pode-se tentar alterar uma informação de dado de algum atendimento específico do paciente dono da rede. Neste caso utilizou-se como exemplo alterar o código do procedimento realizado com o paciente.

O dado do arquivo foi então alterado para o teste, e logo após verificou-se a validade da rede blockchain através do método *isChainValid()*. A Figura 17 mostra o código de alteração do valor e o resultado da verificação em tela.

Figura 17 - Código de alteração do valor e a validação com o método criado

```
// altera o dado do arquivo presente no bloco 1
leonardoChain.chain[1].arquivoSaida.arquivoFinal.body.procedimento.Codigo = '0302030027'; //original = 0302030026
// verifica se o blockchain é valido
console.log('O blockchain é válido?', leonardoChain.isChainValid() ? 'Sim' : 'Não');
```

Fonte - Elaborado pelo autor

A verificação pelo método é realizada verificando se o valor do *hash* do bloco atual é diferente do valor do *hash* caso calculamos novamente o valor. A Figura 18 mostra a etapa do código que realiza essa função.

Figura 18 - Código de verificação da *hash* do bloco

```
// verifica se o HASH do bloco atual ainda está válido
if(currentBlock.hash !== currentBlock.calculateHash()){
    return false;
}
```

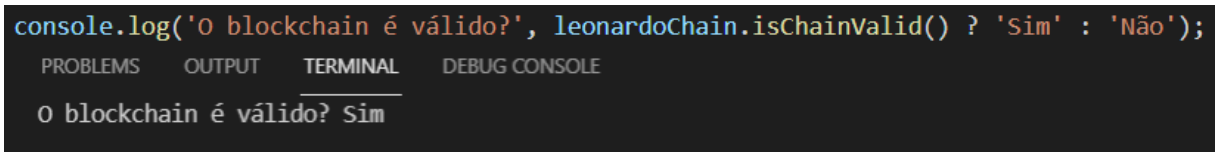
Fonte - Elaborado pelo autor

Sendo assim, quando se altera o valor de um dado presente em um bloco na rede, essa ação será automaticamente penalizada e se tornará inválida pelo código do programa. Dito isto, cabe salientar que todos os nodos da rede, pacientes e provedores, possuem cópias autorizadas de todos os blocos, desta forma a validação das informações pode acontecer.

### 3.5.3 Assinaturas com as *hashs*

Os testes de assinaturas utilizando os *hashs* de cada bloco foram realizados utilizando o método criado *isChainValid()*. Primeiramente verificou-se, após a criação e adição dos blocos, se a rede é válida. A Figura 19 mostra o resultado do primeiro teste.

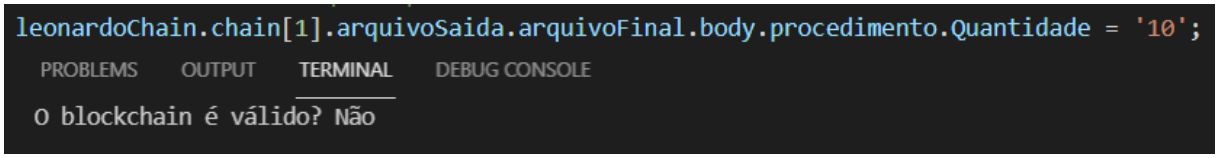
Figura 19 - Código do teste da rede e a saída no terminal




Fonte - Elaborado pelo autor

Após realizar a validação, alterou-se o valor da variável quantidade do arquivo 1 do bloco 1. E novamente verificou-se a validade da *blockchain*. Como é possível ver na imagem 20, o sistema retorna negativamente para a validade da rede.

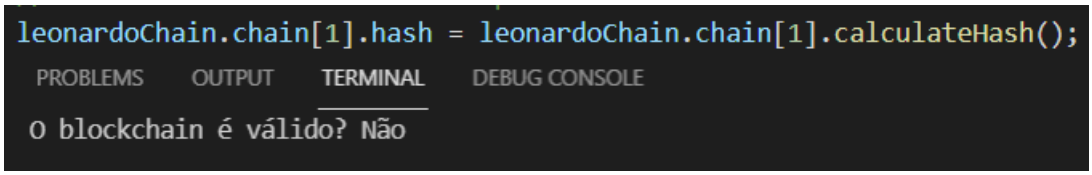
Figura 20 - Verificação da validade da rede após alteração da quantidade




Fonte - Elaborado pelo autor

Para burlar o sistema, pode-se pensar em recalcular a *hash* do bloco 1 e realizar novamente a validação da rede. Para isso, utilizou-se o método *calculateHash()* para o bloco após a alteração dos dados. O resultado é visto na Figura 21 a seguir.

Figura 21 - Resultado em tela no terminal validando a *blockchain*




Fonte - Elaborado pelo autor

O resultado deste teste mostra que mesmo que seja realizado um novo cálculo da *hash* do bloco após a alteração do dado, o sistema, através de uma verificação

completa, realiza a validação do *hash* utilizando os dados dos outros blocos da rede. Ou seja, caso um bloco seja alterado o seu *hash* é alterado e as informações desse *hash* não estarão de acordo com o valor do *hash* de outros blocos na rede, o que torna toda a rede inválida.

### 3.5.4 Autenticação de acesso através das chaves

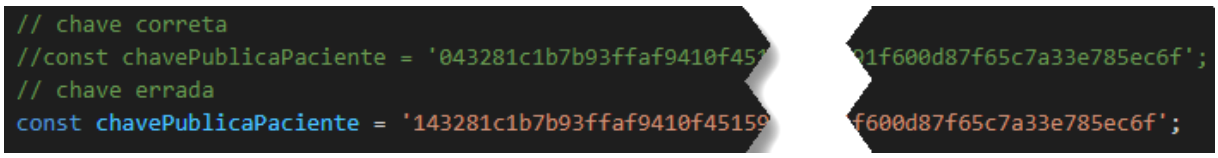
Para estes testes foi necessário alterar o valor da chave pública do paciente no começo do código da classe principal. Pois neste caso está se simulando a incorreta informação da chave de acesso para a adição de um novo bloco na rede privada do paciente.

Sendo assim, pode-se simular que:

- minhaChave = chave do usuário/nodo
- chavePublicaPaciente = chave pública providenciada pelo paciente

Caso a chave seja errada ou o nodo esteja tentando gerar uma chave de forma maliciosa, é realizada uma verificação pelo código através da biblioteca *elliptic* revalidando o par de chaves. Isso ocorre no momento de criação ou acesso à rede *blockchain*. A Figura 22 mostra a tentativa de uso de uma chave errada no código.

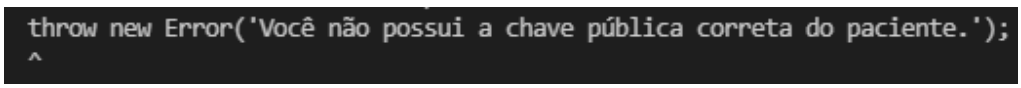
Figura 22 - Código contendo a informação de chave incorreta




Fonte - Elaborado pelo autor

O resultado da verificação e comprovação de que a chave pública está errada é mostrado na Figura 23. Como pode-se verificar, o próprio sistema anuncia que a chave está errada e não é possível adicionar um novo bloco ou acessar a rede do paciente.

Figura 23 - Erro gerado pelo programa após verificar que a chave é incorreta




Fonte - Elaborado pelo autor

Portanto, esse teste de validação mostra que para realizar o acesso ou adicionar um novo bloco a rede, o nodo provedor deve possuir a autorização necessária. Isso pode se dar de várias formas, principalmente via APIs e SDKs de segurança mais robustas, porém para fins didáticos durante este trabalho foi utilizado esse método de par de chaves via biblioteca específica

## ANÁLISE DE DADOS

Como este trabalho propõe uma aplicação de uma rede *blockchain* privada para usuários de um sistema de saúde, pode-se verificar, juntamente com os testes realizados, que a segurança de dados é de fundamental importância nesta aplicação. Dito isto, como o paciente é o detentor das suas informações e dados, cabe às instituições, tanto públicas quanto privadas, requisitar o acesso ao usuário. Como neste trabalho foi utilizado a criação de um par de chaves criptografadas para simulação do processo, um detalhamento mais elaborado e completo deve ser feito para que esta etapa possa ser sem por cento confiável para a população.

Atualmente os dados dos pacientes do SUS são armazenados em bancos de dados locais e também no banco de dados nacional. Por isso a informação é centralizada nestes locais. O usuário pode obter acesso, mas sempre com o poder central público ditando como isso ocorre. Com a utilização do *framework* proposto, cada usuário detém o poder dos seus dados pessoais, tendo acesso ao seu histórico a qualquer momento sem a centralização da informação. Ainda assim nada impede que a gestão pública também tenha o acesso aos dados, porém não mais com tanto poder de escolha.

O processo de armazenamento das informações dos atendimentos clínicos e hospitalares permanece o mesmo, contudo, foi adicionado um novo bloco nesta etapa para que além de ser salvo localmente, o dado passa por um novo tratamento sendo adicionado a um bloco e, por fim, à rede privada do paciente. Isto garante que a informação esteja descentralizada e que não seja possível manipulá-la ou edita-la após estar adicionada ao bloco e a rede.

Todas essas questões fazem com que a ideia de utilização de uma rede *blockchain* em um sistema de saúde seja interessante, pois permite que os dados e informações nunca sejam manipulados e editados sem a autorização do paciente, além de deixar o controle das informações em posse da população. Isso permite também que um controle eficaz de gestão orçamentária e dos sistemas públicos possa usufruir deste sistema.

Os testes realizados mostram que um sistema neste nível é possível e viável. Cabe à gestão pública definir parâmetros e requisitos para que um sistema de alto nível seja criado. Todas as opções podem então ser adicionadas para que torne a

aplicação robusta e interessante tanto para os gestores e provedores de saúde quanto para a população que usufruirá da tecnologia empregada.

## CONCLUSÃO

Após a realização deste trabalho pode-se concluir que a ideia de aplicação de uma técnica como a *blockchain* no sistema de saúde pública do Brasil é aplicável. Isto é possível pois o fluxo e tratamento das informações e dados utilizados pelo governo brasileiro permite essa integração de tecnologia.

O método e controle de rastreamento do *blockchain* foi definido como sendo uma rede com permissões e privada. Ou seja, cada paciente possui controle e poder sob seus dados e informações de atendimentos passados e que estão adicionados em sua rede. Desta forma possibilita uma maior autonomia da população na questão de segurança e também um aumento na eficácia de atendimentos clínicos.

Cada sistema estudado em sua particularidade, e também no todo, possui um relacionamento diferente com cada parte do processo na saúde pública no Brasil. Porém, cada um possui uma ligação com sistema mestre que controla o fluxo destas informações e dados através dos diversos sistemas menores e seus armazenamentos. Como, por exemplo, o sistema SIA e SIH. Ambos são de complexas características porem possuem arquivos e dados específicos.

O sistema proposto neste estudo visou utilizar cada característica dos arquivos provenientes destes sistemas para que seja possível adicioná-lo a um bloco que posteriormente é adicionado a rede *blockchain* particular de cada paciente. Portanto, pode-se tanto utilizar arquivos gerados por médicos quanto por outros profissionais da saúde que fazem o atendimento mas que utilizam os já existentes aplicativos e softwares do governo.

Utilizando métodos específicos de manipulação de arquivos e segurança da informação, pode-se afirmar que a qualidade de controle e histórico dos pacientes pode ser melhorada, levando a diagnósticos precisos e com maiores chances de sucesso. Este correto controle pode ainda, como efeito positivo, contribuir para uma gestão orçamentária de maior qualidade e eficácia para o setor público do país. O estudo ainda permitiu evidenciar que atualmente o processo de tecnologia da informação do sistema de saúde pública do Brasil é de alto nível, e como consequência permite espaço para diversas ideias novas de integração aos sistemas existentes.

Como trabalhos futuros, sugere-se a criação de uma interface de usuário que comunique-se e tenha como base o sistema *back-end* desenvolvido. Desta forma a

criação de arquivos de saída e adição de novos blocos na rede  pode ser feita de forma amigável. Outra sugestão é a criação de uma rede de comunicações ponto-a-ponto com diversos nodos na rede que compartilham uma mesma *blockchain*. Desta forma possibilitando os testes de conexão e autenticação de chaves de acesso. Por fim, ainda é possível realizar um maior aprofundamento na aplicação e desenvolvimento de um sistema com arquivos de saída fidedignos aos encontrados hoje nos aplicativos de saúde pública, para que o gerenciamento e armazenamento das informações em banco de dados possa ser analisada e testada.

# REFERÊNCIAS


BASHIR, Imran. **Mastering Blockchain**: Distributed ledger technology, decentralization, and smart contracts explained. Second Edition. ed. rev. e atual. Birmingham, UK: Packt Publishing Ltd., 2018. 911 p. ISBN 978-1-78883-904-4.

CERQUEIRA, Daniel R. C.; ALVES, Paloma P.; COELHO, Danilo S. C.; REIS, Milena V. M.; LIMA, Adriana S. **Uma Análise da Base de Dados do Sistema de Informação Hospitalar entre 2001 e 2018**: Dicionário dinâmico, disponibilidade dos dados e aspectos metodológicos para a produção de indicadores sobre violência. Rio de Janeiro: Instituto de Pesquisa Econômica Aplicada – ipea, 2019. 160 p. ISBN 978-85-7811-357-5.

CHEN, Hannah S. et al. Blockchain in Healthcare: A Patient-Centered Model. **Biomedical Journal of Scientific & Technical Research**, [s. l.], v. 3, ed. 20, p. 15017-5022, 8 ago. 2019. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6764776/pdf/nihms-1046136.pdf. Acesso em: 26 nov. 2020.

CONASS. **Legislação do SUS**. Brasília: Conselho Nacional de Secretários de Saúde, 2003. 604 p. ISBN 85-89545-01-6.

CONASS. **SUS: avanços e desafios**. 1. ed. Brasília: Conselho Nacional de Secretaria de Saúde, 2006. 164 p. ISBN 85-89545-07-5

CONASS. **Ciência e Tecnologia em Saúde**: Coleção Para Entender a Gestão do SUS. 1. ed. Brasília: Conselho Nacional de Secretários de Saúde, 2011. 291 p. ISBN 978-85-89545-61-7.

CONASS. **Sistema Único de Saúde**: Coleção Para Entender a Gestão do SUS. 1. ed. Brasília: Conselho Nacional de Secretários de Saúde, 2011. 291 p. ISBN 978-85-89545-61-7.

DE AZEVEDO-MARQUES, Paulo Mazzoncini et al. Integração RIS/PACS no Hospital das Clínicas de Ribeirão Preto: uma solução baseada em "web". **Radiologia Brasileira**, São Paulo, v. 38, ed. 1, p. 37-43, 26 fev. 2004. DOI 10.1590/S0100-39842005000100009. Disponível em: http://www.scielo.br/pdf/rb/v38n1/23365.pdf. Acesso em: 27 nov. 2020.

DIMITROV, Dimiter V. Blockchain Applications for Healthcare Data Management. **Healthcare Informatic Research**, [s. l.], v. 1, ed. 25, p. 51-56, 31 jan. 2021. DOI 10.4258/hir.2019.25.1.51. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6372466/pdf/hir-25-51.pdf. Acesso em: 25 nov. 2020.

GUPTA, Manav. **Blockchain For Dummies®**: IBM Limited Edition. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2017. 51 p. ISBN 978-1-119-37123-6.

KUO, Tsung-Ting; KIM, Hyeon-Eui; OHNO-MACHADO, Lucila. Blockchain distributed ledger technologies for biomedical and health care applications. **Journal of the American Medical Informatics Association**, Oxford, UK, v. 24, ed. 6, p. 1211–1220, 8 set. 2017. DOI 10.1093/jamia/ocx068. Disponível em: https://academic.oup.com/jamia/article/24/6/1211/4108087?searchresult=1. Acesso em: 19 set. 2020.

MAGNAGNAGNO, Odirlei Antonio. **MECANISMOS DE PROTEÇÃO DA PRIVACIDADE DAS INFORMAÇÕES DE PRONTUÁRIO ELETRÔNICO DE PACIENTES DE INSTITUIÇÕES DE SAÚDE**. 2015. 138 f. Dissertação (Mestrado) - Curso de Administração e Negócios, Faculdade de Administração, Contabilidade e Economia, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2015.

MINISTÉRIO DA SAÚDE (Brasil). Secretaria de Atenção à Saúde; Departamento de Regulação, Avaliação e Controle; Coordenação-Geral de Sistemas de Informação (ed.). **RAAS – Registro das Ações Ambulatoriais de Saúde: Manual de Operação do Sistema**. 1.2. ed. Brasília, DF: [s. n.], Outubro 2012. 33 p.

MINISTÉRIO DA SAÚDE (Brasil). Secretaria de Atenção à Saúde; Departamento de Regulação, Avaliação e Controle; Coordenação-Geral de Sistemas de Informação (ed.). **APAC – Autorização de Procedimento Ambulatorial: Manual de Operação do Sistema**. 1. ed. Brasília, DF: [s. n.], Setembro 2012. 23 p.

MINISTÉRIO DA SAÚDE (Brasil). Secretaria de Atenção à Saúde; Departamento de Regulação, Avaliação e Controle; Coordenação-Geral de Sistemas de Informação (ed.). **Manual Técnico Operacional SIA/SUS: Sistema de Informações Ambulatoriais** 1. ed. Brasília, DF: [s. n.], Março 2010. 69 p.

MINISTÉRIO DA SAÚDE (Brasil). Secretaria de Atenção à Saúde; Departamento de Regulação, Avaliação e Controle; Coordenação-Geral de Sistemas de Informação (ed.). **BPA – Boletim de Produção Ambulatorial: Manual de Operação do Sistema**. 1. ed. Brasília, DF: [s. n.], Janeiro 2017. 104 p.

MINISTÉRIO DA SAÚDE (Brasil). Secretaria de Atenção à Saúde; Departamento de Regulação, Avaliação e Controle; Coordenação-Geral de Sistemas de Informação (ed.). **SIH – Sistema de Informação Hospitalar: Manual Técnico Operacional do Sistema** 1. ed. Brasília, DF: [s. n.], Setembro 2012. 25 p.

MINISTÉRIO DA SAÚDE. Departamento de Informática do SUS - DATASUS. **SIHD2 - Sistema de Informações Hospitalares Descentralizadas 2**: Manual de Operação. 1.0. ed. Rio de Janeiro, RJ: Processo de Documentação de Sistemas – PDOC, 2009. 213 p. Disponível em: ftp://ftp2.datasus.gov.br/public/sistemas/dsweb/SIHD/Manuais/MNL-PDOC-SIHD2-ManualOperacao-Edicao1.0.pdf. Acesso em: 21 mar. 2021.

PAIM, Jairnilson Silva et al. **O que é o SUS**. Rio de Janeiro: Editora Fiocruz, 2015. 93 p. ISBN 978-85-7541-453-8

PERCHE, Moacyr Esteves (coord.). **DATASUS e-SUS Hospitalar**. Brasil: Ministério da Saúde, [ca. 2015]. Disponível em:

http://www.cosemssp.org.br/downloads/INFORMACAO-CONGRESSO-E-SUS-HOSPITALAR.pdf. Acesso em: 12 set. 2020.

PORTAL DA SAÚDE. **DATASUS, 2020. e-SUS Hospitalar**. Disponível em: http://datasus1.saude.gov.br/sistemas-e-aplicativos/hospitalares/hospub. Acesso em: 12, setembro 2020.

PORTAL DA SAÚDE. **DATASUS, 2020. Histórico/Apresentação**. Disponível em: http://datasus1.saude.gov.br/datasus. Acesso em: 10, outubro 2020.

SOLHA, Raphaela Karla de Toledo. **Sistema Único de Saúde**: Componentes, Diretrizes e Políticas Públicas. 1. ed. São Paulo: Érica, 2014. 120 p. ISBN 978-85-365-1323-2.

ZHAO, Weijie. Blockchain technology: development and prospects. **National Science Review**, Oxford, UK, v. 6, ed. 2, p. 369-373, 13 nov. 2018. DOI 10.1093/nsr/nwy133. Disponível em: https://academic.oup.com/nsr/article/6/2/369/5181348. Acesso em: 16 set. 2020.