



UNIVERSIDADE FEDERAL DO TOCANTINS  
CAMPUS UNIVERSITÁRIO DE PALMAS  
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO  
MESTRADO EM MODELAGEM COMPUTACIONAL DE SISTEMAS

Flávio Fernandes de Melo

UM *BACK-END* PARA UM SISTEMA DE ACREDITAÇÃO EM SAÚDE  
prototipação e testes para a validação conceitual

PALMAS – TO

2021

FLÁVIO FERNANDES DE MELO

UM *BACK-END* PARA UM SISTEMA DE ACREDITAÇÃO EM SAÚDE

prototipação e testes para a validação conceitual

Dissertação apresentada à Universidade Federal do Tocantins – UFT, como requisito parcial para a obtenção do grau de Mestre em Modelagem Computacional de Sistemas

Orientador: Professor Doutor Patrick Letouze  
Moreira

PALMAS – TO

2021

## FICHA CATALOGRÁFICA

**Dados Internacionais de Catalogação na Publicação (CIP)**  
**Sistema de Bibliotecas da Universidade Federal do Tocantins**

---

M528b Melo, Flávio.

UM BACK-END PARA UM SISTEMA DE ACREDITAÇÃO EM SAÚDE: prototipação e testes para a validação conceitual. / Flávio Melo. – Palmas, TO, 2021.

138 f.

Dissertação (Mestrado Acadêmico) - Universidade Federal do Tocantins – Câmpus Universitário de Palmas - Curso de Pós-Graduação (Mestrado) em Modelagem Computacional de Sistemas, 2021.

Orientador: Patrick Letouze Moreira

1. Blockchain. 2. Back-end. 3. Smart contracts. 4. Acreditação. I. Título

**CDD 004**

---

TODOS OS DIREITOS RESERVADOS – A reprodução total ou parcial, de qualquer forma ou por qualquer meio deste documento é autorizado desde que citada a fonte. A violação dos direitos do autor (Lei nº 9.610/98) é crime estabelecido pelo artigo 184 do Código Penal.

**Elaborado pelo sistema de geração automática de ficha catalográfica da UFT com os dados fornecidos pelo(a) autor(a).**

FLÁVIO FERNANDES DE MELO

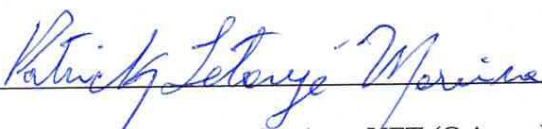
UM *BACK-END* PARA UM SISTEMA DE ACREDITAÇÃO EM SAÚDE  
prototipação e testes para a validação conceitual

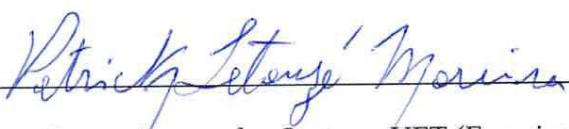
FOLHA DE APROVAÇÃO

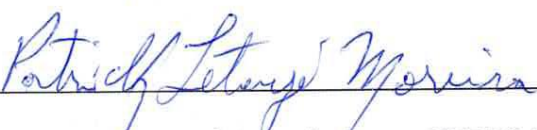
Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional de Sistemas como requisito para obtenção do Título de Mestre em Modelagem Computacional de Sistemas e aprovada em sua forma final pelo orientador e pela Banca examinadora.

Aprovada em: 26 / 08 / 2021

Banca Examinadora:

  
\_\_\_\_\_  
Professor Doutor Patrick Letouze Moreira – UFT (Orientador)

  
\_\_\_\_\_  
Professor Doutor George França dos Santos – UFT (Examinador Interno)

  
\_\_\_\_\_  
Professor Doutor Sergio Manuel Serra da Cruz – UFRRJ (Examinador Externo)

*Dedico esse trabalho a  
minha mãe Miriene, que  
sempre me incentivou  
a buscar meu melhor.*

## AGRADECIMENTOS

Primeiramente a Deus, que me concedeu saúde e sabedoria para conduzir os tantos afazeres. À minha família, em especial a minha mãe Miriene, que me apoiou nessa jornada. Ao orientador de projeto, Professor Dr. Patrick Letouze Moreira, pelos direcionamentos necessários para realizar esse trabalho. Aos professores da banca examinadora: Dr. George França dos Santos, e Dr. Sergio Manuel Serra da Cruz, por terem aceitado avaliar essa dissertação e pelas valiosas sugestões para enriquecer esse projeto. Aos amigos, meus sinceros agradecimentos pela ajuda e incentivo. E aos que direta ou indiretamente, deram sustentação para conclusão dessa etapa. Minha imensa gratidão!

## RESUMO

Este trabalho tem como propósito de desenvolver o *back-end* de um protótipo funcional do sistema de acreditação internacional em saúde (IAS) a fim de realizar sua validação conceitual. O entendimento deste trabalho para *back-end* envolve a própria rede *blockchain*, os contratos inteligentes e materiais de apoio para a utilização destes em um sistema web. Nesse sentido, é esperado uma ferramenta que auxilie na criação e instalação de uma rede *blockchain* privada, na compilação de *smart contracts* e na integração destes com um sistema web escrito em linguagem JAVA. Elaborou-se então, um *script* para automatizar a criação e configuração de uma rede *blockchain*, e outro para compilação de contratos que juntamente com roteiros de orientação compõem a ferramenta utilizada para implementar o protótipo funcional do IAS para realizar sua validação conceitual por meio de uma simulação de estudo de caso, com intuito de demonstrar como o sistema pode apoiar o processo de acreditação profissional, além de facilitar a busca de informações dos profissionais acreditados garantindo a qualquer interessado a possibilidade validá-las e afastar a possibilidade de falsificação.

**PALAVRAS-CHAVE:** *blockchain*, *back-end*, *smart contracts*, acreditação.

## ABSTRACT

This work aims to develop the back-end of a functional prototype of the international accreditation system for health professionals (IAS) in order to carry out its conceptual validation. The understanding of this work for back-end involves the blockchain network itself, the smart contracts and support materials for the use of these in a web system. In this sense, a tool is expected to assist in the creation and installation of a private blockchain network, in the compilation of smart contracts and in the integration of these with a web system written in JAVA language. He then elaborated a script to automate the creation and configuration of a blockchain network, and another to compile contracts that together with guidance scripts make up the tool used to implement the functional prototype of IAS to perform its conceptual validation through a simulation of case study, to demonstrate how the system can support the professional accreditation process, in addition to facilitating the search for information from accredited professionals.

**Key Words:** blockchain, back-end, smart contracts, accreditation.



## LISTA DE ILUSTRAÇÕES

### APÊNDICES

Apêndice 1- Relatório de avaliação das plataformas “ <i>blockchain</i> ” .....	99
Apêndice 2 - Artigo publicado no Educomp 2021 sobre os scripts de instalação da <i>blockchain</i> .....	106
Apêndice 3 – Roteiro para <i>deploy</i> e utilização de contratos inteligentes ..	117
Apêndice 4 – Cenários para simulação de uso .....	120
Apêndice 5 – Questionários aplicados no estudo de caso instrumental ....	122
Apêndice 6 – Código fonte do script de automação da rede <i>blockchain</i> ...	126
Apêndice 7 – Código fonte dos <i>smart contracts</i> e do protótipo do IAS ...	126
Apêndice 8 – Vídeos tutoriais dos <i>scripts</i> . .....	126
Apêndice 9 - Impact Analysis of sisu at the Federal University of Tocantins .....	127
Apêndice 10 – Parking space management using internet of things .....	134

## FIGURAS

Figura 1 - Representação do Front-End e Back-End.....	23
Figura 2 - Fluxograma do processo de acreditação.....	26
Figura 3 - Blocos encadeados.....	31
Figura 4 - Desenvolvimento do blockchain.....	34
Figura 5 - Definição da classe Person em schema.org.....	39
Figura 6 - O processo de desenvolvimento de protótipos.....	42
Figura 7 - Estrutura do IRPM.....	46
Figura 8 - Diagrama Aquisição Evolucionária.....	47
Figura 9 - Incorporação de EA ao IRPM.....	48
Figura 10 - Padrão de arquitetura MVC.....	49
Figura 11- Arquitetura MVC incorporado à Aquisição Evolucionária.....	50
Figura 12 - Padrão de arquitetura MVC incorporado ao EA-IRPM.....	51
Figura 13 - Fluxograma das etapas do MVC EA-IRPM para este projeto..	52
Figura 14 - Fluxo geral dos scripts para iniciar um nó.....	62
Figura 15 - Funções de cada nó.....	64
Figura 16 - Retorno das informações do log de uma transação.....	70
Figura 17 - Decodificador ABI online.....	71
Figura 18 - Primeira versão do diagrama de casos de uso de Pessoa.....	72
Figura 19 - Primeira versão do diagrama de casos de uso da Organização.....	72
Figura 20 - Primeira versão do diagrama de casos de uso do Certificado...	73
Figura 21 - Diagrama de caso de uso da pessoa.....	74
Figura 22 - Diagrama de caso de uso da organização.....	74
Figura 23 - Diagrama de caso de uso da certificação de acreditação.....	75
Figura 24 - Estrutura utilizada para desenvolvimento dos <i>smart contracts</i> .....	77
Figura 25 - Fábrica de contratos do tipo Certificado.....	78

Figura 26 - Tela do apresentando os dados da criação do contrato.....	79
Figura 27 - Transação de implantação enviada pela aplicação WEB.....	79
Figura 28 - Modelagem dos <i>smart contracts</i> em diagrama de classes. ....	82
Figura 29 - Pares conectados na rede <i>blockchain</i> . ....	84

## GRÁFICOS

Gráfico 1 - Gráfico de Likert dos questionários A e B.....	88
Gráfico 2 - Pesquisa de opinião sobre o IAS.....	90

## TABELAS

Tabela I - Mapeamento entre Bitcoin e IAS.....	29
Tabela II - Benchmark das plataformas <i>blockchain</i> .....	55
Tabela III - Máquinas utilizadas no teste do <i>script</i> de automação. ....	83
Tabela IV - Cenários propostos para teste de funcionalidade do protótipo. 86	
Tabela V - Tempo de espera para criar um <i>smart contract</i> . ....	91

## LISTA DE ABREVIATURAS E SIGLAS

ABI - Interface binária de Aplicação

ABNT - Associação Brasileira de Normas Técnicas

API - Interface de Programação de Aplicação

CFMV - Conselho Federal de Medicina Veterinária

Cgcre - Coordenação Geral de Acreditação

CONMETRO - Conselho Nacional de Metrologia, Normalização e Qualidade Industrial

EA - Aquisição Evolucionária

EBS - Conselho Europeu de Cirurgia

GPL - Licença Pública Geral

GPU - Unidade de Processamento Gráfico

HTTP - Protocolo de Transferência de Hipertexto

IAF - Fórum Internacional de Acreditação

IAS - Sistema de Acreditação Internacional

IDE - Ambiente de Desenvolvimento Integrado

IEC - Comissão Eletrotécnica Internacional

INMETRO - Instituto Nacional de Metrologia, Qualidade e Tecnologia

IRPM - Gerenciamento de Projetos de Pesquisa Interdisciplinar

ISO - Organização Internacional de Normalização

MVC - Modelo-Visão-Controle

NBR - Norma Brasileira Registrada

OA - Organismo de Acreditação

OAC - Organismo de Avaliação de Conformidade

OMS - Organização Mundial de Saúde

OMT - Técnica de Modelagem de Objetos

ONA - Organização Nacional de Acreditação

OOSE - Engenharia de Software Orientada a Objeto

P2P - Ponto a ponto

PLD - Dispositivo Lógico Programável

PoC - Prova de conceito

PoS - Prova de Participação

PoW - Prova de Trabalho

PPA - Arquivo de Pacote Pessoal

RA - Análise de Requisitos

RPC - Chamada Remota de Procedimento

SBAC - Sistema Brasileiro de Avaliação da Conformidade

SINMETRO - Sistema Nacional de Metrologia, Normalização e Qualidade Industrial

SPDX - Troca de Dados de Pacotes de Software

TI - Tecnologia da Informação

UEMS - União Europeia de Médicos Especialistas

UML - Linguagem de Modelagem Unificada

URL - Localizador Uniforme de Recursos

WSL - Subsistema do Windows para Linux

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>18</b>
	1.1 MOTIVAÇÃO .....	20
	1.2 JUSTIFICATIVA.....	20
	1.3 O PROBLEMA .....	20
	1.4 OBJETIVOS.....	20
	1.4.1 <i>Objetivo Geral</i> .....	20
	1.4.2 <i>Objetivos Específicos</i> .....	21
	1.5 ESTRUTURA DO TRABALHO.....	21
<b>2</b>	<b>FUNDAMENTOS .....</b>	<b>22</b>
	2.1 <i>FRONT-END E BACK-END</i> .....	22
	2.2 O SISTEMA DE ACREDITAÇÃO INTERNACIONAL EM SAÚDE .....	24
	2.2.1 <i>Utilização da acreditação</i> .....	24
	2.2.2 <i>Processo da acreditação</i> .....	26
	2.2.3 <i>Benefícios da acreditação</i> .....	27
	2.2.4 <i>Sistema de Acreditação Internacional IAS</i> .....	27
	2.3 <i>BLOCKCHAIN</i> .....	30
	2.3.1 <i>Estrutura</i> .....	30
	2.3.2 <i>Mecanismo de consenso</i> .....	32
	2.3.3 <i>Propriedades</i> .....	33
	2.3.4 <i>Evolução da blockchain</i> .....	33
	2.4 ETHEREUM .....	35
	2.5 SMART CONTRACTS .....	37
	2.6 METADADOS.....	38
	2.7 PROVA DE CONCEITO .....	40
	2.8 PROTOTIPAÇÃO .....	42
	2.9 ESTUDO DE CASO INSTRUMENTAL .....	44
<b>3</b>	<b>METODOLOGIA .....</b>	<b>45</b>
	3.1 EA-IRPM .....	45
	3.2 MVC EA-IRPM .....	49



		17
3.3	A ESTRATÉGIA DE DESENVOLVIMENTO DO SISTEMA .....	51
3.3.1	<i>Iniciação</i> .....	52
3.3.2	<i>Planejamento</i> .....	53
3.3.3	<i>Execução</i> .....	53
3.3.4	<i>Controle</i> .....	54
3.3.5	<i>Encerramento</i> .....	54
3.4	BACK-END.....	54
3.4.1	<i>Rede blockchain</i> .....	55
3.4.2	<i>Modelagem dos smart contracts</i> .....	57
3.4.3	<i>Integração com sistema web</i> .....	58
<b>4</b>	<b>RESULTADOS</b> .....	<b>60</b>
4.1	ANÁLISE DOS PRODUTOS E RESULTADOS.....	60
4.1.1	<i>Scripts automatizados para rede blockchain</i> .....	60
4.1.2	<i>Script para compilação dos smart contracts</i> .....	67
4.1.3	<i>Versão preliminar dos smart contracts</i> .....	71
4.1.4	<i>Versão final dos smart contracts</i> .....	80
4.2	TESTES DO BACK-END COM O PROTÓTIPO FUNCIONAL .....	82
4.2.1	<i>Teste em ambiente de homologação do script de automação</i> .....	83
4.2.2	<i>Teste automatizados dos smart contracts</i> .....	85
4.2.3	<i>Estudo de caso instrumental</i> .....	86
4.3	CHAMADAS ASSÍNCRONAS NAS CRIAÇÕES DOS CONTRATOS .....	91
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>92</b>
5.1	TRABALHOS FUTUROS .....	94
	REFERÊNCIAS .....	95
	APÊNDICES .....	99

# 1 INTRODUÇÃO

A preocupação com a qualidade não é uma questão que surgiu recentemente, tanto na venda de produtos quanto na prestação de serviços é notável um crescimento na busca pela qualidade e na exigência dos clientes que buscam uma boa experiência pelo que estão adquirindo/utilizando.

Com a evolução da comunicação, principalmente com a crescente utilização da internet e redes sociais, expressar um descontentamento com estabelecimentos que lhe propõe uma experiência negativa torna-se uma tarefa muito fácil e muitas vezes incentivada pelo sentimento de revolta e traição da confiança conferida pelo consumidor é um reflexo natural escrever reclamações e críticas sobre a situação a que foram submetidos, a fim de alertar a outros consumidores do estabelecimento.

Nesse contexto, segundo Maximiano 2010, para as instituições, custos relacionados com a má qualidade, ou seja, despesas com retrabalho e dispêndios com processo ineficiente, não podem mais ser tolerados, já que comprometem a imagem da organização perante a sociedade, podendo levar a perda de clientes e de mercado.

O cliente ao contratar um serviço, gosta de ser tratado com atenção e se sentir seguro de que receberá um serviço de qualidade, cabe a instituição repassar essa segurança, seja por um bom atendimento que será transmitido em forma de indicações de seus próprios clientes altamente satisfeitos ou através de certificados que comprovem e transmitam essa segurança da qualidade de seus serviços àqueles que ainda não conhecem seu trabalho.

Já os hospitais lidam diariamente com atendimentos emergenciais e devem estar preparados para tomar decisões rápidas sobre materiais, equipamentos, espaços físicos e profissionais, sem qualquer plano de contingência específico para tal condição. Para garantir uma resposta eficiente pode-se utilizar processos de certificação para que alguns padrões de alta qualidade sejam seguidos.

O surgimento do surto mundial da doença COVID-19 causada pelo coronavírus (SARS-COV-2) fez com que vários planos de controle devessem ser rapidamente criados para lidar com a situação de uma nova pandemia que foi declarada em março de 2020 pela Organização Mundial de Saúde (OMS).

Considerando a necessidade de garantir qualidade dos serviços a fim de passar uma maior segurança aos pacientes pode-se destacar o programa de acreditação, já bastante utilizado na área da saúde, mas que é utilizado também em outras áreas. Um profissional acreditado para lidar com o acompanhamento de pacientes infectados pelo COVID-19, traria uma maior confiança aos pacientes, que saberiam que suas decisões seguiram recomendações reconhecidas como um padrão de qualidade.

Considerando a necessidade de desenvolvimento de estratégia de incremento da qualidade dos serviços podemos destacar o programa de acreditação, muito presente na área da saúde, mas que é utilizado em outras áreas.

Segundo Costa 2006, acreditação é o reconhecimento formal por um organismo de acreditação, que um organismo de certificação ou inspeção, atende a requisitos previamente definidos e demonstra ser competente para realizar suas atividades com confiança. Logo, ser um profissional acreditado, serve como garantia e o qualifica de suas competências relacionadas à área em que foi acreditado.

Para Casanova (2009), acreditação é o processo pelo qual um profissional ou um especialista em treinamento alcança ou satisfaz um nível de competência e qualidade. Dessa forma, para se tornar um profissional acreditado, este deve cumprir uma série de requisitos que deverão ser comprovados pelo reconhecimento de uma instituição em que uma pessoa concluiu um processo de credenciamento.

Ainda em seu artigo Casanova (2019), explica que no contexto da acreditação de profissionais competentes para transplantes de órgãos foi criada em 2007 a Divisão de transplantes, fruto de dois anos de trabalho do Grupo de Trabalho de Transplante da Seção de Cirurgia da União Europeia de Médicos Especialistas (UEMS) e do Conselho Europeu de Cirurgia (EBS). O principal objetivo da Divisão é garantir o melhor padrão de atendimento no transplante de órgãos na Europa, garantindo que o treinamento em cirurgia de transplante seja mantido no mais alto nível.

Dada essa relação de confiança entre as partes, profissional a ser acreditado e instituição acreditadora e o cenário proposto por Souza Júnior et al. (2019), uma aplicação web com estrutura de rede social para acreditação profissional e consulta sobre as certificações pelos pacientes, com alcance mundial criada com base na tecnologia *blockchain* nomeado como *International Accreditation System (IAS)*.

A tecnologia *blockchain*, proporciona um ambiente para interações confiáveis e a formação de uma rede descentralizada, aspectos importantes para um sistema deste serviço, visto que poderia ser utilizado por qualquer instituição que desejasse e fosse interessante para o funcionamento do sistema.

## **1.1 MOTIVAÇÃO**

Para auxiliar na tarefa de enfrentar as ameaças a saúde com serviços de qualidade, este trabalho pretende criar o *back-end* para uma aplicação web a ser integrada a um sistema com base em uma rede social, com o intuito de viabilizar o processo de acreditação de forma segura e rastreável através da tecnologia *blockchain*.

## **1.2 JUSTIFICATIVA**

Dadas as ameaças a saúde como a pandemia gerada pelo COVID-19, torna-se necessário que profissionais e instituições de saúde tenham o devido treinamento e compartilhem informações e de disseminação de métodos confiáveis. A acreditação, pode promover a confiabilidade nos profissionais e instituições que estão preparados para enfrentar essa e outras ameaças, além disso, facilitando e aumentando a rede de profissionais acreditados que possam atuar em tarefas fora da região em que é reconhecido.

## **1.3 O PROBLEMA**

Nesse contexto, este projeto busca responder ao seguinte problema de pesquisa: a criação do “*back-end*” do Sistema de Acreditação em Saúde, baseada na tecnologia *blockchain*, pode trazer maior segurança aos pacientes que esperam ser atendidos por profissionais que tenham conhecimento e habilidades necessárias para prestar um serviço com qualidade?

## **1.4 OBJETIVOS**

Na sequência são apresentados os objetivos gerais e específicos.

### **1.4.1 Objetivo Geral**

Desenvolver o "*back-end*" de um protótipo funcional do Sistema de Acreditação Internacional em Saúde para testes de validação conceitual.

### 1.4.2 Objetivos Específicos

Nesta subseção são enumerados os objetivos específicos:

1. Implementar uma versão básica de uma rede *blockchain* para avaliar o uso da plataforma.
2. Elaborar um roteiro de instalação e configuração para referências futuras e de base para o desenvolvimento do Sistema de Acreditação em Saúde
3. Modelar e desenvolver os *smart contracts* na linguagem nativa da rede *blockchain* de acordo com base nos metadados básicos necessários para os processos de certificação e acreditação do Sistema de Acreditação em Saúde.
4. Implementar os *smart contracts* no protótipo funcional do Sistema de Acreditação em Saúde.
5. Testar o protótipo funcional com um estudo de caso instrumental.

## 1.5 ESTRUTURA DO TRABALHO

A pesquisa compõe-se de seis capítulos distribuídos da seguinte forma:

- No capítulo I (INTRODUÇÃO) serão abordados: a introdução, a justificativa, a problematização, a delimitação e os objetivos.
- O capítulo II (FUNDAMENTOS) é iniciado com a apresentação dos conceitos de *front-end* e *back-end* seguido de uma exposição sobre a prática da acreditação profissional no Brasil e no Mundo, apontando sua caracterização e conceito, processos e benefícios tanto na perspectiva do profissional quanto na da sociedade. Se sequência, a tecnologia *blockchain*, bem como suas características, estrutura e funcionamentos; as características sobre a plataforma escolhida para desenvolvimento do projeto e seus *smart contracts*, os quais são a base deste estudo. Finalizando com conceitos necessários para o desenvolvimento de um sistema web.
- O capítulo III (METODOLOGIA) descreve a metodologia de pesquisa utilizada (MVC EA-IRPM), assim como uma descrição das fases e etapas executadas no decorrer do trabalho.
- O capítulo IV (RESULTADOS) apresenta resultados obtidos, destacando os passos necessários para utilização dos produtos criados.
- Finalmente o capítulo V (CONCLUSÃO), apresentando as considerações finais e conclusões obtidas no projeto.

## 2 FUNDAMENTOS

Este capítulo tem como objetivo apresentar a revisão bibliográfica pertinente ao tema de pesquisa abordado. A expectativa é que esta revisão contribua com a geração de conhecimento para o planejamento e execução do projeto de pesquisa. Serão abordados os campos de pesquisa: tecnologia *blockchain*, acreditação profissional, desenvolvimento de sistemas web e os conceitos necessários para realização deste projeto.

### 2.1 FRONT-END E BACK-END

Amaral e Neris (2016), relatam que na computação, *front-end* e *back-end* são termos generalizados que se referem às etapas inicial e final de um processo de software. O *front-end* fica responsável pela coleta e adequação das entradas fornecidas pelo usuário a fim de repassá-las de uma forma adequada para que o *back-end* possa utilizar. Em contrapartida o *back-end* fica responsável por validar e construir os dados vindos do *front-end* à medida que executa as regras de negócio da aplicação.

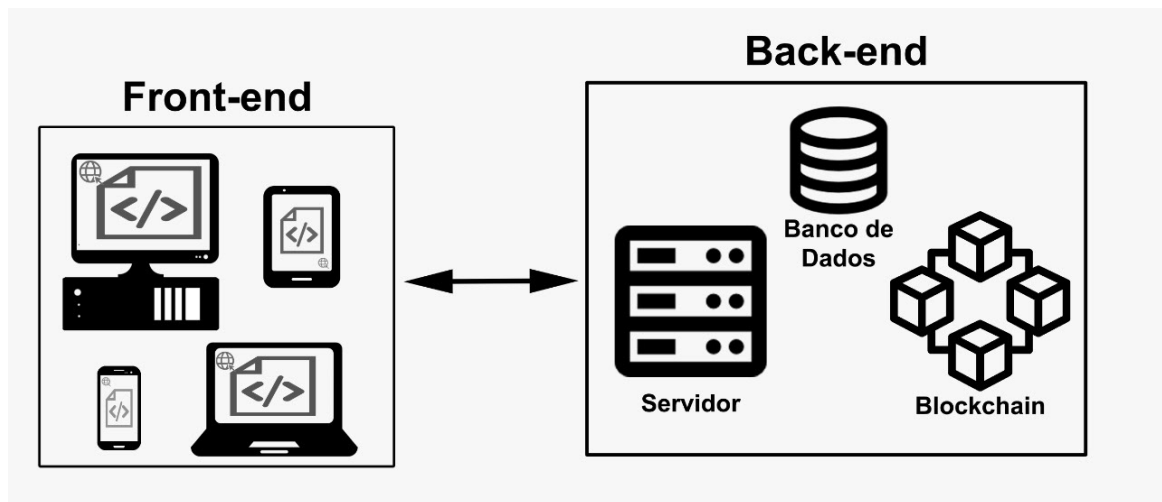
Para Andrade (2018), o *front-end* de uma aplicação é todo o código responsável pela apresentação do software (*client-side*). Em se tratando de aplicações web, é exatamente o código do sistema que roda no navegador. Citrus7 (2017), define *front-end* como a primeira camada na qual o usuário se depara ao se acessar um site, uma intranet ou um sistema web. Em seu trabalho, Filipova e Vilão (2018), argumentam que, ainda que seja comum se referir a um aplicativo ou página da web rodando em um navegador, qualquer aplicação que exponha uma interface gráfica ou até mesmo interface de linha de comando pode ser considerada *front-end*. Defendendo ainda que quaisquer softwares os quais sejam executados do lado do cliente são aplicações *front-end*, incluindo assim além dos aplicativos da web, aplicativos móveis e aplicativos para TV como exemplos *front-end*.

O *back-end*, por sua vez, é definido por Andrade (2018) como a parte de uma aplicação que roda no servidor (*server-side*), sendo o que determina e garante todas as regras de negócio, acessa o banco e define a segurança e escalabilidade de uma aplicação. Para Filipova e Vilão (2018) o *back-end* é a parte do sistema que é responsável por receber e tratar as solicitações dos usuários, disponibilizando todas as operações necessárias para os aplicativos clientes, o *back-end* geralmente é executado em servidores dedicados ou hospedados em serviços de nuvem.

Stewart (2020), afirma que o desenvolvimento *back-end* refere-se ao lado do servidor no desenvolvimento de uma aplicação, focando principalmente em como uma aplicação web funciona. Este tipo de desenvolvimento usualmente consiste em três partes: um servidor, uma aplicação e um banco de dados.

No presente estudo adiciona-se a este conceito comum mais uma parte ao *back-end*, a rede *blockchain*, a qual pode ser assim considerada data a finalidade de tratar as solicitações dos usuários sendo responsável por realizar o armazenamento e recuperação de dados. Dessa forma, a estrutura do projeto, em termos de *front-end* e *back-end*, é apresentada na Figura 1.

Figura 1 - Representação do Front-End e Back-End.



Fonte: Elaborado pelo autor (2021).

No contexto da *blockchain* este projeto preocupa-se em desenvolver um método replicável para a instalação e configuração de uma rede *blockchain* privada, a modelagem dos *smart contracts*, para apoiar a validação e distribuição dos dados referentes aos certificados de acreditação gerados pelo sistema, de modo que sejam definidos dados chaves para identificação das organizações e profissionais seguindo para este fim, os modelos do *schema.org*. Adicionalmente, a modelagem deve garantir que dados extras, relevantes as peculiaridades de cada instituição, possam ser registrados caso sejam necessários sem que estes interfiram no funcionamento padrão do sistema.

Dessa forma, este projeto que consiste no desenvolvimento do “*back-end*” para o Sistema de Acreditação Internacional, podemos considerar como seus componentes:

1. Rede *blockchain* privada.
2. Configurações da rede e servidor.
3. Modelagem dos *smart contracts*

4. Modelagem das entidades do sistema em conformidade com os modelos do Schema.org.

## **2.2 O SISTEMA DE ACREDITAÇÃO INTERNACIONAL EM SAÚDE**

Oficialmente o termo acreditação foi introduzido no Sistema Brasileiro de Avaliação da Conformidade (SBAC) através da resolução nº 5, de 10 de dezembro de 2003, do Conselho Nacional de Metrologia, Normalização e Qualidade Industrial – CONMETRO, que dispôs sobre a alteração do termo “Credenciamento” para “Acreditação” para expressar reconhecimento de competência de organismos de avaliação da conformidade no âmbito do Sistema Nacional de Metrologia, Normalização e Qualidade Industrial (SINMETRO).

Para a língua portuguesa a palavra acreditação tem origem etimológica no verbo “acreditar” com o sufixo “ação” sendo definida pelo dicionário Priberam como: reconhecimento oficial de uma pessoa ou entidade para o desempenho ou realização de algo; autorização para o exercício de uma atividade (ACREDITAÇÃO, 2019).

De acordo com a ABNT NBR ISO/IEC 17011:2019, a acreditação é o processo em que um organismo de avaliação da conformidade atesta que uma terceira-parte demonstra competência para realizar tarefas específicas de avaliação da conformidade. Em resumo, o objetivo desta norma é especificar os requisitos gerais para organismos de acreditação (OA) que avaliam e concedem acreditação para organismos de avaliação de conformidade (OAC), a fim de transmitir confiança nos produtos, serviços e processos por ela acreditados.

Para a instituição, o processo de acreditação busca melhorar o gerenciamento da instituição e garantir uma assistência de qualidade, com segurança e eficiência. Para o profissional, segundo Casanova (2019), o processo de acreditação é aquele pelo qual um profissional ou um especialista em treinamento alcança ou satisfaz um nível de competência e qualidade.

### **2.2.1 Utilização da acreditação**

A acreditação como um mecanismo de avaliação da melhoria da qualidade pode ser utilizada por qualquer área do conhecimento e em qualquer profissão, no entanto, a principal área que a utiliza é a da saúde.



Através da aplicação de normas nacionais e internacionais, o governo, compradores e consumidores podem ter confiança nos resultados de calibrações e ensaios, nos relatórios de inspeção e nas certificações fornecidas pelos organismos de acreditação.

Os organismos de acreditação estão estabelecidos em muitos países com o objetivo principal de assegurar que os organismos de avaliação de conformidade estejam sujeitos à supervisão por um organismo competente. Para reforçar a aceitação de seus certificados de acreditação assinam acordos entre seus pares além das fronteiras nacionais, criando, assim, uma estrutura para apoiar o comércio internacional por meio da remoção de barreiras técnicas.

Neste sentido, temos a figura da *International Accreditation Forum (IAF)*, uma associação mundial de organismos de acreditação e avaliação da conformidade, outros organismos que atuam nas áreas de sistemas de gestão, produtos, serviços, pessoas e outros programas similares de avaliação de conformidade. Sua principal função é desenvolver um único programa mundial de avaliação de conformidade, capaz garantir aos utilizadores a competência e imparcialidade do organismo acreditado.

Dentre os organismos de acreditação podemos destacar a *International Accreditation Service, Inc (IAS)*, com sede nos Estados Unidos da América, que credencia uma ampla gama de empresas e organizações, incluindo entidades governamentais, universidades, empresas comerciais e associações profissionais. Pioneira na profissão de credenciamento, o IAS desenvolve ativamente novos programas de credenciamento para organizações que buscam demonstrar o mais alto nível de competência e serviço em seus respectivos setores.

No Brasil temos a Coordenação Geral de Acreditação (Cgcre) é o organismo de acreditação brasileiro de organismos de avaliação da conformidade e órgão oficial de monitoramento de conformidade segundo os princípios das Boas Práticas de Laboratório, como estabelecido na Lei 12.545/2011 (BRASIL, 2011) e no Decreto 6.275/2007 (BRASIL, 2007).

Existem também, no Brasil, instituições que realizam a acreditação de organismos de avaliação da conformidade em suas áreas específicas, dentre elas podemos citar:

- Organização Nacional de Acreditação (ONA), que desde 1999, trabalha para que as instituições de saúde no Brasil adotem práticas de gestão e assistenciais que levem à melhoria do cuidado para o paciente;
- Sistema Nacional de Acreditação de Cursos de Graduação em Medicina Veterinária representada pelo Conselho Federal de Medicina Veterinária (CFMV) se certificam de que os cursos seguem padrões adequados de ensino;

- Associação Brasileira de Psicoterapia e Medicina Comportamental (ABPMC) que abriga profissionais de diversas áreas, envolvidos com ensino, pesquisa e prestação de serviço em Análise do comportamento em vários contextos, que certifica como acreditados os profissionais com qualificação de nível superior que trabalhem com o conhecimento científico e filosófico da Análise do Comportamento e do Behaviorismo Radical.

### 2.2.2 Processo da acreditação

O processo de acreditação pode variar para cada instituto que realiza um diagnóstico organizacional verificando uma série de pontos, a maioria deles em comum, porém cada uma com seu método e peculiaridades próprias.

De acordo com Piratelli-Filho (2011), para o INMETRO, como o processo de acreditação é de natureza voluntária, este tem início com a solicitação formal da acreditação, seguida de uma análise crítica da solicitação e indicação da equipe de avaliação, visita de pré-avaliação, análise da documentação, comparações Inter laboratoriais, avaliação inicial, decisão sobre a acreditação e manutenção da acreditação realizada a cada dois anos. O fluxograma deste processo pode ser observado na Figura 2.

Figura 2 - Fluxograma do processo de acreditação.



Fonte: Adaptado de Piratelli-Filho (2011).

### 2.2.3 Benefícios da acreditação

Como foi definido anteriormente a acreditação tem como objetivo transmitir confiança e segurança nos produtos, serviços e processos por ela acreditados, trazendo assim uma série de benefícios tanto para o acreditado quanto para os usuários dos produtos/processos/serviços.

Dessa forma a acreditação traz consigo um atestado de qualidade e pelo fato de que este é um processo que deve ser renovado com uma certa periodicidade, a busca pela qualidade também é dada de forma contínua acarretando outros benefícios mencionados por Mamédio (2014), como por exemplo:

1. Vantagem competitiva;
2. Reconhecimento de seguradoras, associações, colaboradores e outros financiadores, gerando a confiança da comunidade;
3. Recrutamento de profissionais, com reestruturação e/ou implantação de um processo de educação e qualificação profissional;
4. Fortalecimento do trabalho em equipe;
5. Esforços rumo à melhoria contínua;
6. Gerenciamento do risco;
7. Clara definição da missão institucional e perfil assistencial, abrangendo todos os departamentos e serviços da instituição (próprios, terceirizados, unidades em diferentes locais), estabelecimento de uma estrutura e sistema de gerenciamento da qualidade e alinhado às estratégias da direção e da instituição.

### 2.2.4 Sistema de Acreditação Internacional IAS

Souza Junior et al. (2019) através de um artigo publicado no "*International Journal of Information and Education Technology*" no ano de 2019, descreve um sistema web com uma estrutura de rede social que pretende prestar um serviço a profissionais e pacientes de saúde. Este sistema baseado na tecnologia *blockchain* atua no processo de acreditação de profissionais de saúde com uma abrangência internacional, sendo nomeado assim como *International Accreditation System (IAS)*.

O primeiro objetivo proposto pelo IAS é fornecer uma plataforma baseada na Web para apoiar o processo de acreditação e certificação é baseado na premissa de que “a globalização diminuiu as fronteiras da prática profissional” visando suportar esses processos em nível internacional de forma que seja um sistema é uma organização sem fins lucrativos,

autônoma e descentralizada, criada e gerenciada por um coletivo de sociedades profissionais de saúde, de forma a manter a autonomia e independência e, portanto, é uma nova aplicação para *blockchain*.

A seguir está descrito o funcionamento básico do IAS segundo Souza Junior et al. (2019):

1. **Uma nova sociedade profissional solicitaria fazer parte do IAS:** com sua aceitação, essa sociedade definiria seus processos de acreditação e certificação dentro do sistema.
2. **Todos os membros desta nova sociedade seriam membros do IAS:** um membro pode ser associado a qualquer número de sociedades, e profissionais e instituições podem ser membros.
  - a. As instituições podem solicitar a certificação e conceder acreditação a membros profissionais.
  - b. Os profissionais podem solicitar a acreditação.

Este seria o funcionamento básico e há ainda algumas perspectivas que podem ser incorporadas ao sistema IAS com base na tecnologia *blockchain*, como por exemplo:

1. Além de compartilhar processos de acreditação, as sociedades poderiam compartilhar conhecimento;
2. Possibilidade de os pacientes poderem ter acesso ao histórico de acreditação dos profissionais de saúde e ao histórico de certificações das instituições;
3. Implantação de um processo de acreditação pessoal entre pares, ou seja, os profissionais de saúde poderiam acreditar nas interações da prática clínica profissional.

No entanto, para implantação destas perspectivas é necessária uma avaliação mais aprofundada dessas possibilidades e deve ser realizada pelo coletivo de sociedades.

O IAS deve ser composto não só por sociedades e profissionais, mas também composto por divisões dentro das sociedades. São estas divisões as responsáveis pelo processo de acreditação, definindo as regras para certificação, como critérios de elegibilidade e exame. Assim, no que diz respeito aos elementos de rede e suas conexões, as sociedades não precisam estar conectadas a todas as sociedades. Os profissionais podem estar conectados a mais de uma sociedade, mas a pelo menos a uma. E uma divisão pertence a uma sociedade.

Não descartando a possibilidade das perspectivas anteriormente mencionadas possam ser adicionadas ao IAS, temos alguns novos conceitos que podem ou não ser aplicados: um profissional pode estar conectado a outro profissional e as divisões podem estar conectadas a outras divisões e sociedades. Isso significa que os profissionais podem acreditar outros profissionais, ou seja, digamos que dois profissionais tenham trabalhado juntos, eles podem

registrar esse fato e isso funcionaria como uma acreditação pessoal. E para as divisões, a possibilidade de poderem trabalhar juntas para uma acreditação combinada.

Levando em consideração esses cenários, o RA geral para o sistema IAS pode ser descrito da seguinte forma:

1. Organização autônoma descentralizada.
2. Confiabilidade.
3. Elementos de rede: sociedades, divisões e profissionais.
4. Possibilidade de credenciamento entre qualquer nó da rede.

Quanto ao RA específico para o núcleo do sistema, por razões de simplificação, foi considerado apenas o funcionamento básico sem as perspectivas adicionais, é a própria tecnologia *blockchain*, como implementada pelo Bitcoin. Em outras palavras, um mapeamento entre aplicativos ou serviços, como o RA Específico, é apresentado na Tabela I.

Tabela I - Mapeamento entre Bitcoin e IAS

<b>Bitcoin</b>	<b>Sistema Internacional de Acreditação</b>
<b>Financiadores</b>	Profissionais e divisões
<b>Carteira</b>	Registro de acreditação
<b>Mineradora</b>	Sociedade profissional
<b>Transação</b>	Transação
<b>Bloco</b>	Dados de acreditação

Fonte: Adaptador de Souza Junior et al. (2019).

Outro aspecto importante descrito na proposta do IAS, é referência ao sistema web desenvolvido por Letouze et al. (2017), um sistema chamado RGM - Reporting Guidelines in Medicine, em inglês, este é um sistema web para a criação, desenvolvimento e gerenciamento de diretrizes em saúde. O gerenciamento das diretrizes desenvolvidas e registradas no sistema, permite que as interações sejam salvas e possibilita rastrear seu histórico. E como o sistema permite que a definição dos parâmetros de medição seja mostrada graficamente, é possível avaliar o desenvolvimento das diretrizes.

Neste contexto, o RGM poderia ser a base para o desenvolvimento do IAS, por meio de uma estratégia de aquisição evolutiva, devido ao fato deste sistema ter implementado dentro de si a estratégia EA-IRPM de Letouze (2011b), que é adequada para integrar a tecnologia *blockchain* com contratos inteligentes, para fornecer flexibilidade e configurabilidade aos processos de acreditação das sociedades. Possui uma estrutura de rede social hierárquica, contemplando instituições, profissionais e, se desejado, pacientes. Fornece

uma arquitetura centrada no usuário com as funções gerais de um sistema da web, que satisfaz o RA Geral para o IAS.

## 2.3 BLOCKCHAIN

A tecnologia de *blockchain* foi criada em 2008, primeiramente descrita por um artigo publicado por Satoshi Nakamoto, que descreve o funcionamento de uma moeda inteiramente digital, que poderia circular sem a necessidade de uma autoridade central para validar as transações.

Segundo Nakamoto (2008), a tecnologia *blockchain* funciona como um tipo de livro razão distribuído com recurso de imutabilidade entre nós em uma rede *peer-to-peer* (uma rede em que os nós agem como clientes e servidores para os outros nós da rede) baseado em um protocolo de consenso, no qual cada nó pode manter a mesma razão sem uma autoridade centralizada com *hashes* (“endereços” que mapeiam dados grandes e de tamanho variável para pequenos dados de tamanho fixo) criptográficos e assinaturas digitais garantindo a integridade das transações em cada bloco.

Para Preukschat et al. (2017), podemos considerar a rede *blockchain* composta pelos seguintes os elementos:

- **Nó:** equipamentos de informática pertencentes a uma rede *peer-to-peer* responsável pelo armazenamento e distribuição das transações realizadas em tempo real.
- **Protocolo de consenso:** software de comunicação entre os nós que definem um conjunto de regras comuns para alcançar uma perfeita operabilidade.
- **Rede *peer-to-peer*:** comunicação na rede *blockchain* sempre ocorre diretamente entre pares.
- **Sistema descentralizado:** a rede *blockchain* não é controlada por nem um ponto central, todos os computadores conectados são todos iguais uns aos outros, ou seja, não há hierarquia entre os nós que trabalham em comum acordo como um único computador.

### 2.3.1 Estrutura

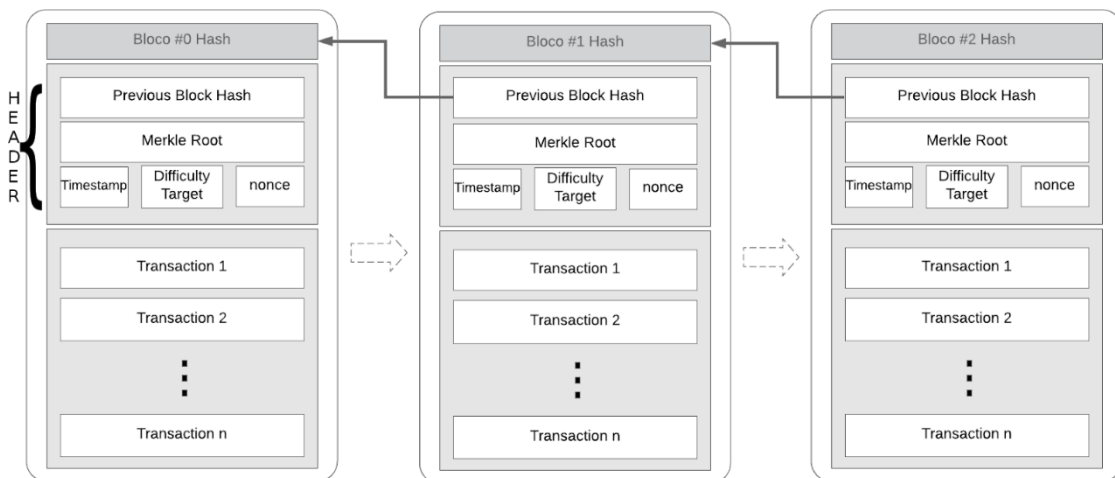
Quanto a estrutura do *blockchain*, esta é construída por blocos ligados por uma lista encadeada de forma que cada bloco referência o seu antecessor, garantindo assim que para a

modificação de informações gravadas em cada bloco exija um grande poder computacional, tornando essa ação computacionalmente impraticável.

Antonopoulos (2014) descreve um bloco sendo composto por um identificador (*block hash*), definido pela dupla aplicação do algoritmo SHA-256 em seu cabeçalho, o *block hash* do bloco anterior, o conjunto de todas as transações, juntamente com um conjunto de informações que compõem seu cabeçalho.

A estrutura do cabeçalho pode ser dividida em três conjuntos de dados de acordo com o seu propósito. O primeiro chamado de *Previous Block Hash*, composto com o *hash* do bloco anterior, garante a conexão entre todos os blocos da *blockchain*. O segundo é campo *Merkle Root*, usado para resumir de maneira eficiente o conjunto de transações do bloco. Por fim, o conjunto dos campos *timestamp*, *difficulty target*, e *nonce* são referentes ao processo de mineração, representando respectivamente, hora aproximada da criação do bloco, dificuldade alvo do algoritmo utilizada no bloco e o contador utilizado pelo algoritmo.

Figura 3 - Blocos encadeados.



Fonte: Adaptado de Antonopoulos (2014).

A Figura 3 apresenta um exemplo da estrutura dos três conjuntos de dados do bloco e do encadeamento entre eles, sendo comum para a identificação do bloco, além do *hash* duplo criado pela criptografia, o número da posição em que ele se encontra na *blockchain*, no entanto, a posição não pode ser admitida com identificador único, visto que mesmo sendo invariável, a posição pode ser disputada simultaneamente por dois ou mais blocos, durante a inserção.

### 2.3.2 Mecanismo de consenso

Um mecanismo de consenso é um algoritmo que serve para criar um bloco em um ambiente descentralizado de forma consensual entre os nós da rede P2P. (ALIAGA e HENRIQUES, 2017, p. 2). São responsáveis por manter a integridade e segurança das redes *blockchain*. O primeiro algoritmo a ser criado foi o *Proof of Work* (PoW), desenvolvido e descrito por Nakamoto (2008) este foi implementado no Bitcoin como forma de evitar as chamadas falhas bizantinas (*byzantine faults*), que ocorrem, especificamente de sistemas de computação distribuída, quando é uma condição de um ou mais componentes falharem e não há informações precisas sobre a falha de um componente ou se as informações do sistema estão corretas dificultando para os outros componentes tomar a decisão de declará-lo falido e excluí-lo da rede. Para isso, é necessário chegar a um consenso sobre qual componente falhou em primeiro lugar.

Como definido por Bashir (2017), o mecanismo de consenso é como um conjunto de passos que são dados pelos nós que compõem a rede para entrar em consenso a respeito de um valor. Para as redes *blockchain* que são sistemas distribuídos e não dependem de uma autoridade central, a validação das transações só deve acontecer mediante a aprovação dos computadores participantes da rede. Fica a cargo dos protocolos de consenso assegurar que as regras da rede estão sendo seguidas e que todas as transações ocorrem de forma confiável, além de garantir que o gasto pela transação ocorra apenas uma vez.

Introduzido por Nakamoto (2008) em seu artigo, o algoritmo PoW utilizado no Bitcoin é baseado em *hash* do cabeçalho do bloco, com a criptografia SHA-256, cujo valor deve ser encontrado de tal forma a seguir os parâmetros definidos pela dificuldade indicada na rede. Assim, quanto maior o poder computacional, mais rápido será a resolução dos problemas e conseqüentemente a construção dos blocos no *blockchain*, portanto, devido a essa demanda de computadores superpotentes, o PoW acaba privilegiando os nós que possuem mais recursos financeiros disponíveis para investir em hardware dedicado (*asics - application-specific integrated circuit*), que tem desempenho muito melhor que processadores de propósito geral ou GPUs.



### 2.3.3 Propriedades

Como apresentado por Greve et al. (2018), as principais propriedades da tecnologia *blockchain* que contribuem para o desenvolvimento de aplicações e sistemas são as seguintes:

1. **Descentralização:** De acordo com Greve et al. (2018), este é o principal interesse em aplicações com a tecnologia. Os sistemas descentralizados estabelecem confiança entre as partes, descartando a necessidade de uma entidade intermediária central.
2. **Disponibilidade e Integridade:** Todos os dados e transações são replicados em diferentes nós a fim de manter o sistema disponível e consistente.  
**Transparência e Auditabilidade:** Todas as transações são registradas e são disponibilizadas publicamente, o que as tornam possíveis de serem auditadas. Além disso, os códigos da tecnologia costumam ser abertos e passíveis de verificação.
3. **Imutabilidade e Irrefutabilidade:** Uma vez que a transação foi registrada, esta não pode ser alterada ou desfeita. Atualizações são possíveis a partir da geração de novas transações garantindo assim um registro de todas as mudanças.
4. **Privacidade e Anonimidade:** A rede possibilita a privacidade aos usuários com cada usuário gerenciando suas próprias chaves e cada nó servidor armazena apenas fragmentos criptografados de dados do usuário. As transações ocorrem entre os usuários da *blockchain* com conhecimento apenas dos “endereços”, cabendo ao usuário querer se identificar ou não.
5. **Desintermediação:** A *blockchain* possibilita a integração entre diversos sistemas de forma direta e eficiente. Assim, é considerada um conector de sistemas complexos (sistemas de sistemas), permitindo a eliminação de intermediários de maneira a simplificar o projeto dos sistemas e processos.

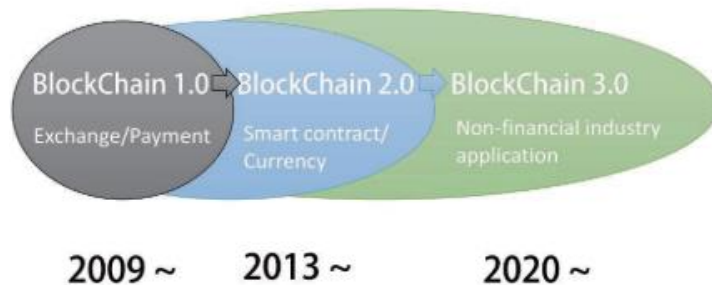
Adicionalmente Iansiti e Lakhani (2017), adiciona a *blockchain* como uma de suas principais características a **Lógica Computacional** devido à natureza digital dos registros, as transações realizadas na rede *blockchain* podem ser vinculadas a uma lógica computacional, ou seja, programadas com que chamamos de *smart contracts*, definido regras que devem ser executadas assim que as transações forem efetivamente adicionadas a rede.

### 2.3.4 Evolução da *blockchain*

Essa tecnologia despertou grande interesse mundial dadas essas características, e com a sua popularização, diversas criptomoedas começaram a surgir, dentre elas Bitcoin, Ether e Litecoin, porém essa tecnologia não se limita apenas a transações financeiras e pode ser utilizada para registrar virtualmente tudo que possua valor.

No ano de 2013, começaram a surgir novas aplicações para a *blockchain* além das aplicações financeiras. De acordo com Cheng et al. (2018), o surgimento da Ethereum *smart contracts* impulsionou a tecnologia *blockchain*, que passou a ser conhecida como *blockchain 2.0*. Como apresentado na Figura 4, a *blockchain 1.0* foi adotado principalmente pela Bitcoin para resolver problemas relacionados a criptomoedas e pagamentos descentralizados e a *blockchain 2.0* focado no empregado para transformar ativos através de contratos inteligentes, criando valor através do surgimento de alternativas ao Bitcoin. E já se espera que a evolução não pare, avançando para uma *blockchain 3.0* que possa permitir maior flexibilidade e escalabilidade a rede.

Figura 4 - Desenvolvimento do blockchain.



Fonte: Cheng et al. (2018).

Lin (2017), define um *smart contract* como um contrato digital escrito em código-fonte executado por computadores, que integra o mecanismo à prova de adulteração de *blockchain*.

Com a utilização dos *smart contract* da plataforma Ethereum, é possível a utilização de códigos de programação que pode definir regras estritas e consequências da mesma forma que um documento legal tradicional, estabelecendo as obrigações, benefícios e penalidades que podem ser devidas a qualquer das partes em várias circunstâncias diferentes.

Os *smarts contracts* da plataforma Ethereum são escritos na linguagem Solidity, uma linguagem de programação semelhante ao JavaScript. Depois que um contrato inteligente programado pela Solidity for concluído, um complemento chamado *solc* é necessário para transformar o código de Solidity em *bytecode* de contrato e em seguida, as instruções compiladas são implantadas em uma *blockchain* da Ethereum.

## 2.4 ETHEREUM

Dentro da indústria de *blockchain*, o termo capitalização de mercado (ou capitalização de mercado) se refere a uma métrica que mede o tamanho relativo de uma criptomoeda e segundo ao site CoinMarketCap (2020), que exibe a capitalização das criptomoedas em tempo real, o Ether, moeda da *blockchain* Ethereum, é uma das maiores (segunda posição) em questão de capital de mercado, no entanto, é importante entender que o Ethereum é mais do que uma moeda digital.

O projeto Ethereum foi introduzido, em 2013, por Vitalik Buterin. Este projeto foi financiado, em 2014, através de um *crowdfunding* e, em 2015, entrou definitivamente em funcionamento. Segundo Buterin (2014), o objetivo da criação da plataforma Ethereum é melhorar e unir os conceitos de *scripting*, *altcoins* (moedas alternativas ao Bitcoin) e *meta-protocols* aos que já existiam no sistema do Bitcoin, e permitir que os desenvolvedores criem aplicativos que utilizem os recursos do *blockchain* e tenham escalabilidade, padronização, facilidade de desenvolvimento e interoperabilidade oferecida por esses diferentes paradigmas, todos ao mesmo tempo.

Vitalik Buterin, foi um dos programadores envolvidos no desenvolvimento do Bitcoin, quando constatou que a *blockchain* criada apresentava algumas limitações quanto sua utilização, sendo unicamente para transações da moeda virtual. Dessa forma, a plataforma Ethereum foi criada para ser uma expansão da tecnologia *blockchain* já existente para além de nativamente ser um sistema de pagamento, com a criação da moeda ether, permitir executar e ativar contratos inteligentes (*smart contracts*) o que marcou o início da tecnologia *blockchain* de segunda geração.

De acordo com Wood (2014), se o Bitcoin foi criado como uma alternativa de moeda digital em relação ao sistema monetário tradicional, o Ethereum, por sua vez, foi concebido para ser uma rede *peer-to-peer* de máquinas virtuais sendo criada como uma plataforma para criar e executar contratos inteligente ou aplicações descentralizadas sobre *blockchain*. Assim, a diferença fundamental da Ethereum é desta ser uma plataforma programável, onde contratos ou aplicações podem ser construídos de forma descentralizada.

A Ethereum utiliza a tecnologia *blockchain* para armazenar todas as transações, sendo estas verificadas e validadas pelo processo de *mining* (mineração) por todos os utilizadores do sistema através do *proof-of-work* (PoW). Porém, a Ethereum está mudando para

um mecanismo de consenso denominado prova de participação (*proof-of-stake* ou PoS) a partir do PoW. Este sempre foi o plano, pois é uma parte fundamental da estratégia da comunidade para dimensionar o Ethereum por meio das atualizações Eth2, que se refere a um conjunto de atualizações interconectadas que tornarão o Ethereum mais escalonável, mais seguro e mais sustentável.

O processo PoS é descrito por Mascarenhas, Vieira e Ziviani (2018), como um processo no qual a validação de transações e acordos, pode ser efetuado por qualquer utilizador com crédito no sistema e diferentemente do PoS os participantes do processo de validação não competem entre si para conseguir validar o novo bloco e sim um validador é eleito aleatoriamente para essa tarefa. No entanto, a validação é feita por turnos e votação, sendo que os votos dos utilizadores não têm o mesmo peso, dependendo do tamanho do seu depósito em moeda digital.

As vantagens que advêm do PoS são:

1. Aumento da segurança: Além da segurança tradicional do *blockchain*, neste tipo de validação o sistema evita que os usuários se sintam motivados a fraudar o algoritmo já que no momento que uma transação fraudulenta é detectada, o detentor do nó corre o risco de perder tudo, ou uma parte, do dinheiro investido. Além disso, perde o direito de participar nas próximas disputas de validação.
2. Redução do risco de centralização: Os usuários são cada vez mais encorajados a participar do processo de validação já que é relativamente fácil e barato fazê-lo. Aliado ao fato da rede contar com o sistema de escolhas aleatórias a rede se torna mais democrática e descentralizada
3. Aumento da eficiência energética: Uma das principais críticas ao PoW é em relação aos gastos com energia elétrica, já que uma grande quantidade de energia é gasta para resolver um problema matemático que serve apenas para encontrar um novo bloco, isso tem um alto custo, porque requer muita energia e tentativas mal-sucedidas.

Ao iniciarmos uma nova rede privada da Ethereum atualmente podemos também escolher como mecanismo de consenso prova de autoridade (*proof-of-authority* - PoA), denominado como Clique.

O protocolo de consenso Clique é especificado em EIP-225. O conjunto inicial de signatários autorizados é configurado no bloco genesis. Os signatários podem ser autorizados e desautorizados usando um mecanismo de votação, permitindo assim que o conjunto de signatários mude enquanto a *blockchain* opera. (ETHEREUM.ORG, 2019)

Desse modo, o mecanismo de consenso Clique é um sistema de prova de autoridade onde novos blocos podem ser criados apenas por 'signatários' autorizados, ideal para redes privadas em que não estarão abertas para mineração com participação pública.

## 2.5 SMART CONTRACTS

O termo *smart contracts* de acordo com Sillaber e Walth (2017) foi inicialmente proposto anos antes do *blockchain*, em 1994 por Nick Szabo que o descreveu como a execução de contratos por computador entre duas partes que possa ser assegurada sem a necessidade de terceiros.

Segundo Christidis e Devetsikiotis (2016), podem ser definidos como as cláusulas de um contrato que são codificadas e podem ser incorporadas a um hardware ou software com auto execução. Os *smart contracts* se assemelham aos contratos legais tradicionais que representam um acordo de vontades firmado por duas ou mais pessoas, capaz de criar, modificar ou extinguir direitos através de transações com segurança sem necessariamente se conhecerem e confiarem umas nas outras.

Koulu (2016) define que um *smart contract* consiste em três componentes distintos: o próprio acordo contratual, a administração das pré-condições necessárias para que as obrigações contratuais ocorram e a execução efetiva do contrato.

- **Acordo contratual:** As obrigações contratuais das partes envolvidas são negociadas e transformadas em código de programa executável. As partes são identificadas por meio de suas contas de *blockchain* (carteiras), e as transações retratam obrigações cumpridas entre elas. O código é então implementado e armazenado no *blockchain*.
- **Administração de pré-condições:** Todos os nós participantes, incluindo mineradores, são capazes de executar contratos inteligentes. O que restringe sua execução é a avaliação se as pré-condições definidas no *smart contract* são atendidas ou não.
- **Execução do contrato:** Se as pré-condições forem atendidas, o contrato é executado e as transações são realizadas pelos nós participantes. A execução correta é garantida através do protocolo de consenso. Portanto, os contratos inteligentes são autoexecutáveis, o que significa que os ativos digitais são alocados de forma autônoma de acordo com os termos contratuais predefinidos.

Como os *smart contracts* são códigos de programa estes também possuem linguagens de programação para serem codificados. A plataforma Ethereum, por exemplo, em sua página oficial para desenvolvedores encontramos Solidity como linguagem para o desenvolvimento de *smart contracts*. Uma linguagem de alto nível, fortemente tipada, com bibliotecas, herança e orientada a objetos inspirada pelo C++, Python e JavaScript. Atualmente

a linguagem Vyper, que se parece com Python, também é suportada, porém, Solidity é a mais utilizada pela rede.

## 2.6 METADADOS

O termo metadados foi primordialmente usado no contexto dos sistemas de banco de dados para descrever e controlar a gestão e o uso dos dados (SAYÃO, 2010, p. 2). Segundo Grácio (2012), os metadados referem-se à um conjunto de dados, chamados ‘elementos’, cujo número varia de acordo com o padrão adotado, e que descreve o recurso, possibilitando a um usuário ou a um mecanismo de busca acessar e recuperar esse recurso. A finalidade principal dos metadados é documentar e organizar de forma estruturada os dados das organizações, com o objetivo de minimizar duplicação de esforços e facilitar a manutenção dos dados (IKEMATU, 2001, p. 1).

Alves (2010), afirma que a construção adequada e aplicação do padrão de metadados correspondente ao tipo de ambiente informacional garantem maior efetividade nos sistemas e, conseqüentemente, uma recuperação com melhores resultados.

Nesse sentido, para Ouchi e Simianato (2018), quando se tem o objetivo de disponibilizar conjuntos de dados na Web é aconselhado considerar a descrição desses dados escolhendo, desde o início, um ou mais padrões de vocabulários que seguem padrões internacionais de metadados.

Para Ikematu (2001), o interesse sobre o assunto de metadados cresceu dada a necessidade:

- das pessoas em definirem melhores formas de encontrar e avaliar informações na Internet e nas intranets;
- e dos sistemas de gerenciamento de conhecimento integrando informações de fontes múltiplas e aplicações precisam oferecer maior facilidade de pesquisa e manutenção.

Em uma iniciativa conjunta dos buscadores Google, Yahoo, Bing e Yandex, em junho de 2011, surgiu o Schema.org como uma proposta compartilhada e colaborativa com o objetivo de criar, manter e promover esquemas de dados estruturados para a internet. (SCHEMA.ORG, 2020)

Desde então, o Schema.org forneceu um local central para a documentação dos esquemas de dados estruturados, junto com exemplos de uso. O vocabulário do Schema.org contém classes para descrever os tipos mais populares de conteúdo da Web, incluindo perfis pessoais, resenhas de filmes, saúde e medicina, esportes, arquivos, bibliotecas e bibliografia, automóveis, ofertas de produtos e muito mais. A Figura 5 apresenta parte da documentação da classe Person, a qual lista os atributos válidos de uma pessoa, seu significado e valor esperado.

Figura 5 - Definição da classe Person em schema.org.

**Person**  
 A Schema.org Type  
 Thing > Person [more...]

A person (alive, dead, undead, or fictional).

Property	Expected Type	Description
<b>Properties from Person</b>		
<b>additionalName</b>	Text	An additional name for a Person, can be used for a middle name.
<b>address</b>	PostalAddress or Text	Physical address of the item.
<b>affiliation</b>	Organization	An organization that this person is affiliated with. For example, a school/university, a club, or a team.
<b>alumniOf</b>	EducationalOrganization or Organization	An organization that the person is an alumni of. Inverse property: <b>alumni</b>
<b>award</b>	Text	An award won by or for this item. Supersedes <b>awards</b> .
<b>birthDate</b>	Date	Date of birth.
<b>birthPlace</b>	Place	The place where the person was born.
<b>brand</b>	Brand or Organization	The brand(s) associated with a product or service, or the brand(s) maintained by an organization or business person.
<b>callsign</b>	Text	A <b>callsign</b> , as used in broadcasting and radio communications to identify people, radio and TV stations, or vehicles.
<b>children</b>	Person	A child of the person.
<b>colleague</b>	Person or URL	A colleague of the person. Supersedes <b>colleagues</b> .
<b>contactPoint</b>	ContactPoint	A contact point for a person or organization. Supersedes <b>contactPoints</b> .
<b>deathDate</b>	Date	Date of death.
<b>deathPlace</b>	Place	The place where the person died.
<b>duns</b>	Text	The Dun & Bradstreet DUNS number for identifying an organization or business person.
<b>email</b>	Text	Email address.
<b>familyName</b>	Text	Family name. In the U.S., the last name of a Person.
<b>faxNumber</b>	Text	The fax number.

Fonte: (SCHEMA.ORG, 2020)

Em comparação com outros vocabulários, segundo Bizer, Meusel e Primpeli (2020), analisando o desenvolvimento da adoção de classes do schema.org, há um aumento contínuo no número de Dispositivos Lógicos Programáveis (PLDs) que adotam as classes do schema.org, principalmente nos últimos três anos, enquanto o número de sites implantando as classes do Facebook Open Graph Protocol, o segundo vocabulário mais utilizado, permanece aproximadamente constante. Sendo em 2019, a classe com maior adoção do Schema.org, a classe Product, presente em 1,27 milhões de PLDs, enquanto a mais utilizada do Open Graph Protocol, a classe Breadcrumb, apresenta adoção em 312 mil PLDs.

De acordo com Schema.org (2020), o Open Graph Protocol atende muito bem a finalidade pela qual foi desenvolvido que é descrever as informações das classes utilizadas pelo

Facebook, mas não fornece as informações no mesmo nível de detalhamento apresentado pelo vocabulário do Schema.org, em específico quando se trata das entidades da área da saúde.

Levando esses fatos em consideração, o Schema.org foi aqui escolhido por ser o vocabulário que mais cresce ao passar dos anos, por ser adotado pelos maiores buscadores da internet e fornecer as informações de forma mais detalhada para as entidades necessárias para um sistema de acreditação em saúde.

Vale notar que essa definição de classe sugerida pelo Schema.org segue o mesmo conceito de atributos de classes da programação orientada a objetos, podendo então estes esquemas servirem como base para modelagem de classes voltadas para o desenvolvimento de sistemas.

Desta forma, um sistema que visa ser utilizado internacionalmente de forma colaborativa com várias instituições autônomas e independentes, a definição de um padrão para a descrição dos metadados torna-se necessária. Utilizando estes esquemas já definidos e internacionalmente reconhecidos, qualquer instituição que queira compartilhar, ao passo que consome, os dados de um sistema colaborativo, não terá muitas dificuldades de integração dada a interoperabilidade que este tipo de modelagem propõe.

## **2.7 PROVA DE CONCEITO**

Prova de conceito, do inglês *Proof of Concept* (PoC), é um termo utilizado por Carsten (1989), para a realização de um determinado método com o objetivo de verificar se algum conceito ou teoria tem potencial prático. Para Silva (2015), é um termo utilizado para denominar um modelo prático que possa provar o conceito (teórico) estabelecido por uma pesquisa ou artigo técnico.

Em Tecnologia da Informação (TI), segundo Pressman e Maxim (2016), o termo pode ser relacionado ao desenvolvimento de um protótipo como ferramenta para provar a viabilidade de uma nova tecnologia no contexto de software.

No desenvolvimento de software de acordo com Nathali (2020), o objetivo é descobrir quais as tecnologias serão mais eficazes e apropriadas no desenvolvimento do produto, provando que a ideia a ser desenvolvida poderá ser realmente construída.



Segundo Linhares (2016), uma prova de conceito é necessária e deve ser empregada em situações que considerem:

- **Escopo do projeto não bem entendido** – quando o escopo não é familiar aos envolvidos no projeto, a prova de conceito deve explorar soluções possíveis, mas também pode ajudar a entender e esclarecer os requisitos necessários;
- **Experiência de projeto** – quando o grupo de trabalho tem pouca experiência anterior, levando em consideração que todo projeto é uma atividade única e não repetitiva, não sendo possível basear-se em arquiteturas e tecnologias existentes;
- **Requisitos complexos** – quando qualquer requisito é considerado complexo, ou classificado como particularmente oneroso, mesmo que o domínio seja familiar e o projeto tenha similaridade com outros existentes;
- **Alto risco** - quanto maior o risco, mais esforço é necessário, uma vez que há a expectativa de resultados mais realistas dos modelos produzidos e avaliados. Entretanto, é preciso reconhecer que nem todos os riscos podem ser eliminados.

Silva (2015), destaca que a PoC pode ser realizada de diversas formas, dentre elas, enumerando em forma de lista as tecnologias conhecidas que pareça adequada ao projeto; criação de um esboço do modelo conceitual da solução a ser validada; simulações da solução através de ferramentas de software e criação de protótipos executáveis.

A utilização da PoC no início de um projeto pode trazer uma série de benefícios, devido ao fato de antecipar muitos dos erros que passariam despercebidos até o início das próximas etapas fossem iniciadas. Nesse sentido, Nathali (2020), cita como benefícios:

- **Reduzir gastos desnecessários:** Saber previamente que uma ideia não é viável em termos de tecnologia ou de mercado, antes de assumir um grande risco, certamente irá economizar muito tempo e investimento.
- **Identificar falhas e evitar a Dívida Técnica:** É possível verificar se escolhas como a arquitetura do código a ser utilizada e os objetivos que a solução deverá atingir, para garantir que o código seja construído de maneira sólida.
- **Evita incertezas entre as partes envolvidas:** Permite verificar se a ideia é realmente executável, e que há usuários interessados na sua utilização. Ampliando as expectativas do time envolvido e as possibilidades de satisfação sobre o projeto.

O método PoC não elimina todas as falhas que podem acontecer em um projeto, apresentando se o conceito é viável, garantindo a oportunidade de ser corrigido e aprimorado no próximo processo. Sendo assim, é uma metodologia recomendada para evitar imprevistos durante a execução do projeto, comprovando o bom entendimento do escopo e que os requisitos estão bem definidos.

## 2.8 PROTOTIPAÇÃO

De acordo com Bacurau, Leal e Ramos ([2011]), a prototipação, no contexto de sistemas de informação, consiste na produção de sistemas simplificados (protótipos) com a finalidade de realizar experimentações e verificações para avaliar e validar suas funcionalidades.

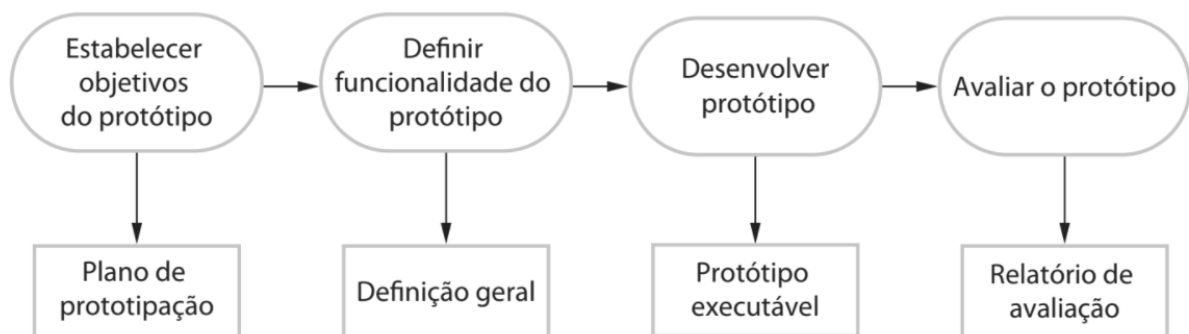
Pressman e Maxim (2016), alertam que, muitas vezes, o cliente define uma série de objetivos do software desejado, mas não consegue dar muitos detalhes sobre os requisitos ou o desenvolvedor pode não ter certeza sobre alguns detalhes do sistema. Tornando assim, a prototipação uma opção válida para mitigar essa questão.

Segundo Queiros et al. (2019), protótipos são classificados conforme o nível de fidelidade, podendo ser de:

- **Baixa fidelidade:** utiliza materiais como papel, madeira e cartolina no lugar de telas eletrônicas e metal, durante a etapa inicial de desenvolvimento, para compreensão de requisitos.
- **Média fidelidade:** versão aprimorada do protótipo de baixa fidelidade. Utiliza ferramentas computacionais para criação do protótipo com funções limitadas a alguns recursos para avaliação de cenários.
- **Alta fidelidade:** utiliza materiais que se espera que estejam no produto final e geralmente são construídos em linguagem de programação. Dessa forma, desenvolvido e apresentado no computador.

Um modelo de processo para desenvolvimento de protótipos presente na Figura 6, apresentado por (SOMMERVILLE, 2011).

Figura 6 - O processo de desenvolvimento de protótipos.



Fonte: (SOMMERVILLE, 2011).

Primeiramente, os objetivos da prototipação devem ser explicitados desde o início do processo como, por exemplo, se o seu desenvolvimento tem a finalidade de prototipar a

interface de usuário, a validação dos requisitos funcionais ou demonstrar aos gerentes a viabilidade da aplicação. O mesmo protótipo não pode cumprir todos os objetivos. A falta dessa definição pode resultar na perda dos benefícios esperados do desenvolvimento do protótipo pela falta de entendimento sobre a função do protótipo.

O próximo passo é decidir o que colocar e, talvez mais importante ainda, o que deixar de fora do sistema de protótipo e seguir para o processo de desenvolvimento. Para reduzir os custos e o tempo de desenvolvimento do protótipo pode-se optar por relaxar os requisitos não funcionais, padrões de confiabilidade e qualidade de programa.

Após desenvolvido, o estágio final do processo é a avaliação do protótipo. Durante esse estágio, provisões devem ser feitas para o treinamento do usuário que avaliarão e fornecerão um retorno (feedback), que servirá para aprimorar os requisitos.

A construção de protótipos permite ao desenvolvedor demonstrar uma funcionalidade de forma rápida a fim de elucidar os requisitos alinhando com as expectativas do cliente. No entanto, Sommerville (2011), advertem que o aproveitamento do protótipo para a construção da versão final do software costuma ser desaconselhável devido aos seguintes motivos:

1. Pode ser impossível ajustar o protótipo para atender requisitos de desempenho, proteção, robustez e confiabilidade, que foram ignorados durante o desenvolvimento do protótipo.
2. Mudanças rápidas durante o desenvolvimento inevitavelmente significam que o protótipo não está documentado, dificultando a manutenção a longo prazo.
3. As mudanças durante o desenvolvimento do protótipo provavelmente terão degradado a estrutura do sistema. O sistema será difícil e custoso de ser mantido.
4. Padrões de qualidade organizacional geralmente são relaxados para o desenvolvimento do protótipo.

Nesse contexto, Pressman e Maxim (2016), afirma que o segredo é definir as regras do jogo desde o início; todos os envolvidos devem concordar que o protótipo é construído unicamente com o objetivo de ser um mecanismo para definição de requisitos. Portanto, será descartado (pelo menos em parte) e o software final é arquitetado visando qualidade.

Podemos utilizar para criação de um protótipo a técnica de *Toy Example* (também chamado de *Toy Problem*), a qual para Chaudhuri *et al.* (2015), consiste no desenvolvido de um modelo simples para imitar um problema real mais complexo a fim de testar e avaliar os

métodos a serem utilizados. Esta técnica será utilizada neste trabalho a fim de demonstrar e compreender o funcionamento das interações do sistema web com a rede *blockchain*.

## 2.9 ESTUDO DE CASO INSTRUMENTAL

Segundo Yin (2015), estudo de caso é um método de pesquisa que investiga um fenômeno atual dentro de seu contexto no mundo real, em que as fronteiras entre o fenômeno e o contexto não são claramente definidas ou na situação em que múltiplas fontes de evidência são usadas. Atribuindo assim, o objetivo de explorar, descrever e explicar o evento ou fornecer uma compreensão profunda do fenômeno.

Para a utilização deste método, Dencker (2007), ressalta que o estudo de caso é recomendado na fase inicial das pesquisas científicas, dada a possibilidade de o pesquisador levantar dados que serão úteis na formulação de hipóteses e na reformulação de seu problema de pesquisa.

Stake (2001, p. 433), por sua vez, alerta que o pesquisador deve estar ciente do tipo de conhecimento que se pretende adquirir com este método. Ao utilizar o método de estudo de caso, a ênfase está na compreensão, fundamentada basicamente no conhecimento tácito que, segundo Yin (2015), tem uma forte ligação com intencionalidade, o que não ocorre quando o objetivo é meramente explanação, baseada no conhecimento proposicional. Dessa forma, o estudo de caso pode ser uma desvantagem quando a explanação, ou a busca de um conhecimento proposicional, seja o objetivo de um estudo, mas quando o objetivo é a compreensão, ampliação da experiência, a desvantagem desaparece.

O estudo de caso pode ser classificado de acordo com suas finalidades em: intrínseco, instrumental e coletivo. Em particular no estudo de caso instrumental, segundo Yin (2015), é utilizado quando se examina um caso para se compreender melhor outra questão, algo que não é exclusivamente o caso em si, orientar estudos ou ser instrumento para pesquisas posteriores.

Nesse sentido, a utilização do estudo de caso auxiliará a compreender o funcionamento mais amplo do sistema, utilizando o estudo de caso instrumental para direcionar os testes do protótipo funcional do sistema web baseado no artigo de Souza Junior *et al.* (2019).

### 3 METODOLOGIA

Neste capítulo, está descrito o processo de elaboração do “*back-end*” para o Sistema de Acreditação Internacional, demonstrando algumas metodologias utilizadas para isso.

O procedimento de construção do “*back-end*” segue, a priori, a seguinte sequência:

- a) Definição da Plataforma de *blockchain* a ser utilizada;
- b) Desenvolvimento de *scripts* de instalação e configuração da rede *blockchain*.
- c) Desenvolvimento de material de instalação e utilização dos *scripts* desenvolvidos.
- d) Mapeamento dos requisitos básicos para o protótipo inicial (*Toy Example*).
- e) Desenvolvimento de *smart contracts* e um protótipo inicial para teste da tecnologia.
- f) Adequações para integração com o sistema web.
- g) Testes de funcionalidade da *blockchain* e dos *smart contracts*.
- h) Teste com usuários através de um estudo de caso instrumental.

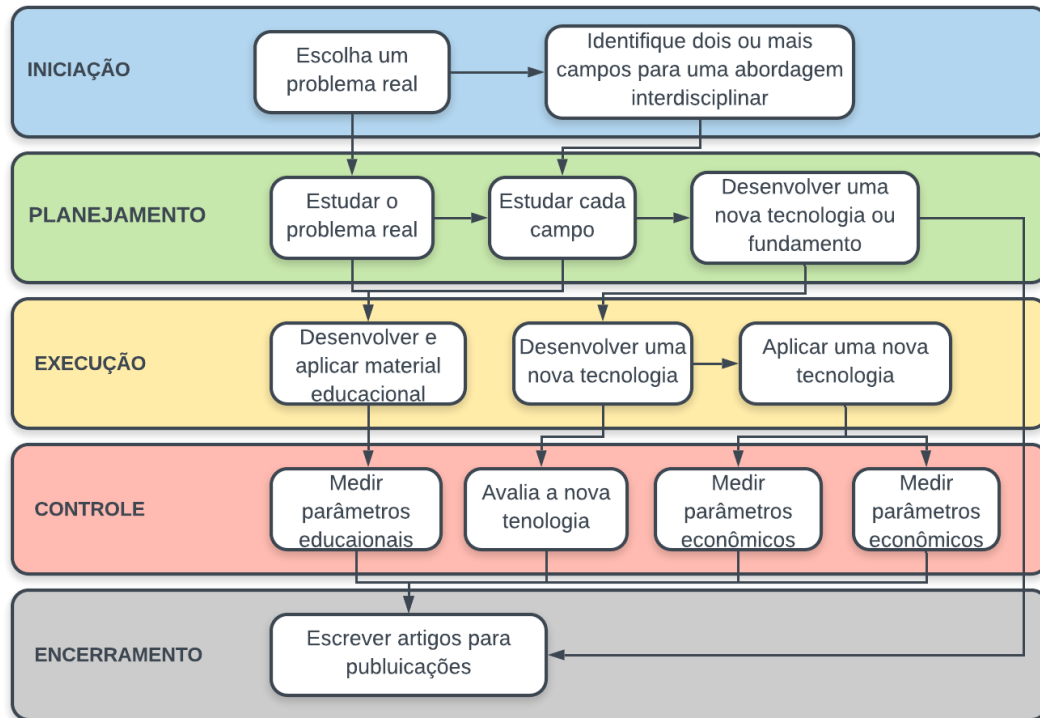
Como este projeto tem característica interdisciplinar, a metodologia escolhida foi a MVC EA-IRPM de Letouze et al. (2012), metodologia de desenvolvimento de sistemas que voltado para projetos interdisciplinares. O resultado esperado é o *back-end* do protótipo funcional de um sistema web para o IAS.

#### 3.1 EA-IRPM

Interdisciplinar é um adjetivo que qualifica o que é comum a duas ou mais disciplinas ou outros ramos do conhecimento. É o processo de ligação entre as disciplinas. Dessa forma, um projeto interdisciplinar é um projeto que aborda duas áreas distintas, buscando a interação entre disciplinas construindo conhecimento comum, que muitas vezes não seria adquirido sem o enfoque de áreas distintas.

A metodologia de “Aquisição Evolucionária - Gerenciamento de Projetos de Pesquisa Interdisciplinar” (EA-IRPM) de Letouze (2012) é aconselhada para projetos de sistemas com alto grau de interdisciplinaridade. O EA-IRPM é uma combinação do IRPM proposto por Letouze (2012), com a estratégia de “Aquisição evolucionária” de Lightsey (2001).

Figura 7 - Estrutura do IRPM.



Fonte: Adaptado de Letouze (2011).

O IRPM, representado pela Figura 7, é composto por cinco fases: iniciação, planejamento, execução, controle e encerramento, cada etapa tem seus processos. Os processos de cada uma das fases estão descritos na lista abaixo.

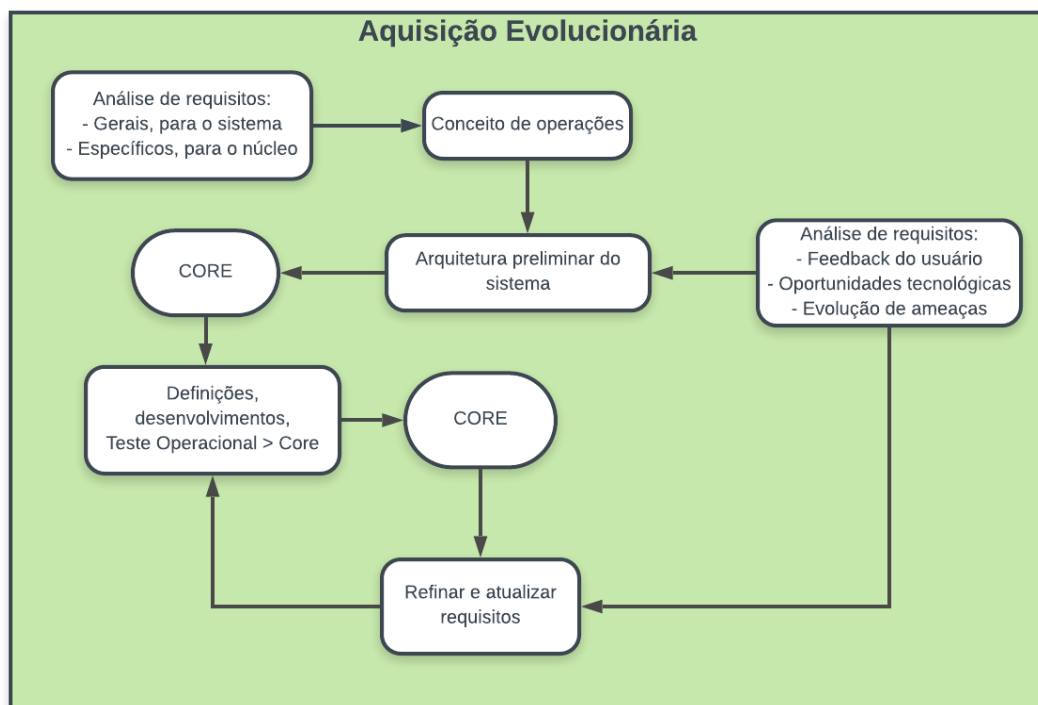
- **Iniciação:** determinar objetivos do projeto, entregas e saídas do processo, documentar restrições e suposições do projeto, definir estratégias, identificar critérios de desempenho, determinar requisitos de recursos, definir o orçamento e produzir uma documentação formal, identificar dois ou mais campos para uma abordagem interdisciplinar, a fim de documentar suas restrições e suposições.
- **Planejamento:** refinar o projeto e fazer um estudo mais profundo do problema e os campos escolhidos podem ser executados. Estes estudos devem promover um novo fundamento ou metodologia, criar uma estrutura analítica do projeto (EAP), desenvolver o plano de gerenciamento de recursos, refinar as estimativas de tempo e custo, estabelecer controles de projeto, desenvolver o plano do projeto e obter a aprovação do plano.
- **Execução:** comprometer recursos, implementar recursos, gerenciar o progresso, comunicar o progresso e implementar procedimentos de garantia de qualidade.
- **Controle:** medir o desempenho, inclusive, dos novos parâmetros educacional, tecnológico, econômico e social estabelecidos no Planejamento, refinar os limites de controle, adotar ações corretivas, avaliar a eficácia das ações corretivas, garantir a conformidade do plano, reavaliar os planos de controle, responder aos gatilhos dos eventos de risco e monitorar a atividade do projeto.

- **Encerramento:** obter a aceitação de resultados, documentar as lições aprendidas, facilitar o fechamento, preservar registros e ferramentas de produtos e liberar recursos, dependendo dos resultados dos parâmetros medidos os artigos podem ser escritos.

A metodologia EA proposta por Lightsey (2001), é iniciada pelo processo de análise de requisitos. Os requisitos são inicialmente divididos em requisitos gerais e requisitos específicos, sendo os requisitos gerais definidos para o sistema como um todo e os requisitos específicos definidos para a criação do núcleo inicial do sistema.

A Aquisição Evolucionária (EA), apresentada na Figura 8, começa com a análise de requisitos, na qual são definidos os requisitos gerais para o sistema e os requisitos específicos para o núcleo inicial do sistema. No passo seguinte, projetamos a partir de uma análise de requisitos do feedback dos usuários, oportunidades tecnológicas e avaliação de ameaças. Da arquitetura preliminar do sistema, desenvolvemos o primeiro núcleo do sistema.

Figura 8 - Diagrama Aquisição Evolucionária.



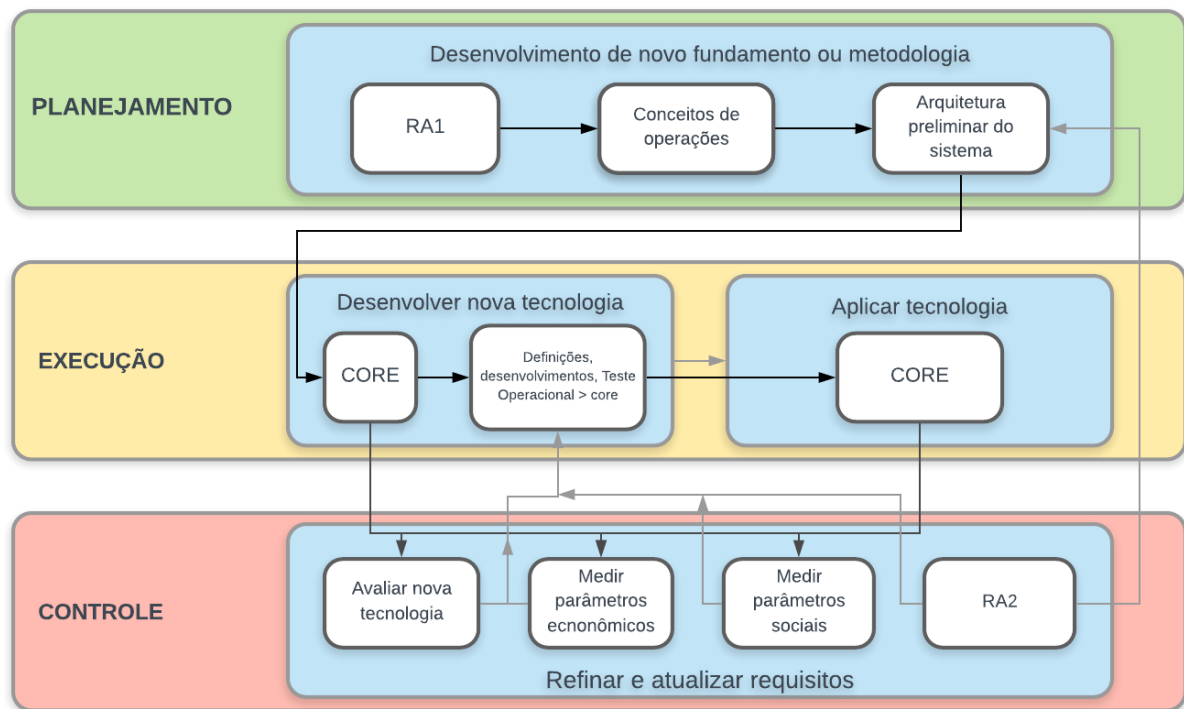
Fonte: Adaptado de Lightsey (2001).

A partir do núcleo inicial desenvolvido, novas definições e desenvolvimentos como testes operacionais podem resultar em uma nova versão do núcleo. Então, com experiência e uso, novos refinamentos e atualizações de requisitos podem ser identificados e usados para desenvolver um novo núcleo ou melhorá-lo.

Para propor o EA-IRPM, Letouze (2012), combinou sua estratégia de gestão de projetos interdisciplinar com a metodologia de desenvolvimento de sistemas Aquisição Evolucionária, unindo os processos e etapas da EA à três das fases do IRPM, Planejamento, Execução e Controle, como vista na Figura 9 cuja as setas pretas apresentam o sentido principal do fluxo e as cinzas indicando o retorno de informações. Neste modelo, os RA significam Análise de Requisitos de:

- a) geral para o sistema e específica para o núcleo; e
- b) feedback do usuário, tecnologia oportunidades e ameaças em evolução.

Figura 9 - Incorporação de EA ao IRPM.



Fonte: Adaptado de Letouze (2012).

Assim, na fase de planejamento, mais especificamente depois de estudar o problema real através das visões dos campos interdisciplinares escolhidos na fase de Iniciação, procura-se desenvolver um novo fundamento ou metodologia para gerar a arquitetura preliminar do sistema. Isso é, começando com RA 1, considerando RA 2 se disponível, para construir a arquitetura preliminar do sistema.

Assim, na fase de planejamento, mais especificamente depois de estudar o problema real através das visões dos campos interdisciplinares escolhidos na fase de Iniciação, procura-se desenvolver um novo fundamento ou metodologia para gerar a arquitetura preliminar do sistema. Isso é, começando com RA 1, considerando RA 2 se disponível, para construir a arquitetura preliminar do sistema.



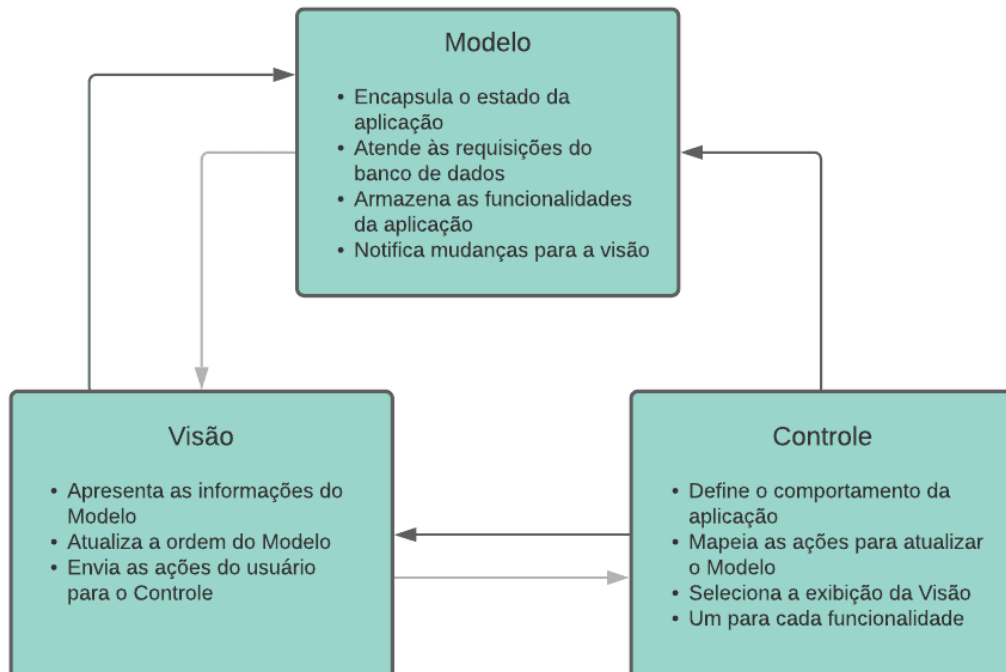
Então a fase de execução começa com o desenvolvimento de uma nova tecnologia, que consiste em implementar o núcleo a partir da arquitetura preliminar do sistema, seguida por novas definições e desenvolvimentos para realizar testes operacionais. Posteriormente, a nova tecnologia é aplicada em uma situação da vida real, ou seja, o núcleo deve ser colocado em produção.

A fase de controle é sobre refinar e atualizar requisitos, o que implica avaliar tecnologia, medir parâmetros econômicos e sociais, e verificar o feedback dos usuários, oportunidades tecnológicas e ameaças em evolução, ou seja, RA 2.

### 3.2 MVC EA-IRPM

O padrão Model-View-Controller (MVC) é uma arquitetura de software amplamente difundida para a criação de sistemas web que pretende separar a lógica de negócios (modelo), a interface do usuário (view) e a entrada do usuário (controller). Esta arquitetura fornece uma maneira de dividir as funcionalidades de forma que cada camada seja independente e tenha suas responsabilidades, a camada chamada de modelo contém todos os metadados do sistema, a camada chamada de controle é responsável por todo o processamento de dados necessário e última camada é chamada de visão pois é responsável pela apresentação dos dados aos usuários. A Figura 10 apresenta o padrão MVC.

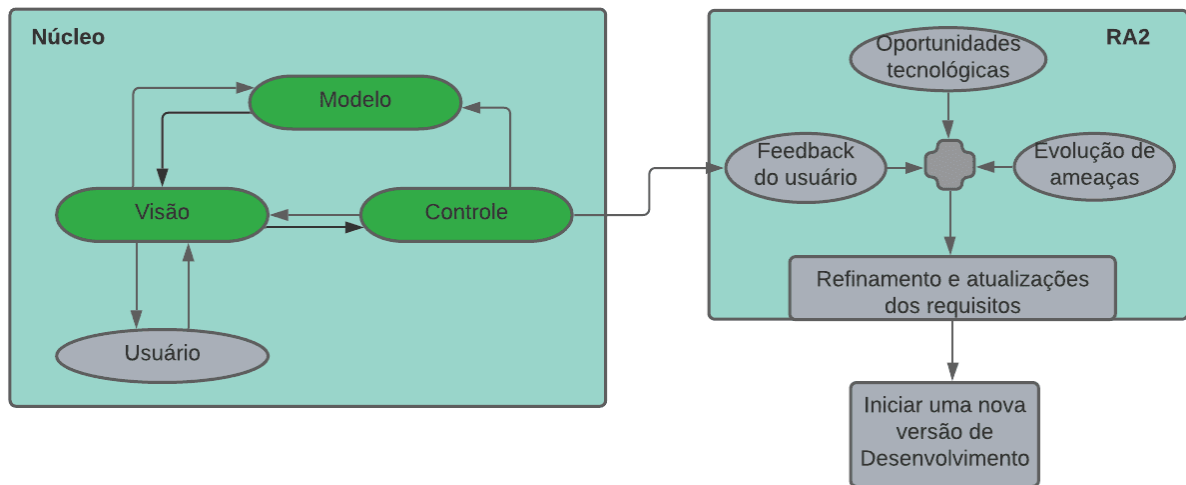
Figura 10 - Padrão de arquitetura MVC



Fonte: Adaptado de Letouze *et al.* (2012).

Para melhor conformidade com os padrões de desenvolvimento de sistemas para Web, Letouze *et al.* (2012), propõem a combinação do padrão MVC com a metodologia EA-IRPM (*Evolutionary Acquisition Interdisciplinary Research Project Management*) criada também por Letouze (2012), a fim de proporcionar ao desenvolvimento dos softwares mais flexibilidade e escalabilidade ao sistema. Produzindo um novo método chamado de MVC EA-IRPM.

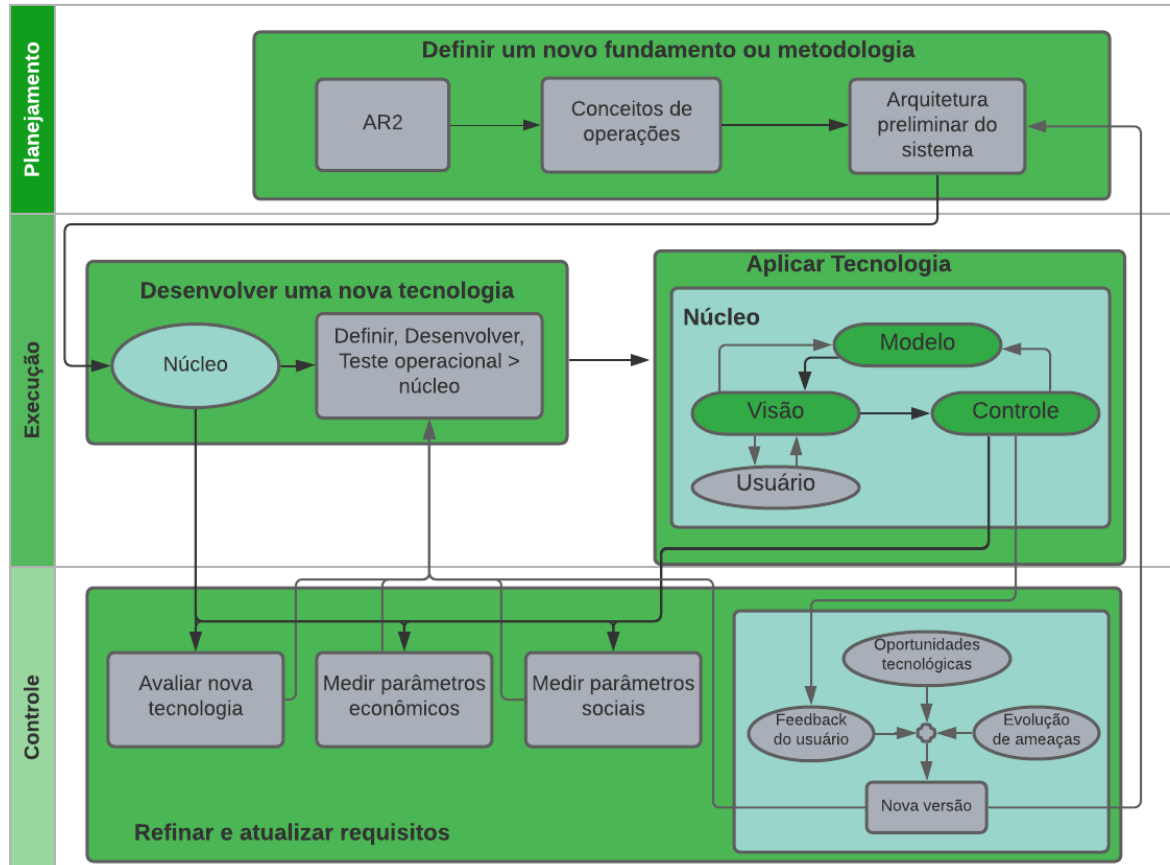
Figura 11- Arquitetura MVC incorporado à Aquisição Evolucionária.



Fonte: Adaptado de Letouze *et al.* (2012).

O MVC EA-IRPM surge da incorporação do MVC ao método de Aquisição Evolucionária presente no EA-IRPM e pode ser visto na Figura 11. O padrão MVC é então utilizado para o desenvolvimento do núcleo e se conecta à Análise de Requisitos RA2 por meio de feedback do usuário, que deve ser um banco de dados independente sistema, possibilitando a atualização e refinamento de novos requisitos. Dessa mesma forma, o MVC também pode ser incorporado ao EA-IRPM, como apresentado na Figura 12.

Figura 12 - Padrão de arquitetura MVC incorporado ao EA-IRPM.

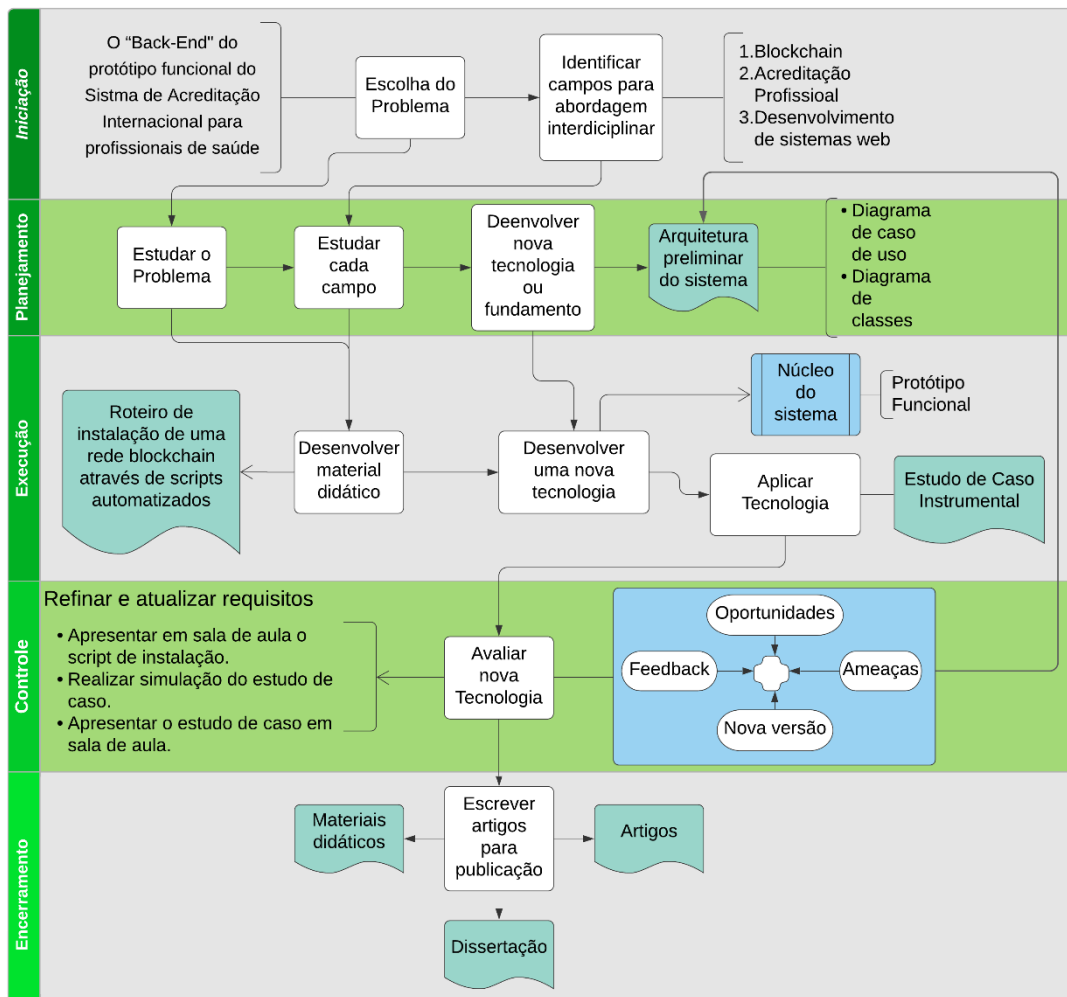


Fonte: Adaptado de Letouze et al. (2012).

### 3.3 A ESTRATÉGIA DE DESENVOLVIMENTO DO SISTEMA

Devido à natureza interdisciplinar deste trabalho, com o emprego da metodologia MVC EA-IRPM, serão seguidos os processos e etapas por esta definidas, adaptando-as às situações conforme se apresentem durante a execução deste projeto. A Figura 13 apresenta o fluxograma das etapas seguindo o MVC EA-IRPM aplicadas para este projeto.

Figura 13 - Fluxograma das etapas do MVC EA-IRPM para este projeto.



Fonte: Elaborado pelo autor (2021).

A seguir uma breve descrição das atividades a serem desenvolvidas durante cada fase do projeto.

### 3.3.1 Iniciação

Durante a fase de **iniciação**, pela metodologia EA-IRPM, temos o processo de escolha de um problema e identificação de dois ou mais campos para uma abordagem interdisciplinar, definidos assim como: a “tecnologia *blockchain*”; a “acreditação profissional” e “desenvolvimento de sistemas web”.

Ainda neste processo temos a definição do problema e objetivos do projeto. Limitando assim, este trabalho como a parte referente ao *back-end* do Sistema de Acreditação Internacional em Saúde baseado em tecnologia *blockchain*.

Assim, o esperado como resultado é uma infraestrutura básica de uma rede *blockchain* privada configurada e operacional para suportar o sistema Web, levando em consideração os *smarts contracts* que garantiram o tratamento dos dados referentes a acreditação profissional a serem armazenados na cadeia de blocos.

O *front-end* desse sistema não será abordado neste trabalho e maiores detalhes dos componentes do *back-end* serão apresentados na seção posterior.

### 3.3.2 Planejamento

Prosseguimos assim para a fase de **planejamento**, realizando estudos sobre o problema em si, através de revisões literárias sobre o processo de acreditação profissional e um levantamento das tecnologias que vão ser a base para a solução proposta, primeiramente sobre a própria tecnologia *blockchain*, as plataformas *blockchain* disponíveis, métodos de criação e utilização das redes *blockchain* e seus *smart contracts* e os conceitos. A seguir, em um segundo momento, é realizado o estudo sobre a acreditação profissional, sobre seus conceitos, processos e o impacto que esta prática pode proporcionar no cenário atual durante a realização deste trabalho, alinhando esses conceitos com o sistema proposto por Souza Junior et al. (2019).

Como a proposta do trabalho é o desenvolvimento de um sistema de suporte internacional, uma preocupação a ser observada é com os metadados no modelo do sistema, a fim de garantir a compatibilidade dos dados necessários para acreditação de diversas organizações médicas pelo mundo. Logo, um levantamento sobre a utilização de modelos de dados como o Schema.org é parte desse processo.

Com estes levantamentos podemos realizar a análise de requisitos dando origem aos diagramas de classe e casos de uso definindo a arquitetura preliminar do sistema que será desenvolvido.

### 3.3.3 Execução

A fim do planejamento prosseguimos para a fase de **execução** na qual dar-se-á início ao desenvolvimento do projeto. Espera-se três ciclos para esta etapa, sendo elas:

1. Criação e configuração de uma rede *blockchain* privada.
2. Desenvolvimento de *smart contracts*.
3. Integração da rede *blockchain* e dos *smart contracts* com protótipo funcional do Sistema de Acreditação Internacional em saúde.

Como resultado dessas etapas o esperado é uma rede *blockchain* com *smart contracts* funcionais a fim de criar uma versão preliminar do sistema proposto. Com esta versão inicial do núcleo do sistema será criado um estudo de caso para testes e prova de conceito do projeto. Outro produto esperado consiste em uma versão de *script* automatizado para a criação e configuração da rede *blockchain*, acompanhado de um material didático a ser aplicado em sala de aula a fim de receber *feedbacks* para próxima etapa.

### 3.3.4 Controle

A quarta fase da metodologia é o **controle**, nesta etapa os produtos desenvolvidos são avaliados a fim de identificar possíveis oportunidades tecnológicas, ameaças levando em consideração os *feedbacks* recebidos para o desenvolvimento de um novo núcleo do sistema incluindo ainda novos requisitos que possam ter sido ignorados nas fases anteriores. Espera-se nesta fase, a aplicação dos materiais didáticos em sala de aula a fim de recolher não somente as opiniões dos alunos como criar perspectivas sobre o conceito apresentado. A qualificação deste projeto de mestrado também será utilizada para controle sobre o modelo construído, reavaliando os pontos ressaltados nas considerações da banca avaliadora.

### 3.3.5 Encerramento

Por fim, na fase final chamada de **encerramento**, serão desenvolvidas as atividades de escrita da documentação do sistema e término da dissertação do programa de Mestrado Profissional de Modelagem Computacional de Sistemas da Universidade Federal do Tocantins, apresentando os devidos resultados e conclusões obtidas após finalizarmos o projeto. Além da dissertação, espera-se publicar artigos relacionados à pesquisa realizada e elaborar um manual didático do sistema desenvolvido.

## 3.4 BACK-END

O *back-end* desenvolvido por este projeto servirá como uma base para o desenvolvimento do Sistema de Acreditação Internacional. Porém, diferindo-se do significado tradicional do termo, podemos considerar como seus componentes:

1. Rede *blockchain* privada.
2. Configurações da rede e servidor expressos em forma de um manual de instalação.
3. Modelagem dos *smart contracts* em alinhamento com a modelagem das entidades do sistema em conformidade com os modelos do Schema.org.

4. Alinhamento das versões das ferramentas utilizadas na rede *blockchain* e *smart contracts* com as ferramentas do sistema web.

### 3.4.1 Rede *blockchain*

Durante o planejamento algumas observações importantes evidenciadas pelo foram levantadas sobre as necessidades deste projeto em relação a rede *blockchain*. Para a comunicação do sistema web com uma *blockchain* verificou-se a necessidade do suporte nativo aos *smart contracts* a fim de criar regras mais rígidas quanto a inserção de dados, outra necessidade é o suporte à interação com sistemas JAVA para garantir a integração com o sistema base, o RGM. Para escolher a *blockchain* que atende a estes requisitos realizou-se um *benchmarking* comparando as principais plataformas disponíveis. Foram analisadas as redes Bitcoin, Ethereum, Hyperledger Fabric, Quorum, EOS e R3 Corda.

A Tabela II apresenta um comparativo das características levantadas para a escolha da plataforma deste projeto, dentre elas estão: proposta da plataforma, tipo de rede se permite ou não a participação de partes sem ser previamente autorizadas, protocolos de consenso, interfaces de programação de aplicação (API) disponíveis e o suporte para *Smart Contracts*.

Tabela II - Benchmarking das plataformas *blockchain*.

	<b>Bitcoin</b>	<b>Ethereum</b>	<b>Hyperledger</b>	<b>Quorum</b>	<b>EOS</b>	<b>R3 Corda</b>
<b>Principal uso</b>	Criptomoeda	Plataforma genérica de <i>blockchain</i>	<i>Blockchain</i> voltada para empresas	Para aplicativos que requerem alto nível de privacidade.	Criar uma plataforma escalável para dapps em escala industrial	Plataforma especializada para indústria financeira (ativos digitais)
<b>Tipo de rede</b>	Não permissionada	Não permissionada ou permissionada	Permissionada	Permissionada	Permissionada	Permissionada
<b>Consenso</b>	PoW	PoW, PoS	Kafka, PoET, BFT	QuorumChain, RAFT (baseado)	DPOS	RAFT, BFT
<b>Smart Contracts</b>	Limitado	Sim	Sim	Sim	Sim	Sim
<b>APIs</b>	Bitcoin-cli (RPC)	Java, Python, Javascript, Go, Rust, .NET, Delphi	CLI, REST, Java e Node.js	Ferramentas familiares da Ethereum	Javascript, Swift, Java	Kotlin, Java
<b>Código Aberto</b>	Sim	Sim	Sim	Sim	Sim	Sim

Fonte: Elaborado pelo autor (2021).

A Ethereum foi escolhida por garantir as restrições mencionadas e após as comparações notou-se que, diferente das outras plataformas, apresenta a possibilidade de criar uma rede não provisionada seria a ideal para atingir objetivos futuros do projeto inicial, já que este tipo de rede é projetado para permitir a participação pública (por exemplo, alguns aplicativos que dependem de dados gerenciados pelos usuários).

Vale ressaltar que a plataforma Ethereum suporta nativamente os contratos compilados na linguagem Solidity, porém não dispõem, por padrão, no seu conjunto de ferramentas o compilador de códigos desta linguagem, sendo necessário a instalação a parte do Solidity Compiler.

Com a plataforma de *blockchain* escolhida deve-se definir a forma de instalação e configuração a serem empregadas. Como um dos objetivos do projeto é a apresentação de um produto que sirva como material didático, a replicação deste deve ser garantida. Optou-se assim, que a instalação deveria ser realizada por meio de *scripts*. Após um levantamento dos modos de instalação da rede, foram identificadas três formas distintas para a instalação da rede *blockchain* da Ethereum:

- através de sistemas de gerenciamento de pacotes;
- através da compilação de códigos fontes e;
- através de download de arquivo binário já compilado.

No primeiro caso, os sistemas de gerenciamento de pacotes do Linux e do MacOS podem auxiliar na instalação da rede Ethereum, precisamos para isso, adicionar um repositório PPA no caso do Linux ou instalar o Homebrew no caso do MacOS, sendo que para o sistema da Microsoft, este meio de instalação não está disponível. A problemática deste modo ficaria a cargo de seguir tutoriais desatualizados do Ethereum que poderiam indicar versões não mais suportadas em sistemas operacionais mais recentes, devendo fazer a correção das versões manualmente à medida que forem identificadas versões não mais existentes ou incompatíveis com dependências instaladas.

Para o segundo modo, algumas dependências são requeridas, sendo necessário baixá-las antes de se iniciar o processo de instalação. Aqui novamente, podemos ter problemas quanto a versão das dependências e do sistema operacional da máquina, o que no futuro poderia ser um complicador quanto a utilização das mesmas dependências utilizadas em um tutorial já que estas poderiam apresentar depreciação e incompatibilidade ao passo que estas forem sendo atualizadas.



O último meio de instalação é através do download de arquivo binário, deve-se baixar o arquivo compactado e extraí-lo para sua utilização, este meio tem menores riscos de problemas com dependências, assim basicamente o problema que pode ocorrer é escolher um arquivo desatualizado e incompatível com seu sistema operacional, o que geralmente pode ser contornado baixando a versão mais atual do arquivo.

No entanto, todos os três meios têm em comum a desvantagem de não ter um único arquivo, ou um único comando 100% funcional em todos os sistemas operacionais, já que para cada um deles existe uma série de comandos específicos e/ou um link exclusivo para download dos arquivos necessários. Para este projeto, o intuito é fornecer um ambiente configurado e pronto para uso com menor esforço para instalá-lo. Assim, a fim de tornar os passos únicos para instalação e configuração da rede decidiu-se no primeiro momento pela utilização do Docker, que através de um *script* único criou-se um contêiner Linux Ubuntu em uma versão 19.04 com seus comandos de instalação e configuração já predefinidos através do repositório PPA da Ethereum, já que o sistema operacional e sua versão serão sempre o mesmo, a desvantagem anterior não se aplica a esta abordagem.

A abordagem do Docker, no primeiro momento pareceu eficiente, uma vez que foi possível criar e configurar nós da rede totalmente funcionais, mas para a comunicação de containers em máquinas diferentes até com o mesmo sistema base, são necessárias configurações adicionais de infraestrutura que aumentaram consideravelmente a complexidade do *script* fugindo da ideia inicial de simplicidade na instalação, então decidiu-se procurar outra abordagem.

Mesmo com as diferenças entre os sistemas operacionais anteriormente citados, para a confecção de um novo *script* foi retirado o container Docker e adicionados todos os comandos necessários para criar e configurar o ambiente nos três sistemas operacionais escolhidos, ficando a cargo do *script* primeiramente reconhecer qual o sistema operacional o usuário está utilizando e escolher qual a série de comandos deve ser executada. Para simplificar a quantidade de comandos, a abordagem selecionada foi o download de um arquivo binário que também é escolhido de acordo com o sistema em que for executado.

### **3.4.2 Modelagem dos *smart contracts***

Primeiramente para verificar o funcionamento dos *smart contracts* e a correta configuração da rede *blockchain* privada, contratos simples são criados e integrados a rede para

testes através das próprias ferramentas da plataforma Ethereum. A ferramenta em questão é a mesma que inicia a rede, o Geth, utilizando a conexão gerada pelo Geth ao servidor de aplicação externado pela rede *blockchain*, será possível por linha de comandos interagir com os contratos instanciados dentro da rede.

Antes criar os *smart contracts* que gerenciaram os dados do sistema de acreditação, deve-se modelar quais dados devem ser registrados e como estes serão inseridos e recuperados posteriormente. É necessário então, um alinhamento com a modelagem do protótipo a ser construído, tanto ao primeiro protótipo que servirá para prova de conceito quanto ao protótipo funcional objetivo deste projeto.

No fim, deseja-se um sistema que seja flexível às necessidades de cada instituição que fará uso dele, mas devemos definir um núcleo que seja comum a todas essas organizações. A fim de garantir uma padronização na modelagem dos dados, analisou-se os modelos propostos pelo Schema.org, definido as entidades essenciais resultando em um diagrama de classes.

Para o protótipo inicial a modelagem será centrada em três tipos de objetos: profissionais de saúde, instituições médicas e certificado de acreditação. Para cada um destes objetos será criado um contrato para gerenciar os dados que serão inseridos na *blockchain* levantando as principais funcionalidades modelando-as em diagramas de casos de uso, separando as funcionalidades do sistema web e aquelas que devem ser executadas pelos *smart contracts*.

Com o protótipo inicial finalizado, será observado o funcionamento básico do sistema, através de testes com estudo de caso, validando o conceito da ideia e tecnologias utilizadas. Após os testes e com as experiências tiradas com a utilização deste exemplo levantam-se os requisitos específicos para a construção do sistema web pretendido para este projeto.

### **3.4.3 Integração com sistema web**

Após a aprovação do primeiro protótipo iniciamos a etapa final de integração dos *smart contracts* com a protótipo funcional do Sistema de Acreditação Internacional em Saúde. Durante esta etapa, serão remodelados os contratos para seguir a modelagem final do sistema, redefinindo os campos necessários para armazenagem de dados.

Uma preocupação prevista será a conformidade das versões das ferramentas utilizada, os contratos em Solidity, por exemplo, devem indicar a versão do compilador e esta deve ser compatível com o driver de conexão do sistema JAVA. O alinhamento das versões de ambos deve ser observado para que funcione corretamente a comunicação com o servidor da *blockchain*, mais especificamente a versão do pacote Web3j responsável por criar uma API Java para se comunicar com a rede *blockchain*.

Para a conexão do sistema com a *blockchain*, é necessário compilar os *smart contracts* em classes JAVA. Com os contratos expressos em classes JAVA podemos utilizá-los da mesma maneira que qualquer outro objeto, nas quais as funções do contrato são transformadas em métodos das classes. Essa conversão é realizada com outra ferramenta externa ao kit da plataforma Ethereum, e para este fim utilizaremos a ferramenta web3j-cli para gerar automaticamente as classes JAVA a partir de arquivos binários do Solidity.

Quanto a rede privada não se espera que seja necessárias alterações visto que esta deve ser construída para suportar o suporte com qualquer aplicação, desde que esteja funcional durante a validação nenhuma alteração é esperada.

## 4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos com o desenvolvimento desta pesquisa. Estes resultados foram baseados na estratégia de desenvolvimento indicada no capítulo III. A seguir serão detalhados estes resultados e na sequência apresentadas as considerações gerais com os trabalhos futuros propostos ao fim deste projeto.

### 4.1 ANÁLISE DOS PRODUTOS E RESULTADOS

Os resultados a seguir mesmo formando uma ferramenta única foram divididos em quatro conjuntos a fim de simplificar o entendimento dos produtos desenvolvidos. Em cada subseção detalhamos essas etapas em que foram desenvolvidas, ou aplicadas, cada uma das partes desta ferramenta.

#### 4.1.1 Scripts automatizados para rede *blockchain*

O primeiro produto esperado para este projeto era um meio de reproduzir o processo de criação e configuração da rede *blockchain* a ser utilizada no protótipo funcional do IAS. Como forma de garantir a sua reprodução em qualquer ambiente computacional, foram desenvolvidos *scripts* para a realização desta tarefa.

Vale ressaltar que antes de utilizar este produto, caso o sistema operacional seja o Windows da Microsoft, primeiramente deve-se instalar o Git através da URL <https://git-scm.com/download/win> ou caso utilize o Windows 10 o mais indicado seria ativar o Subsistema do Windows para Linux (WSL) seguindo as instruções oficiais em <https://docs.microsoft.com/pt-br/windows/wsl/install-win10>.

Além do arquivo do *script* o produto ainda conta com os seguintes componentes:

- start.sh (arquivo principal)
- genesis.json (configurações iniciais da rede *blockchain*)
- .accountpassword (senha da carteira de teste)
- .privatekey (chave privada da carteira de teste)

A seguir explicamos as principais funcionalidades de cada um destes componentes:

#### 4.1.1.1 Arquivo Genesis

Para iniciar uma nova cadeia precisamos definir o bloco inicial com algumas configurações que indicaram como novos blocos serão inseridos, dentre estas definições destacamos:

- **config**: a configuração da *blockchain*. Em suas definições temos:
  - “**chainId**”, um identificador utilizado na proteção contra-ataques de repetição. Por exemplo, se uma ação é validada combinando certo valor que depende do ID da cadeia, os atacantes não podem obter facilmente o mesmo valor com um ID diferente.
  - “**ethash**”, indica que o protocolo de consenso é o PoW, pode ser substituído por “**clique**” caso queira que o protocolo de consenso seja o PoA.
- **coinbase**: é um endereço onde todas as recompensas coletadas com a validação de bloco bem-sucedida serão transferidas. Uma recompensa é uma soma do pagamento pela mineração e dos reembolsos da execução de transações de contrato. Como é um bloco de inicial, o seu valor não é relevante. Para todos os próximos blocos, o valor será um endereço definido pelo mineiro que validou esse bloco.
- **difficulty**: dificuldade de mineração, para desenvolvimento e testes defina esse valor baixo para que você não precise esperar muito pelos blocos de mineração.
- **gasLimit**: o limite do custo do gás por bloco.
- **nonce**: é o número de transações enviadas de um determinado endereço. É usado em combinação com **mixhash** para provar que uma quantidade suficiente de computação foi realizada neste bloco.
- **mixHash**: um *hash* de 256 bits que, combinado com o “**nonce**”, prova que uma quantidade suficiente de computação foi realizada no bloco. A combinação de “**nonce**” e **mixhash** deve satisfazer uma condição matemática.
- **parentHash**: é o *hash* do cabeçalho do bloco pai. Familiar a um ponteiro para o bloco pai necessário para formar uma cadeia real de blocos. Um bloco de gênese não possui um bloco pai, portanto, o resultado será apenas neste caso igual a 0.
- **alloc**: esse parâmetro é usado para pré-financiar alguns endereços com *ether* (criptomoeda da rede Ethereum). Ele contém dois parâmetros, o endereço da carteira que deve ser um *hash* de 160 bits e o número de *ether* com o qual uma conta deve ser financiada.

O arquivo Genesis definido para a criação da rede *blockchain* deste trabalho conta com duas contas já pré-financiadas para não ser necessário criar uma conta manualmente e colocá-la para minerar a fim de receber fundos necessários para realizar transações.

#### 4.1.1.2 Arquivo do script

A ferramenta apresenta um script executável nomeado de **start.sh**, o qual é responsável pela implantação dos nós da rede *blockchain*. Para melhor controle de suas

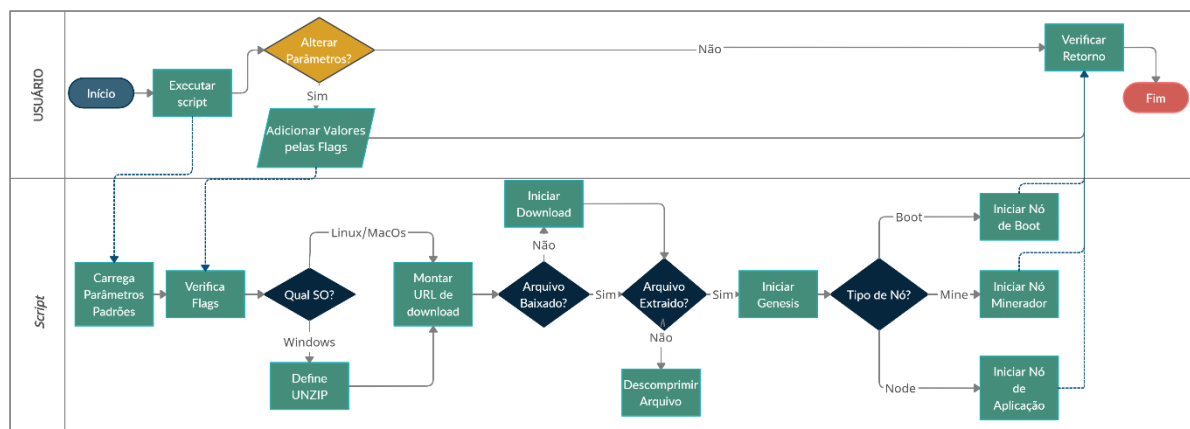
funcionalidades o *script* suporta a criação de três tipos diferentes de nós: o primeiro é nó de **boot** (bootnode), o qual deve ser instanciado apenas uma única vez e apenas em uma máquina, e o segundo é o nó de **aplicação** que deverá ser iniciado em pelo menos uma máquina para ser a interface de acesso das aplicações web com a rede *blockchain* e o último o nó **minerador** responsável pela mineração dos novos blocos a serem inseridos na *blockchain*, para este nó espera-se que seja iniciado em todas as organizações médicas que façam parte do IAS.

As tarefas dos nós foram divididas para melhor observar as funcionalidades e tarefas executadas por cada um, de forma a tentar se aproximar de uma rede de múltiplas máquinas bem como veríamos com a rede em produção.

O processo executado pelo *script* para iniciar cada nó é basicamente o mesmo, com as diferenças apenas nas configurações necessárias para especialização de cada nó.

A Figura 14 apresenta o fluxograma dos processos executados pelo usuário e pelo *script* ao iniciar cada um dos nós componentes da rede *blockchain*.

Figura 14 - Fluxo geral dos scripts para iniciar um nó.



Fonte: Elaborado pelo autor (2021).

Para a execução dos nós é necessária a definição de alguns parâmetros referentes à conexão da rede. Todos os parâmetros estão definidos nas primeiras linhas do *script* que podem ser editados dentro do arquivo ou alterados durante a execução com a utilização de flags suportadas pelo *script*, os parâmetros, seguidos da flag para alterar seu valor, são:

- **NODETYPE (-t):** Indica qual o tipo de nó que deve ser iniciado, **boot** para nó de boot, **node** para nó de aplicação e **mine** para nó de mineração. Por padrão está definido como **node**.
- **VERSION (-v):** Versão do arquivo binário do Ethereum a ser instalado.
- **NETWORKID (-n):** Deve ser o mesmo valor do "chainId" presente no arquivo genesis, valores diferentes nos arquivos interferem na conexão dos nós.

- **DATADIR (-d):** Pasta no computador em que os arquivos da rede serão armazenados. Por padrão: `$PWD/$TYPE` (pasta\_de\_trabalho\_atual/tipo\_do\_nó), dessa forma se estiver sendo executado dentro da pasta `"/home/user"` o **DATADIR** seria `"/home/user/boot"`.
- **BOOTNODEKEY (-k):** Um nó de inicialização pede uma chave hexadecimal e através dela será gerado um ID para conexão de outros nós, deixamos esse valor pré-definido para podermos ter certeza da URL de conexão que será utilizado pelos demais nós. Esse valor pode ser gerado pelo comando: `bootnode -genkey bootnode.key`.
- **BOOTNODEIP (-i):** O IP da máquina em que será instanciado o nó de Boot.
- **BOOTNODEPORT (-p):** A porta em que o nó de Boot deverá expor à rede. Por padrão **30301**.
- **BOOTNODEID (-b):** Deve ser o id ("enode") gerado pela execução do nó de Boot, se não foi alterado o parâmetro **BOOTNODEKEY** este valor já está configurado.
- **MYNODEPORT (-m):** Porta em que será executada a rede no computador que está iniciando o nó. Por padrão: **30303**.

Dessa forma, podemos executar o script na máquina somente indicando qual o tipo de nó queremos iniciar.

Ao executar o **start.sh** por linha de comando, caso não sejam passados nenhum argumento, a rede será instanciada com todos os parâmetros padrões, dentre eles o que pode inviabilizar a utilização da rede, caso incorreto, é o IP da máquina de boot, logo devemos nos certificar que este parâmetro foi definido corretamente.

Para iniciar os nós podemos executar o *script* por linha de comando, como por exemplo:

- `./start.sh -t boot`, para iniciar nó de boot
- `./start.sh -t mine`, para iniciar nó minerador
- `./start.sh -t node`, para iniciar nó de aplicação

Adicionalmente podemos utilizar as outras flags para alterar os parâmetros, como por exemplo:

- `./start.sh -t node -i 192.168.0.155`, para iniciar nó de aplicação e alterar o parâmetro do IP do nó de boot
- `./start.sh -t mine -d $HOME/.ethereum/private/mine -m 30306`, para iniciar nó minerador alterando pasta dos arquivos da rede e a porta de execução do nó.

Depois de definidos os parâmetros, o *script* identifica qual o sistema operacional que está sendo utilizado e seleciona os comandos adequados para baixar, descompactar e executar a rede.

Em seguida, com o conjunto de ferramentas pronto para ser utilizado, a próxima ação do *script* é a criação do bloco inicial da rede, já que este bloco deve ser igual em todos os nós para que estes possam se sincronizar, essa ação é realizada por todos os tipos de nó.

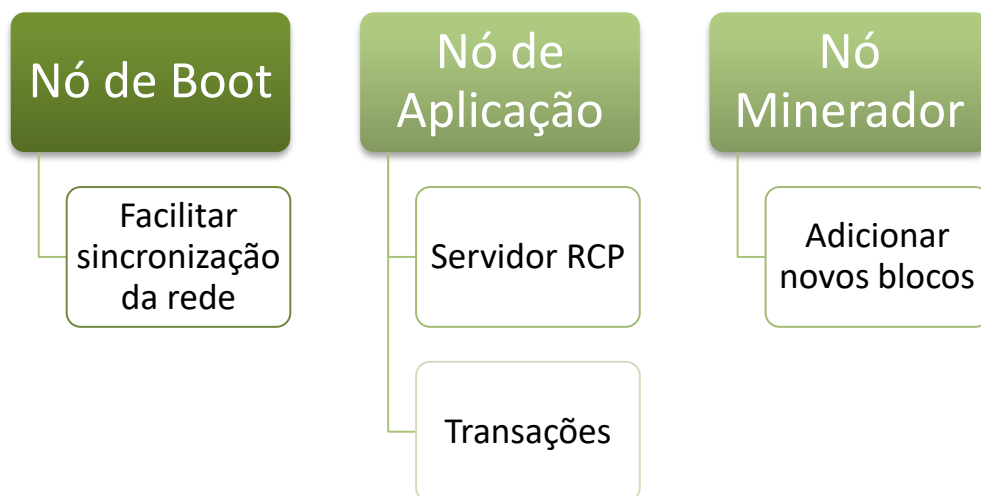
Após confirmada a existência do bloco inicial temos a primeira diferença entre o fluxo dos diferentes nós. Para os nós de aplicação e minerador, devemos criar pelo menos uma conta inicial que realizará as primeiras transações. Assim, o *script* verifica se já há alguma conta criada na máquina e caso contrário importa uma conta já criada anteriormente, cujo *hash* da carteira já conhecemos. Neste momento utilizamos uma conta já existente pois já queremos que esta conta tenha *ether* em sua carteira para financiar as transações que serão executadas sem a necessidade de esperar pelos rendimentos da mineração.

Por fim, o último passo é selecionar o comando para iniciar o nó da rede através do tipo definido nos parâmetros. Sendo assim, cada tipo de nó tem seus argumentos específicos que definem o seu funcionamento.

#### 4.1.1.3 Definição das tarefas de cada nó

Um único nó da rede *blockchain* poderia assumir todas as tarefas que foram divididas entre os três tipos de nós criados pelo *script*. Porém, a divisão destas funcionalidades auxiliou na automação do processo de criação e configuração da rede com múltiplos nós. De acordo com Figura 15 podemos entender melhor a função de cada um.

Figura 15 - Funções de cada nó.



Fonte: Elaborado pelo autor (2021).

Primeiramente para um nó de **boot** a funcionalidade principal é definir um link de conexão, o chamado “enode”, previamente conhecido para automação da conexão de novos nós



na mesma rede. Vale ressaltar que cada nó da rede *blockchain*, mesmo que não seja um nó de **boot**, é identificado por um “enode” e este é utilizado em conjunto com o parâmetro **networkid** para garantir a correta comunicação entre os pares. Esse último, garante ao nó que ao receber uma mensagem de seu par, ele encerrará as conexões em caso de incompatibilidade.

Este nó de **boot** não é obrigatório para o funcionamento da rede, mas para que múltiplos nós possam operar como um sistema distribuído pelo qual a rede *blockchain* se propõem será necessário que, após a criação do primeiro nó, seja alterado o endereço do “enode” nos scripts ou que a conexão seja feita manualmente pelo console da ferramenta Geth.

Já o que define que um nó será de **aplicação** é a ativação do servidor de Chamada Remota de Procedimento (RPC, acrônimo de Remote Procedure Call). Como definido por Wang e Xie (2002), o RPC é um protocolo em que um programa pode usar para solicitar um serviço de um programa localizado em outro computador da rede. Esse servidor de RPC quando ativado utiliza uma API controlada por acesso, de modo que somente usuários conhecidos possam interagir com o servidor. O script assume a responsabilidade pela configuração da API, definindo o endereço da porta, o domínio que pode acessá-la e as funcionalidades que estarão disponíveis para interação com o servidor RPC.

Para realizar transações dentro da rede, funcionalidade esperada para o nó de **aplicação**, estas precisam ser iniciadas por uma conta financiadora, ou seja, uma carteira tenha *ether* suficiente para pagar a mineração, e por padrão a cada solicitação de transação é necessária a autorização mediante senha. Como queremos realizar testes de automação do processo de configuração e funcionalidade da rede, a conta pré-existente é desbloqueada para transações para que não seja necessário fazer a autorização mediante senha de toda transação. Este desbloqueio deve ser suprimido ao executar a rede em produção.

Ao executar o nó de **aplicação**, o esperado é a saída “HTTP server started” indicando que o servidor HTTP foi ativado e qual é o endereço de acesso configurado para a API, característica existente apenas nesse tipo de nó.

Por sua vez, o nó **minerador** tem como característica principal o início automático da mineração, mas para que se inicie devemos deixar indicado uma das carteiras já criadas para que receba as recompensas pelo trabalho de mineração. Podemos também definir o número de *threads* que queremos que sejam executadas de forma paralela, indicando o número total de núcleos de processador que serão utilizados nessa tarefa, por padrão deixamos apenas duas *threads*. A saída esperada da execução de um nó minerado tem como característica o início do

trabalho de mineração indicado pela saída “Commit new mining work”. Uma característica que podemos notar nesse momento é que o processo de mineração, mesmo sem receber novas transações, permanece criando blocos continuamente mesmo que vazios.

Após a criação de mais de um nó, para verificarmos se eles estão funcionando em conjunto como esperada de uma rede distribuída, podemos observar os logs de saída dos nós de boot ou aplicação se estes estão importando novos blocos minerados pelo nó minerador indicado pelo registro “Imported new chain segment”.

#### **4.1.1.4 Roteiro de Instalação**

Foram criados um manual de utilização desta ferramenta e vídeo tutorial demonstrando sua utilização, ambos estão disponíveis no Apêndice 6 – Código fonte do script de automação da rede *blockchain* Apêndice 6.

Esse material desenvolvido foi transformado em artigo intitulado “Scripts de Instalação de uma rede *blockchain* como Recurso Didático para Metodologias Ativas de Ensino de Computação” submetido e aprovado na EduComp 2021. Neste artigo, descrevemos o processo de criação e utilização dos scripts e propomos a sua utilização em sala de aulas para o ensino da tecnologia *blockchain* e das diversas disciplinas correlatas utilizadas para sua criação e configuração.

Podemos citar como exemplo de outras disciplinas em que poderia ser abordado essa parte do trabalho, uma aula de Segurança em Tecnologia da Informação, na qual seria interessante ser apresentada a criptografia empregada na rede *blockchain* como uma técnica de proteção para comunicação segura, ou aulas sobre Banco de Dados, pode se fazer um paralelo entre as duas tecnologias para indicar as diferenças e em que situação devemos utilizar cada uma dessas tecnologias.

Ainda nesse sentido, o script como um todo é um bom exemplo de algoritmo podendo ser utilizado em aulas como Introdução a Programação e Algoritmos. As verificações do sistema operacional, se o download ou descompressão do arquivo já foram executadas podem demonstrar o funcionamento de estruturas de seleção.

Em aulas de Sistemas Operacionais fazendo uso do script pode-se abordar chamadas de sistema, explicar o que são processos, seus estados, execução em primeiro e segundo plano e o que os diferencia dos programas. Detalhes como o redirecionamento de

portas, o servidor HTTP do nó de aplicação e as permissões de acesso à API da *blockchain* poderão também ser utilizados nas disciplinas que abordam configurações de redes.

#### 4.1.2 Script para compilação dos *smart contracts*

O desenvolvimento do protótipo do IAS prevê a interação com a rede *blockchain*, essa interação dar-se-á pela utilização dos *smart contracts*. Seguindo o modelo de desenvolvimento dos scripts para criação e configuração da rede *blockchain* privada, procurou-se criar outro roteiro com base em *script* para manipulação de um *smart contract*. Assim, foram adicionados ao projeto dos *scripts*:

- um exemplo de contrato, arquivo **Profissional.sol**, escrito na linguagem Solidity, que tem como objetivo guardar e recuperar informações de profissionais de saúde;
- um *script*, arquivo **contract.sh**, para compilar o contrato e prepará-lo para implantação na *blockchain*;
- e um roteiro com explicações de como utilizar o script e interagir com o contrato já implantado na rede.

Os novos arquivos foram colocados dentro da pasta **solidity** do repositório mencionado no tópico anterior.

##### 4.1.2.1 *Smart contract de teste*

Para o teste dos *smart contracts* foi desenvolvido o contrato **Profissional.sol** para armazenar as informações de um profissional de saúde. Ao desenvolver um contrato em Solidity uma recomendação feita pelo Solidity Compiler é o uso da identificação de licença seguindo o padrão SPDX. Cada arquivo deve começar com um comentário indicando sua licença, como por exemplo: “SPDX-License-Identifier: GPL-3.0”. A não definição da licença gera um aviso indicando a sua falta durante a compilação do contrato.

Após essa definição, deve-se especificar para qual versão do Solidity o código-fonte foi escrito, no nosso exemplo definimos o código compatível com as versões de 0.4.22 até a versão mais recente da linguagem 0.8.1 (versão atual no momento da escrita deste). Caso seja indicada uma versão incompatível com a versão do compilador utilizado, será gerado um erro e a compilação irá falhar.

Um contrato escrito em Solidity é uma coleção de código (suas funções) e dados (seu estado) que reside em um endereço específico na *blockchain* da plataforma Ethereum. Declaramos então o nome do contrato depois da palavra **contract**, de forma parecida a uma

classe JAVA. Dentro do corpo do **contract** declaramos as variáveis que receberam os dados enviados ao contrato, nesse caso os dados do profissional a ser registrado e posteriormente suas funções. Neste exemplo, o contrato define as funções e que podem ser utilizadas para recuperar ou modificar o valor da variável, funções **getProfessionalDetails** e **setProfessionalDetails**.

#### 4.1.2.2 Compilando os contratos

A fim de demonstrar o processo de compilação dos contratos pelo Solidity Compiler, foi criado o script **contract.sh**. Como no script de instalação, este arquivo contém alguns parâmetros:

- **GETHPATH**, esse deve informar a localização da ferramenta Geth, pode ser deixada em branco se o esquema de arquivos dessa ferramenta não for alterado.
- **VERSIONGETH**, esse parâmetro deve ser o mesmo que o utilizado nos scripts para criação da rede.
- **SOLCVERSION**, define a versão do compilador da linguagem solidity. Padrão: **'v0.8.1'**
- **CONTRACT**, define o arquivo do contrato inteligente a ser compilado. Padrão: **Professional.sol**

Para executar este script e compilar o contrato inteligente digite no terminal dentro da pasta **solidity** do projeto. Este script primeiramente verifica o sistema operacional para montar a URL do compilador de contratos Solidity e dá permissão de execução a este arquivo após o download ser concluído.

Após o download o contrato é compilado gerando dois arquivos. O primeiro é um arquivo binário simples com extensão “.bin” e o segundo um arquivo mais complexo, de extensão “.abi”, definido como Application Binary Interface (ABI), que segundo Ethereum.org (2021), é a maneira padrão de interagir com contratos no ecossistema Ethereum, tanto de fora da *blockchain* quanto para interação de contrato a contrato. Os dados são codificados de acordo com seu tipo, conforme descrito nesta especificação. A codificação não é auto-descritiva e, portanto, requer um esquema para ser decodificada.

Para importá-los para a *blockchain*, estes precisaram ser transformados em variáveis JavaScript. Dessa forma, o *script*, então lê o conteúdo destes arquivos compilados e cria variáveis JavaScript que podem ser importadas no console da ferramenta Geth. Caso os parâmetros da ferramenta Geth estejam corretamente definidos no *script* o console é iniciado, caso contrário utilize a ferramenta Geth do seu computador para acessar o console através do comando: **“./geth attach http://localhost:8545”**.

A partir desse momento o script não realiza mais nenhuma ação, dessa forma seguiremos o roteiro criado para exemplificar a manipulação dos contratos dentro da rede.

#### 4.1.2.3 Roteiro de interação com o *smart contract*

O roteiro de interação com o *smart contract*, foi desenvolvido para testar o funcionamento da característica de lógica computacional que a rede *blockchain* proporciona. Para isso é apresentando um passo-a-passo de como implantar e interagir com *smart contract*, sem a necessidade de desenvolver ou instalar ou interface adicional.

Inicialmente é indicado como carregar os arquivos compilados dentro da API conectada pelo Geth, seguindo dos comandos para criar uma variável com o conteúdo compilado do contrato no formato aceito pela rede.

Orienta-se as definições do valor *gas* necessário para o *deploy* do contrato e da conta padrão para realizar as transações, utilizando para este fim os comandos nativos da rede **eth.estimateGas** e **eth.defaultAccount**, respectivamente.

Para criar um objeto JavaScript com a interface de um contrato da plataforma Ethereum, utilizamos a função **eth.contract**, permitindo que seja possível interagir com os *smart contracts* como se fossem objetos JavaScript.

Com a interface anterior, podemos de fato implantar uma nova instância do contrato na rede *blockchain*. Para essa tarefa utilizamos a função **new** do objeto JavaScript que criamos, passando como argumento a variável com o contrato compilado e a quantidade de *gas* que foi estimada. Para não precisarmos carregar o contrato pelo seu endereço posteriormente, orienta-se salvar em uma variável, a referência do contrato retornada pela função.

Toda transação enviada para a *blockchain*, só é efetivada após a mineração, neste momento então, devemos aguardar o término da mineração do novo contrato. Após mineração ser realizada, podemos utilizar todas as funções definidas no contrato, para nosso exemplo executamos a função **setProfessionalDetails** para registrar os dados do profissional no contrato.

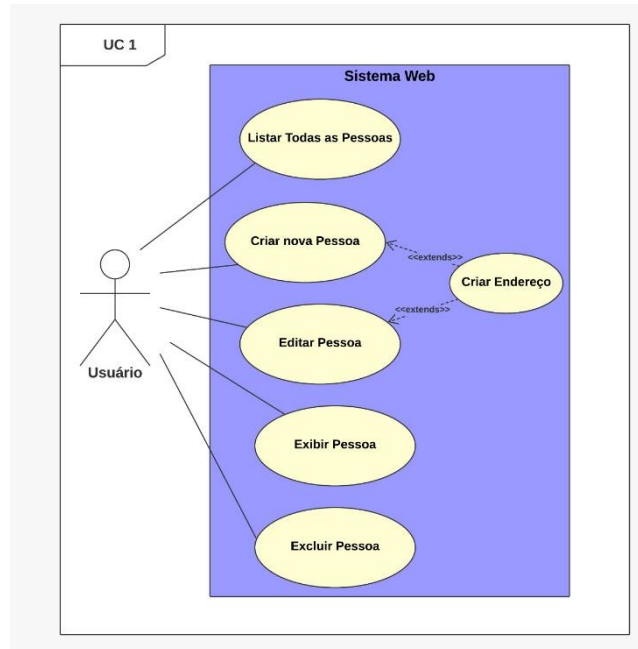
O retorno esperado no console ao executar uma função de um contrato é o *hash* que identifica a transação (**transacionHash**). Outro detalhe importante de se notar é que mesmo os métodos de recuperação de informações não retornam nada na tela além do *hash* da transação. Isso ocorre, porque não é possível, pelo console, obter o valor de retorno de uma função. Para





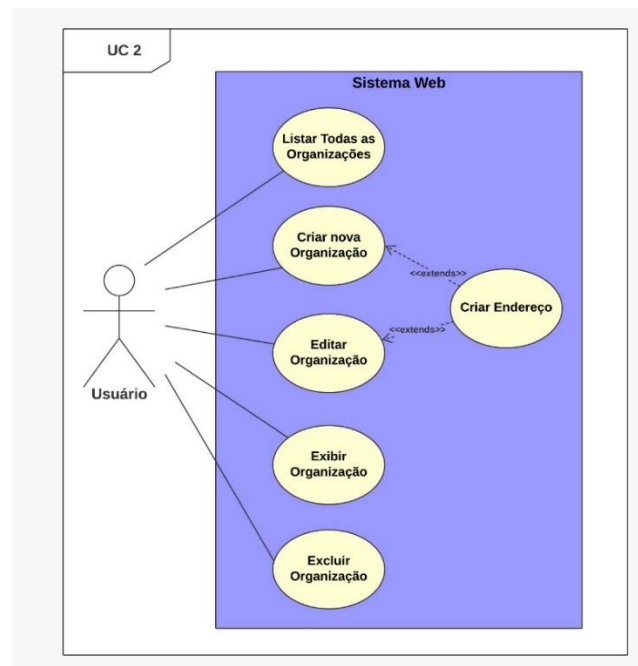
primeiro protótipo as quais foram identificadas as três entidades principais: **CertificadoAcreditação** (referente ao certificado de acreditação); **Organização** (representando as instituições médicas); e **Pessoa** (representando os profissionais que serão certificados). Estas entidades serviram de base para criação de uma primeira versão dos diagramas de casos de uso que podem ser vistos nas Figuras Figura 18, Figura 19 e Figura 20.

Figura 18 - Primeira versão do diagrama de casos de uso de Pessoa.



Fonte: Elaborado pelo autor (2021).

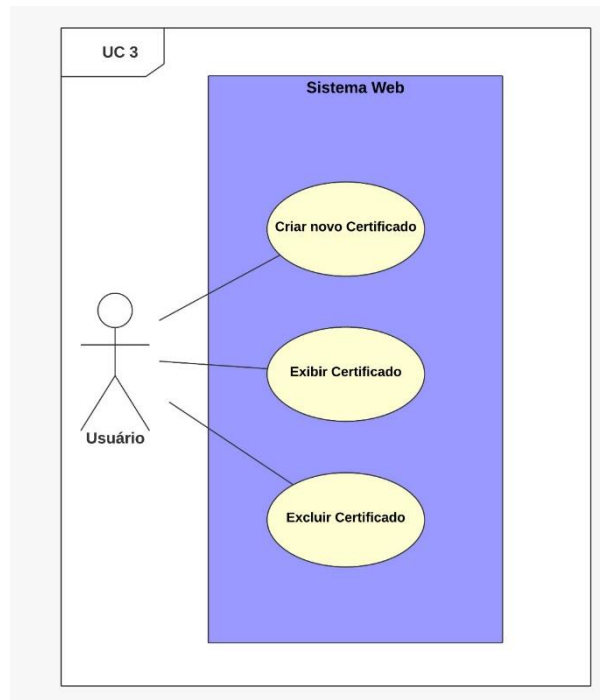
Figura 19 - Primeira versão do diagrama de casos de uso da Organização.



Fonte: Elaborado pelo autor (2021).



Figura 20 - Primeira versão do diagrama de casos de uso do Certificado.

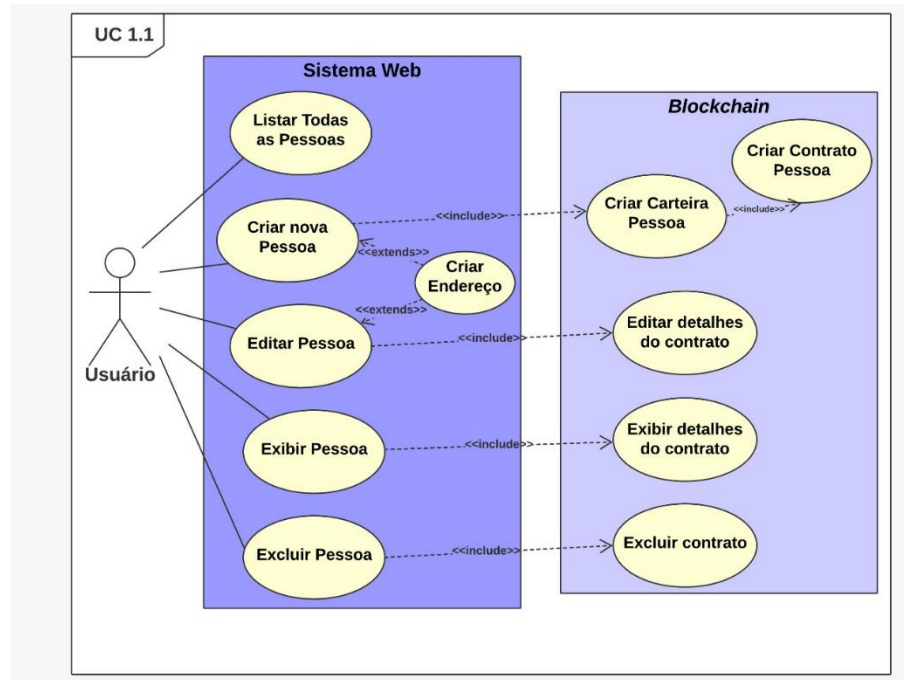


Fonte: Elaborado pelo autor (2021).

Para a versão inicial do protótipo as funcionalidades em que integramos ações à *blockchain* são aquelas que representam o CRUD, acrônimo da expressão do idioma inglês referente às ações de *create* (criação), *read* (consulta), *update* (atualização) e *delete* (exclusão), com exceção da entidade representante da certificação que não possui a opção de atualização dos certificados, caso seja necessária alguma alteração, deve-se realizar a deleção do antigo e criação de um novo.

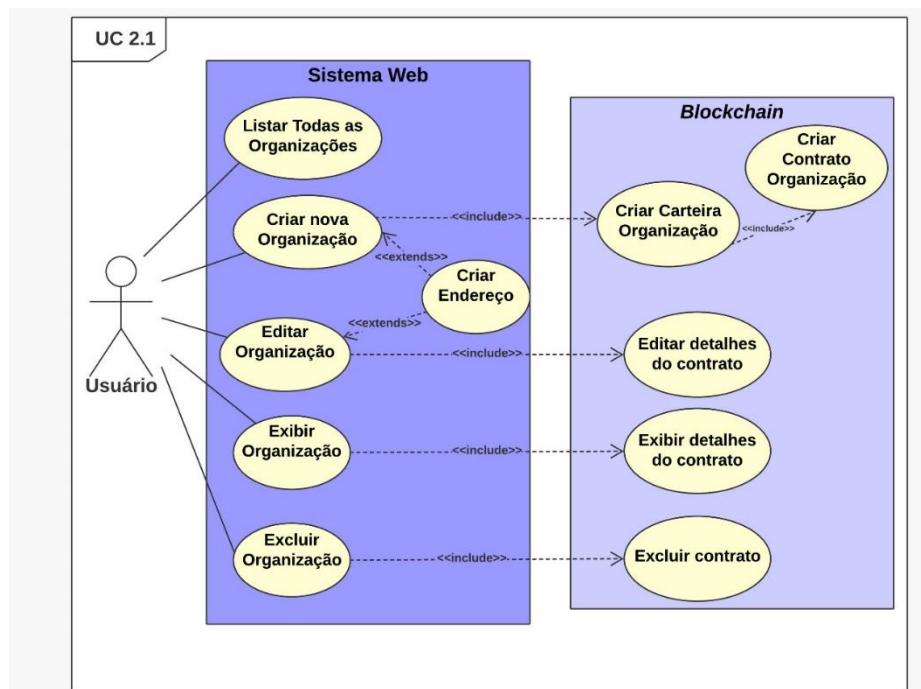
Partindo dessas funcionalidades e após uma análise de como os *smart contracts* iriam apoiá-las, a modelagem de caso de uso foi refeita para adicionar as ações que seriam de responsabilidade dos contratos. Para os diagramas de **Pessoa** e **Organização**, ao se criar um objeto, foi adicionada a ação de gerar novas carteiras sendo executadas diretamente pela interação com a API da *blockchain*, e em seguida, através de um *smart contract*, a criação de uma instancia de contrato para registrar os dados dessas entidades dentro da *blockchain*, disponibilizando funções de recuperação, edição e deleção desses dados. As Figuras Figura 21 e Figura 22 apresentam, respectivamente, o novo diagrama de casos de uso de **Pessoa** e **Organização** com as divisões das funcionalidades referentes ao sistema e à *blockchain*.

Figura 21 - Diagrama de caso de uso da pessoa.



Fonte: Elaborado pelo autor (2021).

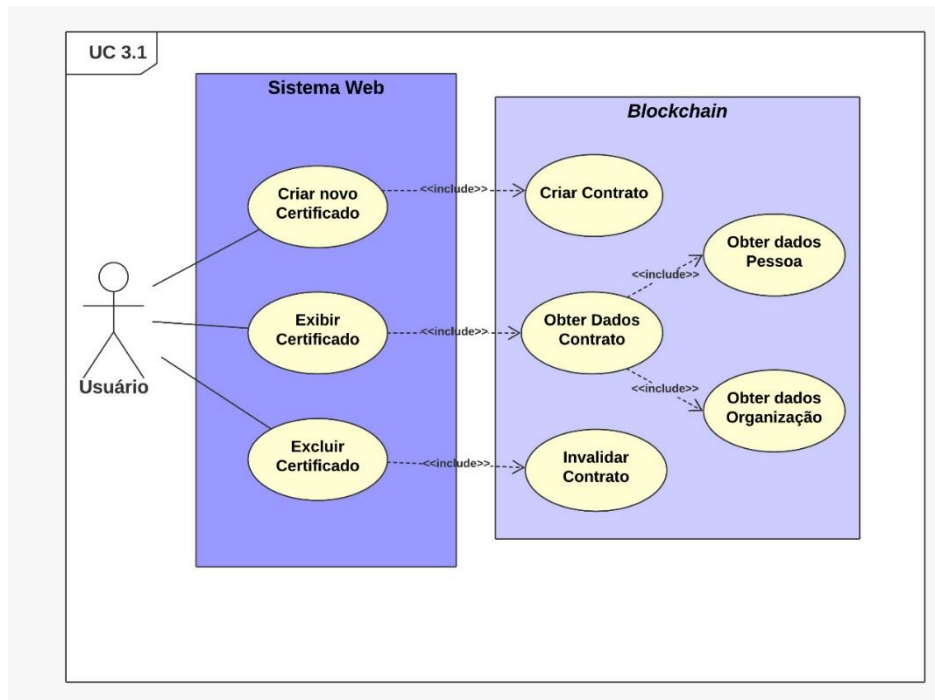
Figura 22 - Diagrama de caso de uso da organização.



Fonte: Elaborado pelo autor (2021).

Já para a entidade **CertificadoAcreditação**, como não foram adicionadas a opção de criação de carteira por não se tratar de um objeto que abstrai uma pessoa, seja física ou jurídica. Dessa forma, como visto na Figura 23, nesse novo diagrama de caso de usos as ações integradas à *blockchain* serão a criação, deleção e consulta.

Figura 23 - Diagrama de caso de uso da certificação de acreditação.



Fonte: Elaborado pelo autor (2021).

Com as funcionalidades definidas passamos ao desenvolvimento dos *smart contracts*. Apresentamos a seguir os aspectos mais importantes de cada um desses contratos e detalhes sobre seu desenvolvimento.

#### 4.1.3.1 Smart Contracts do protótipo inicial

Inicialmente foram propostos para esse primeiro protótipo três contratos, um para cada entidade chave mencionada anteriormente. Cada contrato deveria implementar as funções que foram definidas nos diagramas de casos de uso e registrar os dados levantados pela modelagem, levando em consideração aqueles importantes para identificação e gerar registros de autenticidade dos objetos envolvidos.

Os contratos **Profissional** e **Organização** tem em comum a maioria das variáveis como: nome, cpf, registroMedico, email e telefone. Esses dados em comum são os responsáveis pelo registro dos dados de identificação dos profissionais que serão certificados e organizações participantes do sistema. As variáveis `enderecoHashProfissional` e `enderecoHashOrganizacao` ficam responsáveis por armazenar o endereço *hash* das carteiras criadas para cada um dos objetos.

No caso do contrato **Certificado**, guardamos as informações de identificação do profissional acreditado (cpf), qual a organização que registrou a certificação (reconhecidoPor),

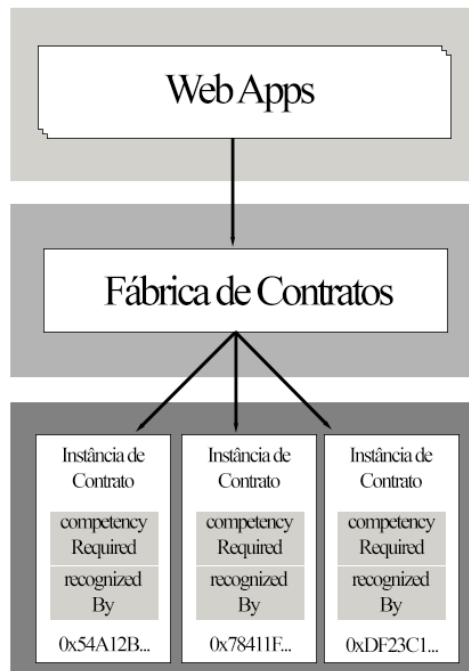
a competência técnica do profissional atestada pela acreditação (`competenciaRequerida`), o título, categoria ou área em que se enquadra a competência atestada (`categoriaCredencial`) e a data de registro do certificado de acreditação (`dataCertificado`).

Quanto as funções de cada um dos contratos temos um construtor que instância o contrato recebendo as informações antes mencionadas e atribuindo esses valores às suas variáveis, a funcionalidade de destruir um contrato, uma função “Get” para recuperação das informações de cada contrato e exclusivamente nos contratos de **Profissional e Organização** a função “Set” para alterar os dados registrados. O contrato **Certificado** não conta com a função “Set” por ser definido que um certificado não pode ser alterado depois de criado, somente invalidado.

No subcapítulo anterior (Script para compilação dos *smart contracts*), foi mencionada a utilização como contrato de exemplo uma versão de “Professional.sol”, durante essa etapa foi observado que a recuperação dos contratos registrados só poderia ser feita mediante o conhecimento do seu *hash* de implantação. Considerando o fato de quando solicitamos a implantação de um contrato ainda não temos conhecimento de qual será seu endereço *hash* isso dificultaria a recuperação do contrato pela aplicação web, que deveria esperar a mineração do contrato ou gravar o *hash* da transação e solicitar esse endereço do contrato posteriormente solicitando primeiramente as informações da transação.

Como há a necessidade de criar e implantar múltiplas instâncias de um determinado contrato, esse processo pode ser muito inconveniente se feito com as abordagens apresentados. Portanto, podemos fazer uso de um padrão de projeto que permita a automação deste processo, criando um contrato que atua como uma “fábrica” de produção de quantas instâncias forem necessárias. Assim, foi definida a estrutura dos novos contratos como exemplificada pela Figura 24.

Figura 24 - Estrutura utilizada para desenvolvimento dos *smart contracts*.



Fonte: Elaborado pelo autor (2021).

Dessa forma, criaremos um contrato fábrica para cada um dos contratos já desenvolvidos que terão especificamente duas funcionalidades:

1. Criar um contrato de seu tipo.
2. Mapear e acompanhar em forma de lista os contratos criados anteriormente.

Com essa abordagem, vamos armazenar a lista de contratos criados em um *mappings* de um endereço identificador, seja ele do profissional ou organização, com um *array* dos contratos criados para esse identificador. Os *mappings* podem ser vistos como tabelas *hash* que são virtualmente inicializadas de forma que todas as chaves possíveis existam e trabalham de forma que cada elemento de sua lista possui uma chave e valor associado, assim podemos realizar uma busca rápida pela chave que desejamos, sem precisar percorrer toda lista ou saber a posição que desejamos consultar.

Esta variável de *mappings* deve ser visualizada publicamente por qualquer pessoa, ou seja, aplicação do *front-end* poderá facilmente recuperar os certificados registrados. A Figura 25 demonstra a implementação de uma fábrica de contratos.

Figura 25 - Fábrica de contratos do tipo Certificado.

```

3 contract CertificadoFactory{
4     mapping(address => Certificado []) enderecosCertificadosRegistrados;
5
6     function createCertificado(address _pessoaResponsavel,
7         string memory _cpf, string memory _categoriaCredencial,
8         string memory _reconhecidoPor, string memory _competenciaRequerida, uint _date) public {
9
10        Certificado newCertificado = new Certificado(msg.sender, _pessoaResponsavel,
11            _cpf, _categoriaCredencial, _reconhecidoPor, _competenciaRequerida, _date);
12
13        enderecosCertificadosRegistrados[_enderecoHashProfissional].push(newCertificado);
14    }
15
16    function getDeployedCertificadoByAddress(address _enderecoHashProfissional)
17    public view returns (Certificado[] memory) {
18        return enderecosCertificadosRegistrados[_enderecoHashProfissional];
19    }
20 }

```

Fonte: Elaborado pelo autor (2021).

Pelo contrato **CertificadoFactory** temos através do *mapping* `enderecosCertificadosRegistrados` todos os certificados registrados em nome de um determinado profissional, pois a cada contrato do tipo **Certificado** criado por essa fábrica, a sua referência é adicionada ao *array* indexado pelo *hash* da carteira do profissional acreditado.

#### 4.1.3.2 Interação com o protótipo web inicial

Para integração com o sistema web do protótipo inicial, utilizou-se do script de compilação de contrato apresentado anteriormente neste trabalho que gerou os arquivos BIN e ABI de cada um dos seis contratos desenvolvidos.

No entanto para interação com um sistema JAVA esses contratos devem ainda ser convertidos em classe JAVA. Partindo dos arquivos compilados e através biblioteca Web3J, escolhida para comunicação com API da *blockchain* no trabalho de Cavalcante (2021), dentro da pasta contendo os contratos compilados podemos utilizar o seguinte comando “web3j generate solidity” passando como argumentos o arquivo ABI (flag -a), o arquivo BIN (flag -b), o diretório de destino das classes JAVA (flag -o) e o nome do pacote em que serão colocadas (flag -p).

Em posse das classes geradas a biblioteca Web3J pode implantar os contratos e recuperar suas informações. Na Figura 26, temos a tela do sistema que retorna os dados do *smart contract* profissional após ser enviado para implantação na rede *blockchain*, dentre os dados temos a carteira do Profissional que será o “dono” do contrato o *hash* da transação que solicitou

a criação do contrato na rede e caso já tenha sido concluído o processo de implantação teremos o *hash* que identifica o contrato.

Figura 26 - Tela do apresentando os dados da criação do contrato.



Fonte: Elaborado pelo autor (2021).

Pela Figura 27, podemos observar o log de atividades do nó de aplicação, dentre as quais podemos observar a submissão da transação que criou o contrato da Figura 26 indicada pelo mesmo *hash* de transação presente na aplicação.

Figura 27 - Transação de implantação enviada pela aplicação WEB.



Fonte: Elaborado pelo autor (2021).

Detalhes da configuração da biblioteca Web3j bem como de sua utilização no sistema podem ser encontradas em Cavalcante (2021).

#### 4.1.4 Versão final dos *smart contracts*

Analisando o funcionamento dos *smart contracts* no protótipo inicial, foram levantadas algumas melhorias a serem realizadas como:

1. **Simplificar o envio de dados:** Como a primeira versão dos contratos era apenas um teste de seu funcionamento não foi levado em consideração questões do tamanho do contrato e o custo para implantá-lo, como o objetivo dos contratos não é ser um centro de armazenagem de dados e sim uma forma de garantir a autenticidade deles, podemos remover do envio de informações os dados que não servirão a esse fim, como telefone e e-mail dos profissionais e organizações.
2. **Alterar trava de criação de contratos “duplicados”:** Foi colocada uma trava para não criar contratos repetidos, mas não estava sendo verificado se o contrato foi invalidado para possibilitar a criação de um novo com a mesma chave.
3. **Contrato Profissional poder criar automaticamente um Certificado:** mesmo sendo uma *feature* opcional estava previsto como uma possível finalidade a acreditação por meio de profissionais que tiveram experiências conjunta com outros que possam atestar a sua competência em alguma área.

O foco dessa nova rodada de desenvolvimento dos *smart contract* é o contrato Certificado e seu relacionamento com o contrato Profissional, enquanto o contrato Organização restringindo a armazenagem de dados a apenas as informações necessárias para a criação dos certificados de acreditação. A seguir discutiremos as versões finais de cada contrato e como estes interagem com o protótipo final do IAS.

##### 4.1.4.1 Alterações nos *smart contracts*

O *smart contract* referente a organização armazenava todas as informações da organização, visto que essa abordagem aumenta o custo para implantá-lo na *blockchain* e como essas informações não eram utilizadas pela rede, optou-se por limitar a informações enviadas apenas ao nome, CNPJ e o identificador da instituição de ensino.

Devido ao fato de um *smart contract* ainda ser localizado mesmo após destruído, foi adicionado também um campo nomeado como “valido” para indicar se o *smart contract* ainda está operacional ou não.

Para o *smart contract* **Profissional**, as mesmas mudanças do *smart contract* Organização foram aplicados, alterando apenas o campo CNPJ para CPF. No entanto, a este contrato foi adicionado uma variável que guarda uma instancia de uma fábrica de contratos do tipo Certificado (**CertificadoFactory**).



Com o **CertificadoFactory** dentro do *smart contract* **Profissional** torna-se possível a interação entre os contratos. Dessa forma, como previsto nas perspectivas esperadas para o IAS no artigo de Souza Junior et al. (2019), foi adicionada a acreditação entre pares de forma que não seja obrigatória ser utilizada, mas demonstre como esta funcionalidade poderia ser utilizada.

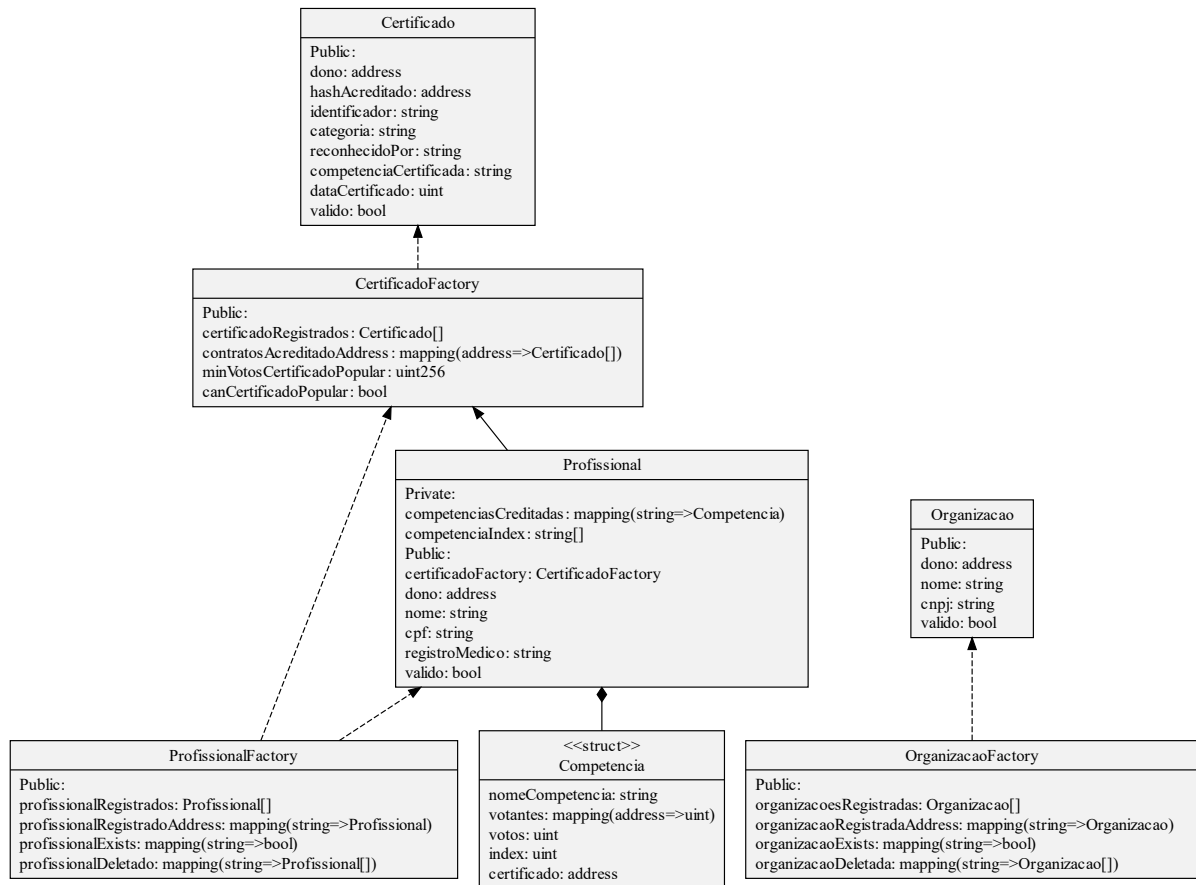
Para a acreditação entre pares cada *smart contract* **Profissional**, que representa um profissional A, teria uma lista de habilidades em que outro profissional B poderia atestar, através de um “voto” de confiança, que participou de ações conjuntas com o profissional A e este apresentou habilidades condizentes a um profissional acreditado. Caberia então ao conjunto de instituições definir quantos desses “votos” de confiança um profissional deveria receber em uma determinada área para ser considerado apto a receber uma acreditação, e quais profissionais acreditados poderiam realizar a ação de conceder esses “votos”.

Assim, ao receber a quantidade necessárias de recomendações o contrato **Profissional** criaria automaticamente um contrato **Certificado**, que se diferenciaria dos criados via sistema apenas pelo órgão acreditador que não seria uma instituição médica registrada e sim o próprio sistema IAS possibilitando indicar que a acreditação foi cedida por indicação profissional e não por avaliação pelas instituições. Dessa forma, a adição dessa funcionalidade, mesmo que seja opcional, explora melhor o potencial dos *smart contracts*, apresentando como estes podem ser auto executados.

O *smart contract* **Certificado** também recebeu a variável que indica que ele está válido mantendo todos os outros atributos anteriores, identificação do profissional, identificação da instituição acreditadora, área acreditada e data da certificação.

Com a possibilidade da acreditação entre pares adicionada no *smart contract* **Profissional**, temos o campo categoria que recebe o valor “PARES” para indicar quando o certificado é concedido por essa nova forma. E para indicar se a acreditação entre pares estará disponível temos uma variável iniciada com o valor “FALSE” que por padrão inativa esse tipo de acreditação, mas poderá ser alterada a qualquer momento pelo IAS.

A sua fábrica de contratos adicionamos também uma variável que controla a quantidade de “votos” que uma acreditação entre pares precisa para gerar um certificado automaticamente, este valor também poderá ser alterado pelo IAS. A visão completa dos *smart contracts* em formato de diagrama de classes pode ser observada na Figura 28.

Figura 28 - Modelagem dos *smart contracts* em diagrama de classes.

Fonte: Elaborado pelo autor (2021).

## 4.2 TESTES DO *BACK-END* COM O PROTÓTIPO FUNCIONAL

Para avaliar se o *back-end* está configurado e se comunica corretamente com o *front-end*, comprovando que o protótipo funcional possibilita a manutenção de processos de acreditação em saúde, realizou-se dois testes.

No primeiro, os *scripts* de automação da rede *blockchain* serão empregados em três máquinas, com acesso público pela internet, representando cada um dos três tipos de nós que o *script* pode criar. Verificando assim, se os três nós são iniciados corretamente e conseguem se comunicar entre si através da internet.

No segundo teste verificamos a integração da rede *blockchain* por meio dos *smart contracts* inseridos no protótipo funcional. O teste de *smart contracts* é um ponto crítico no desenvolvimento de qualquer aplicação que os integrem. Uma vez implantado em uma *blockchain*, esta instância do *smart contract* torna-se imutável. (DESTEFANIS, MARCHESI,

*et al.*, 2018). Se um contrato é implantado com algum erro, não poderá ser alterado posteriormente necessitando uma nova publicação para corrigi-lo. Porém, mesmo corrigido o erro, o contrato anterior ainda estará na *blockchain* e todas as transações realizadas por ele não serão replicadas. Sendo assim, o processo de teste dos *smart contracts* podem trazer grandes benefícios. Dessa forma, este teste avalia se os contratos são criados corretamente e se as funcionalidades retornam os resultados esperados.

A seguir serão detalhados como cada um dos testes foram realizados.

#### 4.2.1 Teste em ambiente de homologação do *script* de automação

Esse primeiro teste tem o objetivo de testar a conectividade dos diferentes nós criados pelo *script*, avaliando se estes conseguem se comunicar corretamente, mesmo quando não estão em uma única rede, através da internet simulando como seria utilizado em uma aplicação em produção. Para este teste, foi criado um ambiente de homologação composto por duas máquinas virtuais e uma máquina física. As máquinas virtuais foram criadas a partir da plataforma Microsoft Azure<sup>1</sup> e para a máquina física foi utilizado um notebook com as especificações descritas na Tabela III:

Tabela III - Máquinas utilizadas no teste do *script* de automação.

Identificação da máquina	Sistema Operacional	Quantidades de CPU's	Memória RAM (GB)	Armazenamento (GB)
Azure B1s	Ubuntu Server 20.04	1	1	4
Azure B1ms	Ubuntu Server 20.04	1	2	4
Dell Inspiron I14-5448-C10	Windows 10	4	8	500

Fonte: Elaborado pelo autor (2021).

As funções dessas máquinas foram distribuídas da seguinte maneira:

1. Azure B1s rodará o *script* se tornando um nó de boot.
2. Azure B1ms ficará com a responsabilidade do nó de aplicação.
3. E o notebook Dell Inspiron I14-5448-C10 por ser a máquina com melhor configuração ficará em cargo do nó minerador.

---

<sup>1</sup> O Microsoft Azure é uma plataforma destinada à execução de aplicativos e serviços, baseada nos conceitos da computação em nuvem.

Primeiramente para iniciarmos os testes, as máquinas virtuais foram criadas e como por padrão na plataforma Azure todas as portas da máquina estão bloqueadas para acesso externo, fazendo necessária a liberação de algumas portas pelo painel de administração da plataforma para que a rede *blockchain* possa se comunicar com a internet, que por padrão são as portas 30301 para a máquina com o nó de boot e 8545 para a máquina com o nó de aplicação. Em seguida foram iniciados em ordem, o nó de boot na B1s e o nó de aplicação na B1ms, a partir desse ponto espera-se que as duas máquinas possam se comunicar e formar um par da rede *blockchain*.

A verificação desse comportamento pode ser conferida através do console da ferramenta Geth que ao se conectar com o nó de aplicação pode solicitar todas as conexões realizadas pela máquina armazenada na variável “admin.peers”. A Figura 29 apresenta o retorno esperado por essa ação na qual podemos notar que o “enode” indicado para o nó que está conectado é o mesmo que o definido para o nó de boot no *script* e o IP público da máquina virtual a qual ela se conecta indicado na variável “remoteAddress”.

Figura 29 - Pares conectados na rede *blockchain*.

```
> admin.peers
[[
  {
    caps: ["eth/63", "eth/64", "eth/65"],
    enode: "enode://4e87faaa0ed677c3ec389f3ac37f8b0e366876f73e72764e3518031daca322768befb783be5c4aea4200f3439f4361571e860c38776142094adc35913964096
59dk416AMdrKMidon0V3CCd12DdWRwgnZd",
    enr: "--Je4QMTLzRVRh3d6Mnghd09iyupL0Lct3xpC_nRWqV-cgq5wb_Jr3QEpl_dgB7_52-Wtte1M6TzLCfwTbFqgILgV0sBg2V0aMf6h65NKqKAgmLkgnY0gmLwhBTJeleJc2Vjc
id: "d36f741bdee84ec72d9407e4cbb09c3c89a6a9315d31e472c69e86fd2bcfa2f7",
    name: "Geth/v1.9.20-stable-979fc968/linux-amd64/go1.15",
    network: {
      inbound: false,
      localAddress: "10.0.0.6:55718",
      remoteAddress: "20.201.122.87:30301",
      static: false,
      trusted: false
    },
    protocols: {
      eth: {
        difficulty: 3551330734,
        head: "0x2da46e5cc1271c3553d6784e20337812cfd64cbb359cc37c3fc307fd35ba4cc",
        version: 65
      }
    }
  }
]]
```

Fonte: Elaborado pelo autor (2021).

A seguir devemos iniciar um nó de mineração que deve ser capaz de sincronizar os blocos minerados com os demais nós. Ao iniciar este nó esperamos pelo começo do processo de mineração para verificarmos se os demais nós está se comunicando corretamente pela internet. Essa verificação pode ser facilmente confirmada pelo *log* dos demais nós que devem começar a importar os blocos minerados apresentando a seguinte informação: “Imported new chain segment”.

A utilização de uma máquina física foi pensada em testar a conexão dos nós fora da rede da Azure, indicando que mesmo em redes diferentes a comunicação entre os nós ocorreria

perfeitamente. Finalizada essa verificação, para garantir o funcionamento contínuo da rede *blockchain* sem depender do notebook utilizado para mineração foi adicionada mais uma máquina virtual do tipo B1ms para fazer o papel do nó minerador, importante ressaltar que podemos ter mais de um nó do mesmo tipo, exceto nó de boot. A adição do segundo nó de mineração não ocasionou nenhum problema a rede e mesmo após o notebook ser desligado a mineração continuou sendo executada apenas pela terceira máquina virtual da Azure.

Dessa forma, podemos concluir que de acordo com os testes realizados o script é capaz de criar uma rede *blockchain* funcional capaz de se integrar novos nós com acesso pela internet. Não excluindo a necessidade, de implantar camadas de segurança no servidor contra acessos não desejados.

#### **4.2.2 Teste automatizados dos *smart contracts***

Com o intuito de realizar os testes com o protótipo funcional e seus *smart contracts*, preparou-se alguns casos de uso referente as suas principais funcionalidades, que podem ser resumir em:

1. Cadastro de parâmetros: Competências, Etapas de Avaliação, Categorias, Níveis de acreditação.
2. Cadastro de organizações, o qual cria e edita contratos contendo os dados de uma organização.
3. Cadastro de profissionais, o qual cria e edita contratos contendo os dados do profissional e a criação de certificados da acreditação por pares.
4. Cadastro de Avaliações e de Grupos de Avaliação.
5. Solicitação de acreditação tanto para profissionais quanto para organizações.
6. Aprovação e criação do certificado de acreditação, funcionalidade que cria contratos que registra a competência de um ser acreditado por uma das instituições participantes do sistema.
7. Avaliação entre pares para registro de acreditação profissional.

Os casos que serão avaliados especificamente por este trabalho são os que interagem diretamente com os *smart contracts*, ou seja, funcionalidades 2, 3, 6 e 7. Estas funcionalidades foram testadas continuamente durante o processo de desenvolvimento, tanto através de ações manuais interagindo com o protótipo, quanto por testes automatizados por meio da ferramenta Remix-tests. O Remix-tests é uma ferramenta para testar *smart contracts* escritos em Solidity que funciona sob o plugin da IDE Remix e que pode ser usado como uma ferramenta de linha de comando.

Durante os testes automatizados foram definidas algumas entradas e conferido se cada uma das funções dos *smart contracts* retornava os resultados esperados, o que auxiliou no controle das alterações para que estas não quebrassem o código, o que é de extrema importância pois como dito anteriormente contratos implantados com erro causam problemas críticos para uma aplicação que os utilizam.

### 4.2.3 Estudo de caso instrumental

Nessa etapa foi realizado um estudo de caso instrumental, pretendendo analisar a viabilidade do desenvolvimento e funcionalidades do sistema. Assim, para cada uma das funcionalidades desenvolvidas para o protótipo funcional, foram descritos cenários de uso indicando quem é o ator da ação e o que é esperado como resultado com o objetivo de realizar testes em um ambiente de homologação através de uma rodada de testes com usuários. Para este fim, a aplicação web e o banco de dados do protótipo funcional do IAS foi implantado no Heroku, uma plataforma própria para disponibilizar aplicações web em várias linguagens como o Java. Os casos escritos para estes testes podem ser conferidos no Apêndice 4 e foram resumidos na apresentando as ações e resultados esperados para cada cenário.

Tabela IV - Cenários propostos para teste de funcionalidade do protótipo.

Cenário	Tipo de Usuário	Ações a serem realizadas	Resultado observado no <i>front-end</i>	Resultado observado no <i>back-end</i>
A	Profissional de saúde	<ul style="list-style-type: none"> <li>• Cadastro de profissional.</li> <li>• Solicitar acreditação.</li> <li>• Conferir solicitação.</li> </ul>	<ul style="list-style-type: none"> <li>• Registro de novo profissional.</li> <li>• Registro de solicitação para acreditação.</li> </ul>	<ul style="list-style-type: none"> <li>• Criação de um <i>smart contract</i> profissional.</li> <li>• Pedido de informação dos dados do <i>smart contract</i>.</li> </ul>
B	Usuário de uma organização participante do IAS	<ul style="list-style-type: none"> <li>• Dar notas para avaliações do processo de acreditação.</li> <li>• Deferir/Indeferir a solicitação de acreditação.</li> </ul>	<ul style="list-style-type: none"> <li>• Registro de notas das avaliações.</li> <li>• Certificado gerado para solicitações deferidas.</li> </ul>	<ul style="list-style-type: none"> <li>• Criação de um <i>smart contract</i> certificado.</li> </ul>
C	Usuário comum (população em geral)	<ul style="list-style-type: none"> <li>• Busca por profissional ou organização</li> <li>• Verificar autenticidade de uma acreditação</li> </ul>	<ul style="list-style-type: none"> <li>• Lista de todas as creditações do profissional</li> <li>• Certificado de autenticidade de uma acreditação</li> </ul>	<ul style="list-style-type: none"> <li>• Pedido de informação dos dados do <i>smart contract</i>.</li> </ul>

Fonte: Elaborado pelo autor (2021).

Devido às recomendações e restrições da pandemia do COVID-19 os testes foram realizados a distância por meio da solução de videoconferência Google Meet, com um grupo

formado por 20 pessoas, todos profissionais de nível superior, de diversas áreas como administração, biblioteconomia, contabilidade, engenharia civil, enfermagem, fisioterapia, desenvolvimento de softwares e zootecnia. Durante os testes, eram apresentados os cenários para os participantes solicitando que estes realizassem as ações pré-definidas. Os testes foram também utilizados para avaliar o *front-end*, parte do trabalho de Cavalcante (2021), como forma de teste de usabilidade do sistema.

Os testes seguiram um roteiro alternando entre questionários e reprodução de ações no protótipo funcional de acordo com o seguinte roteiro:

1. O participante era submetido a uma pesquisa (Questionário A) em que indica seu conhecimento prévio sobre o tema do protótipo.
2. Visualizava um vídeo de apresentação sobre as tecnologias e objetivos do sistema e do processo de acreditação.
3. Submetido novamente a uma segunda pesquisa (Questionário B) composto com as mesmas perguntas do anterior.
4. Apresentava-se o primeiro cenário e era solicitado ao participante que tentasse realizar a ação descrita.
5. Após realizar todas as ações do cenário preenchia outra pesquisa agora como o objetivo de avaliar a experiência que teve com a utilização do sistema.
6. Repetia-se as etapas 4 e 5 para os demais cenários.
7. Respondia a pesquisa final de opinião sobre a utilidade e potencial que o sistema do IAS poderia proporcionar.

Todos os passos eram acompanhados durante a reunião do Meet, solicitando ao participante que compartilhasse sua guia em que o sistema estivesse sendo executado, a fim de realizar a gravação do teste juntamente com o log do nó de aplicação da rede *blockchain*. Dessa forma, a cada teste realizado, os *smarts contracts* foram avaliados pelo ponto de vista do *back-end* levando em consideração os pedidos enviados para a API da *blockchain* e com a observação dos dados registrados na *blockchain* com a utilização da IDE Remix, conectada com a nossa rede privada, que serviu como uma interface de visualização dos dados de um *smart contract*, como por exemplo, no primeiro cenário ao registrar um novo profissional, após confirmada a criação do contrato, através do remix este contrato era aberto e conferido se este foi corretamente criado comparando os dados do contrato com os informados pelo participante.

Após todos os testes realizados aferiu-se que todos os pedidos de interação com *smart contracts* enviados para *blockchain* foram corretamente executados não apresentando falhas na modelagem dos *smart contracts* que impedissem de executar suas funcionalidades.

### 4.2.3.1 Resultados dos questionários

Todos os questionários foram organizados em perguntas objetivas de acordo com a escala de Likert em que o participante é convidado a emitir o seu grau de concordância com aquela frase em uma escala de pontos com descrições verbais que contemplam extremos: “concordo totalmente” e “discordo totalmente”. A utilização dessa escala garante ao participante a se expressar com maior fidelidade sem resumir sua opinião a um “sim ou não”.

No primeiro questionário a ser aplicado (Questionário A), os participantes registraram o nível de conhecimento das áreas envolvidas na criação do protótipo do IAS. O objetivo do segundo questionário (Questionário B) é aferir se após uma breve explicação, utilizando um vídeo de apresentação, esse entendimento passaria por mudanças, algo que poderia ser utilizado posteriormente ao implantar o IAS em um ambiente real para indicar se um vídeo explicativo ajudaria a instruir os usuários quanto a sua utilização.

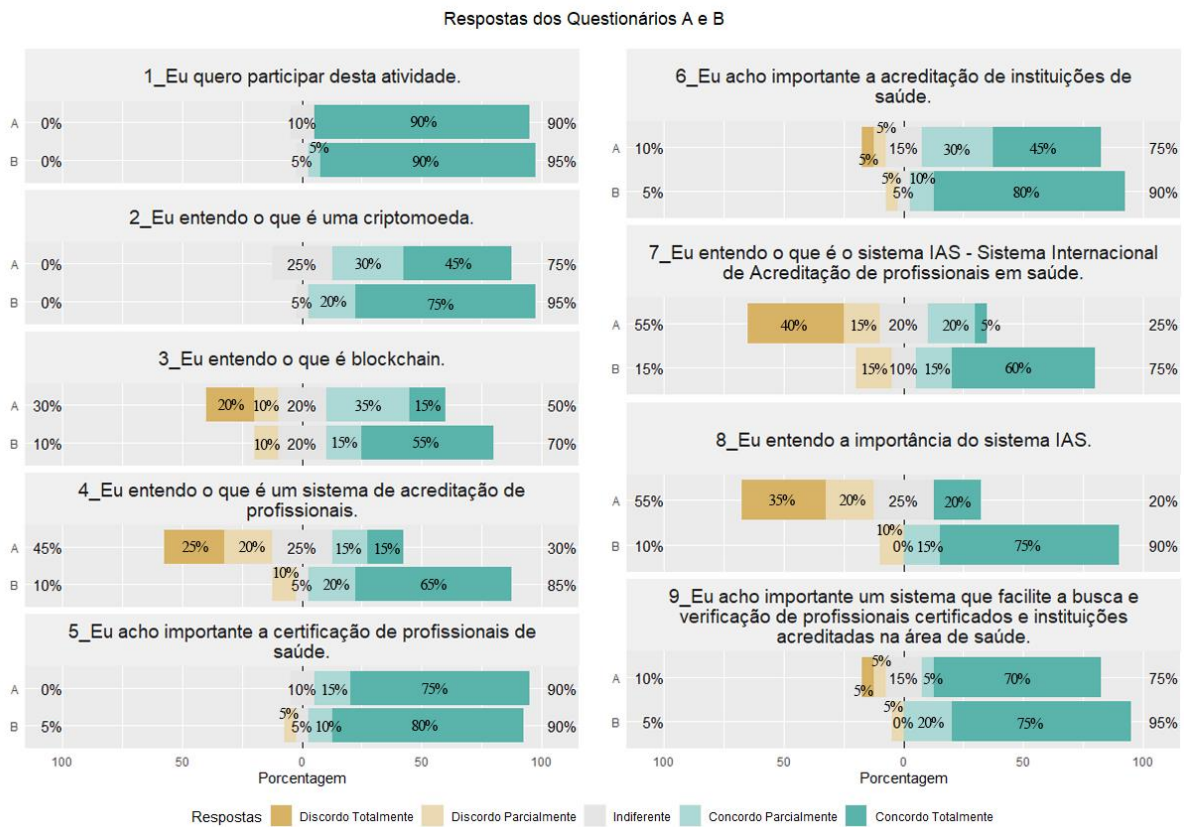


Gráfico 1 - Gráfico de Likert dos questionários A e B.

Fonte: Elaborado pelo autor (2021).

O Gráfico 1 apresenta a variação das respostas obtidas nesses dois questionários agrupadas por pergunta em forma de um gráfico de Likert, o qual é apresentado em



porcentagens primeiramente das variações de resposta obtidas no questionário A seguida da variação de respostas do questionário B.

Em todos os questionamentos podemos observar que após assistir o vídeo de apresentação do sistema, o entendimento das áreas correlatas bem como do próprio sistema aumentou. Podemos notar que 75% dos participantes do teste concordaram, mesmo que parcialmente, entender do que se trata o conceito de criptomoeda, mesmo assim, quando questionados sobre a tecnologia que está por trás delas esse percentual cai para metade. Após assistir ao vídeo a porcentagem das pessoas que concordam em entender o conceito de criptomoedas passar para 95% e para o conceito de *blockchain* passa para 70%, ambos com um aumento de 20%.

No caso da acreditação profissional apenas 30% disseram entender sobre o assunto, com a maioria afirmando não entender o assunto. Após o vídeo o entendimento sobre acreditação profissional sobe para 85%, indicando que uma breve explanação sobre o assunto pode realmente apresentar o conceito básico sobre o assunto. Vale notar, que mesmo não sabendo que a acreditação se assemelha com o conceito de certificação de profissionais da área de saúde, esse último é considerado importante para 90% dos participantes, mantendo-se após o vídeo. A acreditação de instituições de saúde também é considerada importante, primeiramente por 75% dos participantes e posteriormente se igualando ao observado sobre os profissionais da área.

Sem explicação previa do que é o IAS, podemos observar que parecido com o que foi registrado sobre o entendimento da acreditação profissional, 55% registrada que não entendem do que se tratava. E com a explicação em vídeo o entendimento passa de 25% para 75% do total participante. Como não entendiam sobre o sistema o esperado era que a importância de tal sistema também fosse considerada baixa, registrando também 55%, sendo o percentual daqueles que concordam com sua importância de apenas 20%. E novamente temos um grande salto após a explicação do sistema que passa para 90% do total de pessoas participando dos testes.

Dessa forma, podemos levar em consideração que os participantes dos testes acreditam que um sistema para auxiliar na acreditação profissional de saúde pode ser bem aceito ao entrar em funcionamento, afirmação reforçada pela última pergunta sobre ser importante existir um sistema para busca de certificações dos profissionais e instituições da área da saúde que registrou inicialmente 75% e 95% no segundo questionário.

Quanto aos questionários aplicados entre cada cenário, as perguntas estavam mais relacionadas ao funcionamento e usabilidade do *front-end*, sendo assim, serão abordados apenas por Cavalcante (2021). No entanto, junto ao último questionário tínhamos uma pesquisa de opinião, em que podemos destacar três perguntas nas quais os participantes são consultados sobre a utilidade do sistema, a segurança que a tecnologia *blockchain* pode agregar e se com a sua utilização se sentiriam mais seguros em procurar informações sobre a área médica nele disponibilizadas, são elas:

1. Eu acredito que o sistema IAS dar maior segurança na busca de profissionais e instituições para atendimento de saúde.
2. Eu acredito que um sistema de certificação e acreditação é importante para a confiabilidade na disseminação de informações.
3. Eu acredito que o sistema IAS pode ser uma ferramenta importante para o combate a epidemias e pandemias.

Como um dos principais motivos deste projeto era também criar um sistema que promovesse a confiabilidade nos profissionais e instituições de saúde a opinião de usuários quanto a essa confiabilidade que o sistema pode passar é muito importante. As respostas a esses questionamentos são apresentadas no Gráfico 2.

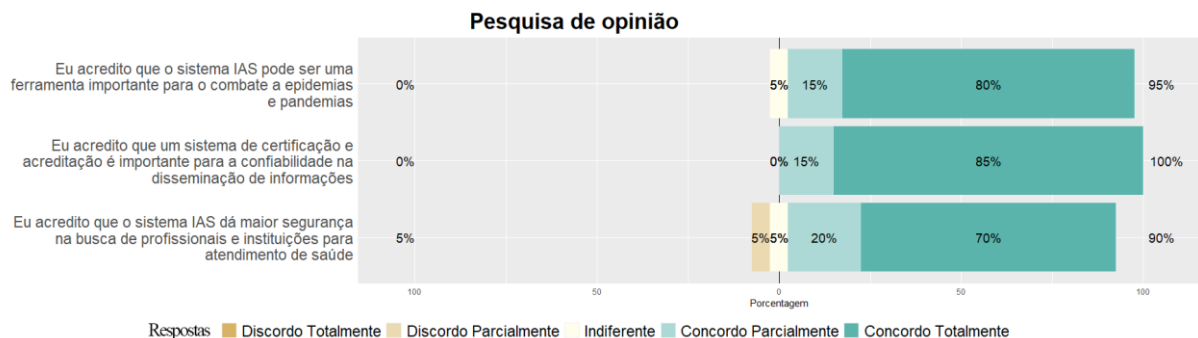


Gráfico 2 - Pesquisa de opinião sobre o IAS.

Fonte: Elaborado pelo autor (2021).

Ao observarmos o gráfico notamos mais 90% dos participantes concordam com todas as afirmações, sendo mais expressiva a confiabilidade que o sistema passa na disseminação de informações sobre os certificados e creditações da área da saúde. Apenas 5% não acredita que o IAS possa trazer mais segurança na busca de profissionais e instituições que possam prestar um serviço médico de qualidade, o que traz fortes perspectivas quanto ao futuro desse projeto.

### 4.3 CHAMADAS ASSÍNCRONAS NAS CRIAÇÕES DOS CONTRATOS

Durante o estudo de caso instrumental, foi encontrada uma possível situação de falha, por causa de uma limitação do Heroku a execução de uma funcionalidade do protótipo poderia encerrar sua execução sem concluir todas as tarefas caso o tempo limite fosse excedido, que por padrão era de 30 segundos. Como os *smarts contracts* só são efetivamente criados após a mineração de um bloco o tempo de retorno do sucesso da operação poderia ser maior que o tempo em que a sessão do sistema ficava aberta sem apresentar a conclusão do processo, mesmo com a criação do *smart contract* concluída. Uma alternativa para diminuir esse tempo de espera é o envio dos pedidos de criação dos contratos de forma assíncrona.

O protótipo foi alterado para enviar todas as transações para a *blockchain* de forma assíncrona (comunicação em que receptor da mensagem não a receberá ou responderá imediatamente) e submetido a uma nova rodada de teste para comparar o tempo das funções que criam um contrato com os tempos observados durante os testes com os usuários em que as funções eram síncronas (comunicação que o transmissor aguarda o recebimento e a resposta do receptor para finalizar a comunicação). A Tabela V apresenta os resultados levantados durante os testes do estudo de caso instrumental com os testes realizados após a alteração para envios assíncronos à *blockchain*.

Tabela V - Tempo de espera para criar um *smart contract*.

	MENOR TEMPO (MS)	TEMPO MÉDIO (MS)	MAIOR TEMPO (MS)
CHAMADAS SÍNCRONAS	16563	16950,65	19617
CHAMADAS ASSÍNCRONA	1024	1368,7	2317

Fonte: Elaborado pelo autor (2021).

Ao comparamos esses dados podemos observar que a troca de abordagem para o envio assíncrono do pedido de criação de contatos traz em média uma redução de aproximadamente 92% no tempo de execução. No entanto, a mudança na forma de envio não é a única alteração que deve ser feita para o completo funcionamento do sistema, como a resposta da *blockchain* ainda não estará disponível no momento, o endereço do contrato deve ser obtido posteriormente, uma opção viável devida à adição das fabricas de contratos na modelagem criada.

## 5 CONCLUSÃO

Neste trabalho foi apresentada a construção de uma infraestrutura básica, o *back-end*, para a criação de um sistema web baseado na tecnologia *blockchain* com o intuito de ser a base para a criação de Sistema Internacional de Acreditação para profissionais de saúde descrito por Sousa Junior et al. (2019), focando-se na forma em que seriam armazenados os certificados de acreditação emitidos pelo sistema, fazendo uso das características da *blockchain*, para garantir a qualquer interessado a possibilidade validá-los e afastar a possibilidade de falsificação.

Com relação aos objetivos traçados, todos foram alcançados como descritos na lista a seguir:

1. **Implementar uma versão básica de uma rede *blockchain* para avaliar o uso da plataforma:** foi concluída com os scripts de automação da rede *blockchain* (Apêndice 6).
2. **Elaborar um roteiro de instalação e configuração para referências futuras e de base para o desenvolvimento do Sistema de Acreditação em Saúde:** concluída através desenvolvido um roteiro para auxiliar a utilização de scripts apresentando em forma de texto e vídeo (Apêndice 8), dando origem ainda a um artigo sobre como utilizá-los em aulas de computação como ferramenta ativa de ensino (Apêndice 2).
3. **Modelar e desenvolver os *smart contracts* na linguagem nativa da rede *blockchain* de acordo com base nos metadados básicos necessários para os processos de certificação e acreditação do Sistema de Acreditação em Saúde:** atingido com o desenvolvimento de três *smarts contracts* que fazem parte do protótipo funcional do Sistema de Acreditação em Saúde (Apêndice 7) e com um script para auxiliar no processo de deploy na rede *blockchain* (Apêndice 3).
4. **Implementar os *smart contracts* no protótipo funcional do Sistema de Acreditação em Saúde:** foi providenciado com a integração dos *smarts contracts* através da conversão para classes JAVA e a conexão do protótipo funcional com a rede *blockchain* (Apêndice 7).
5. **Testar o protótipo funcional com um estudo de caso instrumental:** realizado através dos testes realizados com a *blockchain* e *smarts contracts* e com utilizando os cenários (Apêndice 4) e questionários (Apêndice 5) apresentados ao grupo de participantes, foi possível verificar o funcionamento do sistema, bem como recolher feedback sobre a experiência dos usuários ao utilizá-lo.

Com a criação da rede *blockchain* privada identificou-se também a possibilidade de otimizar o processo de mineração para que este consuma menos recursos computacionais, como por exemplo a adição de um script para controlar a mineração de novos blocos iniciando-a apenas quando novas transações forem enviadas à rede. Alternativamente, pode-se também

trocar o protocolo de consenso para o Clique, mas este deveria ser seguido de uma alteração do funcionamento do sistema, já que para novas instituições serem autorizadas a adentrar na rede como mineradores, estas deveriam ser autorizadas pela ação da maioria das instituições já presentes na rede a como mineradoras.

Quanto a abordagem de *smart contracts* podemos listar algumas limitações que devem ser levadas em consideração na hora da implantação do sistema. A primeira grande limitação é o fato de que o contrato será executado exatamente como escrito, sem qualquer flexibilidade, sendo executados mesmo com erros técnicos de programação, demandando a destruição do contrato existente e *deploy* de uma nova versão para uma correção seja aplicada.

Outro aspecto que podemos ressaltar que devido à natureza das redes *blockchain* as interações que necessitam de validação, como a implantação de um *smart contract*, envolvem um custo. Dada a necessidade do trabalho dos validadores (mineradores), o custo, que na plataforma Ethereum é chamado de *gas cost*, referente a energia gasta para realizarem essa tarefa deve ser remunerada. Dada essa característica, a execução de *smart contracts* só é permitido mediante um pagamento para a rede, proporcional à complexidade do contrato.

Essa última limitação, no entanto, pode ser contornada mais facilmente em uma rede privada. Ao criarmos a rede *blockchain*, podemos definir as contas já pré-financiadas com um montante suficiente para realizar todas as transações.

Ainda nesse contexto, o tamanho do código dos *smart contracts*, podem necessitar a alteração do limite de *gas* utilizado para se implantar uma instância do *smart contract* na rede *blockchain*, o que foi necessário ser feito com a versão final do contrato **Professional.sol**.

No fim, aplicando essas recomendações a rede *blockchain* e os *smart contracts* criados atingiram o objetivo principal para qual foram criados que era a validação conceitual de um protótipo funcional do Sistema de Acreditação Internacional em saúde. Através dos testes realizados pelos estudos de caso instrumental, o sistema se comportou adequadamente realizando as funções necessárias para auxiliar no registro de acreditação. Junto ao feedback positivo apresentado pelos participantes, o conceito do IAS apresenta-se ser bastante útil não só para os profissionais da medicina quanto a sociedade em geral.

## 5.1 TRABALHOS FUTUROS

Como trabalho futuro podemos citar a utilização dos produtos e materiais para o desenvolvimento de uma versão completa do sistema web do Sistema Internacional de Acreditação em Saúde e adicionando suas funcionalidades ao sistema RGM de Letouze et al. (2017).

Ao final deste trabalho a plataforma Ehtereum passou a suportar uma segunda linguagem para o desenvolvimento dos *smart contracts*, a linguagem Vyper, ainda é pouco documentada e pouco utilizada se comparada com a Solidity, futuramente seria interessante avaliar a construção de contratos com essa nova linguagem e se esta pode trazer benefícios maiores que utilizada neste projeto.

Por fim, sugere-se também testes como o outro protocolo de consenso Clique a fim de identificar se o esforço adicional para autorização de novos participantes da rede *blockchain* gera impactos negativos ao processo do sistema, e se estes serão compensados pelos recursos computacionais que seriam poupados pela utilização do novo consenso.

## REFERÊNCIAS

- ACREDITAÇÃO. *In*: Dicionário Priberam da Língua Portuguesa, 2019. Disponível em: <<https://dicionario.priberam.org/acreditaçao>>. Acesso em: 15 dez. 2019.
- ALIAGA, Y. E. M.; HENRIQUES, M. A. A. Uma comparação de mecanismos de consenso em blockchains. **X Encontro de Alunos e Docentes do DCA/FEEC/UNICAMP (EADCA)**, Campinas, 2017.
- ALVES, R. C. V. **Metadados como elementos do processo de catalogação**. Tese (doutorado) - Universidade Estadual Paulista, Faculdade de Filosofia e Ciências. Marília, p. 132. 2010.
- AMARAL, R. A. D.; NERIS, V. P. D. A. Análise comparativa entre frameworks de frontend para aplicações web ricas visando reaproveitamento do back-end. **Revista TIS**, v. 4, n. 1, 2016.
- ANDRADE, T. F. D. Back-end vs Front-end vs Fullstack: Escolha o seu futuro como programador! **AlgaWorks**, 2018. Disponível em: <<https://blog.algaworks.com/back-end-front-end-full-stack/>>. Acesso em: 2020 set. 14.
- ANTONOPOULOS, A. M. **Mastering Bitcoin: Unlocking Digital CryptoCurrencies**. 1. ed. [S.l.]: O'Reilly Media, Inc., 2014. ISBN 1449374042, 9781449374044.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 17011:2019: Avaliação de conformidade – Requisitos gerais para organismos de acreditação credenciando organismos de avaliação de conformidade**. Rio de Janeiro. 2019.
- BACURAU, R. M.; LEAL, B. G.; RAMOS, R. A. **Uma Abordagem para a Construção de Diagramas da UML Concomitante à Prototipação de Interface**, [2011]. Disponível em: <<http://www.univasf.edu.br/~brauliro.leal/pesquisa/Bacurau-5.pdf>>. Acesso em: 14 jan. 2021.
- BASHIR, I. **Mastering blockchain**. [S.l.]: Packt Publishing Ltd, 2017.
- BIZER, C.; MEUSEL, R.; PRIMPELI, A. Microdata, RDFa, JSON-LD, and Microformat Data Sets. **Web Data Commons**, 2020. Disponível em: <<http://webdatacommons.org/structureddata/>>. Acesso em: 30 dez. 2020.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML–Guia do usuário**. Rio de Janeiro: Campus, 2000.
- BRASIL. CONGRESSO. CÂMARA DOS DEPUTADOS. Decreto nº 6.275, de 28 de dezembro de 2007. **Estrutura Regimental e o Quadro Demonstrativo dos Cargos em Comissão e das Funções Gratificadas do Instituto Nacional de Metrologia, Normalização e Qualidade Industrial - INMETRO, e dá outras providências**, Brasília,DF, 2007.
- BRASIL. CONGRESSO. CÂMARA DOS DEPUTADOS. Lei nº 12.545, de 14 de dezembro de 2011, Brasília-DF, 2011.
- BUTERIN, V. A next-generation smart contract and decentralized application platform. **Ethereum Foundation**, v. 3, 2014. Disponível em: <[http://blockchainlab.com/pdf/Ethereum\\_white\\_paper\\_a\\_next\\_generation\\_smart\\_contract\\_and\\_decentralized\\_application\\_platform-vitalik-buterin.pdf](http://blockchainlab.com/pdf/Ethereum_white_paper_a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf)>.
- CARSTEN, B. Carsten's Corner: Let's Define a Few Terms. **Magazine Power Conversion and Intelligent**, n. 38, nov 1989.
- CASANOVA, D. Surgical Accreditation in Liver Transplantation. **Transplantation proceedings**, Elsevier, v. 41, n. 3, p. 998-1000, 2009.
- CAVALCANTE, C. E. A. **O “FRONT END” DO PROTÓTIPO FUNCIONAL DE UM SISTEMA DE ACREDITAÇÃO INTERNACIONAL PARA PROFISSIONAIS DE SAÚDE – UMA**

- VALIDAÇÃO CONCEITUAL POR ESTUDO DE CASO INSTRUMENTAL.** Dissertação (Mestrado) - Programa de Pós-Graduação em Modelagem Computacional de Sistemas, Universidade Federal do Tocantins. Palmas - TO. 2021.
- CHAUDHURI, A. et al. NASA Uncertainty Quantification Challenge: An Optimization-Based Methodology and Validation. **Journal of Aerospace Information Systems**, v. 12, n. 1, p. 10-34, 2015.
- CHENG, J.-C. et al. Blockchain and smart contract for digital certificate. **Proceedings of 4Th Ieee International Conference on Applied System Innovation 2018 ( Ieee Icase 2018 )**, IEEE, p. 1046–1051, 2018.
- CHRISTIDIS, K.; DEVETSIKIOTIS, M. Blockchains and smart contracts for the internet of things. **IEEE Access**, v. 4, p. 2292-2303, 2016.
- CITRUS7. O QUE É FRONT-END E BACK-END? **Citrus7**, 2017. Disponível em: <<https://citrus7.com.br/artigo/o-que-e-front-end-e-back-end/>>. Acesso em: 24 set. 2020.
- COOK, S. et al. **Unified Modeling Language (UML) Version 2.5.1**. Object Management Group (OMG). Milford, p. 754. 2017. (formal/2017-12-05).
- COSTA, A. F. **Processo de acreditação de organismos de certificação utilizado pelo Inmetro: um estudo comparativo com organismos congêneres de diversos países.** Dissertação (Mestrado) - Curso de Sistemas de Gestão, Universidade Federal Fluminense. Niterói, p. 108 f. 2006.
- COSTA, C. A. A aplicação da linguagem de modelagem unificada (UML) para o suporte ao projeto de sistemas computacionais dentro de um modelo de referência. **Gestão & Produção**, v. 8, n. 1, p. 19-36, 2001.
- DENCKER, A. D. F. M. **Pesquisa em turismo: planejamento, métodos e técnicas.** 9. ed. São Paulo: Futura, 2007.
- DESTEFANIS, G. et al. Smart contracts vulnerabilities: a call for blockchain software engineering? **2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)**, IEEE, 2018. 19--25.
- ETHEREUM.ORG. Official Go implementation of the Ethereum protocol. **Go Ethereum**, 2019. Disponível em: <<https://geth.ethereum.org/docs/interface/private-network>>. Acesso em: 09 jan. 2021.
- ETHEREUM.ORG. Contract ABI Specification. **Solidity**, 2021. Disponível em: <<https://docs.soliditylang.org/en/v0.8.1/abi-spec.html>>. Acesso em: 24 jan. 2021.
- FELISBINO, C. M. **Ferramenta para o apoio ensino-aprendizagem do modelo orientado a objetos durante a construção do diagrama de classes.** Dissertação (Mestrado em Computação Aplicada) - Universidade Tecnológica Federal do Paraná. Curitiba, p. 118. 2017.
- FILIPOVA, O.; VILÃO, R. **Software Development From A to Z.** 1. ed. Berkeley, CA: Apress, 2018.
- GRÁCIO, J. C. A. **Preservação digital na gestão da informação: um modelo processual para as instituições de ensino superior.** São Paulo: Cultura Acadêmica, 2012.
- GREVE, F. et al. Blockchain e a Revolução do Consenso sob Demanda. **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos**, 2018.
- IANSITI, ; LAKHANI, K. R. The truth about blockchain. **Harvard business review**, v. 95, n. 1, p. 118-127, jan. 2017.
- IKEMATU, R. S. Gestão de metadados: sua evolução na tecnologia da informação. **DataGramZero-Revista de Ciência da Informação**, v. 2, n. 6, 2001.
- JACOBSON, I. **Object Oriented Software Engineering: A Use Case Driven Approach.** 1. ed. México: Addison-Wesley, 1992.



- KOSBA, A. et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. **2016 IEEE symposium on security and privacy (SP)**, IEEE, 2016. 839-858.
- KOULU, R. Blockchains and Online Dispute Resolution: Smart Contracts as an Alternative to Enforcement. **SCRIPTed**, v. 13, n. 40, 2016.
- LARMAN, C. **Utilizando UML e Padrões**. 2. ed. Porto Alegre: Bookman, 2004.
- LETOUZE, P. Interdisciplinary research project management. **International Proceedings of Economics Development and Research**, v. 14, p. 338-342, 2011.
- LETOUZE, P. Evolutionary acquisition interdisciplinary research project management. **International Proceedings of Economics Development and Research**, v. 30, p. 231-235, 2012.
- LETOUZE, P. et al. Applying MVC to evolutionary acquisition IRPM. **International Proceedings of Computer Science and Information Technology**, v. 45, p. 123--128, 2012.
- LETOUZE, P. et al. Applying the MVC EA-IRPM to Reporting-Guidelines in Medicine: a strategy that is a web system. **2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)**, Orland, FL, USA: IEEE, 2017. 97-100.
- LETOUZE, P.; DA SILVA, V. M.; SOUZA JÚNIOR, J. I. M. Patient-centric healthcare service systems: evidence-based medicine as architecturally significant requirement. **Proceedings of the International Workshop on Software Engineering in Healthcare Systems - SEHS '16**, Austin, Texas: ACM Press, 2016. 26-32.
- LIGHTSEY, B. **Systems engineering fundamentals**. Fort Belvoir, VA: Defense Acquisition University, 2001.
- LIN, X. Semi-centralized Blockchain Smart Contracts: Centralized Verification and Smart Computing under Chains in the Ethereum Blockchain. **Department of Information Engineering, National Taiwan University**, Taiwan, ROC, 2017.
- LINHARES, M. V. D. **Uso de big data e criação de tecnologia (software e hardware), com prova de conceito e validação, para identificar, diagnosticar e prever os fatores de riscos no controle de qualidade da cadeia produtiva e industrial do mel com Prospecção tecnológica visa**. Tese (doutorado) - Universidade Federal da Bahia. Salvador, p. 74. 2016.
- LOBO, E. J. R. **Guia Prático de Engenharia de Software: Desenvolva softwares profissionais com uso UML e “best practices” de gestão**. São Paulo: Digerati Books, 2009.
- MAMÉDIO, D. F. Estratégia como processo em uma organização hospitalar: um diagnóstico dos 5ps de mintzberg. **Revista Eletrônica Científica do CRA-PR-RECC**, v. 1, n. 1, p. 37-52, 2014.
- MASCARENHAS, J. Z.; VIEIRA, A. B.; ZIVIANI, A. Análise da rede de transações do ethereum. **Anais do I Workshop em Blockchain: Teoria, Tecnologias e Aplicações**, Campos do Jordão, 2018.
- NAKAMOTO, S. **Bitcoin: A Peer-to-Peer Electronic Cash System**, 2008. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>. Acesso em: 28 mar. 2019.
- NATHALI, M. PROOF OF CONCEPT NO DESENVOLVIMENTO DE SOFTWARE: O QUE É? **GoBacklog**, 2020. Disponível em: <<https://gobacklog.com/blog/proof-of-concept/>>. Acesso em: 2020 nov. 20.
- OUCHI, M. T.; SIMIONATO, A. C. Descrição de conjuntos de dados na Web com schema.org. **Informação & Tecnologia**, v. 5, n. 1, p. 128-140, 2018.
- PIRATELLI-FILHO, A. Acreditação do laboratório de metrologia dimensional da universidade de Brasília, região centro-oeste do Brasil. **Revista Produção Online**, v. 11, n. 1, p. 96–115, 2011.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software**. 8. ed. [S.l.]: McGraw Hill Brasil, 2016.

- PREUKSCHAT, A. et al. **Blockchain: la revolución industrial de internet**. Barcelona: Gestión 2000, 2017.
- QUEIROZ, V. A. R. D. et al. Oficina de prototipação como ação extensionista: um relato de experiência com jovens de uma comunidade de baixa renda. **Revista de Sistemas e Computação-RSC**, v. 8, n. 2, 2019.
- ROEHR, A.; DA COSTA, C. A.; DA ROSA RIGHI, R. OmniPHR: A distributed architecture model to integrate personal health records. **Journal of Biomedical Informatics**, v. 71, p. 70 - 81, 2017. ISSN 1532-0464.
- SANTANDER, V. F.; CASTRO, J. Desenvolvendo Use Cases a partir de Modelagem Organizacional. **WER**, 2000. 158-180.
- SAYÃO, L. F. Uma outra face dos metadados: informações para a gestão da preservação digital. **Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação**, v. 15, n. 30, p. 1-31, 2010.
- SCHEMA.ORG. **Welcome to Schema.org**, 2020. Disponível em: <<https://schema.org/>>. Acesso em: 29 dez. 2020.
- SILLABER, C.; WALTL, B. Life cycle of smart contracts in blockchain ecosystems. **Datenschutz und Datensicherheit-DuD**, v. 41, n. 8, p. 497--500, 2017.
- SILVA, M. A. D. PROVA DE CONCEITO (PoC) EM PROJETOS. **LinkedIn**, 2015. Disponível em: <<https://www.linkedin.com/pulse/prova-de-conceito-poc-em-projetos-silva-pmp-prince2-practitioner>>. Acesso em: 2020 dez. 19.
- SOMMERVILLE, I. **Engenharia de Software**. Tradução de Ivan Bosnic e karlinka G. de O. Gonçalves. 9. ed. São Paulo: Pearson, 2011.
- SOUZA JUNIOR, J. I. M. D. et al. An International Accreditation System for Healthcare Professionals Based on Blockchain. **International Journal of Education and Information Technologies**, v. 9, n. 7, p. 462--469, 2019.
- SOUZA, T. B. D.; CATARINO, M. E.; SANTOS, P. C. D. Metadados: catalogando dados na Internet. **Transinformação**, v. 9, n. 2, 1997.
- STAKE, R. E. The case study method in social inquiry. In: DENZIN, N. K.; LINCOLN, Y. S. **The American tradition in qualitative research**. Thousand Oaks, CA: Sage Publications, v. 2, 2001.
- STEWART, L. Front End Development vs Back End Development: Where to Start? **Course Report**, 2020. Disponível em: <<https://www.coursereport.com/blog/front-end-development-vs-back-end-development-where-to-start>>. Acesso em: 25 set. 2020.
- VALENTE, M. T. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**. 1. ed. [S.l.]: UmLivro, 2020.
- VAZQUEZ, C.; SIMÕES, G. **Engenharia de Requisitos: Software Orientado ao Negócio**. Rio de Janeiro: Brasport, 2016.
- WANG, S.; XIE, J. Integrating Building Management System and facilities management on the Internet. **Automation in construction**, v. 11, n. 6, p. 707--715, 2002.
- WOOD, G. Ethereum: A secure decentralised generalised transaction ledger. **Ethereum project yellow paper**, 2014. Disponível em: <<http://gavwood.com/paper.pdf>>.
- YEH, L.-Y. et al. E-university applications : A Privacy-Preserving Diploma Notarization Platform in Taiwan. **Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE)**, Athens, 2018. 44-50.
- YIN, R. K. **Estudo de Caso-: Planejamento e métodos**. 5. ed. Porto Alegre: Bookman, 2015.

## APÊNDICES

Apêndice 1 – Relatório de avaliação das plataformas “*blockchain*”.

Modelo de “Relatório Técnico de Avaliação” criado por Prof. Dr. Patrick Letouze Moreira



UNIVERSIDADE FEDERAL DO TOCANTINS  
CAMPUS UNIVERSITÁRIO DE PALMAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
MODELAGEM COMPUTACIONAL DE SISTEMAS

### RELATÓRIO TÉCNICO DE AVALIAÇÃO

ACADÊMICOS:

*Flávio Fernandes de Melo e Carlos Eduardo Alves Cavalcante*

PROFESSOR ORIENTADOR:

*Prof. Dr. Patrick Letouze Moreira*

(Membro permanente do PPG-MCS da UFT)

OBJETO DE AVALIAÇÃO NO CONTEXTO DA PESQUISA PARA A OBTENÇÃO  
DO TÍTULO DE **MESTRE**:

*Seleção de uma plataforma “*blockchain*” para os estudos de viabilidade  
da aplicação do “*blockchain*”*

Palmas/TO  
Novembro/2020

## 1. O Relatório Técnico de Avaliação

O processo de pesquisa para a obtenção do título de Mestre em **Modelagem Computacional de Sistemas** pela Universidade Federal do Tocantins pode eventualmente envolver avaliar um ou mais objetos, sistemas, processos, métodos, estratégias, estudos, equipamentos, softwares e teorias, isto é, o objeto de avaliação do projeto da pesquisa deve ser determinado e esclarecido. Neste sentido, os signatários deste Relatório Técnico de Avaliação (RTA) estabelecem conjuntamente o resultado da análise do objeto avaliado no âmbito de projetos de pesquisa do Programa de Pós-Graduação em **Modelagem Computacional de Sistemas (PPG-MCS)** da UFT.

## 2. Sobre a Escolha do Objeto de Avaliação para a Pesquisa

O orientador membro do corpo docente do Programa de Pós-Graduação em **Modelagem Computacional de Sistemas (PPG-MCS)** da UFT e os alunos regulares signatários deste relatório concordam com o Objeto de Avaliação para a Pesquisa que estabelecem neste Relatório Técnico de Avaliação (RTA).

### 2.1. O Objeto de Avaliação

O Objeto de Avaliação ***Seleção de uma plataforma "blockchain" para os estudos de viabilidade da aplicação do "blockchain"*** é de interesse dos acadêmicos e do professor orientador.

### 2.2. Sobre a Justificativa do Objeto de Avaliação

A Tecnologia ***Blockchain*** é uma lista crescente de registros, chamados de blocos, que são vinculados da seguinte forma: cada bloco contém um hash criptográfico do bloco anterior, uma marcação de data e hora, além dos dados de transação. Este sistema possui arquitetura resistente à modificação de seus dados *ex post facto*. Isto porque a alteração retroativa de dados registrados requer a alteração de todos os blocos subsequentes, e porque esses registros são armazenados de forma descentralizada numa rede *peer-to-peer*. Portanto, entende-se que a tecnologia ***blockchain*** apresenta características arquiteturais desejáveis a dados e serviços que necessitem segurança, imutabilidade e rastreabilidade. Deste modo, faz-se necessário escolher uma plataforma ***blockchain*** para o desenvolvimento desses serviços.

### 3. Sobre o Processo de Avaliação do Objeto para a Pesquisa

O orientador membro do corpo docente do Programa de Pós-Graduação em **Modelagem Computacional de Sistemas (PPG-MCS)** da UFT e os alunos regulares deste mesmo programa de pós-graduação concordam com o método para a avaliação do objeto de interesse para a pesquisa que estabelecem neste Relatório Técnico de Avaliação (RTA).

#### 3.1. O Método de Avaliação do Objeto para a Pesquisa

O método de avaliação do objeto para a pesquisa deve ser adequado a natureza do objeto. Isto implica em uma relação direta, ou seja, o objeto a ser avaliado para a realização da pesquisa deve compor um conjunto de possibilidades de investigação que seja contributivo para a pesquisa em questão. Neste caso, o método escolhido é o **benchmarking de plataformas blockchain**.

#### 3.2. Sobre as Alternativas de Objeto de Avaliação

Recomenda-se que sejam elencadas pelo menos três alternativas pertinentes ao objeto de avaliação, pois caso haja um evento que impossibilite o desenvolvimento da pesquisa com o objeto selecionado, tem-se duas ou mais alternativas para prosseguimento dos trabalhos, e conseqüentemente se reduz o risco de fracasso do processo de **Mestrado**.

#### 3.3. Sobre a Seleção do Objeto de Pesquisa

Para a seleção de objeto de avaliação, recomenda-se que as alternativas sejam comparadas. Sugere-se que os critérios de comparação sejam bem definidos, que a comparação seja tabulada e que a escolha seja justificada.

### 4. Sobre os Critérios de Seleção do Objeto de Pesquisa

Recomenda-se que as alternativas sejam bem definidas. O estabelecimento de critérios deve facilitar a avaliação, quanto propiciar os elementos de comparação que sustentarão a seleção do objeto entre as alternativas. As

Modelo de "Relatório Técnico de Avaliação" criado por Prof. Dr. Patrick Letouze Moreira

características das alternativas com relação aos critérios devem ser especificadas.

#### 4.1.1. Primeiro Critério

O primeiro critério (1C): principal uso.

#### 4.1.2. Segundo Critério

O segundo critério (2C): tipo de rede.

#### 4.1.3. Terceiro Critério

O terceiro critério (3C): consenso.

#### 4.1.4. Quarto Critério

O quarto critério (4C): *smart contracts*.

#### 4.1.5. Quinto Critério

O quinto critério (5C): APIs.

#### 4.1.6. Sexto Critério

O sexto critério (6C): possui código aberto.

## 5. Sobre as Alternativas do Objeto de Avaliação

### 5.1.1. Primeira Alternativa

A primeira alternativa (A1) consiste no **Bitcoin**.

### 5.1.2. Segunda Alternativa

A segunda alternativa (A2) consiste no **Ethereum**.

### 5.1.3. Terceira Alternativa

A terceira alternativa (A3) consiste no **Hyperledger**.

### 5.1.4. Quarta Alternativa

A quarta alternativa (A4) consiste no **Quorum**.

### 5.1.5. Quinta Alternativa

A quinta alternativa (A5) consiste no **EOS**.

### 5.1.6. Sexta Alternativa

A sexta alternativa (A6) consiste no **R3 Corda**.

## 6. Sobre a Comparação das Alternativas de Objeto de Avaliação

Quando possível os critérios estabelecidos devem ser tabulados. Isto consiste num processo de *Benchmarking*.

Modelo de "Relatório Técnico de Avaliação" criado por Prof. Dr. Patrick Letouze Moreira

### 6.1.1. Benchmarking dos Critérios de Seleção de Alternativas de Objeto de Avaliação

Alternativa \ Critério	1C	2C	3C	4C	5C	6C
A1	Criptomoeda	Não permissionada	PoW	Limitado	Bitcoin-cli (RPC)	Sim
A2	Plataforma genérica de blockchain	Não permissionada ou permissionada	PoW, PoS	Sim	Java, Python, Javascript, Go, Rust, dotNet, Delphi	Sim
A3	Blockchain voltado para empresas	Permissionada	Kafka, PoET, BFT	Sim	CLI, REST, Java e Node.js	Sim
A4	Para aplicativos que requerem alto nível de privacidade	Permissionada	QuorumChain, RAFT (baseado)	Sim	Ferramentas familiares da Ethereum	Sim
A5	Plataforma escalável para dapps em escala industrial	Permissionada	DPOS	Sim	Javascript, Swift, Java	Sim
A6	Plataforma especializada para a indústria financeira	Permissionada	RAFT, BFT	Sim	Kotlin, Java	Sim

### 6.1.2. Justificativa da Seleção da Alternativa

**A *Ethereum Blockchain* foi escolhida por garantir as restrições as restrições impostas pelo projeto de origem como: a necessidade de código aberto, suporte à smart contracts e a compatibilidade da rede com a linguagem de programação JAVA. Após as comparações notou-se que a possibilidade de criar uma rede não permissionada seria a ideal para atingir objetivos futuros do projeto inicial, já que este tipo de rede é projetada para permitir a participação pública (por exemplo, alguns aplicativos que dependem de dados gerenciados pelos usuários).**



## 7. Sobre o Resultado da Avaliação

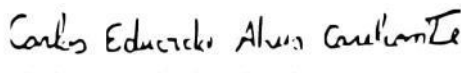
A alternativa selecionada foi:


### **ETHEREUM BLOCKCHAIN**

Neste sentido, de comum acordo, os signatários deste parecer tornam público seu entendimento sobre o objeto avaliado no contexto dos projetos de pesquisa dos acadêmicos **FLÁVIO FERNANDES MELO E CARLOS EDUARDO ALVES CAVALCANTE** sob orientação do **Professor Dr. PATRICK LETOUZE MOREIRA**, para a obtenção do título de **Mestre** pelos acadêmicos signatários.

Palmas, 1 de dezembro de 2020.

  
Flávio Fernandes Melo  
(Aluno regular do PPG MCS - UFT)

  
Carlos Eduardo Alves Cavalcante  
(Aluno regular do PPG MCS - UFT)

  
Prof. Dr. Patrick Letouze  
(Membro permanente do PPG MCS - UFT)

## Scripts de Instalação de uma Rede *Blockchain* como Recurso Didático para Metodologias Ativas de Ensino de Computação

Flávio Fernandes de Melo  
Universidade Federal do Tocantins  
Palmas, Tocantins, Brasil  
meloflavio@uft.edu.br

Carlos Eduardo Alves  
Cavalcante  
Universidade Federal do Tocantins  
Palmas, Tocantins, Brasil  
carlosalves@uft.edu.br

Patrick Letouze Moreira  
Universidade Federal do Tocantins  
Palmas, Tocantins, Brasil  
letouze@uft.edu.br

### RESUMO

Neste trabalho é proposto *Scripts* de instalação de uma rede *Blockchain* como recurso didático para o uso de metodologias ativas de aprendizagem com práticas *hands-on* no ensino de computação. O problema inicial proposto consiste na criação e instalação de uma rede *blockchain* privada. A intenção é disponibilizar um recurso didático que apoie as práticas no ensino de computação em relação a *blockchain* e as disciplinas que utilizem conceitos relacionados a essa tecnologia. Neste intuito, elaborou-se um *script* para a criação e configuração de uma rede *blockchain*, que juntamente com um roteiro de orientação compõem o material didático utilizado para introduzir os conceitos e fundamentos da tecnologia *blockchain* ao mesmo tempo que pode ser utilizado para demonstrar a criação e instalação real de uma rede. Este material pode ser facilmente adaptado para estimular os estudos além da tecnologia *Blockchain*, por exemplo, pode ser adaptado para as disciplinas de Introdução à Computação, Introdução à Programação, Algoritmos e Programação, Sistemas Operacionais, Banco de Dados, Redes de Computadores, Segurança em Tecnologia da Informação entre outras.

### PALAVRAS-CHAVE

*Blockchain*, Metodologias Ativas de Aprendizagem, *Script*

### 1 INTRODUÇÃO

A tecnologia *blockchain* foi apresentada inicialmente por Nakamoto [15] ao descrever uma moeda inteiramente digital, o que permitiria enviar pagamentos online diretamente de uma pessoa para outra, sem a necessidade de passar por uma instituição financeira. A tecnologia rapidamente se popularizou com a criação da criptomoeda Bitcoin, que teve seu bloco inicial criado no início de 2009 e, desde então, expandiu-se em uma escala sem precedentes.

Apesar de seu foco inicial na criação de criptomoedas, Abdellatif [1] afirma que cada vez mais setores, como governos, finanças, saúde, indústrias em geral e entre outros buscam novas possibilidades de uso para esta promissora tecnologia. Grande parte do interesse sobre *blockchain* baseia-se em suas propriedades básicas, que prometem alta segurança, confiabilidade e disponibilidade dos

dados contidos em sua rede, além de promover a descentralização no controle de suas transações.

Com o alto interesse na tecnologia, não demorou muito para que surgissem diversos projetos que a explorassem para além das criptomoedas, por exemplo, Cheng et al. [6], descreve um sistema para o reconhecimento de diplomas de graduação utilizando *blockchain*. Notheisen et al. [17] por sua vez, demonstra a utilização da tecnologia em um sistema para o gerenciamento de ativos do mundo real, como casas e carros. Brave (2019) desenvolveu um navegador com a possibilidade de recompensar os usuários e criadores de conteúdo utilizando uma *blockchain*. Em outros exemplos Souza Junior et al. [25] descreve a utilização de *blockchain* para um sistema internacional de acreditação de profissionais de saúde e Letouze et al. [13] descreve um sistema baseado em *blockchain* para a negociação de precatórios no Brasil.

As possibilidades de uso para a tecnologia *blockchain* são as mais diversas e a perspectiva de evolução e impacto da tecnologia são muito grandes. No entanto, Oliveira e Freitas [18] consideram insuficientes a quantidade de estudos realizados na área até então, o que segundo os autores dificulta a identificação de como ela poderá realmente afetar a sociedade de uma forma mais abrangente, assim necessitando de um maior número de pesquisas sobre o assunto.

Uma das grandes dificuldades na disseminação e utilização de *blockchain* segundo Bornelus, Chi e Shahriar [5] é a considerável curva de aprendizado da tecnologia, uma vez que os fundamentos científicos e computacionais por trás da tecnologia envolvem conhecimentos de múltiplas disciplinas, o que dificulta sua compreensão por pessoas que não estão familiarizadas com estes fundamentos. Diante dessas dificuldades, existem trabalhos que ajudam a difundir o conhecimento dessa tecnologia, como o material de apoio produzido pelo Tribunal de Contas da União [26], elaborado em forma de sumário executivo para auxiliar gestores públicos a avaliar a pertinência do projeto *blockchain* de suas organizações, apresentando a experiência de outras organizações no Brasil e no mundo.

Fomentar o estudo de *blockchain* em sala de aula em cursos de tecnologia é uma alternativa para incentivar novos projetos na área, além de promover a utilização dos conceitos de diversas disciplinas da computação. Porém vale ressaltar que a simples apresentação de conceitos em aulas teóricas pode não ser suficiente, pois como mencionado por Gavaza, Salvador e Do Santos [11], uma disciplina que trata de tópicos que possuem um alto nível de abstração exige bastante esforço dos alunos para sua compreensão.

Para facilitar o aprendizado de assuntos com um alto grau de abstração, como a tecnologia *blockchain*, é preciso buscar alternativas ao ensino tradicional baseada apenas na exposição teórica de seus conceitos. Pinto et al. [8] afirma que para isso é necessário

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

*EduComp '21*, Abril 27–30, 2021, Jataí, Goiás, Brasil (On-line)

© 2021 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

lançar mão de metodologias que busquem envolver mais o aluno no processo de aprendizagem, assim permitindo uma maior relação dos conhecimentos aprendidos em aula com sua utilização prática no mundo real. Neste contexto identifica-se a hipótese de utilização de metodologias ativas de aprendizagem, que contribuem para maior interação dos professores e alunos, permitindo a construção ativa e colaborativa dos conhecimentos.

Entre as metodologias ativas de aprendizado pode-se destacar o Aprendizado Baseado em Problemas (ABP) como metodologia para o ensino de disciplinas complexas. Nessa metodologia um problema é proposto para os estudantes, a solução prática é construída colaborativamente pelos alunos com a supervisão do professor. Neste âmbito, temos os exemplos de Silva et al. [24] que descreve a utilização de ABP para o ensino de urgência e emergência na enfermagem, um estudo realizado na Universidade Federal do Pará. Um outro exemplo de utilização de ABP é o trabalho de Rodrigues e Araújo [9] que relata a utilização da metodologia no ensino das disciplinas de contabilidade de uma universidade particular.

Outras abordagens de aprendizagem ativa também podem ser utilizadas, como o trabalho de Du [10] o qual descreve a utilização de exercícios laboratoriais práticos para o ensino de segurança na computação. Rao e Dave [20] por sua vez, apresenta a utilização de exercícios práticos para o ensino de novas tecnologias como Internet das Coisas (em inglês: Internet of Things, IoT) e *blockchain*. A abordagem descrita por estes autores é conhecida como aprendizado *hands-on*, onde os estudantes são apresentados aos conceitos teóricos e logo em seguida são levados a aplicar os conhecimentos em exercícios práticos. A abordagem *hands-on* de aprendizado pode ser facilmente adaptada e integrada ao ABP, utilizando os exercícios práticos para auxiliar na resolução de um determinado problema, ao passo que constrói gradualmente os conhecimentos necessários.

Neste trabalho propomos a utilização de um *script* - arquivo com conjunto de comandos executados por um interpretador (COSTA, 2010) [7], como um recurso didático *hands-on* no ensino da computação, produto de uma abordagem ABP. O *script* utilizado foi desenvolvido para a criação e configuração automática de uma rede *blockchain* privada, o que permite introduzir os conceitos necessários enquanto realiza-se a demonstração prática da tecnologia. O mesmo *script* pode ser utilizado para o ensino de diversas disciplinas da grade curricular em um curso de ciência da computação apenas alterando o foco da apresentação dos conceitos, uma vez que os fundamentos da tecnologia *blockchain* são compostos de conceitos básicos de várias destas disciplinas, como Introdução à Computação, Introdução à Programação, Algoritmos e Programação, Sistemas Operacionais, Banco de Dados, Redes de Computadores, Segurança em Tecnologia da Informação entre outras.

## 2 FUNDAMENTOS

Nesta seção são descritos alguns dos fundamentos utilizados no trabalho, como *Blockchain*, Metodologias Ativas de Aprendizagem e Aprendizagem Baseada em Problemas.

### 2.1 Blockchain

Segundo Nakamoto [15], a tecnologia *Blockchain* funciona como um tipo de livro razão distribuído, com recurso de imutabilidade entre os nós em uma rede *peer-to-peer* baseado em um protocolo de

consenso. Cada nó pode manter a mesma razão sem uma autoridade centralizada utilizando *hashes* criptográficos e assinaturas digitais garantindo a integridade das transações em cada bloco.

Quanto a estrutura do *Blockchain*, esta é construída por blocos ligados por uma lista encadeada, de forma que cada bloco contenha a referência do seu antecessor, garantindo assim que a modificação de informações gravadas em cada bloco exija um grande poder computacional, tornando essa ação computacionalmente impraticável em grandes redes.

Antonopoulos [2] descreve um bloco sendo composto por um identificador (*block hash*), definido pela dupla aplicação do algoritmo SHA-256 em seu cabeçalho, o *block hash* do bloco anterior, o conjunto de todas as transações, juntamente com um conjunto de informações que compõem seu cabeçalho.

A estrutura do cabeçalho pode ser dividida em três conjuntos de dados de acordo com o seu propósito. O primeiro chamado de *Previous Block Hash*, composto com o *hash* do bloco anterior, garante a conexão entre todos os blocos da *Blockchain*. O segundo é campo *Merkle Root*, usado para resumir de maneira eficiente o conjunto de transações do bloco. Por fim, o conjunto dos campos *timestamp*, *difficulty target*, e *nonce* são referentes ao processo de mineração, representando respectivamente, hora aproximada da criação do bloco, dificuldade alvo do algoritmo utilizada no bloco e o contador utilizado pelo algoritmo.

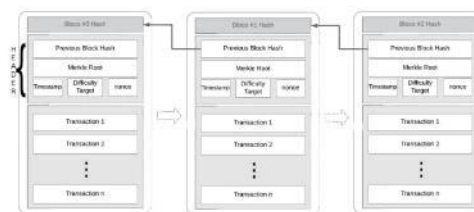


Figura 1: Blocos encadeados. Fonte: Adaptado de Antonopoulos [2].

A Figura 1 mostra um exemplo da estrutura dos três conjuntos de dados do bloco e do encadeamento entre eles, sendo comum para a identificação do bloco, além do *hash* duplo criado pela criptografia, o número da posição em que ele se encontra na *Blockchain*. Sobre as propriedades inerentes aos conceitos confiabilidade e segurança da tecnologia de *blockchain*, Iansiti e Lakhani [12] descrevem cinco princípios básicos, os quais seguem listados abaixo:

- **Banco de dados distribuído:** cada parte em um *blockchain* tem acesso a toda base dados e ao seu completo histórico de transações sem a necessidade de intermediários, no entanto, ninguém pode alterar seus registros individualmente.
- **Transmissão ponto-a-ponto:** a comunicação ocorre diretamente entre os pontos, em vez de serem realizadas de forma centralizada, cada ponto armazena e encaminha as informações aos demais participantes.
- **Transparência e pseudoanonimato:** cada transação e os valores associados são disponibilizadas a qualquer usuário com acesso ao sistema. No entanto, cada nó, ou usuário, em

um *blockchain* tem um “endereço” alfanumérico único que o identifica. Um usuário pode escolher se manter anônimo ou compartilhar provas de identidade com os outros. As transações ocorrem entre estes “endereços” no *blockchain*.

- **Irreversibilidade de registros:** uma vez realizada uma transação e esta transação adicionada ao *blockchain*, os registros não podem ser alterados, uma vez que as propriedades do *blockchain* garantem que cada registro esteja relacionado a todos os registros adicionados antes dele.
- **Lógica computacional:** a natureza digital dos registros significa que as transações de *blockchain* podem ser vinculadas à uma lógica computacional e, em essência, programadas. Assim, os usuários podem determinar algoritmos e regras que vinculam automaticamente transações entre nós.

A introdução de *Smart Contracts*, que funcionam como “um contrato digital que é escrito em código-fonte e executado por computadores, que integra o mecanismo à prova de adulteração de *Blockchain*” (LIN, 2017) [14], propiciou maiores níveis de programabilidade para a tecnologia. A utilização de redes *blockchain* que dispõem desses recursos são ideais para aplicações em novas áreas que diferem de seu foco original das criptomoedas.

Visando estes conceitos e utilizando a linguagem de criação de códigos para automatização de tarefas *Shell Script*, foram desenvolvidos arquivos (*scripts*) contendo instruções que ao serem executadas criam e configuram uma rede *blockchain* privada. Vale ressaltar que estes *scripts* foram escritos para que possam ser executados em máquinas com os sistemas operacionais Windows, Linux e MacOS. Utilizamos para os testes as seguintes versões:

- Windows 7 e 10
- Ubuntu 18.04, Ubuntu 19.04 e Debian 9
- MacOS 10.15

## 2.2 Metodologias Ativas de Aprendizagem

As Metodologias ativas de aprendizagem colocam o aluno como centro do processo de ensino. Conforme Barbosa e Moura em [3], nessas metodologias a aprendizagem ocorre quando o aluno interage com o assunto em estudo das mais diversas formas, como falando, ouvindo, discutindo ou fazendo. O aluno deixa de ser um receptor passivo das informações limitado a memorizar o conteúdo para ele apresentado e torna-se um colaborador ativo do processo de aprendizado, despertando seu pensamento crítico. Neste sentido, Rocha e Lemos em [21] afirmam que nestas metodologias o conhecimento é construído pela interação dos alunos, professores e o ambiente, reforçando a participação do aluno como fundamental para a construção dos conhecimentos.

**2.2.1 Aprendizado Baseado em Problemas.** De acordo com Savery [22], a Aprendizagem Baseada em Problemas (ABP) é uma abordagem instrucional e curricular centrada no aluno que permite que eles conduzam pesquisas, integrem teoria e prática e apliquem conhecimentos e habilidades para desenvolver uma solução viável para um problema pré definido. Orey [19] afirma que em cursos acadêmicos, a ABP é usada como uma ferramenta para ajudar os alunos a compreender a utilidade de um determinado conceito ou estudo.

Segundo Silva et al. [24] nessa metodologia para solucionar o problema apresentado, os alunos devem recorrer aos sete passos do ABP, que são:

- Esclarecer termos e conceitos desconhecidos;
- Definir o problema;
- Analisar o problema baseado em conhecimentos prévios;
- Resumir as conclusões;
- Formular metas de estudo;
- Auto-aprendizado;
- Dividir conhecimentos com o grupo;

O aprendizado nessa abordagem não se limita apenas aos conhecimentos adquiridos, mas também no processo que foi empregado. Dessa forma, o aluno não só aprende resolver o problema proposto, mas como lidar com novas dificuldades que a ele serão apresentadas. Neste sentido Orey [19] afirma que a metodologia ABP é frequentemente abordada em um ambiente de equipe com ênfase na construção de habilidades relacionadas à tomada de decisão consensual, diálogo e discussão, manutenção da equipe, gestão de conflitos e liderança de equipe.

Senna e Lopes [23] ressaltam que a expressão Aprendizagem Baseada em Projeto surge, às vezes, como sinônimo de Aprendizagem Baseada em Problema, por aparecerem na língua inglesa como Project Based Learning e Problem Based Learning utilizando a mesma sigla – PBL, ou as vezes PjBL para o primeiro e PBL para o segundo, e mesmo que o desenvolvimento de um projeto possa ocorrer com a resolução de problemas, uma prática tem como foco o problema, e a outra, o projeto.

De acordo com Bender em [4], a Aprendizagem Baseada em Projetos é uma metodologia de ensino baseada no fato de os alunos confrontarem questões e problemas do mundo real que eles consideram significativos, determinar como abordá-los e, então, agir de forma colaborativa para criar soluções de problemas.

Neste trabalho foi proposto inicialmente um problema, a criação automatizada de uma rede privada de *blockchain*, que servirá como base para um projeto de mestrado e dado a proximidade das duas abordagens de aprendizado a metodologia de Aprendizagem Baseada em Projeto também foi utilizada.

Bender em [4] apresenta como base ou essencial para uma abordagem de Aprendizagem Baseada em Projeto as seguintes palavras ou conceitos:

- **Âncora:** a base para fazer a pergunta que serve para fundamentar a instrução em um cenário do mundo real.
- **Artefatos:** os itens que representam soluções possíveis para o problema ou aspectos da solução do problema, cenários de dramatização são incluídos.
- **Realização autêntica:** representa a ênfase, o tipo de coisas que os profissionais podem esperar fazer na vida real.
- **Debate:** este é um processo pelo qual os alunos passam para formular um plano para as tarefas do projeto.
- **Pergunta de direcionamento:** a pergunta principal que fornece o objetivo geral do projeto.
- **Voz e escolha do aluno:** representa que os alunos devem ter uma palavra a dizer na seleção do projeto e na formulação da questão essencial.

Levando em consideração esses conceitos e que este trabalho representando a primeira fase de um projeto apresenta-se o cenário mostrado na Tabela 1.

Tabela 1: Cenário ABP para fase 1 do projeto

Cenário ABP para Fase 1: Automatização da criação da rede <i>Blockchain</i>	
Âncora	O problema apresentado aos alunos deve ser a necessidade da automatização do processo de criação e preparação de uma rede <i>blockchain</i> privada que possa ser integrada a um sistema web já existente escrito na linguagem JAVA.
Artefatos	Um script que possa ser executado em diferentes sistemas operacionais e um roteiro de como utilizá-lo.
Realização autêntica	A rede <i>blockchain</i> em funcionamento.
Debate	Esse processo deve ser realizado em reuniões periódicas de gerenciamento de projetos.
Pergunta de direcionamento	A automatização do processo de criação e preparação de uma rede <i>blockchain</i> é um ativo valioso para integração de um sistema web em JAVA?
Voz e escolha do aluno	Os alunos devem ajudar a escolher as ferramentas e técnicas para o desenvolvimento do sistema.

### 3 TRABALHOS RELACIONADOS

Alguns trabalhos e abordagens para a introdução e ensino de tecnologia *blockchain* e computação podem ser encontrados na literatura, Rao e Dave [20] utilizaram uma abordagem de aprendizado baseado em exercícios de laboratório (*hands-on*) para ensinar os alunos de graduação os conceitos de IoT, computação em nuvem e também *blockchain*. O projeto consiste na criação de um sistema que deveria obter imagens, salvar registros criptografados imutáveis, transmitindo e armazenando-os na nuvem.

Os autores então dividiram o projeto prático em dois exercícios de laboratório, no primeiro os alunos deveriam realizar a captura da imagem, a transmissão e o armazenamento na nuvem. Para este primeiro exercício foi solicitado aos alunos que estudassem conceitos básicos de comandos Linux e a linguagem de programação Python, além disso foram instruídos sobre o básico da plataforma Raspberry Pi. No exercício prático os alunos então deveriam criar um código em Python para a captura de uma imagem utilizando o módulo de câmera do Raspberry Pi, posteriormente os alunos deveriam codificar a etapa de envio da imagem para uma conta criada no Google Drive.

No segundo exercício os alunos são apresentados previamente aos protocolos de segurança SHA-256, um conjunto de algoritmos de criptografia baseados em funções matemáticas *hash*. Neste exercício os alunos então deveriam converter a imagem capturada no primeiro exercício em uma cadeia de caracteres e então transformá-la

em código *hash* utilizando uma biblioteca de python chamada *hashlib*, segundo os autores utilizando este exercício os alunos puderam entender o fundamento de criptografia e demonstrar a característica de imutabilidade contido na base da tecnologia *blockchain*.

Apesar das afirmações dos autores sugerirem um ensino mais abrangente de *blockchain*, no trabalho descrito apenas foi apresentado o conceito de criptografia comumente usado neste tipo de rede, tópicos como instalação, configuração e o funcionamento real da tecnologia não foram abordados pelos autores, o trabalho apresenta alguns conceitos de segurança da informação, limitando a abordagem aos conceitos de criptografia. Mesmo não sendo explicitamente abordados, conceitos de Redes, Sistemas Operacionais e Programação foram exercitados no citado trabalho.

Uma outra abordagem para o ensino de *blockchain* foi descrita por Negash e Thomas [16], neste trabalho os autores apresentaram um projeto baseado em sete cenários da indústria para transmitir conhecimentos teóricos e técnicos (práticos) de *blockchain* para um conjunto de estudantes de negócios com poucos conhecimentos técnicos. Para exemplificar quatro dos sete cenários propostos pelos autores estão descritos abaixo:

- **Educação:** neste cenário é descrito a utilização de um sistema baseado em *blockchain* para a verificação e autenticação de diplomas, as universidades registram os diplomas numa rede *blockchain* pública que permite a verificação da autenticidade de um diploma posteriormente apresentado.
- **Saúde:** o cenário descreve a possibilidade de utilização da *blockchain* para o armazenamento e controle de prontuários médicos, segundo os autores uma abordagem com *blockchain* permite que pacientes tenham o controle de seus prontuários, permitindo acesso apenas aos dados necessários para cada atendimento.
- **Aviação:** neste cenário é descrito uma oportunidade de negócios onde as passagens aéreas poderiam ser vendidas entre passageiros com o auxílio de um sistema de *blockchain*, onde um indivíduo que comprasse uma passagem poderia vendê-la para outra pessoa diretamente, registrando a transação numa *blockchain* compartilhada com as companhias aéreas.
- **Cadeia de suprimentos:** o cenário descreve a automatização do controle de estoque de empresas, para isso utiliza um sistema *blockchain* baseados em contratos inteligentes com execução semi autônoma onde um pedido de compra pode ser lançado automaticamente quando o estoque da empresa estiver num nível determinado.

Os demais cenários utilizados pelos autores incluem a descrição de sistemas das áreas de Governança, Internet das Coisas (IoT) e FinTech (finanças digitais). Para promover uma experiência significativa aos estudantes os autores projetaram interações reais para para demonstrar a aplicabilidade da tecnologia, para isso utilizaram a infraestrutura da LinuxOne Foundation (com suporte da IBM), utilizando a plataforma Hyperledge-Fabric (plataforma de desenvolvimento *blockchain*), desenvolveram práticas para demonstrar os cenários propostos.

Apesar de uma descrição básica e de alguns exemplos práticos de funcionamento da tecnologia, nesta abordagem o foco é voltado mais para a apresentação das possibilidades de uso da tecnologia *blockchain* do que propriamente para a construção dos sistemas

descritos. Além disso, esta abordagem necessita de mais recursos de infraestrutura para serem aplicadas, o que pode inviabilizar sua utilização em algumas situações.

Uma terceira abordagem para o ensino de *blockchain* é o framework apresentado por Bornelus, Chi e Shahriar (2019), neste propõe a utilização de diversos laboratórios que de forma modular apresentam todos aspectos da aplicação da tecnologia *blockchain*. A descrição dos laboratório *hands-on* está apresentada abaixo:

- **Entendendo a segurança por trás da Blockchain:** segundo os autores o objetivo é apresentar a criptografia por trás dessa tecnologia - são demonstrados tópicos como - árvores Merkle, criptografia de curva elíptica e SHA256.
- **Laboratório prático - Criando seu próprio cripto-sistema:** O objetivo deste laboratório é apresentar aos alunos a plataforma Ethereum, utilizando a criação de contratos inteligentes usando a linguagem Solidity e o Remix, uma ferramenta poderosa para escrita de contratos diretamente no navegador.
- **Passado, presente e futuro:** O objetivo deste tópico é demonstrar os aplicativos de *blockchain* da vida real: são demonstrados exemplos como Bitcoin, AWS Quantum Ledger Database, Azure MS Blockchain, IBM Hyperledger, e a perspectiva de utilizações futuras da tecnologia *blockchain* como o Block-Lattice.
- **Laboratório prático dApps:** O objetivo deste laboratório é aumentar a capacidade de desenvolvimento do aluno, criando um aplicativo descentralizado (d-Apps), para isso são utilizadas ferramentas como Solidity, Ethereum, Truffle, Ganache, Meta Maxis entre outros.

A representação gráfica do framework com o conteúdo completo de cada laboratório é apresentada na Figura 2.

<b>Entendendo a segurança por trás da Blockchain</b>	- SHA256 - Árvore Merkle - Curva Elíptica - Chaves Pública-Privada
<b>Laboratório Prático: Crie Seu próprio cripto-sistema</b>	- Criando seu próprio cripto-sistema parte 1: Usando Solidity, Remix na plataforma Ethereum - Vários Artigos e eventos atuais sobre desenvolvimento blockchain
<b>Passado, Presente e Futuro do desenvolvimento Blockchain</b>	- Bitcoin e outras criptomoedas - Desenvolvimento de aplicações Ethereum - Block-Lattice - Vários artigos e eventos atuais sobre desenvolvimento blockchain
<b>Laboratório Prático: dApp cripto-sistema</b>	- Crie seu próprio cripto-sistema parte 2: Usando Ethereum, código aberto para criar seu ambiente local de desenvolvimento com Truffle e Ganache para lançar dApps

Figura 2: Conteúdos dos laboratórios *hands-on*, Adaptado de Bornelus, Chi e Shahriar [5].

Nesta abordagem a tecnologia *blockchain* é ensinada de forma bastante robusta e avançada, todos os conceitos são apresentados de forma teórica e em sequência são realizadas as atividades práticas para fixação dos conhecimentos apresentados. No entanto é necessário por parte dos alunos um nível mais avançado de conhecimentos teóricos fundamentais, nesta abordagem os professores constroem toda a base teórica para depois utilizarem os laboratórios para as práticas ensinadas, numa abordagem que utiliza a exposição tradicional do conhecimento com atividades mais práticas.

Neste trabalho os alunos devem de antemão terem determinado domínio sobre outras disciplinas de computação, sendo trabalhados conceitos mais avançados nos laboratórios sugeridos pelos autores.

#### 4 METODOLOGIA DESENVOLVIMENTO DOS SCRIPTS

Inicialmente a necessidade de criação de um *script* para inicialização e configuração de uma rede *blockchain* surgiu em um projeto para o desenvolvimento de um sistema, no entanto logo percebeu-se a possibilidade de uso deste *script* como recurso didático, uma vez que diversos conceitos da computação tiveram que ser estudados para sua criação. Dentre as restrições impostas pelo projeto de origem estavam a necessidade de código aberto, suporte à *smart contracts* e a compatibilidade da rede com a linguagem de programação JAVA. Desse modo, o primeiro passo para o desenvolvimento dos *scripts* foi a definição da plataforma *blockchain* a ser utilizada. Foram analisadas as redes Bitcoin, Ethereum, Hyperledger Fabric, Quorum, EOS e R3 Corda.

Na tabela na Figura 2 segue um *benchmark* com algumas características levantadas para a escolha da plataforma deste projeto dentre elas: proposta da plataforma, tipo de rede se permite ou não a participação de partes sem ser previamente autorizadas, protocolos de consenso, interfaces de programação de aplicações (em inglês: Application Programming Interface - API) disponíveis e o suporte para *Smart Contracts*.

A Ethereum *Blockchain* foi escolhida por garantir as restrições mencionadas e após as comparações notou-se que a possibilidade de criar uma rede não permissionada seria a ideal para atingir objetivos futuros do projeto inicial, já que este tipo de rede é projetada para permitir a participação pública (por exemplo, alguns aplicativos que dependem de dados gerenciados pelos usuários).

Com a plataforma escolhida a próxima questão a ser resolvida foi a escolha da forma de instalação que posteriormente deveria ser automatizada. Foram identificadas três formas distintas para a instalação da rede *blockchain* da Ethereum:

- através de sistemas de gerenciamento de pacotes;
- através da compilação de códigos fontes e;
- através de download de arquivo binário já compilado.

No primeiro caso, os sistema de gerenciamento de pacotes do Linux e do MacOS podem auxiliar na instalação da rede Ethereum, precisamos para isso, adicionar um repositório PPA no caso do Linux ou instalar o Homebrew no caso do MacOS, sendo que para o sistema da microsoft esta forma de instalação não está disponível. A problemática deste modo ficaria a cargo de seguir tutoriais desatualizados do Ethereum que poderiam indicar versões não mais suportadas em sistemas operacionais mais recentes, devendo fazer a correção das versões manualmente à medida que forem identificadas versões não mais existentes ou incompatíveis com dependências instaladas.

Para o segundo modo, algumas dependências são requeridas, sendo necessário baixá-las antes de se iniciar o processo de instalação. Aqui novamente, podemos ter problemas quanto a versão das dependências e do sistema operacional da máquina, o que no futuro poderia ser um complicador quanto a utilização das mesmas

Tabela 2: Benchmark das plataformas *blockchain*

	Bitcoin	Ethereum	Hyperledger	Quorum	EOS	R3 Corda
Principal uso	Criptomoeda	Plataforma genérica de <i>blockchain</i>	<i>Blockchain</i> voltado para empresas	Para aplicativos que reque-rem alto nível de privacidade.	Criar uma plataforma escalável para dapps em escala industrial	Plataforma especializada para a indústria financeira (ativos digitais)
Tipo de Rede	Não permissionada	Não permissionada ou permissionada	permissionada	Permissionada	Permissionada	Permissionada
Consenso	PoW	PoW, PoS	Kafka, PoET, BFT	QuorumChain, RAFT(basado)	DPOS	RAFT, BFT
Smart Contracts	Limitado	Sim	Sim	Sim	Sim	Sim
APIs	bitcoin-cli (RPC)	Java, Python, Javascript, Go, Rust, .NET, Delphi	CLI, REST, Java e Node.js	Ferramentas familiares da Ethereum	Javascript, Swift, Java	Kotlin, Java
Possui Código Aberto	Sim	Sim	Sim	Sim	Sim	Sim

dependências utilizadas em um tutorial já que estas poderiam apresentar depreciação e incompatibilidade ao passo que estas forem sendo atualizadas.

O último meio de instalação é através do download de arquivo binário, deve-se baixar o arquivo compactado e extrair-lo para sua utilização, este meio tem menores riscos de problemas com dependências, assim basicamente o problema que pode ocorrer é escolher um arquivo desatualizado e incompatível com seu sistema operacional, o que geralmente pode ser contornado baixando a versão mais atual do arquivo.

No entanto, todos os três meios têm em comum a desvantagem de não ter um único arquivo, ou um único comando 100% funcional em todos os sistemas operacionais, já que para cada um deles existe uma série de comandos específicos e/ou um link exclusivo para download dos arquivos necessários. Para este projeto, o intuito é fornecer um ambiente configurado e pronto para uso com menor esforço para instalá-lo. Assim, a fim de tornar os passos únicos para instalação e configuração da rede decidiu-se no primeiro momento pela utilização do Docker, que através de um *script* único criaria-se um contêiner linux ubuntu em uma versão 19.04 com seus comandos de instalação e configuração já predefinidos através do repositório PPA da Ethereum, já que o sistema operacional e sua versão serão sempre o mesmo, a desvantagem anterior não se aplica a esta abordagem.

A abordagem do docker, no primeiro momento pareceu eficiente, uma vez que foi possível criar e configurar nós da rede totalmente funcionais, mas para a comunicação de containers em máquinas diferentes até com o mesmo sistema base, são necessárias configurações adicionais de infraestrutura que aumentaram consideravelmente a complexidade do *script* fugindo da ideia inicial de simplicidade na instalação, então decidiu-se procurar outra abordagem.

Mesmo com as diferenças entre os sistemas operacionais anteriormente citados, para a confecção de um novo *script* foi retirado o container docker e adicionados todos os comandos necessários para

criar e configurar o ambiente nos três sistemas operacionais escolhidos, ficando a cargo do *script* primeiramente reconhecer qual o sistema operacional o usuário está utilizando e escolher qual a série de comandos deve ser executada. Para simplificar a quantidade de comandos, a abordagem selecionada foi o download de um arquivo binário que também é escolhido de acordo com o sistema em que for executado.

## 5 RESULTADOS

Foram desenvolvidos dois *scripts* cada um com o objetivo de iniciar um tipo de nó e alguns arquivos com configurações e parâmetros que serão utilizados durante a execução dos arquivos.

Antes de começar a utilizá-los, caso esteja utilizando o sistema operacional da microsoft, primeiramente instale o git através da url <https://git-scm.com/download/win> ou caso utilize o windows 10 o mais indicado seria ativar o Subsistema do Windows para Linux (WSL) seguindo as instruções oficiais em <https://docs.microsoft.com/pt-br/windows/wsl/install-win10>.

Os principais componentes do *script* são os arquivos:

- genesis.json
- boot.sh
- start.sh
- .accountpassword
- .privatekey

A seguir explicamos as principais funcionalidades de cada um destes componentes:

### 5.1 Arquivo Genesis

Para iniciar uma nova cadeia precisamos definir o bloco inicial com algumas configurações que indicaram como novos blocos serão inseridos, dentre estas definições destacamos:

- **config**: a configuração da *blockchain*. Em suas definições temos o "chainId", um identificador utilizado na proteção

contra ataque de repetição. Por exemplo, se uma ação é validada combinando certo valor que depende do ID da cadeia, os atacantes não podem obter facilmente o mesmo valor com um ID diferente.

- **coinbase**: é um endereço onde todas as recompensas coletadas com a validação de bloco bem-sucedida serão transferidas. Uma recompensa é uma soma do pagamento pela mineração e dos reembolsos da execução de transações de contrato. Como é um bloco de inicial, o seu valor não é relevante. Para todos os próximos blocos, o valor será um endereço definido pelo mineiro que validou esse bloco.
- **difficulty**: dificuldade de mineração, para desenvolvimento e testes define esse valor baixo para que você não precise esperar muito pelos blocos de mineração.
- **gasLimit**: o limite do custo do gás por bloco.
- **nonce**: é o número de transações enviadas de um determinado endereço. É usado em combinação com *mixhash* para provar que uma quantidade suficiente de computação foi realizada neste bloco.
- **mixHash**: um *hash* de 256 bits que, combinado com o "nonce", prova que uma quantidade suficiente de computação foi realizada no bloco. A combinação de "nonce" e *mixhash* deve satisfazer uma condição matemática.
- **parentHash**: é o *hash* do cabeçalho do bloco pai. Familiar a um ponteiro para o bloco pai necessário para formar uma cadeia real de blocos. Um bloco de gênese não possui um bloco pai, portanto, o resultado será apenas neste caso igual a 0.
- **alloc**: esse parâmetro é usado para pré-financiar alguns endereços com *ether* (criptomoeda da rede Ethereum). Ele contém dois parâmetros, o endereço da carteira que deve ser um *hash* de 160 bits e o número de *ether* com o qual uma conta deve ser financiada.

A seguir na Figura 3 temos o arquivo genesis com duas contas já pré-financiadas para não ser necessário criar uma conta manualmente e colocá-la para minerar a fim de receber fundos necessário para realizar transações.

## 5.2 Execução dos Scripts

A ferramenta apresenta dois *scripts* executáveis o **boot.sh** e **start.sh**, o primeiro responsável pelo nó de Boot (bootnode), o qual deve ser instanciado apenas uma única vez e apenas em uma máquina, e o segundo responsável pela instância de nós de aplicação e mineradores. As tarefas dos nós foram divididas para melhor observar as funcionalidades e tarefas executadas pelos nós da *blockchain*, de forma a tentar se aproximar de uma rede de múltiplas máquinas bem como veríamos com a rede em produção.

O processo executado por cada um dos *scripts* é basicamente o mesmo, com as diferenças apenas nas configurações necessárias para especialização de cada nó.

A Figura 4 mostra o fluxograma dos processos executados pelo usuário e pelos *scripts* ao iniciar cada nó componente da rede *blockchain*.

A seguir discutiremos mais a fundo o funcionamento e peculiaridade de cada um dos *scripts*.

```
{
  "config": {
    "chainId": 2289,
    "homesteadBlock": 0,
    "daoForkBlock": 0,
    "daoForkHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "eip150Block": 0,
    "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
    "byzantiumBlock": 0,
    "constantinopleBlock": 0,
    "petersburgBlock": 0,
    "istanbulBlock": 0,
    "ethash": {}
  },
  "nonce": "0x0",
  "timestamp": "0x00000000",
  "extraData": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "gasLimit": "0x2f000000",
  "difficulty": "0x000000",
  "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "coinbase": "0x0000000000000000000000000000000000000000000000000000000000000000",
  "alloc": {
    "0x0200000000000000000000000000000000000000000000000000000000000000": {
      "balance": "0x2000000000000000000000000000000000000000000000000000000000000000"
    },
    "f09976c76da5e64694278bac91d0e2acd0470d": {
      "balance": "0x2000000000000000000000000000000000000000000000000000000000000000"
    }
  },
  "number": "0x0",
  "gasUsed": "0x0",
  "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
}
```

Figura 3: Exemplo de um arquivo genesis.json

## 5.3 BootNode

Um passo importante para o correto funcionamento de uma rede privada conectada por vários nós e a definição de um nó central o qual os demais se ligaram. Nomeamos o *script* para criação deste nó como **boot.sh**.

Para a execução deste e dos próximos nós faz necessária a definição de alguns parâmetros referentes à conexão da rede. Todos os parâmetros estão definidos no início do *script* e podem ser editados ou passados por meio de *flags* na chamada de sua execução. Os parâmetros referentes ao nó do Boot e as *flags* utilizadas para alterar seus valores ao executar a função são:

- **VERSION (-v)**: Versão do arquivo binário do Ethereum a ser instalado.
- **NETWORKID (-n)**: Deve ser o mesmo valor do "chainId" presente no arquivo genesis.
- **BOOTDATADIR (-d)**: Pasta no computador em que os arquivos da rede serão armazenados. Por padrão: **\$HOME/.ethereum/private/boot**.
- **BOOTNODEKEY (-k)**: Um nó de inicialização pede uma chave hexadecimal e através dela será gerado um ID com um esquema de URL chamado "enode" para conexão de outros nós, deixamos esse valor pré-definido para podermos ter certeza da url de conexão que será utilizado pelos demais nós. Esse valor pode ser gerado pelo comando: **bootnode -genkey bootnode.key**.
- **BOOTNODEIP (-b)**: O IP da máquina em que será instanciado o bootnode.
- **BOOTNODEPORT (-p)**: A porta em que o boot node deverá expor à rede. Por padrão: **30301**.



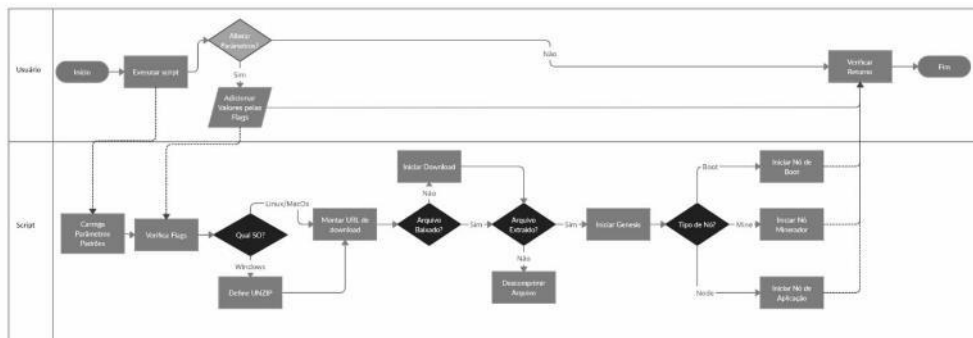


Figura 4: Fluxo geral dos scripts para iniciar um nó.

Depois de definidos os parâmetros, o *script* identifica qual o sistema operacional que está sendo utilizado e seleciona os comando adequados para baixar e descompactar, executar a rede, a Figura 5 mostra um exemplo deste trecho do *script*.

```
UNZIP=$(tar -xvf" #comando para descompactar e arquivo binário (linux e mac)
if [[ "$OSTYPE" == "linux-gnu" ]]; then #verifica se é um sistema linux
  OS="linux"
  EXT=".tar.gz" #extensão do arquivo binário
elif [[ "$OSTYPE" == "darwin*" ]]; then #verifica se é um sistema mac
  OS="darwin"
  EXT=".tar.gz" #extensão do arquivo binário
else #adivinha que se um sistema windows
  OS="windows"
  EXT=".zip" #extensão do arquivo binário
  UNZIP="unzip" #modifica o comando para descompactar o arquivo binário
fi
...
if [ -f "$FILEEXT" ]; then #verifica se é o arquivo já baixado
  echo "Arquivo encontrado"
else
  curl -O $URL #efetua download do arquivo binário
fi
if [ -d "$FILE" ]; then #verifica se é o arquivo foi descompactado
  echo "Arquivo descompactado"
else
  $UNZIP $FILEEXT #descompacta o arquivo binário
fi
```

Figura 5: Comandos de para baixar o arquivo conforme o sistema operacional identificado

Em seguida, na Figura 6 temos o trecho do *script* responsável pelos comandos que executam o nó central.

```
#inicializar a cadeia com o bloco genesis
$FILE/GETH --datadir=$DATADIR init genesis.json
#iniciar o bootnode da rede
$FILE/GETH --datadir=$DATADIR --nodekeyhex=$BOOTNODEKEY --networkid $NETWORKID
--net-estp:$BOOTNODEIP --port $BOOTNODEPORT >boot.log
```

Figura 6: Comandos de execução do nó central

O primeiro comando gera o bloco inicial e o segundo inicia a rede com os parâmetros definidos anteriormente que será executado em background e guardando as saídas da execução no arquivo **boot.log**.

Ao executar o arquivo **boot.sh** por linha de comando, caso não seja passados nenhum argumento a rede será instanciada com todos os parâmetros padrões, dentre eles o que pode inviabilizar a utilização da rede, caso incorreto, é o IP da máquina, logo certifique-se que este parâmetro foi definido corretamente.

Exemplo da utilização do *script boot.sh* é exibido na Figura 7 abaixo.

```
/boot.sh #iniciar com todos os parâmetros padrões
./boot.sh -i 192.168.1.158 #iniciar alterando ip do bootnode
```

Figura 7: Comandos para iniciar o boot.sh

O trecho abaixo apresenta a saída esperada escrita no log, indicando que a rede foi inicializada e qual é o endereço de conexão (enode) de novos nós.

```
INFO [09-24|18:00:36.897] Started P2P networking self
=enode://4e87faaa0ed677c3ec389f3ac37f8b0e366876f73e72
764e3518031daca322768befb783be5c4aea4200f3439f4361571
e860c38776142094adc35913964096b@192.168.1.158:30301
```

Se estiver utilizando o sistema windows certifique-se que tenha instalado o git e execute os *scripts* através do terminal do wsl ou git bash. Uma forma mais rápida de utilizá-lo seria dentro da pasta dos *scripts* clicar com o botão direito do mouse e escolher a opção "Git Bash Here", ou abri-lo através do menu de programas

#### 5.4 Nós de aplicação e mineração

Com o bootnode criado, podemos integrar à redes mais dois tipos de nós, o de aplicação (responsável por externar a API que será utilizado para inserção e consulta dos dados da *blockchain*) e outro nó para mineração dos dados enviados para serem inseridos na rede.

Para estes dois tipos de nós foi criado apenas um *script* sendo indicado qual o tipo de nó deseja ao iniciar o *script*. Dessa forma, a diferença na *script* para os dois tipos de nós é apenas os parâmetros indicados para execução da rede.

Os arquivos criados para este fim são o `start.sh` (*script* executável), `.accountpassword` (contendo a senha da carteira a ser pré-allocada) e `.privatekey` (chave privada da carteira pré-allocada). A senha e a chave privadas foram pré definidas por estarmos importando uma conta ao invés de criar uma nova, já que para pré-financiar uma conta devemos colocá-la no arquivo `genesis.json` antes de iniciarmos a rede.

Como no arquivo anterior, temos no início do arquivo a definição de parâmetros. Os parâmetros referentes a esses nós e as *flags* utilizadas para alterar seus valores ao executar a função são:

- **NODETYPE** (-t): Identifica o tipo de nó, aceita como valores: 'node' para um nó de aplicação, este definido por padrão, e 'mine' para um nó minerador.
- **OPERATIONTYPE** (-o): Aceita os comandos 'start' e 'stop' para, respectivamente, iniciar e para a rede *blockchain*.
- **MYNODEPORT** (-p): Porta em que será executada a rede no computador que está iniciando o nó. Por padrão: **30303**.
- **DATADIR** (-d): Pasta no computador em que os arquivos da rede serão armazenados. Por padrão: **\$HOME/.ethereum/private/node**.
- **BOOTNODEIP** (-i): Deve ser o IP da máquina que está rodando o bootnode.
- **BOOTNODEID** (-b): Deve ser o id ("enode") gerado pela execução do bootnode, se não foi alterado o **BOOTNODEKEY** no `boot.sh` este valor já está configurado.
- **BOOTNODEPORT** (-r): Porta em que está sendo executado bootnode. Por padrão: **30301**.
- **NETWORKID** (-n): É o mesmo "chainId" do arquivo `genesis.json`.

Estes parâmetros podem ser alterados diretamente no *script* ou passado como argumentos em sua execução. Um exemplo é demonstrado na figura 8.

```
./start.sh
./start.sh -t node
./start.sh -t mine -i 192.168.1.18
```

Figura 8: Comandos e parâmetros para iniciar a rede.

No primeiro comando iniciamos a rede com todos os parâmetros pré definidos, no segundo deixamos explícito que queremos iniciar um nó do tipo aplicação, e no último iniciamos um nó minerador indicando um outro valor para o IP do bootnode.

Quanto ao funcionamento do *script*, assim como no anterior após a definição dos parâmetros é identificado o sistema operacional e selecionado os comandos corretos. Em seguida é necessário iniciar a rede *blockchain* com o mesmo arquivo `genesis` do bootnode, e posteriormente o seguinte comando da Figura 9 serve para iniciar o novo nó e o conectando a rede já iniciada.

Neste comando podemos notar que comumente para os dois tipos de nó ao ser iniciados o argumento `-bootnodes` indica a url de

```
#NÓ DE APLICAÇÃO
$FILE/gets --datadir=$DATADIR --bootnodes "enode://$BOOTNODEID:$BOOTNODEIP:$BOOTNODEPORT" --networkid $NETWORKID --port $MYNODEPORT --verbosity=4
--rpc --rpcaddr "0.0.0.0" --rpcapi "eth,eth3,net,admin,debug,personal"
--rpcorsdomain "*" --syncmode="full" $IPC console

#NÓ MINERADOR
$FILE/gets --datadir=$DATADIR --bootnodes "enode://$BOOTNODEID:$BOOTNODEIP:$BOOTNODEPORT" --networkid $NETWORKID --port $MYNODEPORT --verbosity=4
--syncmode="full" --gasprice "0" --etherbase $ADDRESSACCOUNT
--unlock $ADDRESSACCOUNT --password $ACCOUNTFILE --mine
--miner.threads 1 $IPC
```

Figura 9: Comandos para iniciar e conectar um novo nó a rede.

conexão a rede iniciada pelo bootnode e o `networkid` confirma que o ID de todos os nós são iguais para compartilhar as informações.

O que define que o novo nó será de aplicação são os argumentos `"--rpc --rpcaddr --rpcapi --rpcorsdomain"`, responsáveis pela configuração de um servidor responsável pela API de comunicação com serviços externos, dentre estas configurações temos quais as funções que serão liberadas pela API pelo argumento `--rpcapi` e quais endereços IP terão acesso a requisições com `--rpcorsdomain`.

O nó minerador tem como características principais os argumentos `"--etherbase $addressAccount --unlock $addressAccount --password $accountFile --mine"`. Indicando assim, qual o endereço da base de *ether* ou seja o endereço da conta mineradora bem como desativando a conta para realizar as transações e o argumento `--mine` para que já seja iniciada a tarefa de mineração ao iniciar o nó.

Abaixo temos a saída esperada do nó de aplicação quando iniciado, podemos notar que a última linha indica que o servidor HTTP foi ativado, característica existente apenas nesse tipo de nó.

```
INFO [09-24|18:10:56.117] HTTP server started
endpoint=127.0.0.1:8545 cors= vhosts=localhost
```

Enquanto no próximo trecho temos a saída esperada da execução de um nó minerador, este tem como característica o início do trabalho de mineração indicado pela saída "Commit new mining work".

```
INFO [09-24|18:15:13.672] Commit new mining work
number=1 sealhash="c8ecb8...6394dc" uncles=0 txs=0
gas=0 fess=0 elapsed="216.9s"
```

Para utilizar a ferramenta, basta acessar o repositório ([https://github.com/meloflavio/private\\_etheruem\\_scripts](https://github.com/meloflavio/private_etheruem_scripts)) o qual estão descritos o seu funcionamento e apresenta um vídeo tutorial demonstrando sua utilização.

## 6 DISCUSSÕES

O desenvolvimento deste trabalho tinha o objetivo de apresentar um produto educacional destinado àqueles que pretendem iniciar seus estudos práticos na área do *blockchain*. Foram desenvolvidos *scripts* e um tutorial para a criação de um ambiente completo de uma rede Ethereum. Com estes *scripts* não só o ambiente é construído

como também é apresentando uma parte teórica sobre os conceitos necessários para criar uma cadeia de blocos.

Dessa forma, este trabalho pode ser utilizado para introduzir o conceito de *blockchain* bem como explicar seu funcionamento e detalhes necessários para sua configuração resultando em uma aula prática na qual o aluno poderá construir sua própria rede *blockchain*, exemplificando também um sistema distribuído. Todavia, um maior aprofundamento no básico da tecnologia *blockchain* é desejável, pois os conceitos apresentados estão concentrados apenas na estrutura do bloco.

Uma aula de Segurança em Tecnologia da Informação, por exemplo, seria interessante também ser apresentada a criptografia empregada na rede *blockchain* como uma técnica de proteção para comunicação segura. Já em aulas sobre Banco de Dados, pode se fazer um paralelo entre as duas tecnologias para indicar as diferenças e em que situação devemos utilizar cada uma dessas tecnologias. Neste sentido, a análise do *script* pode abordar conceitos de outras disciplinas, o *script* como um todo é um bom exemplo de algoritmo podendo ser utilizado em aulas como Introdução a Programação e Algoritmos, por exemplo as verificações do sistema operacional, se o download ou descompressão do arquivo já foram executadas podem demonstrar o funcionamento de estruturas de seleção.

Em aulas de Sistemas Operacionais fazendo uso do *script* pode-se abordar chamadas de sistema, explicar o que são processos, seus estados, execução em primeiro e segundo plano e o que os diferencia dos programas. Detalhes como o redirecionamento de portas, o servidor HTTP do nó de aplicação e as permissões de acesso à api da *blockchain* poderão também ser utilizados nas disciplinas que abordam configurações de redes.

A primeira versão deste *script* foi utilizada durante uma aula de Computação e Sociedade, disciplina do primeiro semestre do curso de Ciência da Computação da Universidade Federal do Tocantins, nesta aula foram apresentados cada um dos passos de execução do *script* e os conceitos envolvidos a fim de explicar novas tecnologias e abrir uma troca de informações com os conceitos familiares aos alunos. Neste caso, o maior resultado dessa experiência não é necessariamente o resultado do *script*, mas a exposição de todas áreas de estudos envolvidas em sua execução que possibilita o debate de todas as possibilidades que a computação nos traz.

Durante a apresentação, os alunos e o professor da disciplina puderam discutir cada um dos conceitos apresentados utilizando o *script*, os alunos puderam identificar de uma forma prática a utilização de diversas disciplinas que eles estudarão no decorrer de seu curso de graduação, nesta perspectiva diversos alunos interagiram com perguntas e comentários que demonstravam seus interesses e alguns conhecimentos básicos sobre cada um dos conceitos apresentados, de uma forma orgânica ocorreram debates mais aprofundados sobre os assuntos que os alunos demonstravam maior interesse.

Ao final da aula, alguns dos alunos continuaram discutindo sobre a apresentação, solicitando algumas dicas e materiais sobre as disciplinas que mais lhes chamaram a atenção. Neste momento, foi possível observar também que a apresentação despertou a curiosidade sobre algumas novas possibilidades oferecidas pela tecnologia *blockchain*.

Desse modo, a apresentação dos *scripts* nesta aula serviu não apenas para demonstrar a criação de uma rede de *blockchain*, mas também para ensinar alguns dos conceitos básicos das disciplinas

envolvidas no desenvolvimento dos *scripts*, além disso a apresentação despertou o interesse dos alunos em se aprofundarem nestas disciplinas demonstradas.

## 7 CONSIDERAÇÕES FINAIS

Por fim, este produto educacional, ou recurso didático, pode ser utilizado por outros professores em sala de aula para apresentar o comportamento de rede *blockchain* na prática e discutir os demais conceitos envolvidos. Além disso, o material pode auxiliar as pessoas que estão estudando por conta própria na criação de suas redes *blockchain* privadas iniciais na plataforma Ethereum, já prontas para interação com outros sistemas.

Além dos produtos já descritos neste trabalho, espera-se que este trabalho continue a evoluir, já estão em desenvolvimento para próximas etapas a implantação de exemplos de contratos inteligentes e um tutorial para compor este produto educacional. Essa e outras atualizações serão incorporadas ao repositório no GitHub.

## REFERÊNCIAS

- [1] Tesnim Abdellatif and Kei-Léo Brousmiche. 2018. Formal verification of smart contracts based on users and blockchain behaviors models. In *2018 9th IJITP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 1–5.
- [2] Andreas M Antonopoulos. 2014. *Mastering Bitcoin: unlocking digital cryptocurrencies*. O'Reilly Media, Inc.
- [3] Eduardo Fernandes Barbosa and Dácio Guimarães de Moura. 2013. Metodologias ativas de aprendizagem na educação profissional e tecnológica. *Boletim Técnico do Senac* 39, 2 (2013), 48–67.
- [4] William N Bender. 2012. *Project-based learning: Differentiating instruction for the 21st century*. Corwin Press.
- [5] Bertony Bornelus, Hongmei Chi, and Hossain Shahriar. 2019. A Novel Framework to Teach Hands-on Laboratory Exercises in Blockchains. (2019).
- [6] Jiin-Chiou Cheng, Narn-Yih Lee, Chien Chi, and Yi-Hua Chen. 2018. Blockchain and smart contract for digital certificate. In *2018 IEEE international conference on applied system invention (ICASI)*. IEEE, 1046–1051.
- [7] DANIEL C COSTA. 2010. *Administração de redes com scripts: Bash Script, Python e VRSript*. Brasport.
- [8] Antonio Sávio da Silva Pinto, Marcellene Rodrigues Pereira Bueno, Maria Aparecida Félix do Amaral, Milena Zampieri Sellmann, Sônia Maria Ferreira Koehler, et al. 2012. Inovação Didática-Projeto de Reflexão e Aplicação de Metodologias Ativas de Aprendizagem no Ensino Superior: uma experiência com "peer instruction". *Janus* 9, 15 (2012).
- [9] Edna de Almeida Rodrigues and Adriana Maria Proença de Araújo. 2007. O ensino da contabilidade: aplicação do método PBL nas disciplinas de contabilidade em uma instituição de ensino superior particular. *Revista de Educação* 10, 10 (2007).
- [10] Wenliang Du. 2011. SEED: hands-on lab exercises for computer security education. *IEEE Security & Privacy* 9, 5 (2011), 70–73.
- [11] Luiz Otávio Ramos Gavaza, Lais do Nascimento Salvador, and David Moises Barreto dos Santos. 2017. Uma experiência de aplicação de uma abordagem baseada em problemas no ensino de teoria da computação em sala de aula tradicional. In *Anais do XXV Workshop sobre Educação em Computação*. SBC.
- [12] Marco Iansiti and Karim R Lakhani. 2017. The truth about blockchain. *Harvard Business Review* 95, 1 (2017), 118–127.
- [13] Paola YB Ogawa Letouze, Patrick Letouze, JIM de Souza Junior, Bruna Laisy C Everton, Denise S Araujo, and Gentil Veloso Barbosa. 2020. Court-Ordered Government Debt Payment in Brazil: Perspectives for Blockchain Technology. *International Journal of Social Science and Humanity* 10, 4 (2020).
- [14] Xiuping Lin. 2017. Semi-centralized Blockchain Smart Contracts: Centralized Verification and Smart Computing under Chains in the Ethereum Blockchain. *Department of Information Engineering, National Taiwan University, Taiwan, ROC* (2017).
- [15] Satoshi Nakamoto and A Bitcoin. 2008. A peer-to-peer electronic cash system. *Bitcoin*. URL: <https://bitcoin.org/bitcoin.pdf> (2008).
- [16] Solomon Negash and Dominic Thomas. 2019. Teaching Blockchain for Business. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. IEEE, 1–4.
- [17] Benedikt Notheisen, Jacob Benjamin Cholewa, and Arun Prasad Shanmugam. 2017. Trading real-world assets on blockchain. *Business & Information Systems Engineering* 59, 6 (2017), 425–440.

- [18] Eduardo Oliveira and Angilberto Freitas. 2020. Os porquês da tecnologia blockchain ainda não ter sido popularizada: um ensaio teórico. *Revista Gestão & Tecnologia* 20, 1 (2020), 332–343.
- [19] Michael Orey. 2010. *Emerging perspectives on learning, teaching and technology*. CreateSpace North Charleston.
- [20] A Ravishankar Rao and Riddhi Dave. 2019. Developing hands-on laboratory exercises for teaching STEM students the internet-of-things, cloud computing and blockchain applications. In *2019 IEEE Integrated STEM Education Conference (ISEC)*. IEEE, 191–198.
- [21] Henrique Martins Rocha and Washington de Macedo LEMOS. 2014. Metodologias ativas: do que estamos falando? Base conceitual e relato de pesquisa em andamento. *IX Simpósio Pedagógico e Pesquisas em Comunicação, Resende, Brazil: Associação Educacional Dom Boston* 12 (2014).
- [22] John R Savery. 2015. Overview of problem-based learning: Definitions and distinctions. *Essential readings in problem-based learning: Exploring and extending the legacy of Howard S. Barrows* 9 (2015), 5–15.
- [23] Célia Maria Piva Cabral Senna and Graziela Miê Peres Lopes. [n.d.]. Aprendizagem baseada em projetos como forma de inclusão. ([n. d.]).
- [24] Elianny Sousa Silva, Brenda Jamille Costa Dias, João Lucas Moraes Souza, and Mariana Souza de Lima. 2019. Aprendizagem baseada em problema aplicada no ensino de urgência e emergência na enfermagem: um relato de experiência/Learning based on a problem applied in emergency and nursing education in nursing: an experience report. *Brazilian Journal of Health Review* 2, 4 (2019), 2525–2529.
- [25] José Itamar Souza Junior, Denise Sampaio de Araujo, Gentil Veloso, and Patrick Letouze. 2019. An international accreditation system for healthcare professionals based on blockchain. *International Journal of Information and Education Technology* 9, 7 (2019), 462–469.
- [26] Brasil. Tribunal de Contas da União. 2020. Levantamento da tecnologia blockchain. (2020). <https://portal.tcu.gov.br/levantamento-da-tecnologia-blockchain.htm>

## Roteiro para *deploy* e utilização de contratos inteligentes

Esse script foi desenvolvido com o objetivo de demonstrar o funcionamento de um contrato inteligente com a rede *blockchain* privada da plataforma Ethereum criada com esta ferramenta.

Como nos arquivos principais o script contém alguns parâmetros:

1. GETHPATH, esse deve informar a localização da ferramenta Geth, pode ser deixada em branco se o esquema de arquivos dessa ferramenta não for alterado.
2. VERSIONGETH, esse parâmetro deve ser o mesmo que o utilizado nos scripts para criação da rede.
3. SOLCVERSION, define a versão do compilador da linguagem Solidity. Padrão: 'v0.8.1'
4. CONTRACT, define o arquivo do contrato inteligente a ser compilado. Padrão: Profissional.sol

Para executar este script e compilar o contrato inteligente digite no terminal dentro da pasta *solidity* do projeto:

```
./contract.sh #Compila e prepara o contrato para ser utilizado na rede blockchain
```

Este script primeiramente verifica o sistema operacional para montar a URL do compilador de contratos Solidity e dá permissão de execução a este arquivo após o download ser concluído.

Após o download o contrato é compilado gerando dois arquivos, um abi e outro bin.

Para utilizá-los dentro da *blockchain*, estes precisaram ser transformados em variáveis Javascripts. Dessa forma, o script pega o conteúdo destes arquivos compilados e cria variáveis Javascript que podem ser importadas no console da ferramenta Geth.

Caso os parâmetros da ferramenta Geth estejam corretos o console é iniciado, caso contrário utilize a ferramenta Geth do seu computador para acessar o console através do comando:

```
./PATH_GETH/geth attach http://localhost:8545 #Conecta-se com o servidor  
RPC do geth e abre um console de comandos, o servidor localhost:8545 é o valor padrão
```

### Comandos dentro do geth

Dentro no console digite a seguinte sequência de comandos:

1. Carregar os arquivos compilados utilizando a API

```
loadScript('ProfessionalAbi.js')
```

```
loadScript('ProfessionalBin.js')
```

2. Criar variáveis com o conteúdo dos contratos compilados

```
var comp = "0x" + ProfessionalBin
```

```
var abi = ProfessionalAbi
```

3. Definir valor de gas necessário para o *deploy* do contrato e a conta padrão para realizar as transações

```
eth.defaultAccount = eth.coinbase
```

```
var gas = eth.estimateGas({from: eth.coinbase, data: comp})
```

4. Utilizar a função **contract** para criar um objeto Javascript com a interface de um contrato da plataforma Ethereum.

```
var factory = eth.contract(abi)
```

```
#Isso permite que você interaja com contratos inteligentes como se fossem objetos JavaScript.
```

5. Criamos uma instância do contrato e o enviamos para mineração.

```
var professionalContract = factory.new({data: comp, gas: gas}, function(e, contract) {
  if(!e) {
    if(!contract.address) {
      console.log("Contract transaction send: TransactionHash: " + contract.transactionHash + " waiting
to be mined..");
    } else {
      console.log("Contract mined! Address: " + contract.address);
      instance = web3.eth.contract(abi).at(contract.address);
      console.log("Current val: " + instance.get.call());
      gas = instance.set.estimateGas()
      console.log("Gas: " + gas);
      instance.set.call({gas: gas}, function(error, result) {
        console.log("RESULT: " + result);
        console.log("ERROR: " + error);
        console.log("Current ---" + instance.get.call());
        if(!error) {
          console.log("RESULT ---" + result);
        } else {
          console.log("ERROR ----" + error);
        }
      });
    }
  } else {
    console.log(e);
  }
});
```

Neste momento devemos aguardar o término da mineração e a mensagem que informa o endereço em que o contrato foi registrado.

```
Contract mined! Address: 0x0000000000000000 #Exemplo de retorno
```

Com a confirmação da mineração podemos utilizar a variável contrato em que guardamos a referência do *smart contract* para interagir com ele. Assim, podemos executar o método **setProfessionalDetails**:

```
professionalContract.setProfessionalDetails(eth.accounts[0], "Profissional
1", "000000000000", "123CRMTO", "profissional@gmail.com", "63 99999-9999")
```

Esse método não retorna nenhuma informação e mesmo que chamemos o método **getProfessionalDetails**, não será retornado nada na tela. Isso ocorre, porque não é possível obter o valor de retorno de uma função, devemos criar os eventos relevantes no contrato inteligente para que retornem os valores que queremos.

No nosso contrato de exemplo temos então o seguinte evento "**showDetails**" responsável por nos entregar as informações do contato quando executado o método "**getDetails**".

```
contract Professional {

    event showDetails(string stringDetails);

    function getDetails() public {
        stringDetails = string(abi.encodePacked("Profissional - Nome: ", nome, ", CPF: ", cpf, ",
Registry: ", registry, ", Email: ", email, ", Telephone: ", telephone));
        emit showDetails(stringDetails);
    }
}
```

Executando o método "**getDetails**" guardamos o hash da transação em uma variável:

```
transactionHash = professionalContract.getDetails()
```

E para visualizarmos o retorno do evento precisamos acessar as informações da transação:

```
eth.getTransactionReceipt(transactionHash).logs[0].data
```

A informação retornada está codificada conforme a especificação ABI do contrato, muitas das APIs ao serem utilizadas já fazem a decodificação como não estamos utilizando nenhuma API com essa função para decodificarmos facilmente a informação podemos entrar na página: <https://adibas03.github.io/online-ethereum-abi-encoder-decoder/#/decode>. E colamos o retorno do último comando indicando que o que está codificado é uma "string". Veremos assim o resultado:

```
Profissional - Nome: Profissional 1, CPF: 000000000000, Registry: 123CRMTO, Email:
profissional@gmail.com, Telephone: 63 99999-9999
```

## Apêndice 4 – Cenários para simulação de uso

**Cenário 1**

**Tipo de Usuário:** profissional de saúde

---

**Ações:**

- Cadastro de profissional;
  - Solicitação de Acreditação;
  - Conferir resultado da solicitação;
- 

**Resultados Esperados:**

- Criação de um novo registro de profissional;
  - Criação de um registro de solicitação de acreditação;
  - Verificação do resultado de uma solicitação;
- 

**Contextualização do cenário:** Sendo você um profissional de saúde interessado em uma acreditação nas competências X, Y ou Z das Instituições W, P ou O, por favor, realizar sua solicitação utilizando o sistema.

---

**Cenário 2**

**Tipo de Usuário:** Usuário de uma organização participante do IAS.

---

**Ações:**

- Cadastrar notas para etapas do processo de acreditação profissional;
  - Deferir/Indeferir a solicitação de acreditação;
- 

**Resultados Esperados:**

- Cadastro correto das notas para cada uma das etapas do processo solicitado;
  - Exibição da solicitação para avaliação com a exibição das notas das etapas avaliadas;
  - Geração automática do certificado caso solicitação deferida;
- 

**Contextualização do cenário:** Sendo você um profissional instrutor da Organização Acreditadora X, realizar a avaliação das cada uma das etapas de uma solicitação cadastrada no sistema. Após o cadastro das notas das etapas, realizar a avaliação final da solicitação deferindo ou indeferindo o processo.



---

**Instruções Técnicas:**

**Avaliar Etapas:** Utilize o menu lateral clicando na opção **Administração** e no submenu **Grupo de Avaliação** para acessar a lista de etapas avaliadas. Utilizando a tabela seleciona a etapa, a nota poderá ser cadastrada clicando na célula correspondente na tabela exibida e posteriormente salva utilizando o botão **salvar**. Repetir o processo para todas as etapas.

**Avaliar Solicitação:** Clique na opção **Solicitação** no menu lateral e utilize a opção **Solicitação de acreditação** para acessar a lista de solicitações, utilize o botão **finalizar** no registro selecionado, por fim selecione uma das opções **Deferir** ou **Indeferir** e salve o resultado;

---

**Cenário 3**

**Tipo de Usuário:** Usuário comum (população em geral)

---

**Ações:**

- Consultar os certificados de acreditação de um profissional ou organização;
- Verificar validade de um certificado de acreditação;

---

**Resultados Esperados:**

- Consultar certificados de acreditação de profissionais ou organizações
- Validar um determinado certificado;

---

**Contextualização do cenário 3.1:**

Precisando realizar um determinado procedimento de saúde utilize o sistema para consultar se uma organização X ou um profissional Y possuem competência para tal.

**Contextualização do cenário 3.2:**

De posse dos certificados de acreditação X e Y, verificar sua validade utilizando o Sistema IAS.

---

## Apêndice 5 – Questionários aplicados no estudo de caso instrumental

**Questionário A (antes do vídeo)**

Responda a esta pesquisa rápida e conte-nos sua opinião. O questionário é composto de afirmações na escala Likert. Nessa escala você tem 5 opções entre:

**(1) Discordo totalmente; (2) Discordo parcialmente; (3) Neutro; (4) Concordo parcialmente; (5) Concordo totalmente.**

Escolha a opção que mais se adequa a sua opinião em relação a cada afirmação.

A1) Eu quero participar desta atividade.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

---

B1) Eu entendo o que é uma criptomoeda.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

---

C1) Eu entendo o que é *blockchain*.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

D1) Eu entendo o que é um sistema de acreditação de profissionais.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

---

E1) Eu acho importante a certificação de profissionais de saúde.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

---

F1) Eu acho importante a acreditação de instituições de saúde.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

---

G1) Eu entendo o que é o sistema IAS - Sistema Internacional de Acreditação de profissionais em saúde.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

---

H1) Eu entendo a importância do sistema IAS.

	1	2	3	4	5	
Discordo totalmente	( )	( )	( )	( )	( )	Concordo totalmente

---

**Questionário B (depois do vídeo)**

Responda a esta pesquisa rápida e conte-nos sua opinião. O questionário é composto de afirmações na escala Likert. Nessa escala você tem 5 opções entre:

**(1) Discordo totalmente; (2) Discordo parcialmente; (3) Neutro; (4) Concordo parcialmente; (5) Concordo totalmente.**

Escolha a opção que mais se adequa a sua opinião em relação a cada afirmação.

A2) Eu vi o vídeo sobre a acreditação/certificação de profissionais e instituições de saúde

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

---

B2) Eu entendo o que é uma criptomoeda.

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

---

C2) Eu entendo o que é *blockchain*.

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

D2) Eu entendo o que é um sistema de acreditação de profissionais.

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

---

E2) Eu acho importante a certificação de profissionais de saúde.

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

---

F2) Eu acho importante a acreditação de instituições de saúde.

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

---

G2) Eu entendo o que é o sistema IAS - Sistema Internacional de Acreditação de profissionais em saúde.

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

---

H2) Eu entendo a importância do sistema IAS.

1    2    3    4    5

Discordo totalmente    ( ) ( ) ( ) ( ) ( )    Concordo totalmente

---

### Feedback do Teste do Sistema

Queremos saber seu feedback para continuar melhorando esse sistema. Responda a esta pesquisa rápida e conte-nos sua opinião. As respostas serão anônimas, com nove questões múltipla escolha apresentando uma afirmação com cinco opções de resposta:

- 1 - Discordo Plenamente;
- 2 - Discordo Parcialmente;
- 3 - Indiferente;
- 4 - Concordo Parcialmente e
- 5 - Concordo Completamente.

Escolha a opção que mais se adequa a sua opinião em relação a cada afirmação. E no final, temos uma pergunta subjetiva para que possa sugerir melhorias e funcionalidades para o sistema IAS.

A3) Eu fiz as atividades de teste do sistema IAS como usuário (TIPO DE USUÁRIO)

	1	2	3	4	5	
Discordo plenamente	( )	( )	( )	( )	( )	Concordo Completamente

---

B3) Como usuário (TIPO DE USUÁRIO), eu achei o sistema de fácil utilização.

	1	2	3	4	5	
Discordo plenamente	( )	( )	( )	( )	( )	Concordo Completamente

---

C3) O sistema funcionou corretamente.

	1	2	3	4	5	
Discordo plenamente	( )	( )	( )	( )	( )	Concordo Completamente

---

D3) Por favor, caso tenha encontrado erros no sistema relate eles aqui.

---

### Pesquisa de opinião

i) Eu gostei de participar dos testes do sistema IAS

	1	2	3	4	5	
Discordo plenamente	( )	( )	( )	( )	( )	Concordo Completamente

---

ii) Eu acredito que o sistema IAS dá maior segurança na busca de profissionais e instituições para atendimento de saúde

	1	2	3	4	5	
Discordo plenamente	( )	( )	( )	( )	( )	Concordo Completamente

---

iii) Eu acredito que um sistema de certificação e acreditação é importante para a confiabilidade na disseminação de informações.

	1	2	3	4	5	
Discordo plenamente	( )	( )	( )	( )	( )	Concordo Completamente

---

iv) Eu acredito que o sistema IAS pode ser uma ferramenta importante para o combate a epidemias e pandemias.

	1	2	3	4	5	
Discordo plenamente	( )	( )	( )	( )	( )	Concordo Completamente

---

Apêndice 6 – Código fonte do script de automação da rede *blockchain*

Link: [https://github.com/meloflavio/private\\_ethereum\\_scripts](https://github.com/meloflavio/private_ethereum_scripts)



Apêndice 7 – Código fonte dos *smart contracts* e do protótipo do IAS

Link: <https://github.com/meloflavio/ias>



Apêndice 8 – Vídeos tutoriais dos *scripts*.

Link: [https://youtube.com/playlist?list=PLL\\_zP8mu-7XQmpRBXEDj\\_Uk7nn8w6HkPO](https://youtube.com/playlist?list=PLL_zP8mu-7XQmpRBXEDj_Uk7nn8w6HkPO)



## Apêndice 9 - Impact Analysis of sisu at the Federal University of Tocantins



ISSN: 2230-9926

Available online at <http://www.journalijdr.com>

# IJDR

International Journal of Development Research

Vol. 11, Issue, 02, pp. 44756-44762, February, 2021

<https://doi.org/10.37118/ijdr.21166.02.2021>

RESEARCH ARTICLE

OPEN ACCESS

## IMPACT ANALYSIS OF SISU AT THE FEDERAL UNIVERSITY OF TOCANTINS

Flávio Fernandes De Melo, Carlos Eduardo Alves Cavalcante, Patrick Letouze Moreira,  
Andreas Kneip and \*José Itamar Mendes De Souza Júnior

Universidade Federal do Tocantins, Palmas - TO, Brasil

### ARTICLE INFO

#### Article History:

Received 11<sup>th</sup> December, 2020  
Received in revised form  
25<sup>th</sup> December, 2020  
Accepted 17<sup>th</sup> January, 2021  
Published online 28<sup>th</sup> February, 2021

#### Key Words:

SISU; ENEM, Entrance Exam,  
Applied Statistics.

\*Corresponding author: José Itamar Mendes  
De Souza Júnior

### ABSTRACT

This work aims to analyze the possible impacts of using the Unified Selection System (SiSU) at the Federal University of Tocantins. Thus, statistical and comparative research was carried out, using data generated from 2008 to 2018. Through statistical data, it was possible to compare results from the use of SiSU and the selection process previously used by the university.

Copyright © 2021, Flávio Fernandes De Melo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Citation: Flávio Fernandes De Melo, Carlos Eduardo Alves Cavalcante, Patrick Letouze Moreira, Andreas Kneip and José Itamar Mendes De Souza Júnior. "Impact Analysis of SiSU at the Federal University of Tocantins", *International Journal of Development Research*, 11, (02), 44756-44762.

## INTRODUCTION

The proposal to create a Unified Selection System (SiSU) aimed, according to the Ministry of Education (MEC) (BRAZIL, 2010b), to achieve greater democratization of opportunities for access to higher education institutions (HEIs) in the country. In this same direction, Nogueira *et al.* (2017) state that the use of SiSU should supposedly produce at least three initial advantages, namely:

- The reduction of the operational costs of the selection processes, which until then were carried out individually by each institution;
- Greater efficiency in filling vacancies, with an increase in the number of possible candidates; and,
- Increase social inclusion, making it possible for the poorest to apply for any vacancy at participating institutions at no additional cost, or the need to travel to take tests in other cities.

In this context, some studies were carried out to analyze the impacts of SiSU, Santos (2011) observed that after the adoption of SiSU at the Federal University of Recôncavo da

Bahia (UFRB), there was a considerable increase in the number of applicants and in the rate of filling in vacancies. Gómez and Torres (2015) identified in their research the reduction of dropout in engineering courses at the Campus Medianeira of the Federal Technological University of Paraná (UTFP). In another research on the topic, Barbosa *et al.* (2017), using binomial statistical tests, found significant variations in the dropout rates at the University of Uberlândia, when analyzing areas of knowledge separately, the author identified, for example, a significant increase in the dropout rate of Exact Sciences and Terra, on the other hand, identified a reduction in the dropout rate in the areas of Human Sciences, Linguistics, Letters and Arts. In turn, Li and Chagas (2017) studied, using data from ENEM from 2009 to 2014 and data from ENADE (National Higher Education Examination) of the years 2007 and 2008, the effects of SiSU on student migration and dropout nationwide. The study was determined through mathematical models that show that the use of the SiSU would increase the likelihood of interstate migration while reducing intra-state migration. Furthermore, according to the authors, the probability of evasion in the first year would also increase by 4.5% with the use of SiSU.

Therefore, the papers cited showed different possibilities, aspects, and impacts of the application of SiSU in each of the federal universities that were studied. Hence, this plurality of possibilities encourages the need to study the effects of SiSU in other universities that adhere to the Unified Selection System proposed by MEC. Thus, it is relevant to assess the impact of SiSU at the Federal University of Tocantins (UFT). Currently, approximately 3,300 vacancies are offered by UFT, distributed in classes of 53 undergraduate courses in its 7 campuses. From 2004 to 2015, the university admission process of students was carried out by the Permanent Selection Committee (COPESE), managed by the university itself. In 2010, in Ordinance No. 2 of the MEC (BRAZIL, 2010a), the use of the National High School Examination Notes (ENEM) was instituted, as a basis for the selection processes of public HEIs in the country. For this, SiSU was created, with national coverage, which allowed interested institutions to replace their individualized selection processes. UFT initially allocated only part of its vacancies to SiSU, maintaining the application of the entrance exam conducted by COPESE, its main selection process. The number of places available for SiSU has been systematically changed since 2010 until reaching the total number of places in 2015, remaining so until 2018.

Given the above, this work aims to verify the possible impacts of the adoption of SiSU by UFT. In pursuit of this purpose, statistics were generated and analyzed for data produced between the years 2008 and 2018. This period includes the implantation and use of SiSU, as the main university selection process.

**METHODOLOGICAL ASPECTS**

In order to carry out the evaluation of the impacts of SiSU at UFT, the data provided by the institution itself will be used, in order to generate evidence that supports conclusions or that leads to further in-depth studies. Thus, in this methodological path, the defined process will be presented to seek to answer whether the adoption of SiSU was a correct decision or not. In this work, the proposal of Santos (2007) was adopted, that is, for a statistical study to produce reliable results, it is necessary to fulfill some essential steps: problem identification, planning, data collection, presentation of information, its analysis and interpretation. Based on these steps, this work analyzes some statistics impacted by the adoption of SiSU by UFT. Aspects such as age, sex, dropout rate by type of selection process, and the entry of academics from other states were analyzed.

In particular, in this work, a special focus was given to new students through the university's affirmative action system. These actions were defined by UFT through resolutions No. 3A / 2004 (CONSEPE, 2004) and No. 14/2013 (CONSUNI, 2013), which instituted the reservation of vacancies for self-declared indigenous and quilombola candidates, respectively. Following the definition of the problem, the raw data was collected and organized. All data studied were requested from the Dean of Undergraduate Studies, which provided a general report of the information used for the academic census. This report was processed and imported into a database, so that the records were organized and grouped year by year, from 2008 to 2018. After obtaining and organizing, the data were then exported in a table format, to facilitate computational processing using algorithms in Software R. The R is a language and environment for statistical computing that offers a wide variety of techniques and graphics (R

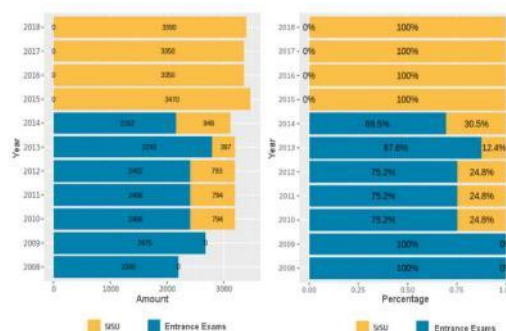
DEVELOPMENT CORE TEAM, 2018). With the aid of R, some metrics were calculated to allow the analysis of the information. Among the statistics generated, basic measures were used, such as: minimum, maximum, and average values. In addition, the standard deviation was determined as a measure of dispersion, whose utility, as described by Feijoo (2010), is to find a value that summarizes the variability of a data set, describing the degree of dispersion around a central position. In the last step, after the data processing has been completed and all predetermined statistics have been calculated, the results have been compiled and presented in a graphic format, facilitating visualization and understanding, which are analyzed and discussed in the following sections.

**DATA ANALYSIS**

In this section, we seek to separately analyze each of the statistics that have been generated regarding the impacts of using SiSU at UFT. In the following subsections, the availability of vacancies, the rate of filling vacancies, the number of calls per process, the age of the freshmen, sex, the origin of the freshmen, and data on the dropout rate of UFT students were analyzed.

**Vacancy:** The vacancies offered by SiSU in 2013 were reduced to almost 12.5%, maintaining this value until the first semester of 2014. For the second period of 2014, the number of vacancies for SiSU has changed again, this time to 50%, which determined a final annual rate of 30.5% in the aggregate of the two semesters. In 2015, 100% of undergraduate courses were allocated to SiSU, which became the university's main selection process. The total availability of places for SiSU remained until 2018.

It is important to note that, due to the non-filling of all vacancies even after making extra calls to the waiting list, as of 2015, the use of Complementary Selective Processes (PSC) was instituted for the remaining vacancies. The PSCs performed by UFT, also use the grades obtained in the ENEM test. However, these processes were entirely managed by the institution itself. Figure 1 presents graphs that represent the distribution of vacancies offered for the selection processes: own entrance exam and SiSU. In the years 2008 and 2009, the UFT selection process was exclusively the entrance exam. In 2010, SiSU started to be used as one of the selection processes, providing almost 25% of the total vacancies, with the same rate remaining until 2012.



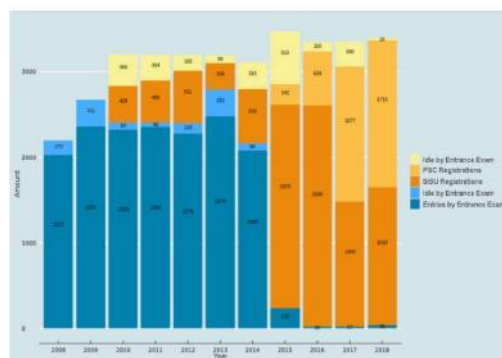
Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

**Figure 1 – Vacancies by Ticket Form.**



**Annual Registration:** In 2008 and 2009, the entrance exam was the only selection process, registering vacancy filling rates of 92.1% and 88.4% respectively. In 2010, already with the use of SiSU, 96.5% of the places destined for the entrance exam were filled, while the occupation of the places destined for SiSU was only 53.9%. In 2011, the entrance exam filling rate rose to 98.0%, in the same year the SiSU occupancy rate increased slightly, ending the year at 61.7%. The occupancy rates for the entrance exams and SiSU in 2012 were 94.8% and 76.9%, respectively. In 2013, there was a significant reduction in the number of places available for SiSU. However, the percentage of SiSU vacancies that were filled this year remained at the same level as the previous year, at 77.6%. Conversely, with a greater offer of vacancies this year, the percentage of occupation by the entrance exam fell to 88.7% of the total. After further changes in the vacancies offered by UFT's selection processes, in 2014 the occupation of vacancies from SiSU decreased to 66.8%.

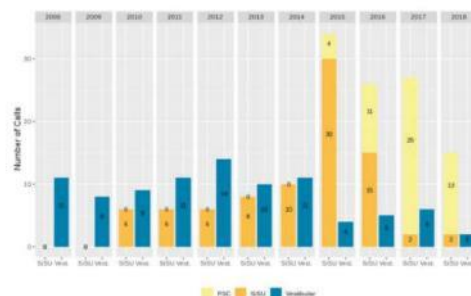
In turn, the occupation of vacancies offered by the entrance exam had significant growth, representing 96.3% of the vacancies offered for this process. As of 2015, the institution began to make the vacancies available to SiSU in full. However, even after some extra calls and the use of the PSC, the occupation of vacancies was low, registering 75.4%. Considering only the filling of vacancies by SiSU, the rate was only 68.4%. In the years 2017 and 2018, The UFT made only the regular call and one call from the SiSU waiting list, leaving all vacancies that remain idle in charge of the PSCs, making the number of enrolled by the latter more expressive than in fact by SiSU. In relation to 2017, we have 42% enrolled through the SiSU process, 45.5% through PSCs, and 8.4% idle vacancies. At the end of 2018, the total number of enrollments by PSCs registered an increase again reaching 49.4% of the total vacancies, against 46.4% of SiSU, thus we have the lowest vacancy rate registered in the analyzed periods, 0.7% of vacancies offered. It is also worth mentioning that, in addition to the use of several PSCs, there were some "extras entrance exams" and vacancy rescheduling processes, courses with lower demands for others with more waiting candidates, which also helped to reduce the number of vacant vacancies in these last few years. the full use of SiSU. Figure 2 shows a graph showing the status of the vacancies that were offered by UFT. The data were grouped separately for the entrance exam, SiSU and PSCs. The idle vacancies of SiSU and PSCs were grouped because they are the same vacancies.



Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 2. Status of vacancies by school year

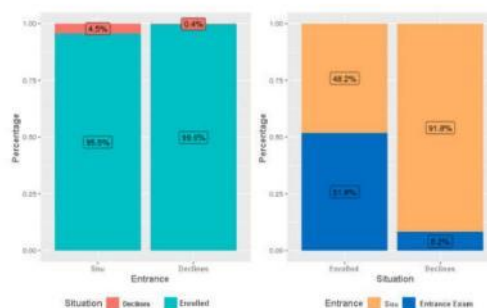
**Quantity of Calls:** In the analyzed period, there was a significant increase in the number of calls and complementary processes. From the information in Figure 3, it can be seen that the number of calls from the entrance exam changes little. There were small fluctuations until 2015, falling from this year onwards since only vacant vacancies were offered in classes that had already started. In the case of SiSU, until 2014 the number of calls was less than the number of calls from the entrance exam "vestibulares". With the use of SiSU as the main means of entry, it was necessary to significantly increase the number of calls to fill the vacancies offered. The number of SiSU calls went from 10 in 2014 to 30 in 2015 with 4 extra calls coming from PSCs, decreasing over the years, reaching the lowest level in 2018 with 2 calls to the SiSU national process and 13 other calls made by PSCs. It is also observed that, with the strategy of making only two calls from students from the process conducted by the MEC, the PSC becomes the process with the highest number of calls.



Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 3. Number of Calls per Entry Process

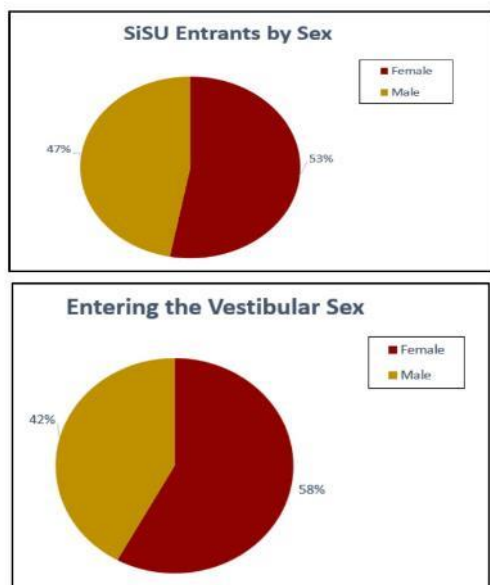
**Declinants:** Another aspect analyzed was the number of students who, although enrolled, for some reason gave up their places at the university before the effective start of the course and requested the cancellation of their enrollment. In this case, the university automatically cancels enrollment and classifies them as declining. The graphs in Figure 4 show the relationship between total enrolled and declining students from 2008 to 2018 grouped by the selection process. It is possible to identify that the percentage of declines in the institution is less than 5%. However, most of the declines in the analyzed period were selected through SiSU, representing 91.8% of the total.



Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 4. Relationship between declining and enrolled students by admission process

**Sex of Members:** Analyzing the data on the sex of the students who entered UFT, during the studied period. It was identified in the graph in Figure 5 that the number of female entrants exceeds the number of males in both forms of selection. The female sex represents 53% of those entering SiSU. Regarding the entrance exam, this percentage rises to 58% of newly enrolled students. It is important to emphasize that the database currently used is not fully adapted to consider new concepts on gender identity. For this reason, the analysis was restricted to the two basic genres already mentioned.



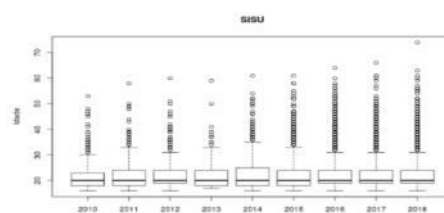
Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 5. Enrollments grouped by gender

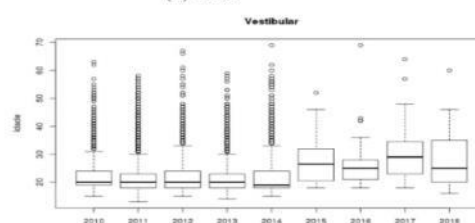
**Age of Members:** In order to perform a better analysis of the age of the freshmen, it was decided to consider only the period from 2010 to 2018, to allow a year-to-year comparison between the forms of selection studied. Also, the outliers of each sample were disregarded in order not to influence the efficiency of the analyzes. Therefore, the box diagram presented in figures 6(a) and 6(b) was used, which facilitate understanding. Considering only students selected by the entrance exam, in Figure 6 (b), there was a greater variation in the mean and in the age deviations.

Initially, the values found remained close to those of SiSU, with an average of 22 years and a maximum of 31 years. From 2015, the values fluctuated more, reaching the maximum age of 48 years in 2017. At the end of 2018, we found an increase in the average to 28 years and a 9.7 standard deviation. Analyzing the entrance exam data, there is a smaller number of outliers compared to SiSU.

**State of Origin of Academic:** One of the great advantages that should be obtained by using SiSU would be the increase in academic mobility in the country. This was one of the main forecasts of the MEC, which were contained in the proposal submitted to the National Association of the Directors of Federal Institutions of Higher Education (ANDIFES), in 2010 (BRASIL, 2010b).



(a) SiSU



(b) Vestibular

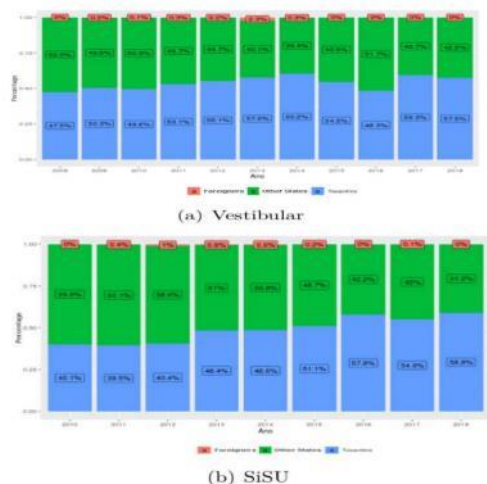
Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 6. Age distribution by admission process

Given this assumption, we seek to verify the real influence of SiSU on the statistics of the naturalness of UFT students. In this study, only interstate migration was analyzed, not considering mobility between municipalities in the state itself. When analyzing the graph in Figure 7 (a), it can be seen that the total number of students from the state itself remained almost always higher than students from other states. Only in the years 2008, 2010 and 2016, students from other states surpassed the total of Tocantins. Despite this, the number of non-Tocantins who entered the university entrance exam remained very expressive, always above 39%. The total number of foreigners selected, in turn, remained low, with the exception of 2013, which registered 2.3%.

The graph in Figure 7 (b) indicates that in the initial years of using SiSU, the MEC's forecast for increasing academic mobility came to fruition. In the period between 2010 and 2014, the total number of students from other states exceeded the number of students from the Tocantins, at a higher rate than that found in the numbers related to the entrance exam. However, from 2015 onwards, the number of new entrants from the Tocantins began to exceed the total number of students from other states, a different behavior than expected with the use of SiSU. The number of foreign students selected through SiSU represented a tiny portion of those entering, almost always being below 1% of the total. In general, the academic community at UFT, whether selected via entrance exam or SiSU, remains very diverse.

The number of non-Tocantins students selected, remains significant in the institution, since more than 39% of all students selected in the analyzed period, were not born in the state. However, the use of SiSU did not have a permanent impact on general academic mobility related to UFT.



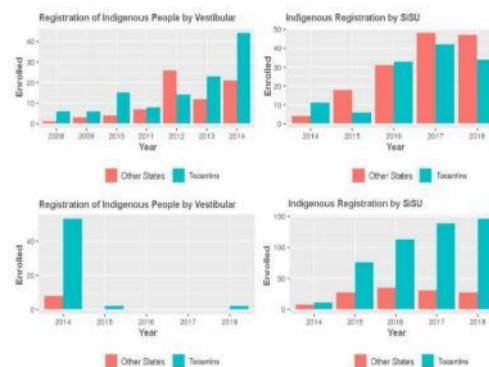
Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 7. Grouping of Origin of Enrolled Persons

**Affirmative Actions:** The main affirmative actions developed by UFT are the reservation of vacancies for indigenous and quilombola candidates. For this reason, special attention was paid to the analysis of the impact of SiSU, in relation to the origin of the newcomers by these actions. The data obtained are compiled in the graphs shown in Figure 8. In detail, the reservation of places for indigenous students began in 2003. During the period studied, most of the indigenous people entering through the entrance exam were from the Tocantins. Only in 2012, the total number of students from outside the state surpassed the number of Tocantins. In contrast, with the use of SiSU, indigenous students from other states had greater access to UFT vacancies. The number of non-Tocantins students surpassed the total number of local students entering SiSU, in most years analyzed, with the exception of the years 2014 and 2016, demonstrating that in the case of indigenous candidates, SiSU represented greater access of students from other states to places reserved by UFT.

In turn, the reservation of places for quilombola students began in 2014, the last year before the use of SiSU as the main means of selection. In this way, the numbers related to the entrance exam after 2014, consider only quilombolas entering through the extra-entrance exam for vacant vacancies, which explains the low number represented in the graph, and all of them being of the Tocantins. However, in relation to students entering through SiSU, it is possible to verify that the total number of quilombola students from the Tocantins always remains well above those from other states. Of the selected, the students' quilombolas born of Tocantins represented 79% of the total quilombolas registered by SiSU, against 21% in other states. However, despite remaining below the total number of Tocantins, the number of quilombolas from other states had a considerable increase in relation to those selected by the entrance exam, from 8 candidates approved in the 2014 entrance exam to an average above 25 with the use of SiSU.

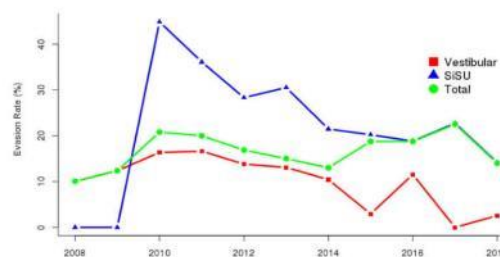
**Evasion:** A negative consequence of using SiSU as a means of entry is the possible increase in evasion, as determined by Li and Chagas (2017).



Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 8. Grouping of enrolled origin (affirmative actions)

The evasion of new students by SiSU appears to be associated with student migration and strategic behaviors, for example, the possibility of selecting other universities as a second option during enrollment at SiSU. To verify if this effect reached UFT, Figure 9 that shows the survey of students who gave up their enrollment at the institution, in the same year they entered. From the graph, we see that in the years 2008 and 2009, when the SiSU had not yet been implemented, the dropout rate represents an average of 11.21% of the total enrolled. In 2010 the dropout of enrolled students, selected by SiSU, registered a rate of 44.9%, raising the total dropout rate to 20.8%. Despite the beginning with such a high rate, over the years it was gradually reduced until 2012, when it obtained a rate of 28.3%, however, remaining quite high. In 2013, the evasion rate at SiSU had a slight jump to 30.5%, but since it represented only 25% of the vacancies available, the total evasion did not suffer major changes, decreasing by 1.9%, since the entrance exam, with greater weight over this year's rate, had a drop of 0.7% in its evasion rate.



Source: Authors' elaboration based on data from PROGRAD, UFT, 2019.

Figure 9. Comparison between dropout rates by entry mode

2014 was initially marked by a decrease in the number of vacancies offered by SiSU in the first semester and an increase in the second. The dropout rate was still higher than the entrance exam, about 21.5% against 10.4% in the annual aggregate. Total evasion, however, declined slightly reaching 13%, as a result of the 9.0% drop in SiSU evasion in relation to the previous year. From 2015 to 2017, with the adoption of 100% of vacancies by SiSU, the dropout rate is basically not affected by new entrants through the entrance exam, since the same is done only to fill idle vacancies.

Thus, the total dropout rate is basically represented through the SiSU, going up to 18.8% and continues to gradually rise until reaching 22.5% in 2017. However, in 2018 the total number of dropout students is calculated at 14, 0%, which represents a decrease of 8.5% compared to the previous year. In comparison to the effect of SiSU on the total dropout rate over the years, we can observe that regardless of the number of vacancies offered by this form of selection, there was an increase in the total dropout rate. According to the numbers surveyed, in the analyzed period, the average dropout rate from the entrance exam (disregarding the years without selection processes) was 13.0%. SiSU, in turn, had an average rate of 26.4% of dropout students. With the use of SiSU, UFT's average total evasion rate rose to 16.6%, resulting in an increase of 3.65% overall, close to that estimated by Li and Chagas (2017) in their article, of 4,5 percentage points.

### FINAL CONSIDERATIONS

One of the impacts of the adoption of SiSU at UFT was the low filling of vacancies offered in the process, which demanded the need to make multiple calls, raising this amount significantly. However, even with the high number of calls, it was still necessary to create and use complementary selection processes in addition to the relocation of vacancies, which generates an additional cost and effort on the part of the university management, not fulfilling one of the objectives intended by SiSU the reduction of operating costs. Another unwanted effect by SiSU, was the increase in the number of declining students related to this process, more than 90% of the total declining students in the studied period were students selected by SiSU. Regarding the characteristics of the freshmen, it is observed that the profile of the students, little changed, given the use of SiSU. It was identified that the average age and the general number of enrolled grouped by sex did not suffer any relevant impacts. Although they did not impact the average age, there was a small increase in the number of older people who joined through SiSU. When analyzing the academic background, it is noteworthy that the share of freshmen from other states has always been quite significant. With the implementation of SiSU, during the first three years, the representativeness of these students grew as an expected consequence of adopting a nationwide process. However, this effect has diminished over the years, demonstrating a temporary impact on the institution. In turn, the migration of students using affirmative actions has had a greater impact due to the adoption of SiSU, enabling a considerable increase in the access of these students to UFT. In the case of indigenous people, students from outside the state started to represent more than half of those selected in the modality referring to this affirmative action by SiSU. Similarly, access to UFT vacancies for quilombola students from other states has increased significantly with the use of SiSU as a selection process, practically tripling the number of quilombola students from other states at UFT. Regarding the permanence of academics, since the implantation of SiSU as a selection process, there was a relationship between the form of admission used and a higher dropout rate.

In this sense, although we cannot determine that it is the only cause, since the implantation of SiSU, the evasion rate of the institution increased, principally after 2015 when 100% of vacancies were allocated to SiSU. Despite some positive impacts with the use of SiSU, such as increasing the inclusion of indigenous and quilombola students from other states and

the initial reduction in the costs of the selection process, some other impacts proved to be quite negative. The reduction in initial costs is overcome by the need for several calls, complementary processes, and relocation of vacancies, thus requiring a great effort to fill the vacancies offered by the institution. The considerable increase in dropout in the first year together with the growth in the number of declines related to SiSU also testify against the effectiveness of the process. These facts maybe some of the motivators for the institution to take the entrance exam again in 2019, reserving only 50% of the places in undergraduate courses for SiSU, seeking to reduce the negative impacts while maintaining the benefits obtained. Thus, it is concluded that the use of SiSU as the only form of admission was not successful as expected, and it is still interesting to be used as a complementary process, making the form of admission of the institution a little more comprehensive. As future work, the mentioned impacts can be further investigated using, for example, individual studies by courses or areas of knowledge. In this way, we could identify a possible occurrence of the Simpson paradox (also known as the reversal paradox or Yule-Simpson effect), which according to Mheen and Shojania (2014) refers to an association or effect found in several subgroups, but which is canceled or reversed when data from these groups are aggregated. The use of new variables can generate situations similar to the results of Barbosa *et al.* (2017) that may differ from the results initially found in this work. For this reason, it is pertinent to deepen the studies carried out to obtain a more precise analysis of the results found.

### REFERÊNCIAS

- Barbosa, J. P. G. *et al.* 2017. A adoção do sisu e a evasão na universidade federal de Uberlândia (*The adoption of the system and evasion at the Federal University of Uberlândia*). Revista Ibero-Americana de Estudos em Educação, v. 12, n. esp., p. 722–738.
- Brasil. 2010. Ministério da Educação (*Ministry of Education*). Portaria normativa nº 2, de 26 de janeiro de 2010 - dispõe sobre o sistema de seleção unificada - SiSU (*Normative Ordinance N° 2, of January 26, 2010 - provides for the unified selection system - SiSU*). Diário Oficial da União.
- Brasil. 2010. Ministério da Educação (*Ministry of Education*). Proposta à Associação Nacional dos Dirigentes das Instituições Federais de Ensino Superior (*Proposal to the National Association of Managers of Federal Institutions of Higher Education*). p. 1, 2010. Disponível em: <[Http://Portal.Mec.Gov.Br/Index.Php?Option=Com\\_Content&View=Article&Id=13318&Itemid=310](http://Portal.Mec.Gov.Br/Index.Php?Option=Com_Content&View=Article&Id=13318&Itemid=310)>.
- Consepe, C. 2004. Resolução do conselho de ensino, pesquisa e extensão - CONSEPE nº 3a/2004 (*Resolution of the teaching, research and extension council*) - Aprova a implantação do sistema de cotas para estudantes indígenas no vestibular da Universidade Federal do Tocantins - UFT (*Approves the implementation of the quota system for indigenous students in the entrance exam of the Federal University of Tocantins*). Boletim Interno UFT.
- Consuni, C. U. 2013. Resolução do conselho universitário - Consuni nº 14/2013 (*University Council Resolution*) - Dispõe sobre a implantação do sistema de cota para os quilombolas em todos os cursos de graduação da Universidade Federal do Tocantins - UFT (*Provides for the implementation of the quota system for quilombolas in all*

- undergraduate courses at the Federal University of Tocantins). Boletim Interno UFT.
- Feijoo, A. M. L.C. 2010. A pesquisa e a estatística na psicologia e na educação [online] (*Research and statistics in psychology and education*). Rio de Janeiro: Centro Edelstein de Pesquisas Sociais, 109p. ISBN: 978-85-7982-048-9. Disponível em: SciELO Books <<http://books.scielo.org>>.
- Gómez, M. R. F., Torres, J. C. Discutindo o acesso e a permanência no ensino superior no contexto do sisu - sistema de seleção unificada (*Discussing access and permanence in higher education in the context of SiSU - unified selection system*). ORG & DEMO, v. 16, n. 1, p.69-88, 2015.
- LI, D. L., Chagas, A. L. S. 2017. Efeitos do Sisu sobre a migração e a evasão estudantil (*Effects of SiSU on student migration and dropout*). In: Encontro Nacional da Associação Brasileira de Estudos Regionais e Urbanos - ENABER. São Paulo: ABER.
- Mheen, P. J. Marang-van de., Shojania, K. G. 2014. Simpson's paradox: how performance measurement can fail even with perfect risk adjustment. [S.l.]: BMJ Publishing Group Ltd.
- NOGUEIRA, Cláudio Marques Martins et al. 2017. Promessas E Limites: O Sisu E Sua Implementação Na Universidade Federal De Minas Gerais (*Promises And Limits: Sisu And Its Implementation In The Federal University Of Minas Gerais*). Educ. rev., Belo Horizonte, v. 33, e161036, 2017. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0102-46982017000100116&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-46982017000100116&lng=en&nrm=iso)>. Access em: 11 dec 2019.
- Development Core Team. R. 2018. A language and environment for statistical computing. r foundation for statistical computing v, austria. 2016. URL <http://www.R-project.org>.
- SANTOS, C. 2007. Estatística descritiva (*Descriptive statistics*). Manual de Auto-aprendizagem, 2 ed. Lisboa: Edições Sílabo.
- SANTOS, J. dos. 2011. Política pública de acesso ao ensino superior: Um olhar sobre a utilização do ENEM/Sisu na Universidade Federal do Recôncavo da Bahia (*Public policy of access to higher education: A look at the use of ENEM/SiSU at the Federal University of Recôncavo da Bahia*). XI Congresso Luso Afro Brasileiro de Ciências Sociais, XI, n. 1.

\*\*\*\*\*

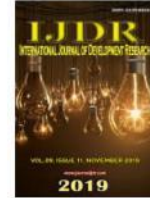


ISSN: 2230-9926

Available online at <http://www.journalijdr.com>

**IJDR**

*International Journal of Development Research*  
Vol. 09, Issue, 11, pp. 31311-31315, November, 2019



RESEARCH ARTICLE

OPEN ACCESS

## PARKING SPACE MANAGEMENT USING INTERNET OF THINGS

**\*Carlos E. A. Cavalcante, Flávio F. Melo, Patrick Letouze, Gentil Veloso, David Prata and Humberto X. Araujo**

Universidade Federal do Tocantins, Brasil

### ARTICLE INFO

#### Article History:

Received 09<sup>th</sup> August, 2019  
Received in revised form  
23<sup>rd</sup> September, 2019  
Accepted 17<sup>th</sup> October, 2019  
Published online 20<sup>th</sup> November, 2019

#### Key Words:

Smart Parking, Esp8266 Micro-controllers,  
Internet of Things, Monitoring of vacancies

\*Corresponding author: **Carlos E. A. Cavalcante**

### ABSTRACT

This paper proposes the use of an IoT system as a tool to reduce the time spent searching for parking spaces. The developed system consists of a monitoring prototype, a real-time database and a smartphone application. The monitoring prototype consists of a development board, based on ESP8266 microcontroller and wireless network modules, connected to ultrasonic sensors for monitoring parking spaces. The microcontroller uploads all data to a cloud-hosted database, which lets you identify which parking spaces are available in real time through a smartphone app. By identifying parking spaces available directly from your mobile phone, you can significantly reduce the time wasted while searching for a parking space.

Copyright © 2019, Carlos E. A. Cavalcante et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Citation: Carlos E. A. Cavalcante, Flávio F. Melo, Patrick Letouze, Gentil Veloso, David Prata and Humberto X. Araujo. 2019. "Parking space management using internet of things", *International Journal of Development Research*, 09, (11), 31311-31315.

### INTRODUCTION

The growth rate of vehicles in circulation in Brazil has been gradually decreasing since 2011, according to reports by the Brazilian Institute of Planning and Taxation (IBPT, in Portuguese) (Amaral, 2018). In 2017, the growth of the vehicle fleet was 1.37%, while in 2015 the growth was 2.67% and in 2011 the growth rate was 8.32%. Even with this reduced growth rate in the number of vehicles, the country has reached a total of over 65 million vehicles currently in circulation in the country, of which 41 million are passenger cars. The growth in the number of vehicles in circulation, especially passenger cars, is increasing the severity of various urban problems such as infrastructure and mobility such as congestion on public roads and the difficulty of finding a suitable parking spot. The number of parking spaces, whether public or private in Brazil, is often insufficient to accommodate the increasing number of vehicles, overloading public roads with cars parked on their side roads. However, the difficulty is often in locating available parking spaces in very large parking lots, wasting a lot of time in this search, and sometimes wasting available parking spaces that were not found by drivers.

The difficulty in finding parking places has caused more than half of Brazilian drivers not to shop in some shops, this data was obtained during the survey conducted in all state capitals, by the Credit Protection Service (SPC Brasil, in Portuguese) and by the National Confederation of Shopkeepers (CNDL, in Portuguese), which sought to analyze the impacts of urban mobility on local retail. From information like this, we can assume that the difficulty in finding a job has a significant and negative impact, not only on people's quality of life, but also on the country's economy. This justifies the search for solutions to avoid wasting time and resources when looking for a parking space, allowing them to be located more quickly and efficiently, thus improving the quality of life a little more and providing the most efficient use of parking lots. Technology has long become an important part of our daily lives, an example of our heavy reliance on technology today. It is related to the extensive use of smartphones, whether to communicate with friends and family through social networks, ordering a quick snack, or finding a place using GPS, the dependence on these devices is increasing, and we are getting used to new ways of interacting with these devices. The 29th Annual Survey on IT Use (Meirelles, 2018) conducted in 2018 by the Getúlio Vargas Foundation (FGV, in

Portuguese), reveals that the number of active smartphones in the country has reached more than 220 million units, the equivalent of more than one device. per inhabitant of the country. However, it is important to clarify that this does not mean that all Brazilians have such a device, since the same person may have more than one device. Similarly, or even because of the widespread use of smart- phones, the internet has become an almost ubiquitous tool in our daily lives, the current generation is almost always connected to the internet, a number that helps us visualize this scenario. released by the National Institute of Geography and Statistics (IBGE, in portuguese) as a result of the National Continuous Household Sample Survey (Continuous PNAD) , which reveals that 70.5% of Brazilian households have access to the Internet, of which 67% use their smartphones to connect to the network. Overall, figures such as those released by this research allow us to believe that the use of a car parking monitoring system based on IoT concepts could be a possible solution to minimize the difficulty of finding a space to park. It is important to highlight the existence of some systems proposals with similar objectives to the system developed in this work. A vacancy identification system was proposed in (Mahdi, 2018), which developed a prototype using pressure sensors, which at the moment a car is standing over them send a signal using Wi-Fi from a NodeMCU microcontroller. notifying the server, in this case Firebase, that the space has been taken.

The (Mahdi, 2018), approach was created by thinking about the traffic situation of Dhaka City in Bangladesh, which is suffering from major congestion caused mainly by cars parked in illegal places, such as in the middle of roads, sidewalks and other areas not allowed for traffic. parking. The application targets both the driver looking for a parking spot and the owners of a particular parking space who wish to rent it during business hours. This way, when a driver selects a vacancy in the application, the information and criteria defined by the vacancy owner are displayed. It is noteworthy that only one prototype was built, not really applied in a real situation, which raises doubts on the size of the sensors that should be used. More complex systems using low-processing microcomputers, infrared sensors, NFC readers, or real-time camera image-based readers are also described in work done in (Kim, 2018; Baroffio, 2015; Abdulkader, 2018; Kodali, 2018). However, the implementation of some of these proposals does not fit the reality of this project, generally due to the high cost of some components or the need for major modifications to local infrastructure. Another system was devised in (da Silva, 2016). where ultrasonic sensors connected to Arduino Uno microcontrollers and Zigbee radios were used to monitor each parking space, sending the data to a central formed with a Zigbee receiver connected directly to a computer. Unlike the previously mentioned, the system developed in this work uses a monitoring prototype with multiple ultrasonic sensors, these sensors are connected to a microcontroller with integrated wireless modules and can identify the occupancy of two slots simultaneously for each prototype built. The microcontroller transmits the data, over the internet, in real time, to an online database, allowing applications for smartphones developed to access this database to obtain the status of each of the monitored parking spots.

### Smart Parkings

The system developed in this work initially aimed to make it faster and more efficient to search for a parking space in a public parking environment. However, considering the

possible use of the system to support private parking environments, an option to reserve an available parking space was developed. To ensure drivers have better access to this information, a real-time communication approach using a smartphone app is required. To check the situation of the parking space, after defining the possible states: available/free, unavailable/busy and reserved. An ultrasonic sensor was chosen for detection of the parked vehicle. The choice of sensor type was determined by the cost-benefit ratio, since this sensor type has good reading efficiency and a relatively more affordable price than the others, such as infrared presence and motion sensors. Noting that these according (Ernani Moura Amaral Filho, 2016), these sensors, despite the names they carry, do not really work due to movement, but with temperature variations, they are therefore calibrated with the temperature of the human body, so they are not suitable for identifying objects. that do not emit this temperature range, for this reason the ultrasonic sensors better fit the system proposal. An ultrasonic sensor will emit a sound wave that, when encountering an obstacle, will reflect back towards the receiver module, being possible to identify the distance between the object and the sensor. object near the sensor, in the case of the system developed in this project, the presence of this object characterizes the occupation of the monitored parking space by a particular vehicle.

The information generated with the sensors is sent using a microcontroller with wireless internet module to an online database, that will record the status of parking spaces. This database can be consulted in real time to check the status of all spots in which sensors are being used. For the display of parking information, a mobile application was developed that receives data from monitored spaces, directly from the online database platform, providing real-time display of the status of each monitored space. This way, the driver can either go directly to the nearest available place or reserve one of them while heading straight for it. It is important to emphasize that once booked, the driver will have a certain time to occupy the reserved place, if the parking space is not occupied by the driver in time, the reservation will be automatically canceled, and the parking space will be available again.

**System Architecture:** The project was gradually developed in stages, first it was necessary to analyze each component individually, to ensure the correct functioning of the projected prototype, and then to test the necessary software for correct communication between all components of the system.

**Development platform:** The platform used for the development of this project was NodeMCU, which according to (de Oliveira, 2017) consists of an ESP8266 module - which is a 32-bit microprocessor with native support for wireless network connections, with a power and programming port, having 10 digital and one analog inputs. The choice of the ESP8266 was motivated by its low cost, coupled with the built-in wireless connection on the microcontroller board, which allows a more convenient way of communicating with a server to receive and send the data sent by the tests.

**Software and Systems:** For programming the chosen microcontroller, Arduino IDE was used, where the default programming language is C / C ++ and works on various platforms and operating systems such as Windows, Linux and macOS, Arduino IDE is ideal for creating various interactive environments and can be configured to use NodeMCU. The

mobile application developed in this project was built using Ionic which is a framework for mobile application development aimed at the development of hybrid applications and rapid development. After installing the application, it is required to login, if the user does not have a registered account yet, can create at this time. Once authenticated the user can select the option to display all monitored parking spaces, the status of each parking space is represented by colors, being green to indicate a available parking space, red for busy and gray for a reserved parking spot. A user can request the reservation of the parking spaces that are available at the moment, after the reservation, if the user who made the reservation no longer wants to use that parking space is possible to make the release, making it available to other users again, as shown in Figure 1. It is important to clarify that a reserved parking space remains unavailable to others until it is released by the driver who reserved it or exceeds the time limit for the user to park in the reserved parking space. The limit was initially set at 15 minutes after the reservation was confirmed. A great benefit of the mobile application is that the parking situation display is in real time, so when any car uses a parking space the application immediately changes the display of the parking space to busy for all users without reloading the page. This synchronization of information between the software and the microcontroller is performed by Firebase - a mobile and web application development platform, more specifically, the real-time database, which provides an API that enables application data is automatically synced across multiple clients. Firebase was chosen for its ability to store and synchronize data across all clients in real time and remain available when the app is offline. The schematic view of the project system can be observed in the Figure 2.

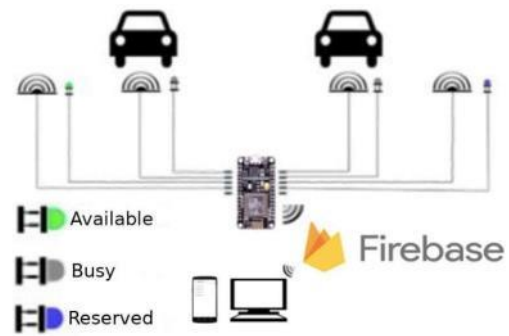


Figure 2. Proposed System

**Components and Modules**

After studying the basic concepts for the project development, it was necessary to organize the components, assemble the equipment and configure the projected prototype. Thus, the prototype works using a model structure constructed from the electronics and materials listed below:

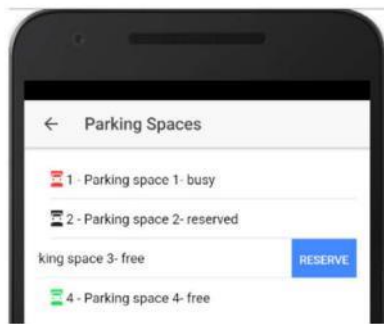
- Jumper wire;
- 1 Protoboard 810 points;
- 1 NodeMCU Esp8266;
- 2 Resistor 330 ohms;
- 4 Resistor 200 ohms
- 2 RGB LED Common Anode;
- 2 Ultrasonic Module HC-SR04.

The Table below shows the average values found in the market between January and February 2019. Of which, jumper wires and resistors, for being sold only in kits with a closed quantity of units can be reused for other projects or in the replication of new ones parking sensor prototypes.

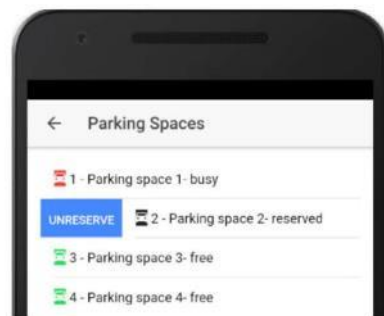
**Table 1. Initial Prototype Budget**

Component	QTY	Price per item	Price(sum)
Jumper wire (Kit 40pcs/lot)	1	\$2,0	\$2,0
Protoboard 810 points	1	\$ 4,0	\$ 4,0
NodeMCU Esp8266	1	\$ 7,6	\$ 7,6
Resistors (100pcs Assorted)	1	\$ 2,95	\$ 2,95
RGB LED Common Anode	2	\$ 0,20	\$ 0,40
Ultrasonic Module HC-SR04	2	\$2,30	\$4,60
<b>TOTAL</b>	-	-	<b>\$21,55</b>

**The Circuit:** The circuit is powered via USB cable by connecting a 5v output power supply to the microcontroller. On the digital pins of the development board are connected 2 LEDs and 2 ultrasonic sensors, these occupy the 10 available digital pins, all being interconnected using a protoboard. The LEDs are powered by 3.3v outputs, but the sensors used in the project need more power to function properly, so they are powered using the 5v output of the microcontroller. The circuit works as follows, according to the implemented algorithm, the ultrasonic sensors are in charge of calculating the distance from the end of the parking space to the car. The microcontroller reads the sensors and analyzes the situation of each monitored parking space. If an object is identified by the sensor within a distance of up to 100 centimeters over a period of 40 seconds, the microcontroller will consider the occupation of the monitored parking space. Consequently, the LED



(a) Reserve parking space



(b) Unreserve parking space

Figure 1. Mobile Application Screens



indicating the status of that spot will go out signaling the unavailability of the parking space that is now busy. On the other hand, when the vehicle moves away from the sensors, the indicator LED will light green, indicating that that space is now available, in parallel with this procedure occurs the synchronization of parking lot status with the Firebase database, which sends this information automatically available to all connected devices. Finally, using a web application, we can also change the status of a parking space from available to reserved, once this process is done by recording the status of that parking space in the database, this information is automatically synchronized with the microcontroller, which changes the LED color to blue, indicating that the location in question is reserved. The following Figure 3 shows the circuit used in the prototype:

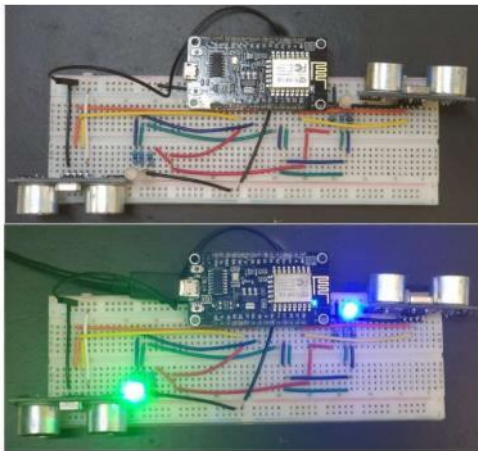


Figure 3. Initial prototype

For behavioral evaluation, the components and the designed system were submitted to the following tests:

**Component Testing:** The objective of this test is to evaluate the correct functioning of the monitoring device components. Initially, attempts were made to connect the microcontroller to wireless internet networks, seeking to receive and send data over the internet using the ESP8266 micro-controller. Subsequently, it was verified that the ultrasonic sensors used, performed the distance readings correctly, for this, objects of different sizes were placed in front of the sensors at different distances, comparing the sensor outputs with the actual distance measured in centimeters.

**System Test:** The purpose of this test was to evaluate if the system was working properly, behaving as expected by the project authors. To perform these tests, the monitoring prototype was placed at a height of 25 centimeters from the ground, under a support platform, the platform was arranged between two parking spaces in opposite directions at a distance of approximately 30 centimeters from the beginning of each In one of the spaces, as shown in the Figure 4, the sensors were configured to identify the presence of objects at a distance of up to 100 cm, for a minimum period of 40 seconds, these reference values were empirically defined by the authors for the realization. of the tests. The vehicles were positioned at four levels of distance in each parking space, observing the

state identified by the sensors through the web monitoring system, as exemplified in the Figure 5.



Figure 4. Test with both parking spaces available







Figure 5. Test with a busy parking space

## RESULTS AND DISCUSSIONS

Behaved as expected, the component test revealed a good accuracy of the sensors in the distance measurement, no differences were obtained greater than 1 millimeter between the distance identified by the sensor and the real distance. The microcontroller wireless module also worked as expected, being able to connect to many wireless networks without any problems, including networks routed through smartphones. The system was able to correctly identify the status of monitored parking spaces, according to the parameters implemented in the identification algorithm, signaling as unavailable, a space in which a vehicle was at a distance of 100 cm or less from the sensors for a period of time. not less than 40 seconds. Communication between the device and the web system occurred almost instantaneously, even when the internet connection was made through low speed networks, providing real-time status of monitored places, this behavior was also verified when using the web system to mark a place as reserved, a process in which the status indication LED was immediately changed to blue by the microcontroller. The states of monitored parking spaces, observing four measuring distances, are described in Table , and the outputs shown by the application for each measured distance are displayed.

Table 2. Monitoring Test Results

Distance App	Output
120 cm	 2 - Parking space 2- free
102 cm	 2 - Parking space 2- free
100 cm	 2 - Parking space 2- busy
50 cm	 2 - Parking space 2- busy

### Conclusion

In this work, an IoT-based system was designed to manage parking spaces in a parking lot. The system was developed using relatively low-cost components to build the necessary equipment. The project presented satisfactory results, reaching the objective of monitoring some parking spaces using ultrasonic sensors, and the internet as a means of providing the information generated to users. Thus, the viability and efficiency in the use of internet of things to control parking spaces has been demonstrated. This is an efficient and relatively low-cost way to control parking spaces, both in terms of system development, maintenance and installation. It was also possible to observe benefits by using the internet, not being necessary to use other communication resources such as bluetooth or radio devices to send the generated information, being sufficient the use of a web system connected using only the internet.

### REFERENCES

- A. D. R. da Silva, C. R. D. Sousa, F. R. Gomes, I. De, O. Régo, J. Cristina, F. Nunes, L. Renata, L. Castelo, V. D. M. Oliveira, N. Mendes, N. Mendes, P. Raissa, R. S. Silva, S. Gabrielle, C. Franco, F. Alves, and F. M. A. D. Araújo, "PROTÓTIPO DE ESTACIONAMENTO INTELIGENTE COM COMUNICAÇÃO SEM FIO," *Mostra Nacional de Robótica (mnr)*, vol. 3, pp. 1–5, 2013. Brasil, Instituto Brasileiro de Geografia e Estatística, "Características gerais dos domicílios e dos moradores 2017," Rio de Janeiro, pp. 1–8, 2018. [Online]. Available: <https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=2101566>.
- D.-h. Kim, S.-h. Park, S. Lee, and B.-h. Roh, "Iot platform based smart parking navigation system with shortest route and anti-collision," in *2018 18th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2018, pp. 433–437.
- Ernani Moura Amaral Filho, "Como funciona o sensor de presença," 2016. [Online]. Available: <http://datalink.srv.br/artigos-tecnicos/como-funciona-o-sensor-de-presenca/>
- F. S. Meirelles, "29a Pesquisa Anual do Uso de TI," *Fundação Getúlio Vargas*, vol. 29, pp. 1–24, 2018. [Online]. Available: <http://caesp.fgv.br/sites/caesp.fgv.br/files/pesti2018gvciaipt.pdf>.
- G. L. do Amaral, C. L. Yazbek, and J. E. Olenike, "FROTA BRASILEIRA DE VEICULOS EM CIRCULAÇÃO," *EMPRESÔMETRO - Inteligência de Mercado*, pp. 1–11, 2018. [Online]. Available: <http://materiais.ibpt.com.br/estudo-frotas>.
- L. Baroffio, L. Bondi, M. Cesana, A. E. Redondi, and M. Tagliasacchi, "A visual sensor network for parking lot occupancy detection in smart cities," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 745–750.
- M. D. Mahdi, Z. H. Anik, R. Ahsan, and T. Motahar, "Ez parking: Smart parking space reservation using internet of things," in *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 2018, pp. 113–118.
- O. Abdulkader, A. M. Bamhdi, V. Thayanathan, K. Jambi, and M. Al-rasheedi, "A novel and secure smart parking management system (spms) based on integration of wsn, rfid, and iot," in *2018 15th Learning and Technology Conference (L&T)*. IEEE, 2018, pp. 102–106.
- R. K. Kodali, K. Y. Borra, S. S. GN, and H. J. Domma, "An iot based smart parking system using lora," in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 2018, pp. 151–1513.
- S. de Oliveira, *Internet das Coisas com ESP8266, Arduino e Raspberry Pi*. NOVATEC, 2017. [Online]. Available: <https://books.google.com.br/books?id=E8gmDwAAQBAJ>

\*\*\*\*\*