

Fundação Valeparaibana de Ensino
Colégio Técnico “Antônio Teixeira Fernandes”
Curso Técnico em Informática

EDILSON RIBEIRO DE ASSIS JUNIOR
JOÃO VITOR RANGEL TEIXEIRA DOMINGUES
KAUANE VITÓRIA DE LIMA FARIA
PEDRO HENRIQUE DOS SANTOS JANUÁRIO

**ESTUDO DE CASO PARA APLICAÇÃO DE BLOCKCHAIN EM CARTÓRIOS
CÍVEIS**

São José dos Campos, SP

2020

Edilson Ribeiro de Assis Junior

João Vitor Rangel Teixeira Domingues

Kauane Vitória de Lima Faria

Pedro Henrique dos Santos Januário

ESTUDO DE CASO PARA APLICAÇÃO DE BLOCKCHAIN EM CARTÓRIOS CÍVEIS

**Relatório final apresentado ao Colégio Univap –
Unidade Centro, como parte das exigências do curso
Técnico em Informática, para obtenção do Título de
Técnico em Informática.**

Orientador: Prof. Bruno Michel Pera

São José dos Campos, SP

2020

EDILSON RIBEIRO DE ASSIS JUNIOR
JOÃO VITOR RANGEL TEIXEIRA DOMINGUES
KAUANE VITÓRIA DE LIMA FARIA
PEDRO HENRIQUE DOS SANTOS JANUÁRIO

**ESTUDO DE CASO PARA APLICAÇÃO DE BLOCKCHAIN EM CARTÓRIOS
CÍVEIS**

Relatório final aprovado para obtenção do título de Técnico em Informática, do Curso Técnico em Informática, do Colégio Técnico “Antônio Teixeira Fernandes”, da Fundação Valeparaibana de Ensino, São José dos Campos, SP, pela seguinte banca avaliadora:

Orientador: Prof. Bruno Michel Pera (FVE):_____

Membro: Prof. Wagner dos Santos Clementino de Jesus (FVE):_____

Membro: Prof. Hélio Lourenço Esperidião Ferreira (FVE):_____

Membro: Prof. Alberson Wander Sá dos Santos (FVE):_____

São José dos Campos, 6 de novembro de 2020

**A Deus, nossos familiares e amigos e ao
nosso orientador.**

AGRADECIMENTOS

Aos nossos pais e familiares pelo apoio e incentivo.

Ao professor Bruno Michel Pera pela orientação, apoio e confiança.

Ao Colégio Técnico Univap Centro, pela oportunidade de fazermos o curso.

“Talento é 1% inspiração e 99% transpiração.”

(Thomas Edison)

RESUMO

Tendo em vista que cartório é a repartição onde funcionam os registros governamentais e são movidos a “Fé pública”, que é possível definir como sendo o crédito dado a documentos emanados de autoridades públicas ou serventuários da justiça. As informações podem sofrer com corrupção de demasiadas formas, o que pode alterar documentos altamente necessários para o funcionamento da sociedade, afetando assim os principais dados de todos os cidadãos.

Desta forma, o objetivo deste projeto é realizar um protótipo simulando de uma maneira simplificada uma *Blockchain*, que é uma estrutura de dados que permite criar um livro-razão de dados digital e testar a confiabilidade e imutabilidade dos dados inseridos, onde será implementada em um ambiente de um cartório virtual.

Será desenvolvida uma interface *web* na linguagem Javascript e HTML (*HyperText Markup Language*) contendo telas para as certidões de casamento, nascimento e óbito, onde o usuário irá inserir os dados necessários para o cadastro da certidão no sistema. Quando um cadastro é realizado, o sistema do replicador coleta os dados do banco de dados principal e os replica para os bancos das camadas 1 e 2, enquanto o sistema do validador funciona de maneira independente a cada 4 segundos validando os dados.

Palavras-chave: Cartório, *Blockchain*, Segurança, Imutabilidade.

ABSTRACT

Bearing in mind that the notary's office is the place where public registries operate and work based on "Public Faith", which can be defined as the credit given to documents issued by public authorities or judicial servants. The data can suffer from corruption in too many ways like changing documents that are highly necessary for the functioning of society, thus affecting the main data of all citizens.

In this way, the objective of this project is to create a prototype simulating in a simplified way a Blockchain that can be defined as a data structure that makes it possible to create a digital ledger, where it will be implemented in a virtual environment and test the reliability and immutability of the inserted data.

A web interface will be developed in Javascript and HTML (HyperText Markup Language) containing screens for the marriage, birth, and death certificates, where the user will insert the necessary data for the registration of the certificate in the system. When a registration is made the system from the replicator, it collects the data from the main database and replicates it to the layer 1 and layer 2 banks, while the validator system works independently every 4 seconds validating the data.

Keywords: Notary's office, Blockchain, Security, Immutability.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de hashes em cada camada de banco de dados.....	18
Figura 2 - Código binário na tabela ASCII da palavra "BOLA"	20
Figura 3 – Código binário concatenado dividido em grupos de 6.....	20
Figura 4 - Divisão dos grupos com o último bloco de 6 completo.....	20
Figura 5 - Binário convertido em grupos de 8.....	21
Figura 6 - Decimais referentes aos binários de 8 (bits)	21
Figura 7 - Palavra "BOLA" convertida em Base64.....	21
Figura 8 - Diagrama de Caso de Uso.....	22
Figura 9 - Diagrama de Classe	23
Figura 10 - Diagrama de Sequência Certidão de Casamento	24
Figura 11 - Diagrama de Sequência Certidão de Óbito.....	25
Figura 12 - Diagrama de Sequência Certidão de Nascimento.....	26
Figura 13 - Diagrama Hierárquico	27
Figura 14 - Tabela Custo do Sistema	27
Figura 15 – Imagem representativa as três camadas de bancos de dados com registro concluído	29
Figura 16 - Exemplo de discrepância entre os bancos de dados	30
Figura 17 -Verificação da veracidade do dado	30
Figura 18 - Página "Menu"	31
Figura 19 - Primeira parte da página “Certidão de Nascimento”	32
Figura 20 - Segunda parte da página “Certidão de Nascimento”	32
Figura 21 - Primeira parte da página “Certidão de Casamento”	33
Figura 22 - Segunda parte da página “Certidão de Casamento”	33
Figura 23 - Primeira parte da página “Certidão de Óbito”	34
Figura 24 - Segunda parte da página “Certidão de Óbito”	34
Figura 25 - Final da página “Certidão de Óbito”	35

SUMÁRIO

1. INTRODUÇÃO	13
1.1 OBJETIVOS	15
1.1.1 Objetivos Gerais:	15
1.1.2 Objetivos Específicos:	15
2. FUNDAMENTAÇÃO TEÓRICA.....	16
2.1 HTML	16
2.2 CSS.....	16
2.3 BOOTSTRAP	16
2.4 FIREBASE	16
2.5 JAVASCRIPT	16
2.6 VISUAL STUDIO CODE	16
2.7 BLOCKCHAIN	17
2.8 SMARTCONTRACTS.....	17
2.9 NOSQL	17
2.10 TECNOLOGIA P2P	17
2.11 BASE64.....	18
2.12 HASH.....	18
3. METODOLOGIA.....	19
3.1 DIAGRAMAS	22
3.1.1 DIAGRAMA DE CASO DE USO (UML):.....	22
3.1.2 DIAGRAMA DE CLASSE:.....	23
3.1.3 DIAGRAMA DE SEQUÊNCIA:.....	24
3.1.3.1 Diagrama de Sequência Certidão de Casamento	24
3.1.3.2 Diagrama de Sequência Certidão de Óbito.....	25
3.1.3.3 Diagrama de Sequência Certidão de Nascimento.....	26
3.1.4 DIAGRAMA HIERÁRQUICO:	27

3.2 ANÁLISE DE CUSTO-BENEFÍCIO	27
3.3 HARDWARE MÍNIMO PARA UMA BOA EXPERIÊNCIA:.....	28
4. RESULTADOS E DISCUSSÃO	29
4.1 Resultados	29
4.2 Manual do usuário.....	31
4.3 Discussão	35
5. CONCLUSÃO.....	36
REFERÊNCIAS.....	37

INTRODUÇÃO

Cartório é a repartição onde funcionam os tabelionatos, os escritórios de notas, os registros públicos, da justiça, e onde se mantêm os respectivos arquivos. Desde a época do Brasil colonial os cartórios já estavam presentes na organização de informações individuais, porém somente no ano de 1988 a instituição foi consolidada, pelo artigo 236 da Constituição Federal (COSTA, 2020).

O professor José Maria Othon Sidou, no dicionário jurídico da Academia Brasileira de Letras Jurídicas, conceitua corrupção como: “Devassidão, depravação, em suas diversas modalidades; improbidade no trato de coisas públicas, na condição ativa ou passiva.” Os males da corrupção são imensuráveis. Corrupção se torna um grande peso e um entrave³ ao desenvolvimento (MENDES, 2020).

Por conta de os cartórios serem regidos por um sistema de fé pública, onde uma pessoa passível de corrupção pode ser designada para comandar um local que se passa dados altamente necessários para o funcionamento da sociedade o sistema de controle do país fica vulnerável à perda e corrupção de materiais.

Tendo os fatos mencionados em vista, o presente projeto busca desenvolver uma aplicação web que terá uma tela para a inscrição de dados para certidão de nascimento, certidão de casamento e certidão de óbito e uma tela de menu. Sendo programado na linguagem Javascript pela plataforma *Visual Studio Code* com o banco de dados Firebase e a tecnologia *Blockchain*, para auxiliar no combate à corrupção e na segurança da identidade dos brasileiros.

Originalmente, *Blockchain* era somente o termo da informática para a estruturação e compartilhamento de dados. Hoje, *Blockchain* são aclamados como a "quinta evolução" da computação (LAURENCE, 2019).

Uma *Blockchain* é uma estrutura de dados que torna possível criar um livro-razão de dados digital e compartilhá-lo em uma rede de grupos independentes. Há muitos tipos diferentes de *Blockchain*: As públicas, como a bitcoin, que são abertas à participação de qualquer um, tem o código aberto para a sua comunidade e são administradas por meio de um *token* nativo;

As permissionadas, como o *Ripple*, que controlam funções que pessoas podem desempenhar dentro da rede, são sistemas amplos e difusos que usam *token* nativo e seu código central pode ou não ser aberto;

As privadas, tendem a ser menores, não utilizam *token*, normalmente são protegidos por associações que tem membros conceituados e informações comerciais confidenciais.

Todos os três tipos de *Blockchain* usam criptografia, que permite a qualquer participante de qualquer rede determinada pode gerenciar o livro-razão de um jeito seguro, eliminando uma autoridade central que pode ser corrupta (GONÇALVES, 2017).

1.1 OBJETIVOS

1.1.1 Objetivos Gerais:

Realizar um protótipo desenvolvido em Javascript simulando de uma maneira simplificada uma *Blockchain* que será um cartório virtual e testar a confiabilidade e imutabilidade dos dados lá inseridos.

1.1.2 Objetivos Específicos:

Implementar um sistema de cadastro utilizando uma aplicação *web* com banco de dados Firebase, que fará a simulação de uma *Blockchain* de 3 elos. O banco de dados recebe os dados do usuário digitados nos campos das telas de documentação referente as: Certidão de nascimento, casamento e óbito. O sistema deve contar com três camadas de banco de dados e para ter uma maior segurança serão criptografados com *hashes*, onde haverá uma dependência dos blocos conectados entre si, ou seja, o próximo bloco de dados a ser criado necessita do anterior.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 HTML

A linguagem *HTML* foi criada por Tim Barners Lee na década de 1990, com especificações da linguagem são controladas pela W3C. Se consiste em uma linguagem de marcação utilizada para produção de páginas *web*, que permite a criação de documentos que podem ser lidos em praticamente qualquer tipo de computador e transmitidos pela *internet* (FREEMAN; ROBSON, 2014).

2.2 CSS

O *CSS (Cascading Style Sheets)* é uma linguagem que complementa, formata e estiliza o *HTML* organizando melhor as linhas e adicionando novas possibilidades ao código, como alterando as margens, fontes, posições e até mesmo a própria estrutura do site (VIERA, 2020).

2.3 BOOTSTRAP

O *Bootstrap* é um *framework* que tem suporte a *HTML*, *CSS* e JavaScript. Que possibilita a criação de projetos responsivos. O que leva a eficiência deste, vem do fato de apenas um *framework* suportar qualquer dispositivo. De uma maneira fácil e eficiente, o *Bootstrap* adapta seus projetos *Web* com um simples código (BOOTSTRAP, 2020).

2.4 FIREBASE

É uma plataforma para a criação de aplicativos móveis e de aplicações para a *web*, esta plataforma foi fundada no ano de 2010 por James Tamplin e Andrew Lee, porém atualmente é comandada pela empresa *Google* (BORGES. et al, 2017).

2.5 JAVASCRIPT

É uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma e é uma das principais tecnologias da *World Wide Web*. Criada por Brendan Eich e surgiu no dia 4 de dezembro de 1995 e a versão mais recente foi lançada no ano de 2018 (MICHAEL, 2008).

2.6 VISUAL STUDIO CODE

Visual Studio Code é um editor de código-fonte que inclui suporte para depuração, controle *Git* incorporado, realce de sintaxe, complementação inteligente de código, *snippets* e refatoração de código, além de ser especialmente desenvolvido pela Microsoft para Windows, *Linux* e *MacOs* lançado no dia 29 de abril de 2015.

2.7 BLOCKCHAIN

O conceito da *Blockchain* começa a deixar claro que vai muito além da inovação tecnológica. Por conta do seu sistema P2P (*peer-to-peer*), que causa uma descentralização nas transações, todos os participantes das redes são iguais, sendo assim possível a criação de um livro-razão de dados virtual criptografado por *hashes* onde o sistema fica mais seguro e com menos chance de ser comprometido.

O sistema apresenta imutabilidade (sendo muito difícil remover ou alterar o que está salvo), distribuição igualitária (visto que não possui uma autoridade central gerenciando o fluxo de dados, uma maneira de manter a integridade dos dados é ter uma rede de usuários independentes em lugares diferentes) e criptografia (privatização de informações pessoais), além de outras diversas capacidades (BASSOTTO, 2020).

2.8 SMARTCONTRACTS

Segundo Nick Szabo, jurista e criptógrafo renomado, um contrato inteligente como uma ferramenta capaz de tornar as redes de computador seguras e formais, combinando protocolos com interfaces de usuário. Ou seja, um aplicativo ou programa que é executado em uma *Blockchain*. Normalmente, eles funcionam como um acordo digital que é aplicado com um conjunto específico de regras. Essas regras são predefinidas por códigos computacionais, que são replicados e executados em todos os nós da rede.

É simplesmente um programa executável, com funções e dados especificados em um endereço na Blockchain. Inviolável e independente do controle externo, este software assegura que um contrato será executado como foi escrito (GONÇALVES, 2017).

2.9 NOSQL

O termo *noSQL* foi criado pensando em projetos *open source* (código aberto, de livre utilização) de armazenamento de dados sem utilizar inteiramente o modelo relacional (ALEXANDRE, 2010). Assim, o que pode ser entendido como "não apenas *SQL*", apresenta alto desempenho e performance em bancos de dados não relacionais com dados não estruturados. Diferente do *SQL*, que é uma linguagem de consulta, o *noSQL* precisa ser acompanhado por uma linguagem de programação para manipular o banco (DEV MEDIA, 2020).

2.10 TECNOLOGIA P2P

P2P é uma arquitetura de rede aonde todos as máquinas conectadas nele são tanto servidores quanto clientes, assim é permitido que o serviço dos dados seja compartilhado sem a

necessidade de um servidor central. Essa tecnologia que possibilita a *Blockchain* ser descentralizada, tendo mais segurança dos dados, pois se existisse apenas um servidor central seria mais fácil de invadi-lo (DE FREITAS, 2002).

2.11 BASE64

Base64 é um método para codificação de dados para facilitar a transferência na Internet, codificação MIME (*Multipurpose Internet Mail Extensions*) para transferência de conteúdo. É utilizado frequentemente para transmitir dados binários por meios de transmissão que lidam apenas com texto. É constituído por 64 caracteres ([A-Z], [a-z], [0-9], "/" e "+") que deram origem ao seu nome. Seu intuito é deixar a informação mais fácil de ser transmitida (JOSEFSSON. et al, 2006).

2.12 HASH

Hash, ou resumo da mensagem é um algoritmo que recebe qualquer comprimento de entrada (de dados/mensagem) e mescla a entrada para produzir uma saída pseudoaleatória de largura fixa. A *hash* impossibilita a reconstrução da mensagem de entrada a partir de uma saída *hash* (função de uma única via). O que mais diferencia a *hash* de uma criptografia comum é que a *hash* não é reversível, já a criptografia é reversível (JUNIOR, 2012).

Na figura 1, pode-se observar as camadas do banco de dados onde estão presentes as *hashes* referentes aos dados de cada camada, no banco de dados principal o nó principal contém somente os dados, já na camada 1 o nó principal contém os dados e a *hash* do banco de dados principal e na camada 2 o nó principal contém os dados e a *hash* da camada 1.

Figura 1 - Exemplo de hashes em cada camada de banco de dados



Fonte: o autor (2020).

3. METODOLOGIA

Na realização deste projeto, primeiramente, criou-se um banco de dados na plataforma para a criação de aplicativos móveis e de aplicações para a *web*, chamado Firebase (BORGES. Et al, 2017).

Após isso, montou-se a tela de certidão de nascimento na linguagem de marcação utilizada para produção de páginas *web*, chamado *HTML* (FREEMAN. et al, 2014). E a linguagem que complementa, formata e estiliza o *HTML*, chamado *CSS* (VIERA, 2020). E institui-se a ligação entre está e o banco de dados, disponibilizando a função inserir.

Assim, foi realizado o mesmo procedimento na montagem das telas de certidão de óbito e de casamento. Em seguida, estabeleceu-se a função de busca no banco de dados a partir do nó principal (nome da mãe no caso da certidão de nascimento, RG do finado na certidão de óbito e nome do presidente da celebração na certidão de casamento).

Após alguns dias foi reorganizado o código para um maior entendimento e foi inserida no código a função deletar para cada tela, onde são deletadas as informações pelo nó principal. Após um hiato de um mês foi finalizada a programação do sistema de validação e segurança dos dados, imutável, descentralizado e transparente, chamado *Blockchain* (BASSOTTO, 2020).

Primeiro foi programado o replicador que insere os dados do banco de dados principal e os envia para o banco de dados “camada 1” e do banco “camada 1” para o banco “camada 2”. Logo depois foi programado o validador que realiza uma comparação dos dados dos três bancos a cada quatro segundos, esse tempo foi estimado para a preservação das máquinas dos integrantes do grupo, se for encontrado uma variação em algum dado o validador a compara e substitui pelo valor da maioria.

Um erro lógico foi encontrado, para fazer a busca é necessário utilizar o nó principal do Firebase, porem, devido a criptografia ao se fazer alteração ou exclusão de um dado as informações seriam comprometidas, pois estariam de baixo do mesmo nó. Para que a Blockchain funcionasse corretamente não seria possível o funcionamento dos botões buscar, excluir e alterar. O erro consistia no nó principal, o sistema de banco de dados Firebase que foi utilizado para a realização do projeto contém uma criptografia para cada nó, assim quando se buscava no replicador os dados, ele replicava somente o nó criptografado pelo Firebase. Por esse motivo os botões foram excluídos juntamente com suas funções.

O sistema de dados utilizado é não relacional, chamado de noSQL, onde não há tabelas de linhas e colunas e a manipulação de dados ocorre por comandos em linguagem de programação, no caso Javascript. A escolha desse recurso foi baseada na alta escalabilidade

que garante um bom gerenciamento e alta disponibilidade de dados com um baixo custo de manutenção e a possibilidade de ter um banco de dados com atualizações em tempo real (BERNADETTE. et al, 2011).

Utilizou-se o modelo de criptografia Base64 devido a sua melhor eficiência e interação com a linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma, chamada Javascript e, pois, seu número de bytes é sustentável para o sistema (MICHAEL, 2008).

O funcionamento do algoritmo de codificação Base64 usa uma lógica simples de codificação, primeiramente é encontrado na tabela ASCII o código binário correspondente a cada letra da palavra “BOLA”, exemplo usado no artigo de referência, como pode-se ver na figura abaixo:

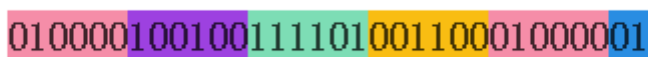
Figura 2 - Código binário na tabela ASCII da palavra "BOLA"



Fonte: FERNANDES (2020).

Logo depois é concatenado todos os códigos binários e divididos em grupos de 6, como pode-se ver na figura abaixo:

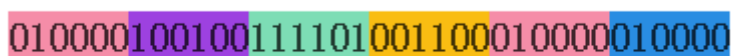
Figura 3 – Código binário concatenado dividido em grupos de 6



Fonte: FERNANDES (2020).

Na palavra “BOLA”, em específico, o último bloco da divisão concatenada do código binário tem seu tamanho menor, esse espaço vazio é completado por “zeros” a direita, nesse exemplo são 4 “zeros”, como pode-se ver na figura abaixo:

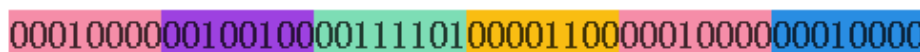
Figura 4 - Divisão dos grupos com o último bloco de 6 completo



Fonte: FERNANDES (2020).

Logo após dividir em grupos de 6 (*bits*) é convertido em grupos de 8 (*bits*), colocando dois “zeros” no início de cada bloco, como pode-se ver na figura abaixo:

Figura 5 - Binário convertido em grupos de 8



Fonte: FERNANDES (2020).

Com todos os blocos de 8 (*bits*) completos, é consultado na tabela ASCII o número decimal referente aos binários, obrigatoriamente os grupos precisam ser completos de 24 *bits*, caso ocorra de um grupo não conter 24 *bits* é adicionado o marcador “=”, que tem seu valor nulo, como pode-se ver na figura abaixo:

Figura 6 - Decimais referentes aos binários de 8 (*bits*)



Fonte: FERNANDES (2020).

Por fim, é consultado a tabela de conversão do Base64 referente aos decimais, assim pode-se ver na figura abaixo a palavra “BOLA” criptografada pelo algoritmo Base64:

Figura 7 - Palavra "BOLA" convertida em Base64



Fonte: FERNANDES (2020).

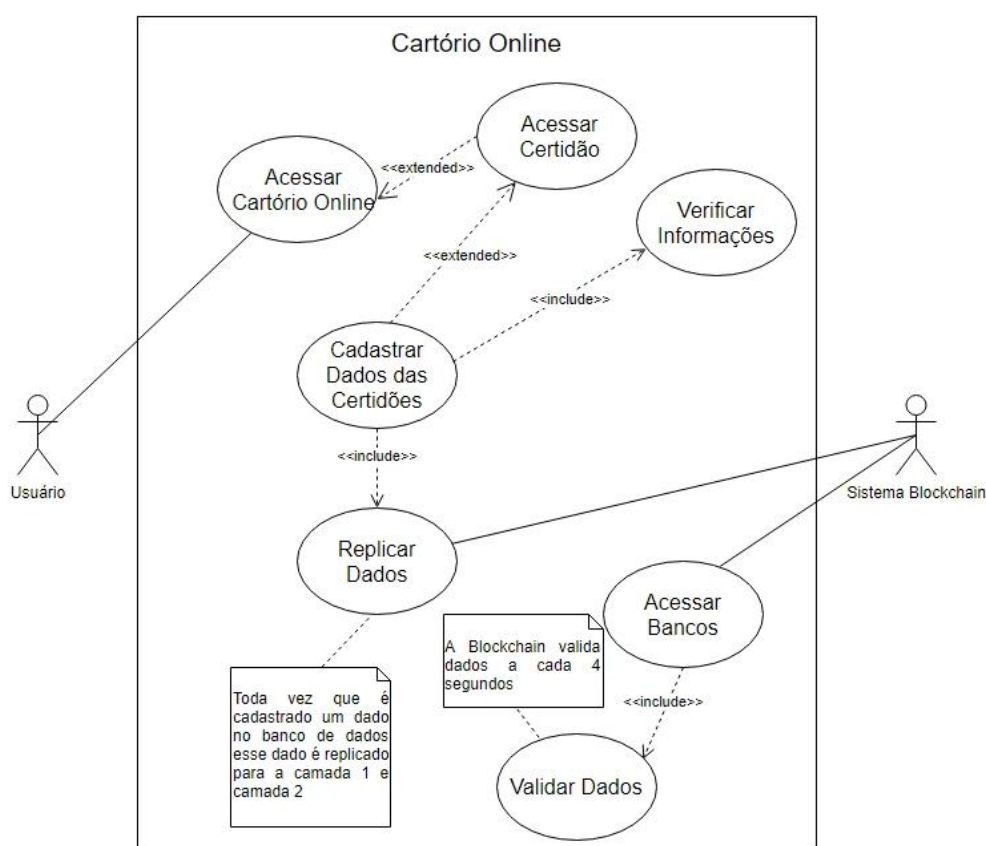
3.1 DIAGRAMAS

3.1.1 DIAGRAMA DE CASO DE USO (UML):

O usuário tem três ações possíveis na *Blockchain*: acessar uma certidão (de nascimento, de casamento ou de óbito), inserir informação e cadastrar dados. A primeira ação pode ser seguida pela segunda ou pela terceira, sendo que, a inserção no banco é verificada e notificada para o usuário, dependendo do resultado e o cadastro é sempre replicado e validado pelo sistema.

Ao acessar a página principal o usuário é apresentado a uma interface web, que está diretamente conectada ao banco de dados Firebase, esta página não faz nenhum tipo de inserção em camada que não seja a camada gênese e a *Blockchain* é a responsável por efetuar a replicação dos dados, ele se aplica acessar certidão e cadastrar dados. A página replicar dados conforme já indicado pelo próprio nome, efetua a transição de dados em sua completude para as outras camadas. A classe validadora garante que os dados não foram alterados fazendo comparação entre eles a cada 4 segundos.

Figura 8 - Diagrama de Caso de Uso

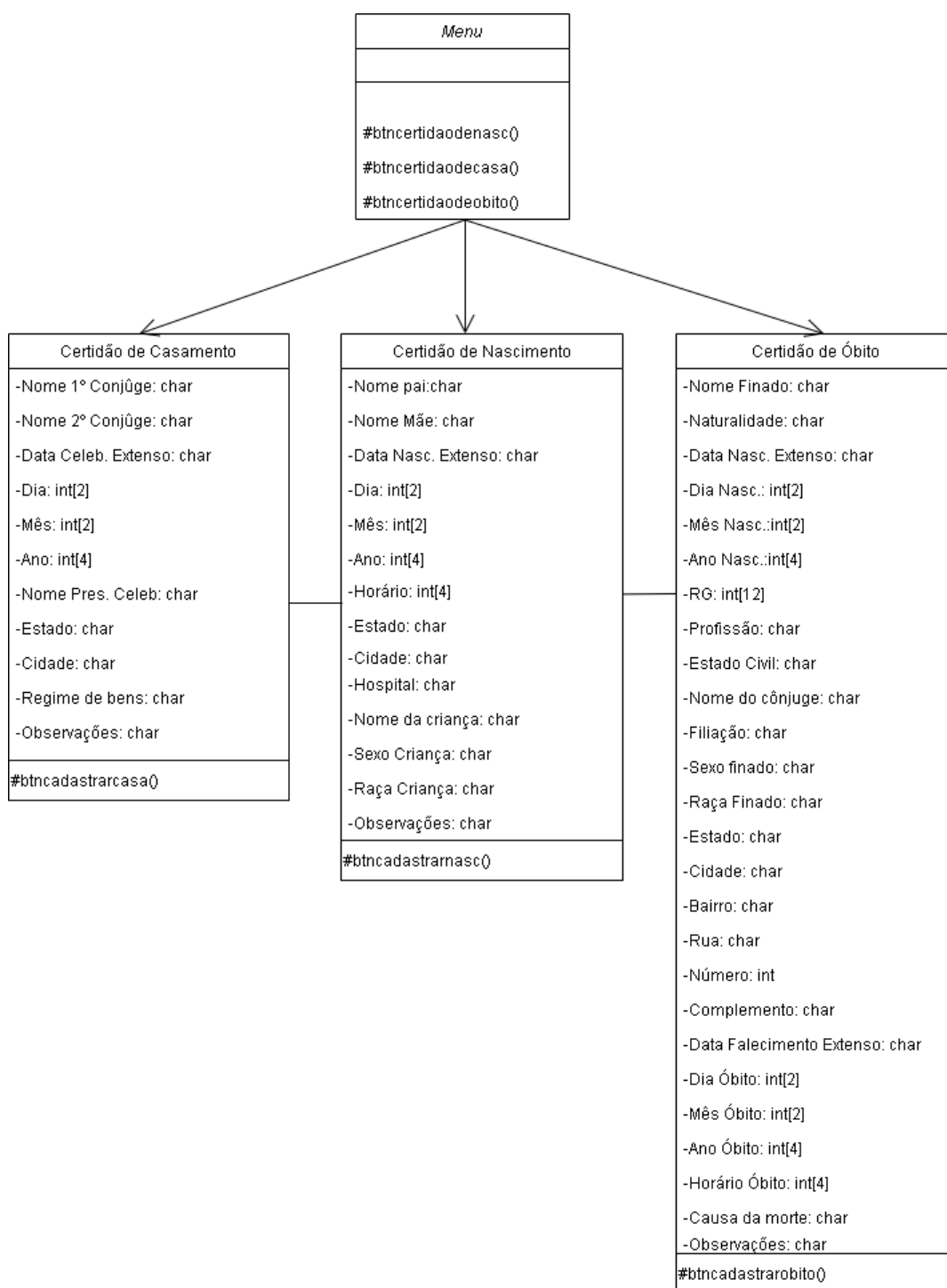


Fonte: o autor (2020)

3.1.2 DIAGRAMA DE CLASSE:

Na figura 8 pode-se observar o diagrama de classe, onde está presente a classe “Menu” que apresenta 3 botões sem passagem de parâmetro, além das classes “Certidão de Casamento”, “Certidão de Nascimento” e “Certidão de Óbito”.

Figura 9 - Diagrama de Classe



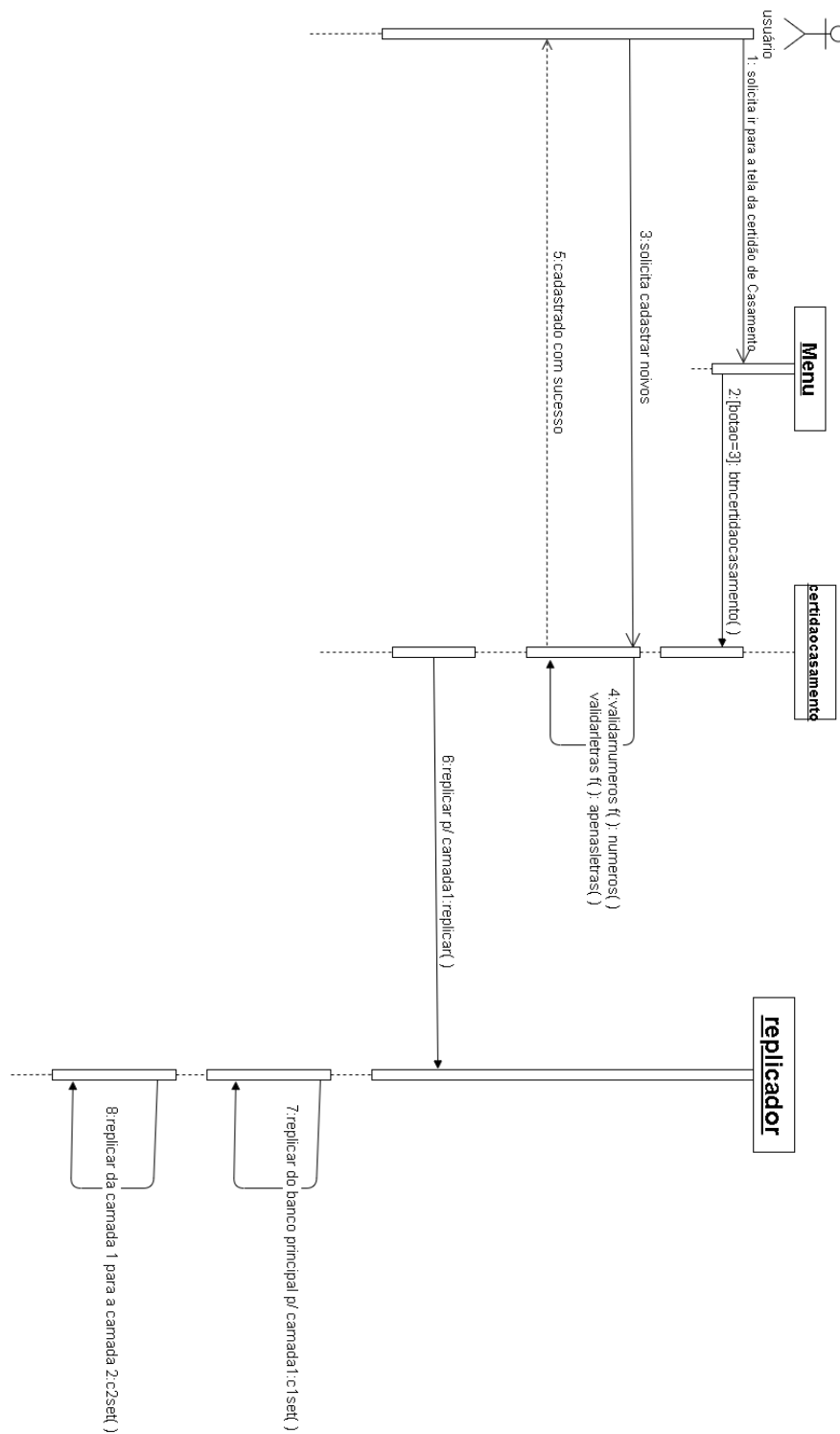
Fonte: o autor (2020)

3.1.3 DIAGRAMA DE SEQUÊNCIA:

3.1.3.1 Diagrama de Sequência Certidão de Casamento

Ao clicar acessar a certidão de casamento, o usuário realiza o cadastro, após a validação dos campos (verificação de apenas números e letras nos campos permitidos).

Figura 10 - Diagrama de Sequência Certidão de Casamento

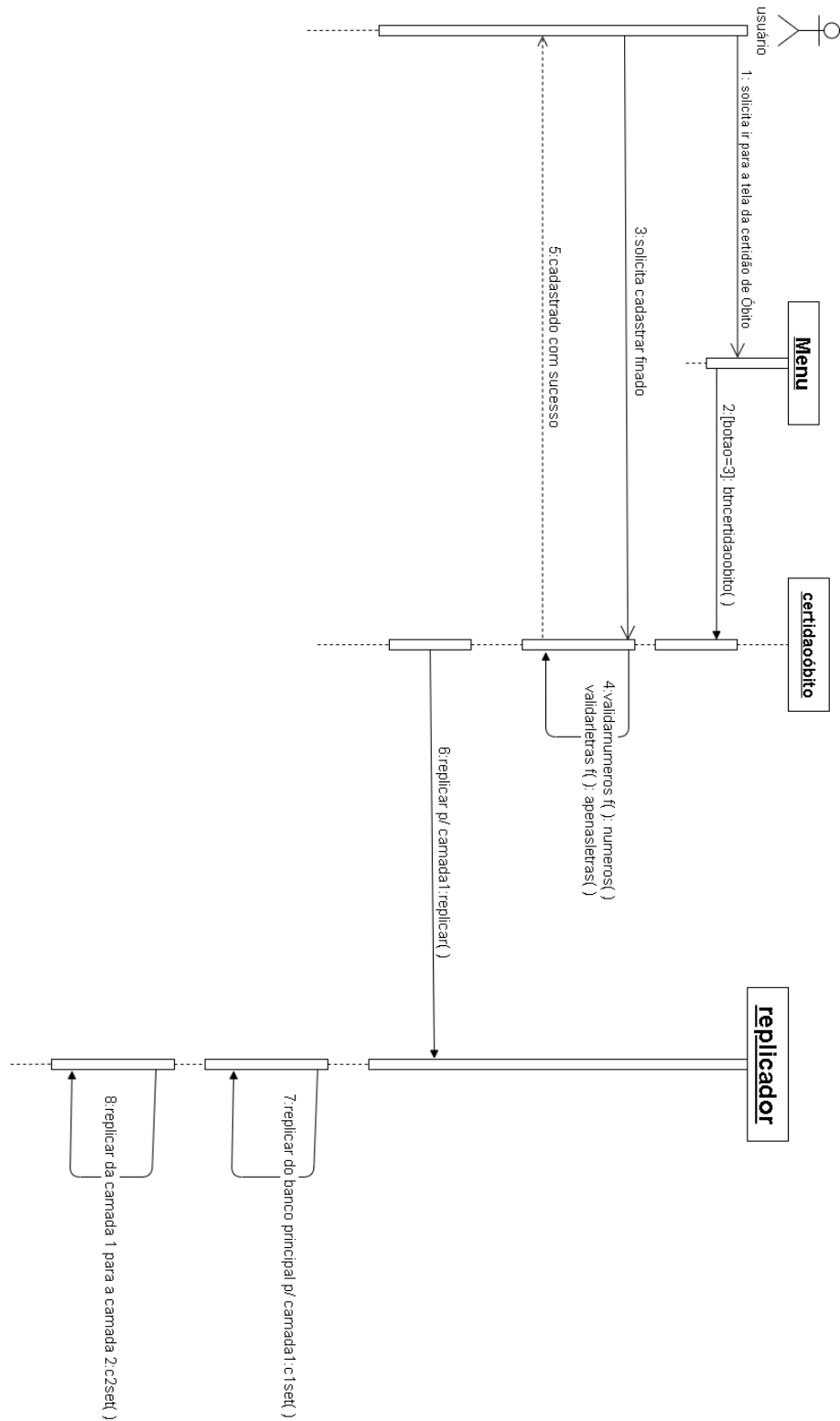


Fonte: o autor (2020)

3.1.3.2 Diagrama de Sequência Certidão de Óbito

O processo no cadastro de finados é o mesmo de cônjuges, visto que o usuário solicita cadastro e após aprovação, as informações são replicadas.

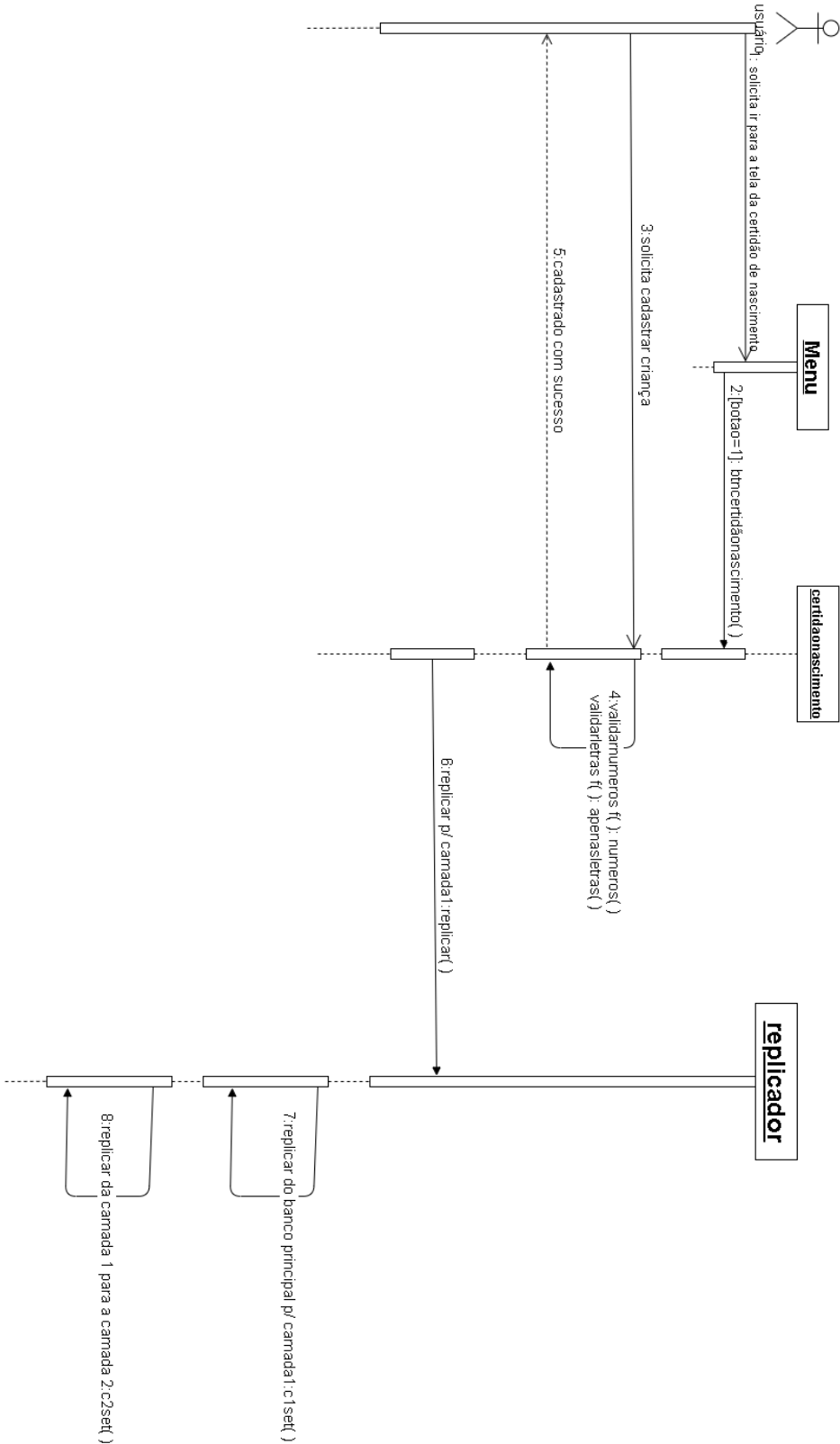
Figura 11 - Diagrama de Sequência Certidão de Óbito



3.1.3.3 Diagrama de Sequência Certidão de Nascimento

Na tela de certidão de nascimento, o procedimento é o mesmo apresentado anteriormente.

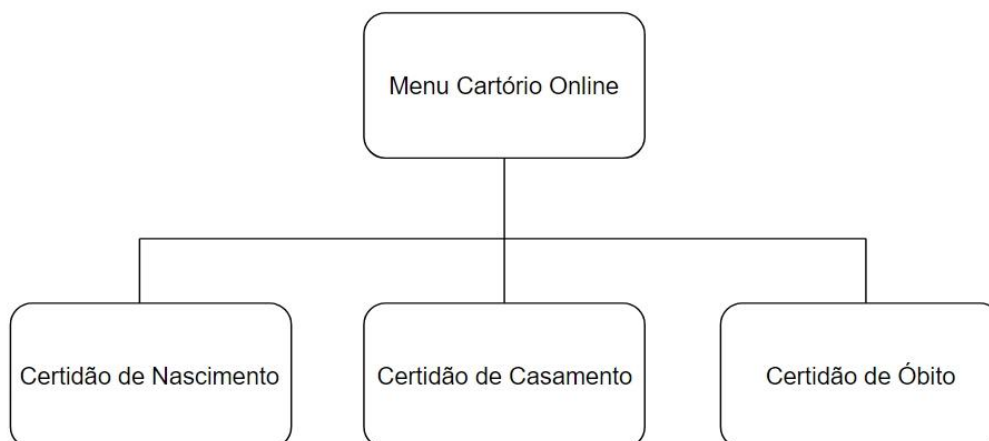
Figura 12 - Diagrama de Sequência Certidão de Nascimento



3.1.4 DIAGRAMA HIERÁRQUICO:

Todas as referências que entram no sistema passam pelo banco principal, que herda os bancos denominados “camada1” e “camada2”, respectivamente de acordo com a linha de sucessão, e são verificadas, replicadas e validadas, criando uma corrente segura.

Figura 13 - Diagrama Hierárquico



Fonte: o autor (2020)

3.2 ANÁLISE DE CUSTO-BENEFÍCIO

A análise de custo/benefício do cartório apresentada a seguir, leva à conclusão de que a *Blockchain* é mais barata, considerando o trabalho do programador Javascript e do servidor mais energia elétrica, em relação ao sistema físico, avaliando o custo geral na instituição. Além disso, é inegável que a virtualização traz uma segurança quase imbatível enquanto a fisicalidade traz consigo a sujeição dos funcionários à erros. Ademais, há a facilidade e acessibilidade na realização de cadastros e na utilização do modelo.

Figura 14 - Tabela Custo do Sistema

Método	Método Online(Blockchain)	
Despesa	Programador Javascript	Servidor
Valor/Mês	≅R\$5.850,00	≅R\$2.500,00 + energia
Valor/Hora	≅R\$39,00	≅R\$16,66 + energia
Horas trabalhadas/em uso	168 horas	168 horas
Valor total	≅R\$6.552,00	≅R\$2.798,88 + energia

Fonte: o autor (2020)

3.3 HARDWARE MÍNIMO PARA UMA BOA EXPERIÊNCIA:

Processador: Processador de 1 giga-hertz (GHz) ou mais rápido ou Sistema em um chip (SoC);

RAM: 1 gigabyte (GB) para 32 bits ou 2 GB para 64 bits;

Espaço em disco rígido: 16 GB para um sistema operacional de 32 bits ou 32 GB para um de 64 bits;

Placa gráfica: DirectX 9 ou posterior com driver WDDM 1.0;

Resolução de tela: 1366x768.

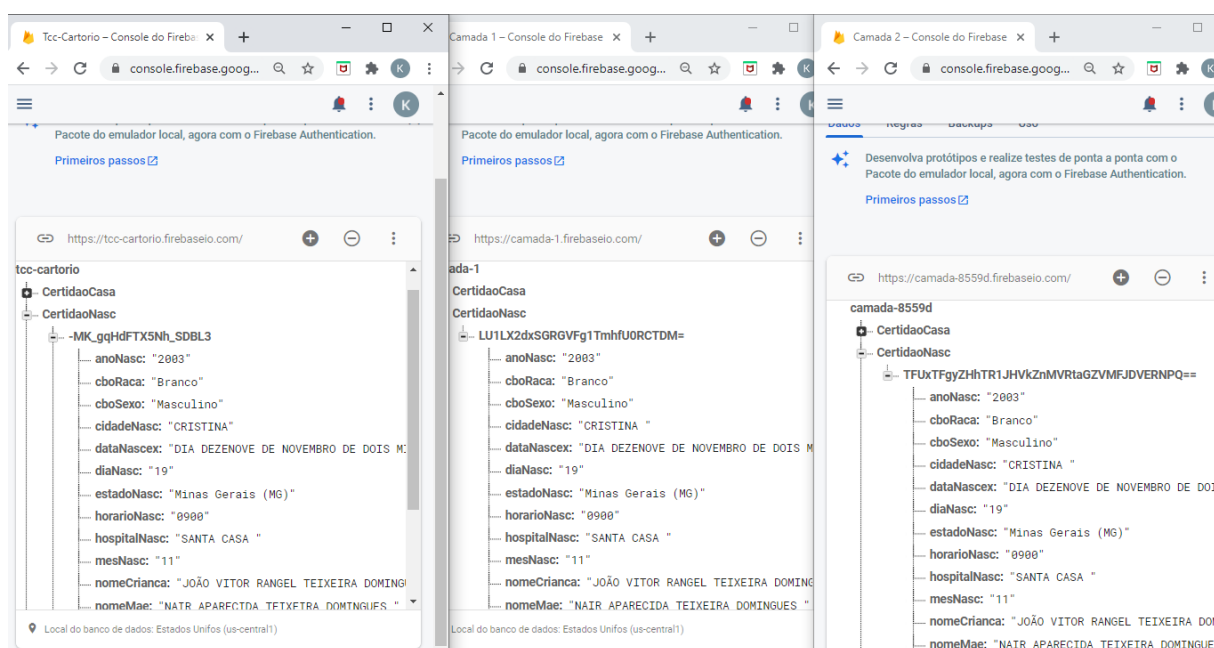
4. RESULTADOS E DISCUSSÃO

4.1 Resultados

O projeto de um cartório civil online usando *Blockchain* consegue efetuar o cadastro de nascimentos, de casamentos e de óbitos. Todos os dados, fornecidos pelo usuário para criação de uma certidão, são recebidos por um sistema web, onde ocorre replicação (“replicador.html”), na qual são transferidos para os outros bancos de dados, assim já os tornando imutáveis e, em seguida, o sistema de validação (“validador.html”), que realiza de dados a cada 4 segundos, utiliza um conjunto de *arrays* das informações verificando a integridade dos elementos entre os bancos.

Ao realizar um cadastro, todos os dados são registrados no banco principal e em seus dois sucessores, ou camada 1 e 2, por meio do replicador, como mostra a figura abaixo.

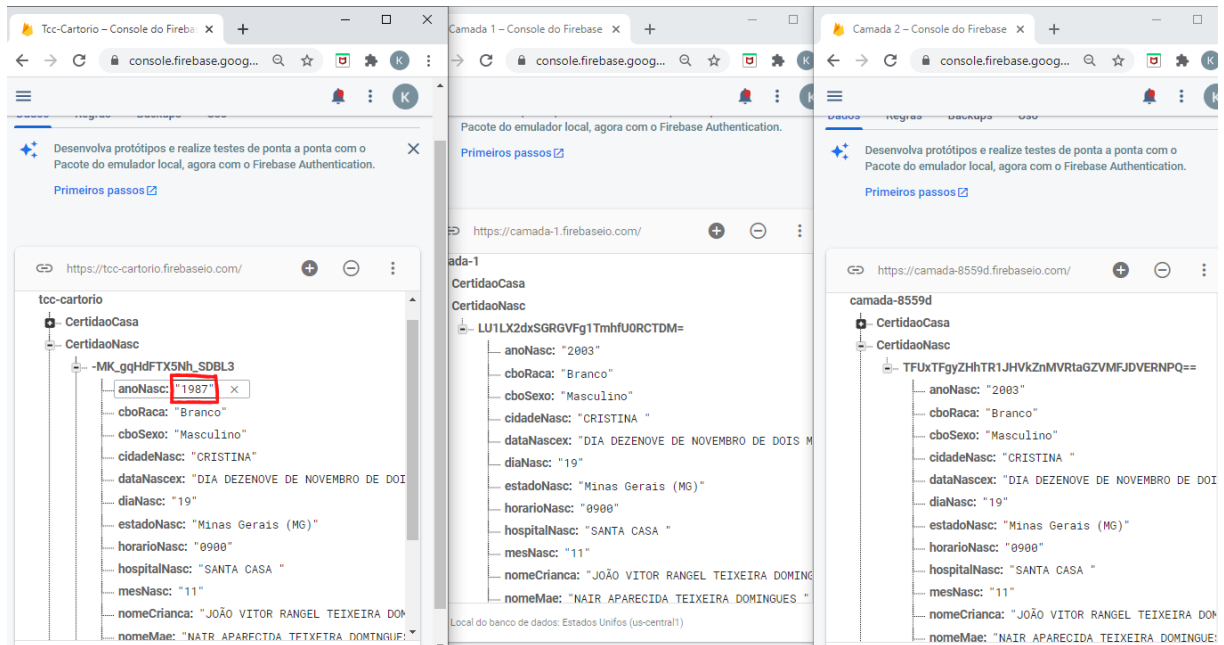
Figura 15 – Imagem representativa as três camadas de bancos de dados com registro concluído



Fonte: o autor (2020)

Se algum elemento for modificado no banco, é realizada a validação e correção conforme comparação entre as camadas.

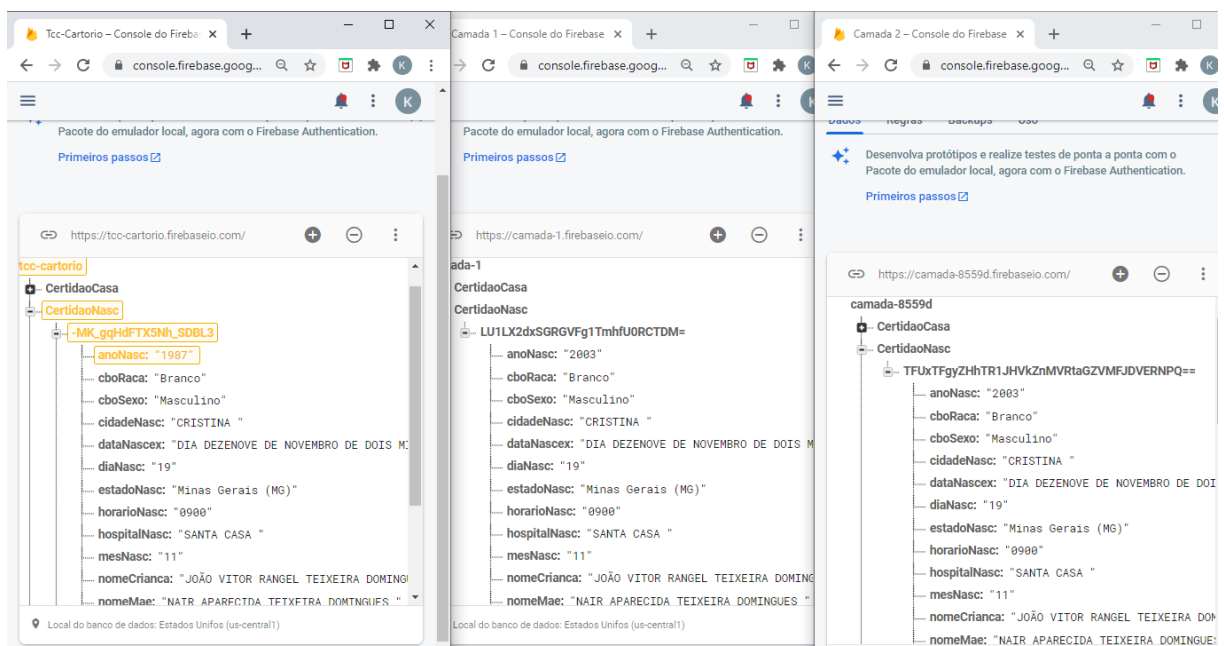
Figura 16 - Exemplo de discrepância entre os bancos de dados



Fonte: o autor (2020)

No momento em que há discrepância, o dado será voltado para a maioria, ou seja, se houver um banco com a letra “A” e dois com a letra “B”, o valor será mudado para “B”, totalmente livre de interferência humana. A figura a seguir exemplifica a situação, tendo em vista que após conferência a condição voltou à da figura 15.

Figura 17 - Verificação da veracidade do dado



Fonte: o autor (2020)

Com este estudo, foi constatado que a Blockchain pode ser usada para gerenciar cartórios cíveis virtualmente utilizando as linguagens de programação: *HTML*, *CSS* e *Javascript* e a plataforma *Firebase*. Na situação apresentada foram implantados 3 elos de bancos de dados garantindo uma maior segurança com confiabilidade e imutabilidade das informações.

O emprego das linguagens neste trabalho resulta aos participantes a agregação de experiência e conhecimentos extras com o uso da tecnologia de 5º geração.

4.2 Manual do usuário

Primeiramente deve-se abrir o site em seu navegador, na página “menu” (figura abaixo), há três botões que direcionam o usuário à certidão de nascimento, de casamento ou de óbito.

Figura 18 - Página "Menu"



Fonte: o autor (2020)

Se o usuário clicar no botão “Certidão de Nascimento” será carregada uma nova página onde encontram-se campos iniciais que condizem aos dados de nascimento (data por extenso, dia, mês, ano, horário, cidade, estado nascimento e hospital).

Figura 19 - Primeira parte da página “Certidão de Nascimento”



REPÚBLICA FEDERATIVA DO BRASIL
REGISTRO CIVIL DAS PESSOAS NATURAIS

Certidão de Nascimento

DADOS DO NASCIMENTO

DATA DE NASCIMENTO POR EXTENSO

DIA MÊS ANO HORÁRIO

Dia um de Janeiro de dois mil e vinte 01 01 2020 00:00

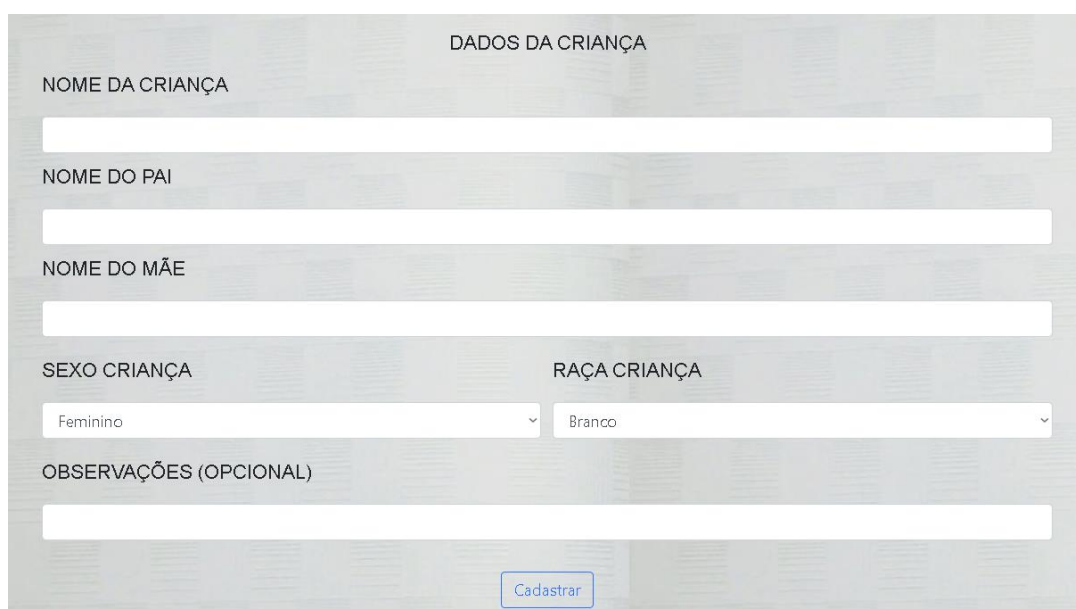
CIDADE NASCIMENTO ESTADO NASCIMENTO HOSPITAL NASCIMENTO

Acre (AC)

Fonte: o autor (2020)

Abaixo, insere-se os dados da criança (nomes da criança, do pai e da mãe, sexo, raça e observações. Para cadastrar os dados da criança no banco de dados basta clicar no botão “Cadastrar” ao final da página.

Figura 20 - Segunda parte da página “Certidão de Nascimento”



DADOS DA CRIANÇA

NOME DA CRIANÇA

NOME DO PAI

NOME DO MÃE

SEXO CRIANÇA RAÇA CRIANÇA

Feminino Branco

OBSERVAÇÕES (OPCIONAL)

Cadastrar

Fonte: o autor (2020)

Se o usuário clicar no botão para a certidão de casamento será carregada uma nova página. Encontra-se campos que condizem aos dados de casamento, inserindo: nome dos cônjuges, data da celebração numérica e por extenso, nome do presidente da celebração, estado, cidade, regime de bens e observações.

Figura 21 - Primeira parte da página “Certidão de Casamento”

REPUBLICA FEDERATIVA DO BRASIL
REGISTRO CIVIL

Certidão de Casamento

DADOS DOS NUBENTES

NOME DO PRIMEIRO CÔNJUGE

NOME DO SEGUNDO CÔNJUGE

DADOS DA CELEBRAÇÃO

DATA DA CELEBRAÇÃO DO CASAMENTO POR EXTENSO

DIA MÊS ANO

Fonte: o autor (2020)

Para cadastrar os dados do casamento no banco de dados basta clicar no botão “Cadastrar”.

Figura 22 - Segunda parte da página “Certidão de Casamento”

NOME DO PRESIDENTE DA CELEBRAÇÃO

ESTADO CASAMENTO CIDADE CASAMENTO

REGIME DE BENS DO CASAMENTO

OBSERVAÇÕES (OPCIONAL)

Fonte: o autor (2020).

Se o usuário clicar no botão para a certidão de óbito será carregada uma nova página com campos que condizem aos dados do finado, incluindo: nome, data de nascimento por extenso, dia, mês, ano, RG, sexo, raça, naturalidade, profissão, estado civil, nome cônjuge e filiação.

Figura 23 - Primeira parte da página “Certidão de Óbito”



Formulário de Certidão de Óbito, primeira parte. O formulário é dividido em seções para coletar dados pessoais do finado. No topo, há o brasão do Brasil e o texto "REPÚBLICA FEDERATIVA DO BRASIL" e "REGISTRO CIVIL". O título principal é "Certidão de óbito". Abaixo, o subtítulo "DADOS FINADO" precede os campos de entrada.

DADOS FINADO

NOME FINADO:

DATA DE NASCIMENTO POR EXTENSO: DIA: MÊS: ANO:

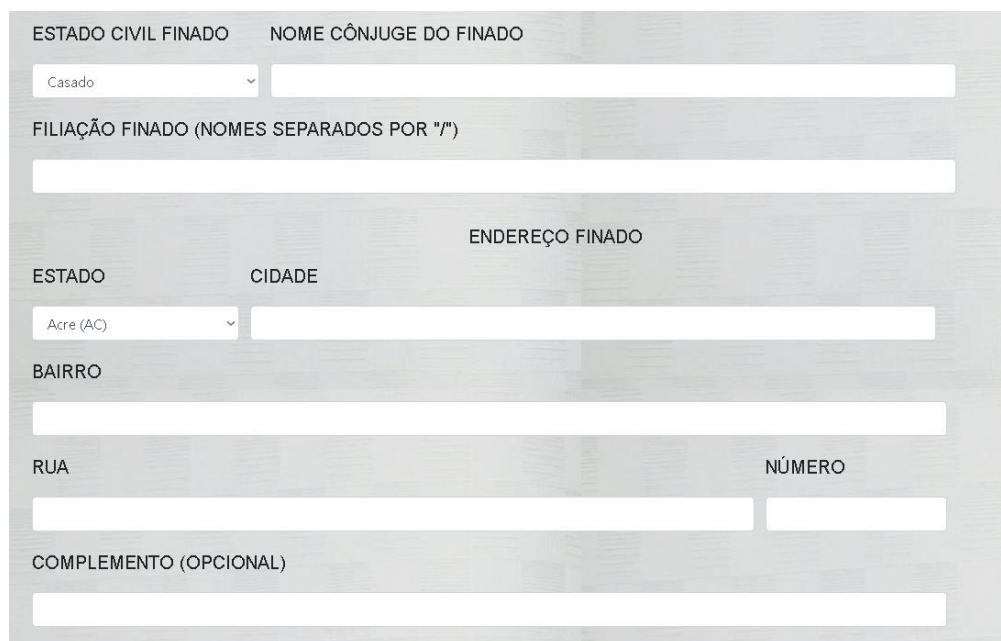
RG FINADO: SEXO FINADO: RAÇA FINADO:

NATURALIDADE FINADO: PROFISSÃO FINADO:

Fonte: o autor (2020)

Também se encontra o endereço (estado, cidade, bairro, rua, número e complemento).

Figura 24 - Segunda parte da página “Certidão de Óbito”



Formulário de Certidão de Óbito, segunda parte. Esta seção coleta informações sobre o estado civil, o cônjuge, a filiação e o endereço completo do finado.

ESTADO CIVIL FINADO: NOME CÔNJUGE DO FINADO:

FILIAÇÃO FINADO (NOMES SEPARADOS POR "/>)

ENDEREÇO FINADO

ESTADO: CIDADE:

BAIRRO:


RUA: NÚMERO:

COMPLEMENTO (OPCIONAL):

Fonte: o autor (2020)

E ao fim da página, informações do óbito (data por extenso, dia, mês, ano, horário, causa da morte e observações) e um botão para cadastrar os dados do finado no banco de dados.

Figura 25 - Final da página “Certidão de Óbito”



O formulário, intitulado "DADOS DO ÓBITO", contém os seguintes campos:

- DATA DO FALECIMENTO POR EXTENSO:** Um campo de texto com o valor "Dia um de Janeiro de dois mil e vinte".
- DIA:** Um campo de texto com o valor "01".
- MÊS:** Um campo de texto com o valor "01".
- ANO:** Um campo de texto com o valor "2020".
- HORÁRIO:** Um campo de texto com o valor "0000".
- CAUSA DA MORTE:** Um campo de texto vazio.
- OBSERVAÇÕES (OPCIONAL):** Um campo de texto vazio.
- Botão:** Um botão azul com o texto "Cadastrar".

Fonte: o autor (2020)

4.3 Discussão

O sistema é um estudo de viabilidade de *Blockchain* na sua forma mais simples. Nele, utilizou-se os conceitos básicos da metodologia, que são a criptografia *HASH*, os dados e uma *TIMESTAMP* (que é a data). No entanto, para um estudo mais aprofundado, deverá levar em conta o custo monetário desses servidores e em quantos cartórios ele deveria ser aplicado para que o valor ficasse atrativo à viabilidade de se atrelar uma criptomoeda e se haveria a opção de se atrelar a uma *Blockchain* pública. O tempo de cada transação deverá ser medido e analisado se vale ou não utilizar a sistemática.

No momento o sistema conta com uma criptografia de Base64, o que é suficiente para resolver o problema de *HASH*. Porém, como o próprio nome já diz, uma criptografia não pode variar de tamanho a cada novo bloco. Sendo assim, se esse sistema fosse implementado em um cartório, provavelmente a partir do quarto ou quinto bloco o índice se torna muito maior que o próprio dado, uma vez que cada novo índice é uma soma do anterior e sua viabilidade seria descartada, inabilitando a segurança.

Uma questão apontada pelo grupo, foi: qual a diferença entre multiplicação SGBDS (Sistema de Gerenciamento de Banco de Dados) e *Blockchain*. Após estudos foi concluído que multiplicação SGBDS somente replica os dados, caso haja um dado modificado, ela irá replicá-lo e não o corrigi-lo. Já a *Blockchain* valida os dados, assim se houver um dado modificado ela só replica após a correção (SINGARETTI, 2018).

5. CONCLUSÃO

Com base nos estudos relacionados ao desenvolvimento de um sistema *Blockchain* confiável e imutável, foi possível desenvolver um projeto que testa a segurança das informações pessoais, especialmente nos cartórios. O desenvolvimento deste *software* em Javascript e Firebase, depende de redes de bancos de dados e de processamento contínuo e isso pode ser um problema quando se trata de espaço em memória, pois o método utilizado para a formação do “nó principal” de cada item faz com que esse por sua vez cresça a cada camada adicionada, sendo possível que em algum momento falte espaço disponível na memória do banco de dados e também uma sobrecarga no processador que rege o sistema da *Blockchain*. Conclui-se que o método utilizado para a criação da *Blockchain* cumpriu com seu objetivo, porém, é sugerido o uso de *hashes* que não aumentem de tamanho, para diminuir o problema de alocação de memória.

Aclamada como a “quinta evolução” da computação, a tecnologia provoca questionamentos nos indivíduos, como: “os cartórios vão acabar?” e “a *Blockchain* vai substituir os cartórios?” e uma resposta possível pode ser assustadora: sim, o uso da estrutura pode sim acabar com o modo antigo de registro. Fazendo uma comparação com a indústria, vê-se que o princípio é quase o mesmo: trocar a mão de obra humana por máquinas, diminuindo o custo geral, considerando o salário pago aos funcionários suas possíveis falhas.

REFERÊNCIAS

- BASSOTTO, L. O que é Blockchain? Como funciona a tecnologia? 2018.
- BOOTSTRAP. Bootstrap Site. Disponível em: <<http://www.getbootstrap.com.br/>>. Acesso em: 04 out. 2020.
- BORGES, Bruno Rodrigues et al. Desenvolvimento de aplicação mobile utilizando metodologia ágil SCRUM. 2017.
- COSTA. M. História dos cartórios no Brasil. 2020.
- DE FREITAS, FREDERICO LUIZ GONÇALVES; BITTENCOURT, GUILHERME. Comunicação entre Agentes em Ambientes Distribuídos Abertos: o Modelo “peer-to-peer”. 2002.
- DEVMEDIA. O que é noSQL? 2020.
- FERNANDES, Henrique Marques. O que é Base64, para que serve e como funciona? 2020.
- FREEMAN, E.; ROBSON, E. Use A Cabeça-Html e CSS. Rio de Janeiro: Alta, 2014.
- GONÇALVES, Pedro Vilela Resende; CAMARGOS, Rafael Coutinho. BLOCKCHAIN, SMART CONTRACTS E ‘JUDGE AS A SERVICE’ NO DIREITO BRASILEIRO. GOVERNANÇA DAS REDES E O MARCO CIVIL DA INTERNET, p. 207, 2017.
- JOSEFSSON, Simon et al. The base16, base32, and base64 data encodings. RFC 4648, October, 2006.
- JUNIOR, Helvio. Introdução à Criptografia (Criptografia, hash, base64 encoding e certificação digital). Março, 2012.
- LAURENCE, Tiana. Blockchain para leigos. Rio de Janeiro: Alta Books, 2019.
- MENDES, Estarela Maia Bravo; DE SOUZA, Marcelo Agamenon Góis. A CORRUPÇÃO. ETIC-ENCONTRO DE INICIAÇÃO CIENTÍFICA-ISSN 21-76-8498, v. 4, n. 4, 2008.
- MICHAEL, Morrison. Use a cabeça! Javascript. 2008.
- SINGARETTI, PAULO. Replicação “Multi-SGBD” com SymmetricDS. 2018.