UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CIÊNCIAS DA COMPUTAÇÃO

Leonardo Gideão Costa Rocha

Blockchain Aplicada em uma Carteira Digital Acadêmica para Facilitar o Controle Curricular Estudantil

Florianópolis
2023

Leonardo Gideão Costa Rocha

Blockchain Aplicada em uma Carteira Digital Acadêmica para Facilitar o Controle Curricular Estudantil

Trabalho de Conclusão de Curso submetido ao Curso de Graduação em Ciências da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para obtenção do título de Bacharel em Ciências da Computação.
Orientador: Lucas Palma, Me. em Ciência da Computação
Coorientador: Prof. Jean Everson Martina, Dr. em Ciência da Computação

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rocha, Leonardo Gideão Costa
Blockchain Aplicada em uma Carteira Digital Acadêmica para Facilitar o Controle Curricular Estudantil / Leonardo Gideão Costa Rocha ; orientador, Lucas Palma, coorientador, Jean Everson Martina, 2023.
112 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciências da Computação. 2. Blockchain. 3. Jornada Estudantil. 4. Contratos Inteligentes. 5. Hyperledger Fabric. I. Palma, Lucas . II. Martina, Jean Everson . III. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. IV. Título.

Leonardo Gideão Costa Rocha

Blockchain Aplicada em uma Carteira Digital Acadêmica para Facilitar o Controle Curricular Estudantil

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Ciências da Computação e aprovado em sua forma final pelo curso de Graduação em Ciências da Computação.

Florianópolis, 09 de Julho de 2023.

Profª. Lúcia Helena Martins Pacheco, Dra.
Coordenadora do Curso

Banca Examinadora:

Lucas Palma, Me. em Ciência da Computação
Orientador
Universidade Federal de Santa Catarina

Johann Westphall, Me.
Avaliador
Universidade Federal de Santa Catarina

Gabriel Estevam de Oliveira, Me.
Avaliador
Universidade Federal de Santa Catarina

Dedico este TCC ao meus familiares e amigos que me acompanharam durante esta jornada.

AGRADECIMENTOS

O desenvolvimento e escrita deste TCC me acompanhou por dois anos da minha vida acadêmica, onde durante este processo tive vários altos e baixos tanto na graduação quanto na minha vida pessoal. Contudo, por mais que a jornada possa ter sido árdua, ela não foi feita sozinha, e graças a pessoas importantes na minha vida foi possível superar cada uma das dificuldades existentes. Sendo assim, dedico esta seção como forma de agradecimento aqueles que possibilitaram de alguma maneira, seja diretamente ou indiretamente, tornar minha vida mais fácil.

Primeiramente, gostaria de agradecer minha família que me acompanha e dá apoio desde o início de tudo e muito antes mesmo de eu pensar estar nesse momento, graças eles foi possível estar onde estou hoje.

Ademais, gostaria de ressaltar a importância dos meus amigos que tive o prazer de encontrar na universidade, Thiago, Nícolas, Luiz, Samuel e André. Entramos na graduação no mesmo período, então vivenciamos cada um dos problemas juntos e mutualmente ajudávamos mutualmente para superá-los.Além disso, vivenciamos momentos de alegria, em cada conquista realizada por um de nós, e foi possível acompanhar nossa evolução como cientista, profissional e indivíduo.

Ressalto, a importância dos meus amigos de Rio Grande/RS que apesar da distância se mostraram presentes em todos os dias da minha graduação, compartilhamos momentos bons e ruins, e graças a eles as dificuldades que tive ou que terei futuramente foram e serão menos complexas. Agradeço a Luana, Lívia, Thiago Lehn, Eliézer, João, Victor e Matheus.

Destaco também a importância do meu amor e companheira Francine por ser minha base nos momentos de insegurança e fazer com que eu sempre acreditasse na minha capacidade.

Finalmente, gostaria de agradecer aqueles que me auxiliaram diretamente no desenvolvimento do meu TCC, sendo estes, meus colegas de trabalho do LabSEC que contribuíram com problemas técnicos existentes, minha amiga Joana que me auxiliou com suas belíssimas sugestões artísticas para meu protótipo e com toda sua experiência para desenvolvimento front-end, a minha amiga e "teacher" Luísa que me ajudou em questões em relação à elaboração desta monografia, e principalmente gostaria de agradecer ao meu orientador e gerente de projeto Lucas, que sempre se mostrou disponível e compreensível com as dificuldades que eu apresentava, além de mostrar o caminho mais efetivo e eficiente para construção deste trabalho.

E um agradecimento ao destino, que impossibilitou que eu trocasse de curso por achar que não teria sido capaz de me formar em computação.

Whereas most technologies tend to automate workers on the periphery doing menial tasks, blockchains automate away the center. Instead of putting the taxi driver out of a job, blockchain puts Uber out of a job and lets the taxi drivers work with the customer directly. (BUTERIN, 2020)

RESUMO

O surgimento da blockchain revolucionou e permitiu o conceito de descentralização quando falamos de criptomoedas. Contudo, com o passar dos anos vislumbrou-se a sua capacidade de aplicação em diversas áreas, principalmente com o surgimento de plataformas como a Ethereum e o Hyperledger Fabric, que permitiram a aplicação de contratos inteligentes para a realização de qualquer tipo de transação entre duas ou mais partes, sem a necessidade de uma entidade central responsável. Sendo assim, o cenário acadêmico onde se demanda confiabilidade, autenticidade e auditoria demonstra-se ideal para utilização desta tecnologia, visto que através dela é possível diminuir burocracias, falhas manuais e fraudes. Nesse contexto, este trabalho visa propor uma reestruturação do modelo desenvolvido pelo Projeto Jornada do Estudante que permite a realização dos processos de auditoria e controle de históricos, currículos e diplomas de maneira descentralizada entre as instituições do Brasil, onde busca-se propor uma nova entidade, o estudante. Sendo assim, o foco desta pesquisa é possibilitar por um aplicativo móvel que os estudantes do ensino superior possam fazer parte do sistema proposto e mantido pelo Projeto Jornada do Estudante, sem que eles precisem ter um conhecimento prévio sobre a tecnologia blockchain. A grande motivação para este desenvolvimento se deu ao fato do estudante ser a principal entidade das universidades, sendo assim, busca-se trazer benefícios de maneira direta a ele, permitindo que o aluno passe a ser uma figura participativa do processo de emissão do seu diploma, ao enviar atividades complementares e/ou estágios. Ademais, outro fator motivador é a possibilidade de trazer ao estudante a capacidade de comprovar de maneira autêntica e confiável a sua participação em uma instituição de ensino superior. Visando um desenvolvimento consistente utilizou-se como base trabalhos relacionados voltados para o desenvolvimento de aplicações móveis que se utilizam de tecnologias blockchain. Junto a isso, utilizou-se o padrão estabelecido pelo governo federal para a interface gráfica. Finalmente, tendo em vista o público alvo da aplicação e os objetivos deste trabalho, realizou-se uma pesquisa utilizando os critérios SUS (System Usability Scale), alcançando um resultado de aceitação em 87% sendo correspondente a um nível excelente de usabilidade, demonstrando que o objetivo principal da proposta de tornar a blockchain acessível para os estudantes foi cumprido.

Palavras-chave: Blockhain. Jornada Estudantil. Hyperledger Fabric. Contratos Inteligentes.

ABSTRACT

The emergence of blockchain revolutionized and allowed the concept of decentralization when talking about cryptocurrencies. However, over the years, its application capacity was glimpsed in more diverse areas, especially with the emergence of platforms like Ethereum and the Hyperledger Fabric, which allowed the application of smart contracts to carry out any transaction kind between two or more parties without the need for a responsible central entity. Thus, the academic scenario, where reliability, authenticity, and auditing are required, is ideal for the use of this technology since through it is possible to reduce bureaucracy, manual flaws, and fraud. In this context, this work aims to propose a restructuring of the model developed by Projeto Jornada do Estudante, which allows the audit and control processes of transcripts, resumes, and diplomas to be carried out in a decentralized manner among the institutions in Brazil, where it seeks to propose a new entity to the student. Thus, the focus of this survey is to make it possible, through a mobile application, for higher education students to be part of the system proposed and maintained by the Projeto Jornada do Estudante without needing previous knowledge about blockchain technology. The great motivation for this development was, since the student is the leading entity of the universities, therefore, it is sought to bring benefits in a direct way to them, allowing the student to become a participatory figure in the process of issuing their diplomas, by sending complementary activities and or internships. Moreover, another motivating factor is the possibility of bringing the student the ability to prove authentically and reliably his participation in an institution of higher education. Aspiring to one consistent development, it is used as a base related to works related to mobile applications that use blockchain technologies. Additionally, it uses the standard graphic interface established by the federal government. Finally, given the target audience of the application and the objectives of this work, a survey was conducted using the SUS (System Usability Scale) criteria, reaching a result of acceptance in 87%, corresponding to an excellent level of usability and demonstrating that the main objective of the proposal, to make blockchain accessible to students, was achieved.

Keywords: Blockchain. Student Journey. Hyperledger Fabric. Smart Contracts.

SUMÁRIO

LISTA DE FIGURAS

1 INTRODUÇÃO

O termo *blockchain* teve repercussão no mundo em 2008, quando uma entidade anônima autointitulada de *Satoshi Nakamoto* (NAKAMOTO, 2008), apresentou um sistema ponto-a-ponto (cada participante da rede funciona tanto como cliente, quanto como servidor, permitindo assim transmissão de informação sem uma entidade central) de transferências eletrônicas, utilizando como base uma moeda digital (criptomoeda), denominada Bitcoin.

Na rede Bitcoin, a confiança não é atribuída a uma entidade centralizada. Sua segurança é proporcionada por protocolos de consenso entre os participantes e incentivos. Em redes *blockchain*, todas as transações são armazenadas em blocos, estes que estão conectados em uma cadeia criptograficamente ligada. Ou seja, o bloco *n+1*, está diretamente relacionado ao seu bloco antecessor *n*. Portanto, qualquer mudança em um bloco é facilmente detectável. Redes como a Bitcoin, por exemplo, garantem a confiança através de dois principais protocolos: (1) Protocolo de Nakamoto (2) Prova de Trabalho. Esses protocolos garantem que os blocos sejam criados somente através de esforço e sorte.

Os blocos são criados através de um quebra-cabeça criptográfico. Para isso, um participante denominado minerador [1] tenta descobrir o número aleatório (*nonce*), que completa a cadeia de blocos. O *nonce* é um atributo de um bloco na rede Bitcoin, composto por 32 *bits*, onde o valor é definido pelos mineradores para que o *hash* obtido seja menor ou igual à dificuldade atual da rede. Como qualquer miníma mudança nos dados irá tornar a solução totalmente diferente, não existe uma forma de prever a combinação que irá gerar o resultado criptográfico correto. Portanto, muitos valores diferentes de *nonce* precisam ser testados.

Sendo assim, a segurança da rede é mantida através da cooperação entre múltiplos participantes. Esses participantes são incentivados a utilizar seu poder computacional a rede para serem recompensados através de criptomoedas. Segundo o autor (ULRICH, 2017), a rede Bitcoin proporciona estimulo à inovação financeira e menores custo de transações.

O sucesso da rede Bitcoin, despertou interesse de diversos pesquisadores e investidores, um deles foi Vitalik Buterin, fundador da Ethereum. Buterin, apresentou, em 2013, em seu *whitepaper* (BUTERIN et al., 2013), um conceito que revolucionou as transações nas redes *blockchain*. Os *smart contracts*, ou contratos inteligentes, apresentados por Buterin, são códigos capazes de serem executados na rede *blockchain* que conseguem representar transações de diversas complexidades. Isto possibilitou suprir a carência que a rede *blockchain* da Bitcoin possui para execução de transações mais complexas, mantendo as características que despertam o interesse em redes *blockchain*.

O grande sucesso das redes *blockchain* proporcionados pela Bitcoin, trouxe um incentivo para estudo desta tecnologia. A partir disso, novas ideias para sua aplicação passaram a existir, indo além do mercado financeiro. O artigo (CROSBY et al., 2016), mostra a versatilidade da tecnologia *blockchain*, sendo aplicada em diferentes áreas. Entre as áreas de aplicação,

[1] minerador em redes *blockchain* são os participantes responsáveis pela criação de novos blocos

pode-se citar:

- *Internet of Things (IoT)* descentralizado, como por exemplo as propostas feitas pelas empresas NetObjx e Helium;
- Aplicações para controle de direitos musicais, como a desenvolvidas pela Digimarc e Mediachain;
- Armazenamento descentralizado, através dos aplicativos como o BitTorrent e Filecoin;
- Soluções anti-falsificação, como a utilizada pela Luxury brands [2];
- Prova de existência para documentos, como a realizada pela Proof of Existence [3].

Vale ressaltar que, apesar de redes *blockchain* terem pontos positivos que despertam o interesse para sua aplicação nas mais distintas áreas, é necessário analisar a sua necessidade bem como suas vantagens e desvantagens.

Neste contexto, é necessário analisar as características de redes *blockchain* e como elas se adéquam a um determinado sistema. Como redes *blockchain* armazenam registros indestrutíveis de todas as transações, elas tendem a precisar de um espaço de armazenamento maior que outras propostas que usam bancos de dados comuns, por exemplo. Portanto, a escalabilidade de redes *blockchain* é mais complexa. Além disso, redes públicas como a Bitcoin demandam um alto empenho de hardware dos mineradores para criação de novos blocos (DELTEC, 2021). Como somente um dos mineradores é recompensado, o alto consumo de energia que foi necessário para criação do Bloco foi desperdiçado para os demais mineradores. Apesar de estar em alta, *blockchain* ainda é uma tecnologia nova em relação as outras, como, por exemplo banco de dados relacionais. Sendo assim, apresenta problemas para integralizar com sistemas mais antigos. Outro problema é que profissionais capacitados nesta área são mais difíceis de encontrar em relação às tecnologias mais consolidadas no mercado (IREDALE, 2022). Portanto, é necessário um estudo minucioso a respeito da aplicabilidade ou não da tecnologia *blockchain* em um determinado sistema.

No artigo *Blockchain and smart contracts for higher education registry in Brazil*, os autores elucidam motivos que possibilitam a utilização de *blockchain* na governança acadêmica (PALMA et al., 2019). No modelo de emissão de diplomas e crédito acadêmicos, abordado pelos autores, a forma que os certificados de graduação são emitidos torna mais difícil a transparência e possibilita que fraudadores possam explorar inconsistências no processo (PALMA et al., 2019). Isto ocorre pelo fato da emissão de certificados ser baseada em um processo burocrático e armazenado em papel. Como descreve PALMA et al., para que um certificado seja emitido, é necessário que diversas etapas sejam realizadas, onde os responsáveis pelo processo verificam a veracidade de cada informação. Após finalizado os processos de validação, cabe ao

[2] https://earlymetrics.com/luxury-brands-using-blockchain-to-fight-counterfeiting/
[3] www.proofofexistence.com

reitor assinar cada um dos diplomas gerados. Dessa forma, possibilitando que, por exemplo, despretensiosamente aprove-se um diploma inválido. Portanto, se faz necessário buscar uma forma de tonar este processo menos burocrático e mais confiável.

Esse problema se agrava, quando contextualizamos o atual cenário do ensino superior no Brasil. Em uma pesquisa realizada pelo CENSO da educação superior, descobriu-se que em 2018 existiam um total de 2537 Instituição de Ensino Superior (IES) no Brasil, sendo 2238 privadas e 299 públicas. Um crescimento considerável em comparação ao ano de 2015, quando existiam 2364 IES. Esse aumento, se deu principalmente pela forma que o mercado de trabalho tem tomado (R7, 2021). De acordo com uma pesquisa realizada pela Organização para a Cooperação e Desenvolvimento Econômico (OCDE), em 2012, constatou-se que o Brasil é o país que mais apresenta diferença salarial entre trabalhadores, com curso superior em relação aqueles que possuem escolaridade inferior. Colocando isso em dados, em uma pesquisa realizada em 2018, pelo Instituto Brasileiro de Geografia e Estatística (IBGE), mostrou que a média salarial de um trabalhador sem ensino superior era de R$1727 (ABMES, 2018). Em contrapartida, um trabalhador com um diploma de ensino superior apresenta em média um salário de R$5110 (ABMES, 2018). Portanto, motivados por essa diferença salarial e tendo em vista a concorrência e exigência do mercado de trabalho, cada ano os cidadãos brasileiros têm buscado mais um diploma.

Contudo, infelizmente, vive-se uma desigualdade social e a menor parcela da população consegue ter a oportunidade de conseguir sua formação. Apenas 34,3% dos brasileiros possuem um diploma obtido em uma IES (SUPERIOR et al., 2019). As grandes diferenças salariais e dificuldades para conseguir acesso ao ensino de qualidade, estimulou um aumento na procura de diplomas e certificações falsificadas, como diz (PALMA et al., 2019). O esquema de falsificações de diplomas não é algo novo, e já existem investigações para tentar combatê-los há bastante tempo. Diversas operações da polícia buscam solucionar esse problema, pode-se citar, por exemplo a Operação “Diploma do Crime” (BRASILIA, 2022). Nesta Operação, foi descoberto um esquema onde criminosos falsificavam assinaturas e carimbos do reitor da Universidade Federal de Pernambuco. Investigações apontaram que o comércio de diplomas gera em média, de mil a três mil reais para os criminosos por documento falsificado (CEARA, 2021). Ademais, em outras apreensões os valores chegaram até cem mil reais, onde um estudante pagou esse valor pelo diploma falsificado de medicina (G1, 2022). Além disso, encontram-se casos de falsificação de diplomas até mesmo em processos seletivos para candidatura a cargos da polícia (PALMA et al., 2020).

Nesse contexto, não só no Brasil como em outros países, começa-se buscar soluções que utilizam *blockchain* e emissão de certificados digitais para que se possa combater os problemas de fraude. Ademais, soluções digitais de emissão e autenticação de diplomas podem diminuir a *blockchain*, custo e proporcionam automatização aos processos. O portal de notícias sobre *blockchain*, Blocknews (CASTRO, 2021), cita dois exemplos de universidades prestigiadas que utilizam de *blockchain* para controle, segurança e emissão no processo de diplomas. Um dos exemplos é a Universidade de Zurique na Suíça e a Universidade Fernando Pessoa em

Portugal. Estas, utilizam um modelo de *blockchain open-source*[4] desenvolvido pelo Massassuchets Institute of Technology (MIT), o Blockcerts [5], que consiste de um padrão para criar, emitir e verificar certificados baseados em *blockchain*. Outro exemplo, que vale a pena ser mencionado, é a plataforma EduCTX [6] utilizada na Universidade de Maribor, na Eslovênia. O EduCTX é uma plataforma descentralizada e distribuída de *blockchain*, que conecta instituições e seus membros, facilitando a gestão de certificados para todos.

Contudo, os projetos de exemplo não abrangem a jornada acadêmica do estudante em toda sua completude. Esses projetos, têm como principal objetivo criar uma forma de validação, automatização e autenticação. Dessa forma, utiliza-se como base para este trabalho o modelo de *blockchain* apresentado na dissertação de mestrado *Bockchain-Based Academic Record System* (PALMA et al., 2019). Em seu trabalho, os autores propõem um modelo distribuído de gerenciamento e emissão de certificados, e controle curricular. Neste modelo, a emissão de certificados é automatizada, onde um estudante ao terminar os créditos e requisitos necessários para conclusão do seu curso, terá um registro desta conquista armazenado na *blockchain*. Como este registro está armazenado na *blockchain*, ele passa a ser:

- *imutável*: Após ser inserido, nenhuma informação a respeito de um certificado poderá ser adulterada;
- *auditável*: Os registros e certificados emitidos por uma IES, podem ser verificados e analisados por todas as demais participantes da rede.

O trabalho proposto pelos autores, foi pioneiro no que diz respeito a utilização de *blockchain* para representação da jornada completa do estudante de graduação. Sendo assim, desde o processo de inscrição, até a emissão do seu diploma de conclusão do curso, utilizando como base os contratos inteligentes para realizarem as operações. No modelo, funcionários de uma IES invocam transações pertencentes a um contrato inteligente para registrar que um aluno concluiu um curso. Então, os contratos inteligentes verificam se todos os requisitos foram atendidos para a conclusão do curso. Caso todos os créditos necessários forem atendidos, o contrato inteligente emite um certificado. Após emitido, o contrato é analisado por todas IES participantes, onde essas podem verificar a existência de transações fraudulentas.

As características e aplicações do trabalho proposto pelos autores, e os problemas presentes no atual modelo, despertaram o interesse do Ministério da Edução (MEC), possibilitando que ele se torne um projeto com financiamento governamental. O financiamento é realizado por um Termo de Execução Descentralizada (TED) entre o MEC e a Universidade Federal de Santa Catarina (UFSC). A partir disso, passou a ser desenvolvida a rede AcadBlock, que consiste em uma *blockchain* privada voltada para IES e que busca unificar, desburocratizar e automatizar os processos acadêmicos no ensino superior. Contudo, apesar da completude do modelo apresentado pelos autores, no que diz respeito a jornada acadêmica de um estudante em graduação,

[4] Open Source é um projeto código livre onde diversos desenvolvedores podem contribuir no desenvolvimento
[5] https://www.blockcerts.org/
[6] https://eductx.org/

seu foco está em ser um facilitador, desburocratizador e um combatente das fraudes voltado para a IES. Assim, o modelo não propõe formas de trazer melhorias e utilizações diretas para o estudante. Neste cenário, o impacto final para o estudante só seria causado por uma melhoria no tempo de emissão de certificados. Portanto, neste trabalho, busca-se trazer benefícios e impactos mais diretos para o estudante de graduação, que agora será um participante direto da rede blockchain.

Neste contexto, é proposto um modelo onde um estudante fará parte da *blockchain* preestabelecida sendo desenvolvida pelo Laboratório de Segurança em Computação (LabSEC), onde ele poderá acessar suas informações acadêmicas como: (1) créditos realizados (2) atividades complementares realizadas; (3) certificados emitidos para ele. Ademais, o estudante poderá enviar informações para a blockchain, como por exemplo uma nova atividade concluída. Isto possibilitará, que o estudante consiga ter: (1) praticidade; (2) agilidade; (3) segurança; (4) uma forma autêntica e incontestável de provar e acompanhar a conclusão de suas atividades e seus certificados.

Para ser possível o ingresso do estudante na rede *blockchain* será necessária uma reformulação e um estudo a respeito de como a arquitetura da rede precisa se adaptar para possibilitar, atribuição de características específicas para o novo membro. Além disso, são necessários estudos a respeito de como este usuário irá interagir com a rede. Ademais, é necessário responder perguntas, como: que informações um estudante pode acessar; que informações um estudante pode adicionar na rede; de que maneira um estudante interage com a rede; quem irá permitir o seu ingresso; e como ele pode ser identificado.

Portanto, para este trabalho, será necessário todo um estudo de como possibilitar a integralização um sistema de contratos inteligentes de uma *blockchain* permissionada a um aplicativo. A partir disso, será possível trazer um impacto direto dos estudos realizados anteriormente. Dessa forma, as burocracias e trâmites que envolvem e atrasam os processos de certificação dos estudantes irão ser facilitadas.

1.1 OBJETIVOS

Esta seção pretende dar o entendimento do que será feito neste trabalho e quais resultados acredita-se que ele irá entregar. Para uma melhor organização esta seção foi subdividida em seções menores onde: 1.1.1 Objetivos gerais, indica de forma genérica o que o trabalho proposto alcança os objetivos específicos e 1.1.2 Objetivos específicos, indica os resultados que pretende-se alcançar através da pesquisa.

1.1.1 Objetivo Geral

Este trabalho desenvolve uma aplicação para dispositivos móveis, que por uma rede *blockchain* possibilitará que os estudantes de ensino superior acessem suas informações aca-

dêmicas, como diplomas, atividades concluídas e estágios realizados de uma maneira segura e prática.

1.1.2 Objetivos Específicos

- Desenvolvimento de contratos inteligentes, que irão integrar aqueles que serão desenvolvidos no LabSEC. Busca-se compreender e determinar a maneira que os estudantes irão participar da rede.
- Analisar o desempenho da utilização de redes *blockchain* na comunicação com dispositivos móveis.
- Definir as políticas e estruturas da rede que mais se adéquam ao cenário de múltiplos estudantes de diferentes IES.
- Contribuir com a literatura atual no que se diz respeito ao estudo da tecnologia blockchain, mais especificamente *blockchain* privadas.

1.2 METODOLOGIA

Na primeira etapa do trabalho, foi utilizado como base uma metodologia exploratória. Onde o foco principal será o estudo das literaturas a respeito dos conteúdos necessários para dar a fundamentação teórica, ter um entendimento claro a respeito do trabalho. A etapa exploratória foi dividida em duas áreas de estudo principais: estudo de *blockchain*, de maneira geral, e os conceitos mais importantes para seu funcionamento, tendo como base redes já conhecidas como a Bitcoin e a Ethereum; para a segunda área, o foco é dar uma fundamentação teórica mais específica para a plataforma Hyperledger Fabric [7], que possui suas particularidades, como maneiras específicas de realizar transações em uma rede *blockchain* e uma estrutura distinta em diversos aspectos a uma rede como o Bitcoin por exemplo. Esta etapa tem como principal objetivo preparar tecnicamente para a etapa de desenvolvimento da *blockchain* e contratos inteligentes.

Na segunda etapa, antes de ser iniciado o desenvolvimento da aplicação será buscado projetos de desenvolvimento similares ao que é buscado neste trabalho. Aplicações do nicho acadêmico serão utilizadas como base para dar uma estrutura organizada à aplicação. Esta etapa, busca organizar previamente os caminhos que serão tomados e a forma que a aplicação deve ter, visando ser o mais satisfatória possível para o usuário final.

Em uma terceira etapa, o principal objetivo é o estudo das linguagens e tecnologias que serão utilizadas para o desenvolvimento dos contratos inteligentes, da aplicação e da rede blockchain. A organização do estudo, se dará principalmente por documentações das linguagens e materiais disponíveis na Internet. Neste processo, o principal objetivo é entender de que

[7] https://www.hyperledger.org/use/fabric

maneira será possível, também, fazer a comunicação entre as diferentes tecnologias e de que forma elas irão funcionar em conjunto.

Além disso, como estamos lidando com um trabalho voltado para o usuário comum, é necessário buscar ao máximo atender às suas expectativas. Dessa forma, a aplicação deve ser capaz de fornecer tudo o que propõe, sem que usuário precise de conhecimentos técnicos. Para isto, ao término do desenvolvimento, será realizada uma pesquisa para definir a avaliação de usabilidade da aplicação. Contudo, levando em consideração que a usabilidade não é um cálculo exato que pode-se definir um valor, utilizaremos como base algumas métricas existentes na literatura. Para a construção do indicativos neste trabalho, será utilizado como base o SUS (*System Usability Scale*), metodologia esta que foi criada por John Brooke em 1986. O SUS, utiliza como base os seguintes critérios de avaliação: efetividade, eficiência e satisfação. Dessa forma, através de um pequeno questionário de dez perguntas, será estipulada uma média, tendo como base as respostas dadas pelo usuário. Esta média calculada, serve como parâmetro para saber se o o aplicativo está cumprindo aquilo que era esperado pelo usuário e se satisfaz o que foi proposto.

2 A REDE BLOCKCHAIN

Neste capítulo são abordados as principais bases teóricas necessárias para entender *blockchain* e seu funcionamento. Para que, dessa forma, seja possível compreender corretamente o que é proposto neste trabalho.

2.1 FUNÇÕES CRIPTOGRÁFICAS HASH

Uma definição comumente utilizada na área da computação é de que: *hash* é uma função matemática, da área da criptografia, onde dada qualquer entrada de tamanho variado, uma saída de tamanho fixo (determinado pelo algoritmo escolhido) será obtida. Diversos algoritmos criptográficos podem ser utilizados para objetivos específicos, dentre eles, pode-se citar: MD5, SHA-256, RipeMD-256, SHA1. Devido ao fato do SHA-256 ser um dos algoritmos de criptografia *hash* mais popular em processo de autenticação e criptografia (NABLE, 2021), e ser base para a rede Bitcoin utilizaremos ele como referência para explicação desta seção.

SHA-256 consiste em uma função *hash* onde dada uma entrada de qualquer comprimento, e independentemente dos dados que estão presentes, o resultado sempre terá um comprimento fixo de 256 *bits*. Famosas aplicações utilizam-se desta função em sua estrutura, um exemplo é a rede Bitcoin, que se utiliza de um esquema de *double SHA-256* (DEV, 2014), onde se aplica o algoritmo criptográficos duas vezes para que se tenha uma camada extra de segurança.

(SOBTI; GEETHA, 2012) define as características de funções hash como sendo:

- *Cálculos rápidos e fáceis*: Independente da entrada que seja passado para uma função hash, o resultado criptográfico será calculado de maneira rápida e sem que seja requirido uma grande capacidade computacional.

- *Determinística*: Não importa a quantidade de vezes que uma mesma entrada for aplicada ao algoritmo, o resultado obtido sempre será o mesmo.

- *Resistência a Pre-Imagem*: Dada uma função *Hash* $H(X) = h$, é inviável determinar o valor X que resultou na função. *"Brute force attacks work on all hash functions independent of their structure and any other working details."* (SOBTI; GEETHA, 2012), essa citação tem como base um dos fatores principais das funções *hash*, o determinismo. Ou seja, temos a garantia que uma entrada $H(x) \rightarrow y$ sempre será igual, não importa quantas vezes executa-se a função. Portanto, por força bruta [1] poderia ocasionar na combinação correta. Contudo, estes testes por força bruta levariam tanto tempo para chegar ao valor da resposta que não trazem impacto para sua segurança. Um entendimento melhor sobre este assunto, será explicado de maneira mais detalhada na Subseção 2.7.1

[1] Na área da criptografia, *Brute Force* ou força bruta, é um ataque exaustivo onde todas as possibilidades são testadas, até a correta seja encontrada.

- *Efeito Avalanche*: Qualquer mínima mudança na entrada da função, irá resultar em um *hash* de resultado totalmente diferente.
- *Resistência a colisões*: Tendo duas entradas diferentes X e Y, é inviável encontrar *hashes* iguais, onde $H(X) \rightarrow h1$ e $H(Y) \rightarrow h2$ resultem no mesmo valor, ou seja $h1 \neq h2$.

2.2 *O QUE É BLOCKCHAIN?*

Este seção foi escrita analogamente ao conceito apresentado por Michael Crosby em seu artigo: *BlockChain Technology: Beyond Bitcoin* (CROSBY et al., 2016).

Uma *blockchain* pode ser vista, fundamentalmente, como um banco de dados distribuído para registro de transações. Estas transações ocorrem entre os participantes de uma rede armazenadas em um livro razão distribuído.

Livro razão, em sua terminologia, consiste em um registro de transações com finalidade de coleta dos dados de maneira cronológica. Nas redes blockchain, ele funciona da mesma maneira já conhecida, sua funcionalidade é armazenar os conjuntos de transações e estados atuais. Dessa forma, é possível obter os registros gerais, como por exemplo: número de ativos disponíveis em uma conta; operações realizadas em um dia especifico e diversas outros.

Antes de entender como as redes *blockchain* funcionam e como armazenam as transações de maneira distribuída, é necessário entender a sua estrutura básica: o bloco.

Na rede blockchain, os blocos são armazenados em uma cadeia, onde cada bloco está criptograficamente ligado ao seu antecessor. Na literatura a respeito de blockchain, os autores definem o nome dos tipos de blocos de maneiras distintas. Portanto, neste trabalho foi escolhido o padrão apresentado por Vitalik Buterin em seu artigo: *Ethereum Whitepaper* (BUTERIN, 2020). Em sua pesquisa, Buterin determina três tipos de blocos: gênesis, validado e inválido. Bloco gênesis representa o primeiro bloco em uma cadeia na blockchain, somente este bloco não está criptograficamente encadeado a um bloco anterior. Blocos válidos são blocos que já tiveram todas as suas transações validadas pelos Mineradores (o termo "mineradores"será explicado detalhadamente na Seção 2.6). Dessa forma, estão aptos para ingressarem na rede. Blocos

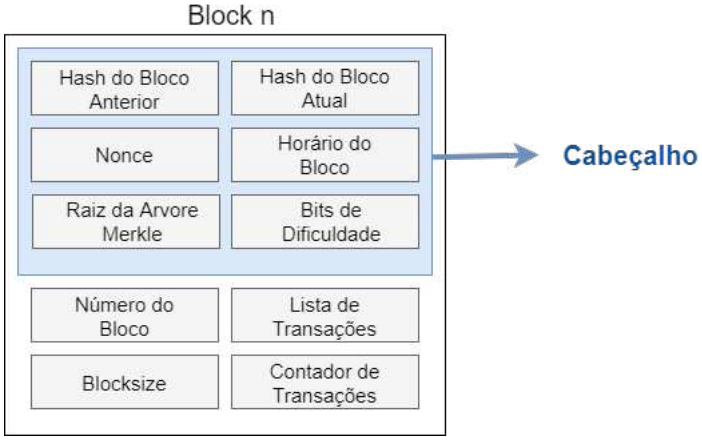


Figura 1 – Exemplo de Estrutura de um bloco

não válidos, são aqueles que ainda não tiveram suas transações verificadas, ou foi encontrada alguma irregularidade nelas.

O fato de um bloco ter uma referência a um bloco anterior é o que cria a ideia de cadeia de blocos. Ou seja, qualquer miníma mudança em um dos blocos pertencente a estrutura, irá impactar diretamente toda a cadeia.

Ademais, blocos podem apresentar estruturas diferentes segundo os objetivos da rede. Sendo assim, para esta seção, utiliza-se o modelo descrito no artigo (ALI et al., 2018), esta estrutura descreve um bloco na rede *blockchain* mais famosa: a Bitcoin.

Os seguintes atributos estão presentes em um bloco na rede Bitcoin: **Cabeçalho**, ***Blocksize***, **Número de bloco**, **Transações** e **Contador de Transações**. Na Figura 1, é possível observar a forma que a estrutura de um bloco se organiza.

- *Blocksize*: É a especificação de um tamanho máximo que irá delimitar a quantidade de transações que poderão ser armazenadas em um bloco;
- *Número de bloco* : É um número responsável por definir a identificação do bloco na cadeia da blockchain;
- *Transações*: Uma lista com todas as transações armazenadas neste bloco;
- *Cabeçalho*: Armazena as principais configurações do bloco, e o hash do bloco antecessor.

O cabeçalho de um bloco se sub-organiza em uma estrutura, onde estão presente os seguintes atributos: **Hash do bloco antecessor**, **Hash da raiz Merkle**, **Horário do bloco**, **Nonce** e **Bits de Dificuldade**.

- *Hash do bloco antecessor*: Contém o *Hash* do cabeçalho do bloco anterior, este atributo cria a relação de encadeamento e dependência que se encontra na *blockchain*.
- *Contador de Transações*: Indica o total de transações que estão sendo armazenada neste bloco.

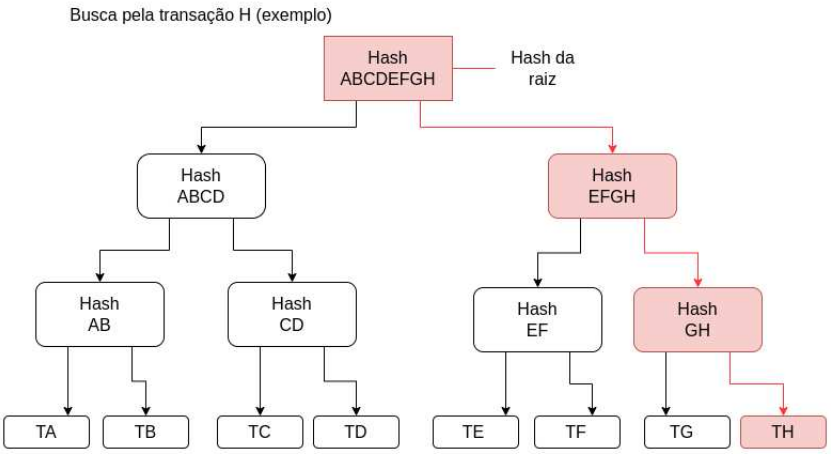


Figura 2 – Buscando uma transação H em uma árvore Merkle.

- *Hash da raiz Merkle*: É o hash que indica a raiz de todos os hashs de transações presentes no bloco. Árvores de Merkle, são estruturas de dados binárias que têm como objetivo, armazenar informações criptografadas de maneira resumida para que seja possível obter uma informação com menos custo e necessidade computacional. O importante é entender, que, graças a ela, é possível fazer buscas na rede sem que seja necessário baixar toda a blockchain. Segundo o site Bit2Me [2] (BIT2ME, 2020), são necessários mais de 300 Gigabytes (GB) para armazenamento da rede Bitcoin. Dessa forma, usuários que desejam só visualizar se uma transação existe, não teriam motivos para arcar com todo esse custo. Além disso, dispositivos com menos capacidade tecnológica, como celulares e tablets, podem verificar as transações realizadas de maneira mais rápida e econômica.

- *Horário do bloco*: Os blocos em uma cadeia são organizados em uma linha cronológica de *Timestamp*. Portanto, quando um novo bloco é adicionado à cadeia deve ser verificado se seu *Timestamp* é maior que o do seu antecessor.

- *Nonce*: *Nonce*, significa número usado apenas uma vez. Este numero aleatório é adicionado ao cabeçalho de um bloco. Na Bitcoin este é o número que um minerador precisa descobrir antes de solucionar um bloco na rede. Este mecanismo pretende proteger as redes do que conhecemos como ataque de repetição [3], dando uma garantia que somente *hashes* diferentes serão gerados.

- *Bits de Dificuldade* : Determina a quantidade de zeros que precisarão aparecer no início do hash gerado pelos mineradores. Este atributo possui a função de manter a rede Bitcoin com a dificuldade ajustada, para serem gerados em média um bloco a cada 10 minutos (SERGEENKOV, 2022).

Na *blockchain* são seguidos dois princípios básicos: imutabilidade e irreversibilidade, isto é: (1) nenhuma transação realizada pode ser modificada; (2) nenhuma transação realizada na rede pode ser apagada.

As redes *blockchain* podem ser desenvolvidas de diferentes maneiras, conforme os objetivos buscados com sua utilização. Na literatura sobre o assunto, alguns autores determinam nomes e tipos diferentes de rede, conforme as suas características. Neste trabalho será utilizado com base os padrões definidos pelos autores: Xiwei Xu, Ingo Weber e Mark Staples no livro *Architecture for blockchain Applications* (XU; WEBER; STAPLES, 2019).

Diferentes sistemas precisam de características apresentadas por diferentes tipos de rede, por exemplo: (1) para uma rede de criptomoedas globais como o Bitcoin necessita-se de uma rede pública, permitindo, assim, uma maior transparência e descentralização (2) um sistema de gerenciamento de alocação de recursos, em uma empresa precisa de uma rede privada, considerando que esses dados não devem ser acessáveis a partes externas a companhia.

[2] https://bit2me.com/pt/
[3] https://academy.bit2me.com/pt/que-es-un-ataque-replay/

Assim, para uma melhor compreensão, deve-se entender as diferenças entre as três principais arquiteturas: Pública, Privada e Consórcio.

2.2.1 Blockchain Pública

Uma rede pública precisa ser descentralizada fundamentalmente, onde qualquer usuário é livre para ingressar, sem que seja necessário a permissão de ninguém. Uma vez que o usuário participa da rede, ele deve conseguir participar de todas as atividades existentes, como: (1) realização de transações; (2) validação de transações; (3) verificação de todas as transações realizadas. Essa política de livre ingresso, é o que torna a rede descentralizada, sendo assim, mais confiável, transparente e autogovernada.

Como este modelo não possui uma entidade central controlando, não é necessário atrelar a confiança a alguém, toda responsabilidade é divida entre os participantes da rede. Sendo assim, redes públicas sempre estarão relacionadas a algum tipo de incentivo, para os usuários continuarem com um bom comportamento, como criptomoedas, por exemplo.

2.2.2 Blockchain Privada

Buscando uma maneira de manter um controle organizado e mais rastreabilidade de transações, empresas se motivaram a utilização da *blockchain*. Entretanto, a necessidade de manter suas transações privadas e gerenciar permissões para cada usuário impossibilitou o uso de redes públicas, sendo necessário, assim, a criação de uma nova estrutura de *blockchain*. Partindo disso, a rede privada traz as características essenciais de uma *blockchfain* pública como comunicação p2p, armazenamento de transações em bloco, segurança com base em criptografia *hash* e auditoria confiável, adicionando uma nova camada de configuração que permite mais privacidade.

Para redes que sigam este modelo, é necessária uma empresa/organização como entidade central, ou seja, um intermediário confiável. É responsabilidade desta autoridade, definir os direitos de cada nó participante da rede (diferentes nós podem ter diferentes permissões).

Em *blockchains* privada, uma entidade conhece o participante que está adicionando a rede. Dessa forma, é fácil identificar transações inválidas, e as relacionar ao nó responsável. Portanto os usuários já possuem um incentivo da própria rede para manterem um comportamento correto, facilitando assim a coordenação mutua. Outro fator positivo em redes privadas, é que devido ao seu tamanho ser bastante reduzido em relação a *blockchains* públicas, os protocolos de consenso (explicado mais detalhadamente na seção 2.7) causam bem menos desperdício.

2.2.3 Blockchain de Consórcio

As *blockchains* de consórcio são classificadas como um meio-termo entre as públicas e as privadas. Sua aplicação é tipicamente realizada entre múltiplas entidades. Nesse tipo de rede, as entidades intermediárias são responsáveis por determinar nós que serão responsáveis pelo consenso na rede, ou seja, irão verificar e determinar as transações que serão válidas para a rede.

2.3 PEER-TO-PEER

Peer-to-peer, é a arquitetura base para desenvolvimento de uma blockchain, onde sua estrutura consiste em nós que podem agir tanto como cliente, quanto como servidor. Isto é o que proporciona que nós compartilhem transações de maneira descentralizada. Na Figura 3, é possível observar uma demonstração de uma rede simples em uma *blockchain* pública como a do Bitcoin, onde nós de usuários estão conectados e todos eles compartilham o mesmo livro razão.

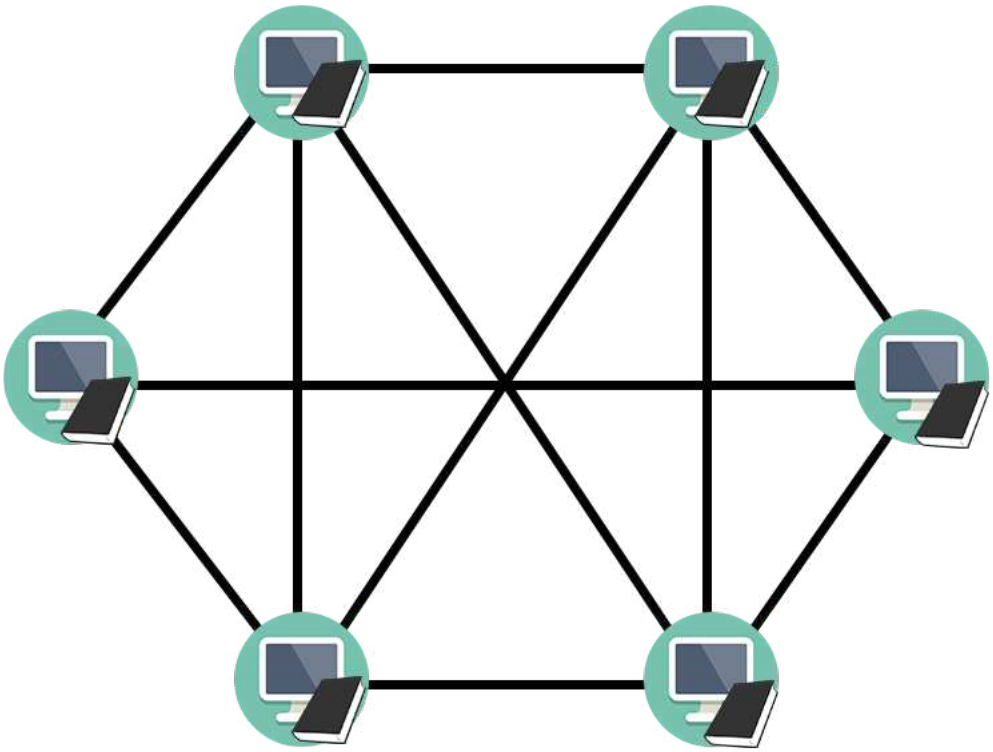
Para o presente trabalho, não é necessário se aprofundar nos detalhes de redes p2p, busca-se dar uma base para entender a forma que os nós se comunicam em uma *blockchain*. Para se aprofundar mais sobre os detalhes a respeito de redes p2p, recomenda-se a leitura do capítulo 2 do livro *Peer-to-peer systems and applications* (STEINMETZ; WEHRLE, 2005).

A não existência de um servidor central é responsável por tornar a rede: tolerante a falhas e descentralizada. Como todos os nós nessa estrutura recebem a mesma responsabilidade e possuem acesso às informações, mesmo que alguns deles possam ficar inativos, a rede irá se manter em pleno funcionamento. Ao efetuar atualizações no livro razão ou validar um bloco, cada nó irá comunicar ao seu nó vizinho para que a informação seja espalhada para todos os participantes da rede.

Em redes p2p, como não existe um emissor central responsável pelo controle, um possível problema seria se usuários mal-intencionados tivessem controle da maior parte (cinquenta e um por cento) dos nós. Dessa forma, eles seriam maioria na decisão e poderiam selecionar transações inválidas ou que os beneficiam de alguma maneira injusta. Em redes blockchain, isso é conhecido como *Sybil attack* ou ataque de maioria (AGGARWAL; KUMAR, 2021). Em redes *blockchain* públicas, esse tipo de ataque é evitado por algoritmos de consenso, uma visão completa dos principais algorítimos será apresentado na Seção 2.7.

2.4 CONTRATOS INTELIGENTES

Apesar de toda a revolução tecnológica proporcionada pelo surgimento do Bitcoin, ele apresenta algumas limitações, principalmente no que diz respeito a transações mais complexas, que vão além de uma “simples” troca de ativos. No artigo introdutório *Ethereum white*



]

Figura 3 – Exemplo Rede p2p aplicada a blockchain pública.

paper, BUTERIN et al., fundador da Ethereum, popularizou um conceito que revolucionaria as negociações na *blockchain*: *Smart Contracts* ou, contratos inteligentes, em português.

Os contratos inteligentes são códigos de programação, responsáveis por definir regras, objetivos e consequências em uma relação entre duas ou mais partes. Basicamente, ele funciona como os contratos que já conhecemos na realidade, onde são estabelecidas metas que devem ser cumpridas e penalidades que serão impostas caso uma das partes não cumpra com o acordo.

Contratos inteligentes possibilitaram expandir os horizontes de possibilidades para a utilização de *blockchain*. No artigo *Security, Performance, and Applications of Smart Contracts: A Systematic Survey* (ROUHANI; DETERS, 2019), são abordados algumas das áreas de aplicação para contratos inteligentes, como: (1) IoT (2) Assistência Médica (3) Cadeia de Recursos (4) Gestão de Processos de Negócio (5) Manutenção de Registros (6) Votação (7) Identidade Digital.

2.5 CHAVES PÚBLICAS E PRIVADAS EM TRANSAÇÕES NA BITCOIN

Em redes *blockchains* como a do Bitcoin, a transparência, segurança e descentralização são os fatores cruciais. Para ser possível realizar transações entre usuários, sem que haja interferência de terceiros maliciosos, são necessários mecanismos de segurança. Dentre os mecanismos utilizados por redes como a Bitcoin, pode-se citar a criptografia de chaves como um dos essenciais. A partir de PKC (public key cryptography), é possível verificar se uma transação realizada foi realmente efetuada pelo dono dos ativos.

Através da sua chave privada, você poderá criar diversas chaves públicas, que serão utilizadas pelo outros participantes da rede para que possam enviar criptomoedas e verificar a autenticidade das suas transações. A partir da chave privada é possível recuperar todas as suas chaves públicas, contudo, a operação inversa não é possível (SECTIGO, 2020). Portanto, caso você perca sua chave privada, seria impossível revogar qualquer direito de posse aos ativos, mesmo que eles sejam seus.

Assim como na criptografia em *hash*, o grande fator que torna as chaves criptográficas tão importantes é a capacidade de ser extremamente rápido e fácil de criar e inviável computacionalmente de se reverter. As chaves funcionam com a ideia de permitir a passagem de informações por um canal público, sem que os dados fiquem expostos. No processo de transações em redes *blockchain*, como o Bitcoin, por exemplo, as chaves contribuem da seguinte maneira:

Iremos definir para o nosso exemplo dois usuários participantes, Leonardo (portador do ativo) e Lucas (destinatário da transação). Na Figura 4 é possível ter uma ideia resumida de como ocorre o processo de transferência de posse um ativo (criptomoeda).

1. Leonardo, deseja enviar a quantidade de 10 Criptomoeda da Rede Bitcoin (BTC) para Lucas, para isso, ele realiza uma transação assinada com sua chave privada, indicando assim, a autenticidade da transação;

2. Para identificar e restringir quem será o beneficiário desta transação, Leonardo insere a chave pública de Lucas na transação;

3. Somente Lucas, o portador da respectiva chave privada, capaz de descriptografar a transação que utilizou sua chave pública como assinatura, irá ter acesso aos BTC.

4. Como prova de autenticidade da transação, Lucas e todos os outros usuários da rede irão utilizar a chave pública de Leonardo, para verificar que esta operação realmente foi realizada por ele (Leonardo).

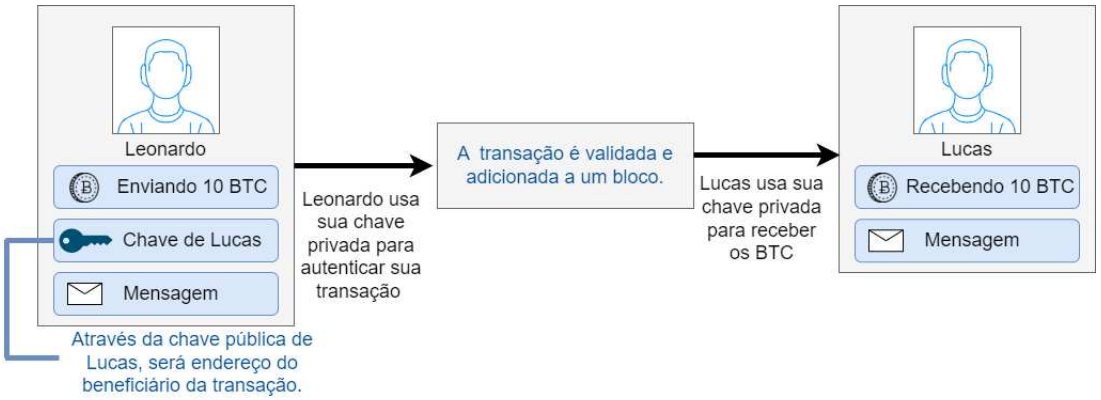


Figura 4 – Transação utilizando PKC.

2.6 MINERADOR

Na rede *blockchain*, mineradores estão sempre à espera de novas transações e blocos, para poderem ser recompensados pelo seu serviço. Na rede Bitcoin, quando uma transação encontra um nó minerador, ela é verificada. Sendo assim, mineradores são responsáveis por confirmar novas transações e criações de blocos, portanto, cruciais para a manutenção do estado da rede e atualização do livro razão de uma *blockchain*.

A mineração na rede Bitcoin é análoga a como conhecemos o processo de garimpo de ouro, por exemplo, onde garimpeiros buscam pelas pedras preciosas através de seu esforço próprio e sorte. A marcante frase de Antoine-Laurent de Lavoisier "Na Natureza, nada se cria, nada se perde, tudo se transforma", se aplica diretamente em criptomoedas. As criptomoedas surgem do trabalho de *hardware* e gasto de energia dos mineradores (computadores), buscando solucionar quebra-cabeças criptográficos.

Sempre que um novo bloco é adicionado a rede, significa para os mineradores que uma rodada foi finalizada e seu vencedor foi anunciado. O vencedor de uma rodada no Bitcoin, ou seja, aquele que foi capaz de resolver o quebra-cabeça criptográfico, é recompensado através de ativos BTC na rede. Além disso, mineradores podem ser recompensados com taxas recebidas na efetivação de transações. Na rede Bitcoin, essa recompensa por validar transação é denominada *coinbase*. Após resolver o quebra-cabeça criptográfico, o minerador constrói o novo bloco adicionando uma referência para o bloco anterior e um resumo das transações presentes na Merkle Tree (XU; WEBER; STAPLES, 2019).

2.7 PROCOLOS DE CONSENSO

Antes do surgimento do Bitcoin, outras moedas digitais descentralizadas não conseguiram ter um sucesso ou se manterem em funcionamento devido alguns problemas estruturais. Isso se deu pela dificuldade de se manter um sistema Bizantino Tolerante a Falhas (BFT) 2.7.2 que acompanhasse o tamanho de crescimento que as redes requisitavam.

Falhas Bizantinas são aquelas que ocorrem em sistemas distribuídos, devido a necessidade de nós estarem em um consenso, contudo, alguns deles podem agir de maneira incorreta e/ou maliciosa. Em um sistema de livro razão totalmente compartilhado, os blocos e transações precisam estar sincronizados em todos os usuários participantes da rede. O Bitcoin conseguiu que a rede crescesse de acordo com suas requisições, eliminando as possíveis falhas bizantinas através do seus protocolos de consenso.

Para que fosse possível expandir a rede, sem que houvesse impacto pelas Falhas Bizantinas, a rede Bitcoin utilizou-se de: (1) Protocolo Bizantino de Tolerância a Falhas, explicado na Subseção 2.7.2 (2) o Consenso de Nakamoto, visto na Subseção 2.7.3 (3) Prova de Trabalho (PoW), abordado na Subseção 2.7.1.

2.7.1 Prova de Trabalho

PoW é o primeiro e principal mecanismo de consenso utilizado em redes *blockchain* (ALI et al., 2018). Seu fundamento passa pelo conceito apresentado anteriormente como mineração. Em redes públicas como a Bitcoin ou Ethereum, todos os participantes da rede podem contribuir para validação de transações e criação de novos bloco. Para um usuário participar deste processo, ele precisará empregar seu poder computacional na rede.

O protocolo de PoW, tem como princípios básicos, ser difícil de solucionar e fácil de verificar. No processo da PoW os mineradores precisam descobrir o número aleatório *Nonce*, que aplicado a função *hash* será capaz de gerar o resultado que completa o bloco. Como na rede Bitcoin o *Nonce* representa um número inteiro, seu valor pode variar entre 0 e 4294967296. Cada bloco possui um *hash* respectivo a ele, este hash será usado como base para mineradores que desejam descobrir o próximo bloco que completa a cadeia de blocos. Junto ao *hash* do bloco anterior, será adicionado o bloco atual com as transações validadas pelo minerador. Com ambos os valores juntos, agora o minerador possui um grande texto que será usado como base para solucionar o problema criptográfico. Os cálculos matemáticos realizados tem como objetivo descobrir o número aleatório (*Nonce*), que, concatenado ao *hash* anterior e transações, será aplicado na função SHA-256 e irá completar o desafio. Ademais, o *hash* gerado pelo minerador precisa atender o pré-requisito de dificuldade da rede, este que foi definido pelos *bits* de dificuldade. Para que um *hash* atenda o requisito de *bits* de dificuldade, é necessário que a solução apresente um certo número de zeros como prefixo.

Na Figura 5 demonstra-se um exemplo de tentativa para solucionar o *hash* criptográfico da rede Bitcoin. Após o quebra-cabeça solucionado, cabe aos outros mineradores participantes da rede validarem a solução. Devido ao elevado número de participantes de uma *blockchain* pública como a Bitcoin, vários mineradores poderiam gerar um resultado válido ao mesmo tempo. Portanto, para que fosse possível entrar em um consenso entre os múltiplos mineradores participantes, um novo protocolo foi necessário, conhecido como: consenso de Nakamoto, este que será abordado na Subseção 2.7.3.

2.7.2 Protocolo Bizantino de Tolerância a Falhas

Em redes *blockchain* públicas, lida-se com um sistema descentralizado, portanto, é necessário manter um acordo entre os participantes a respeito de qual informação é correta e deve ser passada para os demais. Para que isto seja possível, é necessário partir do pressuposto que não existe uma fonte correta de dados, portanto, trata-se todo tipo de informação como uma "promessa". Sendo assim, cada nó poderá agir de forma totalmente independente, sem se preocupar com os demais participantes. Após os nós realizarem suas operações, eles deverão entrar em um consenso, onde irão se totalizando, partindo para um estado com um maior número de decisões favoráveis. Portanto, através deste protocolo é possível garantir segura contra nós maliciosos, desde que estes não constituem a maioria na rede.

2.7.3 Consenso de Nakamoto

Este consenso, representa uma das principais inovações da rede Bitcoin, porque permite facilitar o processo de comunicação descentralizada necessária na rede, solucionando problemas de Falha Bizantina. O Consenso de Nakomoto, permite que múltiplos participantes da rede, através das decisões da maioria (STIFTER et al., 2018) entrem em um consenso.

Como *blockchain* trata-se de um sistema distribuído p2p, a comunicação não acontece de maneira imediata, portanto, é necessário que os diferentes usuários da rede entrem em um acordo. Ao solucionar o quebra-cabeça criptográfico, o minerador adiciona o bloco encontrado a uma cadeia de blocos, podendo gerar bifurcações [4] na blockchain. Em uma rede pública blockchain, como a Bitcoin, não existe uma entidade central capaz de definir quem possui a cadeia de blocos correta, e devido a complexidade de estrutura, quanto mais usuários presentes na rede, mais ramos de cadeia de blocos diferentes são gerados. Portanto, para que fosse possível determinar a ramificação válida, o Consenso de Nakamoto utiliza como regra a seleção da cadeia mais longa. A cadeia mais longa representa aquela que normalmente teve maior computação empregada. Cadeias que tiveram mais computação aplicada, foram aquelas que solucionaram mais problemas criptográficos e adicionaram mais transações, portanto, são aquelas que são mais confiáveis e seguras para serem adicionadas a rede. Na literatura, esta cadeia mais longa é denominada *mainchain*, ou cadeia principal, em português.

[4] Estas bifurcações representam múltiplas cadeias de blocos que podem ser geradas, devido ao elevado número de participantes na rede.

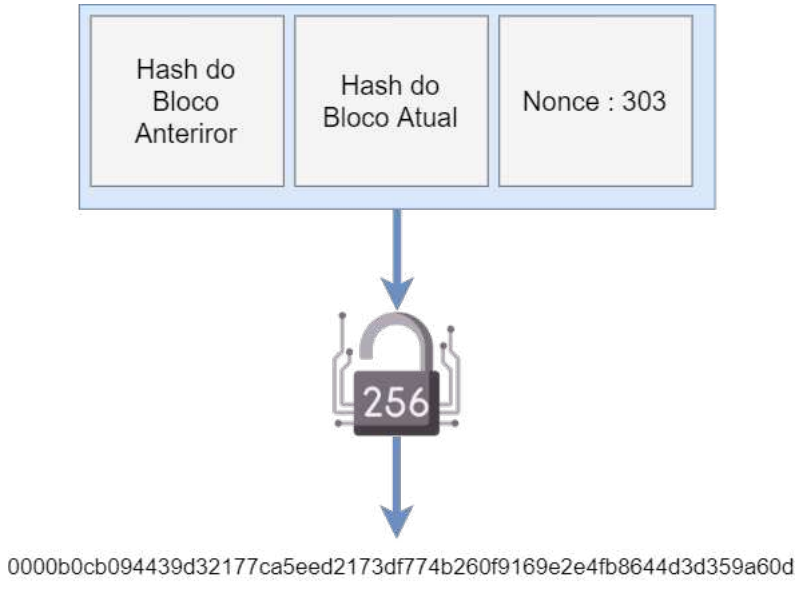


Figura 5 – Processo de Mineração Bitcoin.

2.8 TRANSAÇÕES

Seguindo as fundamentações teóricas dadas nas Seções 2.5 e 2.7, é possível formalizar o conceito de transações na rede blockchain. Transações na rede *blockchain* são irreversíveis, portanto, uma vez validada e confirmada, ela não poderá ser removida dos registros no livro razão. Em transações, dados podem ser adicionados ao seu escopo além do valor de ativos na transferência. Na Ethereum por exemplo, possível registrar transações, variáveis e chamadas de funções (XU; WEBER; STAPLES, 2019).

Logo após uma transação ser assinada por um remetente, ela passa a ser uma proposta para rede, esta é enviada para toda rede. Ao encontrar um nó minerador, ela será validada ou invalidada. Uma vez validada, uma transação é passada para o próximo nó até que seja reconhecida por toda rede. Caso uma transação seja considerada invalida, ela é descartada.

Após uma transação atingir um nó de mineração e ser verificada ela já pode ser adicionada a um bloco. A transação confirmada é adicionada a um bloco criado por um minerador, este bloco é dissipado para toda rede, para que todos os nós completos (que possuem cópias do livro razão), adicionem as novas transações efetuadas. Este processo varia conforme os protocolos de consenso utilizados na rede, no Bitcoin, por exemplo, a transação passaria pelos processos de PoW e Nakamoto. Um bloco que validou a transação pode ser descartado, dado o consenso de Nakamoto, caso isto ocorra, será necessário que a transação seja validada novamente.

3 BLOCKCHAIN NO HYPERLEDGER FABRIC

O grande crescimento de redes *blockchain*, despertou o interesse de empresas na utilização desta tecnologia em suas plataformas. No livro *Blockchain with Hyperledger Fabric:* (GAUR et al., 2020) são destacadas as principais características de redes *blockchain* como: um canal compartilhado para troca de mensagem; proteção contra adulteração; arquitetura escalável; confidencialidade e auditabilidade. As características citadas foram os principais pontos que motivaram companhias a buscarem conhecimento a respeito de redes *blockchain*. Contudo, empresas privadas e organizações buscam um certo grau de confidencialidade e sigilo, que não estão presentes nas redes públicas comuns, já que qualquer participante pode ingressar nelas. Portanto, a identificação dos participantes e atribuição da ideia de "cargos" ou permissões era essencial. Sendo assim, foi necessária uma reformulação das redes *blockchain* não permissionadas, para poderem atender essas necessidades.

3.1 O QUE É O HYPERLEDGER FABRIC ?

O *Hyperledger Fabric* [1] é um projeto de código aberto atualmente mantido pela Linux Foundation. Em sua estrutura, o Hyperledger Fabric, apresenta uma arquitetura extremamente modular e configurável. Esta arquitetura, possibilita adaptar-se as necessidades do usuário do serviço trazendo: inovação; versatilidade; otimização; escabilidade. A IBM, cita como as principais qualidades do Fabric como sendo (IBM, 2020):

- *Permite a utilização de uma rede Autorizada*: somente participantes aprovados podem ingressar na rede blockchain;
- *Transações Confidenciais*: somente participantes com permissão poderão verificar as transações;
- *Arquitetura Conectável e Adaptável*: é possível adaptar a rede *blockchain* de acordo com as necessidades do cliente;
- *Fácil de começar a usar*: utiliza-se de linguagens comuns para desenvolvimentos dos seus contratos, portanto, não é necessário um estudo muito específico dos desenvolvedores.

Fabric é a primeira plataforma com livro razão distribuído com suporte a linguagens de programações comuns, como: Java, Go e Nodejs (Hyperledger Fabric, 2020). Portanto, facilitando o processo de capacitação de funcionários para empresas que desejam utilizar serviços *blockchains*. Outro fator crucial para o sucesso do Hyperledger Fabric é sua característica de possibilitar múltiplos protocolos de consenso e da modularidade na sua arquitetura. Dessa forma, a rede Fabric é capaz de se adaptar as mais específicas necessidades das entidades que

[1] https://www.hyperledger.org/use/fabric

fazem sua utilização. Somado a isto, o Fabric não necessita de uma criptomoeda como incentivo em sua estrutura, pois em sua organização não existe a necessidade de incentivar mineração e/ou a execução de contratos inteligentes.

O Fabric é implementado com uma nova arquitetura para execução e validação de transações denominada *execute-order-validate* (ANDROULAKI et al., 2018), esta que será explicada detalhadamente na seção 3.3.9 .

As transações no Fabric seguem três etapas: (1) *execute*, ou execução em português, é a etapa onde, uma transação é executada e depois é realizado o endosso desta; (2) *order* ou ordem em português, é a etapa onde dado um protocolo de consenso, as transações são ordenadas; (3) *validate*, ou validação em português, é a etapa onde as transações são validadas através das políticas, antes que possam ser registradas no livro razão. No Fabric, políticas de endosso são responsáveis por determinar quais nós, ou quantos deles, precisam dar a garantia que uma transação é valida e, portanto, pode impactar na rede. Dessa forma, as transações na rede Fabric diferentemente de redes *blockchain* públicas comuns, só precisam ser endossada pelo subconjunto predefinido de nós. Esta é uma das características que possibilita a alta escabilidade das redes *blockchain* no Fabric.

3.2 CHAINCODE

No Hyperledger Fabric, *smart contracts* são chamados de *chaincodes*. Em uma rede, diferentes *chaincodes* podem rodar simultaneamente e serem desenvolvidas por qualquer um. Para isso, como medida de segurança, todas as *chaincodes* na rede deverão ser validadas pelas organizações antes de usadas. A utilização de contratos inteligentes é dado em redes *blockchain* como (MAKAROV, 2021): Ethereum [2], Tezos [3], Solana [4]. Em geral, grande parte das redes que utilizam contratos inteligentes necessitam de linguagens de domínio específico, como a Solidty no caso da Ethereum. Esta característica cria uma enorme barreira de entrada e dificulta para desenvolvedores iniciantes no desenvolvimento de contratos inteligentes. Ademais, redes *blockchain* precisam que todas as transações sejam executadas em ordem, portanto, medidas de seguranças complexas são necessárias para que o sistema se mantenha confiável e protegido, dificultando escabilidade e impactando no custo.

Chaincodes na rede Fabric representam o eixo central para execução das aplicações. Elas funcionam de tal maneira que conseguem representar as mais diversas possibilidades, para diferentes tipos transações e objetos. Elas permitem, representação de complexas transações realizadas manualmente, como um programa, tornando assim o processo muito mais eficiente e confiável.

Assim como na rede *Ethereum*, no Fabric foi necessária uma reformulação especial para ser a utilização de contratos inteligentes. Diferentemente da Ethereum, que usa uma má-

[2] https://ethereum.org/pt-br/
[3] https://tezos.com/
[4] https://solana.com/

quina virtual Ethereum Virtual Machine (EVM), no Fabric os contratos inteligentes e toda lógica de execução da rede *blockchain* acontece via contêineres do Docker [5].

Basicamente, contratos inteligentes conseguem realizar as operações primárias de: *put, get and delete*. Ademais, *chaincodes* têm acesso ao registro imutável de transações realizados na rede. As operações básicas realizadas pelos contratos inteligentes são responsáveis por:

- *get*: é a operação de obter o estado atual de um objeto na rede, este valor é obtido por uma chave identificadora utilizada para indexar o objeto.
- *put*: é a operação para adicionar um novo estado no livro razão ou atualizar um já existente.
- *delete*: remover um objeto dos estados do livro razão. Esta operação não retira a transação realizada do registro imutáveis de transações do livro razão.

3.3 COMPONENTES DE UMA REDE HYPERLEDGER FABRIC

3.3.1 Organizações

No Fabric, Organização (ORG) é a entidade que representa os membros que fazem parte da rede. Uma organização representa o individuo, ou grupo de indivíduos, que fazem parte da *blockchain*. Para ela ingressar, uma ORG adiciona sua Membership Service Provider (MSP) rede para definir suas informações. O *MSP* irá definir de que maneira os outros membros participantes poderão verificar se as assinaturas geradas foram realizadas por uma identidade válida. Ademais, ao entrar na rede, os direitos de acesso à informação por esta *ORG* é definido no seu MSP e cabe as outras organizações participantes aceitarem ou não o seu ingresso.

Organizações específicas podem fazer parte de vários canais e possuírem vários pares em sua composição. Ademais, no Fabric é possível definir um conjunto de várias organizações agrupadas logicamente, formando, assim, um consórcio.

3.3.2 Autoridades Certificadoras

Uma ou várias Autoridades Certificadoras são responsáveis por distribuir certificados para o participantes da rede na *blockchain* Fabric. "O Certificado Digital é o documento eletrônico assinado por um meio confiável responsável por credenciar a identidade do participante"(MENKE, 2003). A Autoridade Certificadora (AC) é responsável por assinar digitalmente o certificado, junto a isso, ela vincula o certificado a chave pública do participante da rede. Ademais, é possível que a AC adicione propriedades no certificado digital. Dessa forma, terceiros que desejam se comunicar com o participante e que têm confiança na entidade certificadora, poderão validar através da chave pública a autenticidade de um certificado.

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

Em sua estrutura, o Fabric permite o desenvolvimento de uma hierarquias para a emissão de certificados digitais. Esta organização, permite que ACs raiz emitam certificados para AC intermediárias, possibilitando que o processo fique mais descentralizado e menos rastreável. Como as AC intermediárias tem seus certificados emitidos por uma AC raiz, cria-se uma cadeia múltipla de confiança. Sendo assim, mesmo que por alguma interferência uma AC intermediária venha ser comprometida, as demais não sofrerão consequências.

3.3.3 Membership Service Provider (MSP)

O MSP é um conjunto de pastas de configurações que refere-se a um componente abstrato no sistema, o qual é responsável por fornecer credenciais para os participantes da rede. Este componente é feito de maneira totalmente modular, onde mesmo estando fortemente ligado a outras interfaces na rede, é possível modificá-lo e definir novas configurações sem que outros componentes sejam alterados.

Ou seja, o MSP funciona de tal maneira onde, ele é responsável por definir quais identidades são reconhecidas pela rede. Por exemplo, é o MSP que é responsável por definir se um par tem permissão para endossar a transação, onde ao utilizar a chave pública do certificado do par, esta então é usada para verificar se o MSP reconhece a identidade como uma fonte confiável para o resto da rede (Hyperledger Fabric, 2020).

Ao invés de gerar as certificações de autoridade como a AC faz, o MSP é responsável por conter uma lista de identidades que possuem permissão. Por exemplo, o MSP é responsável por identificar quais AC estão autorizadas para emitir identidades válidas para os membros da rede. Além disso o MSP, no Fabric, se organiza de uma maneira abstrata, onde este se adapta para configurações de todos os tipos, sejam elas para o canal, pares ou ACs.

3.3.4 Pares

Quando falamos de uma rede Fabric, temos como principal elemento em sua composição os pares, ou nós, como também são chamados. Diversos pares na rede Fabric, armazenam cópias dos livros razão e das chaincodes. Esta replicação de dados na rede, é o que permite que ela não tenha um único ponto de falha, ou seja, mesmo que um par venha a não funcionar, a rede pode se manter ativa.

Além dos pares permitirem a comunicação distribuída da rede, são eles que possibilitam a utilização de APIs para comunicação do usuário com a rede *blockchain*.

Quais cópias de livros razão e instâncias de *chaincodes*, serão presentes em um par é definido pelos canais que ele faz parte, esta lógica de canais será melhor explicado na Subseção 3.3.5. A organização de canais e pares na rede Fabric é o que permite que ela seja tão flexível. Na Figura Figura 6, é demonstrada uma estrutura básica de exemplo, nela é possível evidenciar como canais possuem diferentes cópias de livro razão e instâncias de *chaincode*. A

partir disso, é possível criar estruturas tão complexas quanto forem necessárias. Dessa forma, contextos de comunicação privados para pares específicos são possibilitados.

3.3.4.1 Tipos de pares

Na rede Fabric, todos os pares que fazem parte do mesmo canal possuem cópias das mesmas informações, ou seja, todos eles possuem uma cópia do livro razão. Contudo, diferentes pares podem apresentar permissões e funcionalidades distintas. Em sua estrutura, o Fabric determina alguns tipos de pares (Hyperledger Fabric, 2020), dentre eles pode-se citar: par âncora, par de endosso e par líder.

Par âncora, na rede Fabric, representa um ou mais pares, que servem como referência para que os outros pares sejam encontrados através do Protocolo de Gossip, este que será explicado de maneira detalhada na subseção 3.3.4.2. Nas configurações do canal é determinado o endereço para um par âncora; e este irá comunicar a respeito de todos os pares que ele já teve contato, permitindo assim que informação se propague na rede.

Na rede Fabric, pares de endosso são aqueles que participam ativamente do processo de consenso para validação de transações. Ou seja, cabe ao par de endosso confirmar a validade de uma transação para uma determinada política aprovar sua entrada na rede.

O Fabric se organiza de tal maneira onde cada organização possui um par líder, e este possui a função de disseminar a resposta obtida pela ordem de serviço. Ou seja, quando uma transação ou conjunto de transações é confirmada pelo serviço de ordenação, ela é passada a um par líder, e este passa os blocos para os outros pares presentes em uma mesma organização (BACKYARD, 2018). Caso um par líder fique indisponível em uma ORG, um novo será eleito de acordo com um algoritmo pre-estabelecido de seleção.

3.3.4.2 Peer Gossip Protocol

No Fabric, para ser possível escalar a rede sem que houvessem perdas de desempenho e segurança, foi necessário um tratamento especial na maneira que as informações são passadas

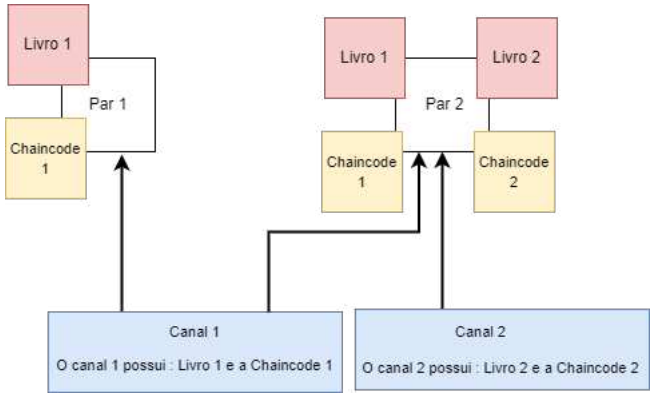


Figura 6 – Exemplo da estrutura dos pares na rede Fabric.

entre os nós participantes. *Gossip Protocol*, ou Protocolo de "fofoca" traduzido pro português, é como o Fabric mantém os dados consistentes entre os múltiplos pares.

Primeiramente, um par líder precisa ser selecionado na rede. Ele é o par que irá se comunicar com o serviço de ordenação do canal, este processo de ordenação é explicado na Subseção 3.3.9. Dessa forma, obtendo os blocos e livro razão atualizado.

Cabe ao *Protocolo Gossip* identificar os nós disponíveis, e aqueles que estão *offlines*. Esse algoritmo permite a disseminação de informação sem que todos os pares se conheçam mutualmente, pois, mesmo que um par não tenha ligação direta com o líder, ele pode ter uma comunicação com outro par que teve uma relação direta ou indireta com o líder. No protocolo, pares recentemente ingressados no canal serão rapidamente atualizados, mantendo a velocidade e escalabilidade da rede. Pares que ficaram indisponíveis, por algum motivo, serão atualizados no momento que se comunicarem com outro par da rede, ou ao se conectarem ao serviço de ordem. Ademais, é possível que um par solicite a informação diretamente, ao invés de esperar que ela chegue para ele, isso pode ser feito por um método de *pull*, neste trabalho não será dado uma explicação profunda sobre como ocorre esse método, para aprofundamento recomenda-se a leitura do material fornecido em (Hyperledger Fabric, 2020).

Para que a rede se mantenha segura, os pares assinam suas mensagens com sua identidade ao enviar informações para os outros. Dessa forma, um usuário mal-intencionado é facilmente identificável e punível em um canal.

Este trabalho não explicará de maneira detalhada como o *Protocolo Gossip* dissemina a informação, busca-se somente dar um entendimento de como a informação se mantém consistente na rede *blockchain* do Fabric. Para o leitor mais curioso, que deseja saber detalhes e complexidades da implementação desse algoritmo, recomenda-se a leitura do artigo *Gossip-Based Peer Sampling* (JELASITY et al., 2007).

3.3.4.3 Comunicação com Aplicações

Como dito anteriormente, o par é o principal componente responsável por possibilitar a comunicação de um aplicativo com os serviços disponíveis na rede blockchain. Em sua documentação, o Fabric determina quatro principais etapas para comunicação entre uma aplicação e a rede, sendo elas: (1) A aplicação se conecta a um par âncora (2) uma proposta de transação é enviada para múltiplos pares de endosso (GAUR et al., 2020) (3) uma resposta da proposta da transação é obtida (4) O Software Development Kit (SDK)[6] do cliente coleta a resposta da transação e a envia para o serviço de ordenação, este que irá ordenar a transação em blocos e distribuir para todos os pares do canal (GAUR et al., 2020). Portanto, para ser possível a comunicação entre pares e aplicações, desenvolvedores utilizam como base o Fabric SDK.

Transações de consulta (leitura de livro razão) são retornadas de maneira imediata. Contudo, transações que precisam fazer atualizações ou inserções no livro razão demandam

[6] Kit de Desenvolvimento é um conjunto de ferramentas cujo intuito é facilitar a vida dos desenvolvedores

uma maior complexidade. Esta complexidade, é definida pela necessidade de comunicação entre aplicativos, pares e ordenadores.

Aplicações que executam transações de atualização em um canal demandam um tratamento diferente. Para ser possível atualizar uma informação no livro razão, o (Hyperledger Fabric, 2020) define que as seguintes etapas precisam ser seguidas:

1. Aplicação conecta-se um par âncora;
2. A proposta de transação é enviada para os múltiplos pares de endosso do canal;
3. A resposta da transação é enviada para a aplicação solicitante;
4. As transações são enviadas para serviço de ordenação para que ele possa ordenar as transações nos blocos.
    4.1. As transações são enviadas para os par líder de cada organização armazenadas em blocos.
    4.2. O par líder repassa os blocos para todos os pares das organizações para serem atualizadas suas cópias do livro razão.
5. A resposta da atualização é enviada para o solicitante (neste caso a aplicação).

3.3.5 Canal

Na arquitetura de uma rede, o Hyperledger se organiza em estruturas denominadas canais. Canais representam uma sub-rede, responsável por agrupar organizações que possuem a mesma finalidade. Além disso, a arquitetura de canais permite que organizações que desejam compartilham informações somente entre si, possam se comunicar. Dessa forma, uma organização que não faça parte do canal, por exemplo, não conseguirá acessar as transações realizadas por este canal. O Fabric possibilita múltiplos canais na rede, isto é o que permite que diversos modelos de negócio coexistam na mesma rede blockchain, sem que um interfira no outro.

Cada canal é organizado a partir: (1) organizações que o compõe; (2) um livro razão compartilhado entre os membros deste canal; (3) chaincodes; (4) nós ordenadores; (5) nó(s) de ancoragem; (6) configurações inicias do canal. Ademais, as organizações pertencentes a um canal irão atribuir a ele, seus pares. Dentre os pares estabelecidos, deve ser escolhido um par líder. O serviço de ordenação estabelecido na rede ordena as transações e as devolve para o par líder, então, o líder distribui os blocos para os outros pares participantes do canal através do Protocolo Gossip.

Toda transação realizada na rede Fabric, será necessariamente executada por um canal. Estas transações precisam ser autenticadas e validadas segundo as políticas definidas no canal. Para isso, todo novo par que desejar ingressar em um canal já existente, precisará de uma identidade válida e autenticada com as configurações preestabelecidas, processo este realizado pelo MSP definido para o canal.

3.3.6 Identidade

Em uma rede *blockchain* no Fabric, todos os participantes (pares, organizações e aplicativos) precisam estar identificados. Por uma identidade, é possível definir as permissões que os participantes poderão ter sobre a rede blockchain. Portanto, uma identidade precisa ser confiável. Para isto, o Fabric utiliza como autoridade confiável o MSP.

Na rede Fabric identidades digitais são definidas a partir de certificados digitais. Certificados digitais são documentos cujo objetivo é atribuir confiança para as transações realizadas em um ambiente digital. As certificações mais comumente utilizadas são aquelas compatíveis com o padrões X.509, estes compostos por: (1) atributos como sujeito, AC emissora e outras informações necessárias como validade e versão do certificado, (2) uma assinatura digital realizada por uma AC confiável, (3) e uma chave pública vinculada o proprietário do certificado (SSL.COM, 2021).

3.3.7 Livro razão

Para facilitar a execução dos contratos inteligentes na rede Fabric, seu livro razão ou *ledger* do inglês, é divido em duas partes: (1) registro imutável com o histórico de todas as transações realizadas, (2) e um estado global organizado como estrutura *chave-valor* [7] que mantém o valor atual de um objeto.

Em seu registro imutável de transações, o livro razão visa armazenar todas as transações realizadas na blockchain. A partir desta estrutura, é possível observar o histórico de transações, que fez com que a rede estivesse no estado atual. Por ser imutável, nenhuma transação realizada será removida do seu log, mesmo que um objeto armazenado na rede possa ser deletado, por exemplo. Sendo assim, uma vez escrita, uma transação será permanentemente auditável.

Esta estrutura permite que seja mais fácil obter um valor atual de um objeto, sem que seja necessária realizar cálculos percorrendo todas as transações realizadas. Na rede Fabric, estados globais são flexíveis e podem armazenar simples objetos ou até mesmo estruturas complexas mais robustas. Diferentemente do registro imutável, um estado global pode ser modificado e/ou deletado.

3.3.8 Politicas

A estrutura flexível do Fabric, permite que sejam estipuladas diferentes configurações de políticas para os diferentes componentes existentes. Ou seja, o Fabric possibilita configurações de política para: canais, organizações, transações, consórcio e entre outras.

No Fabric, uma política define regras que precisam ser respeitadas. para isto as políticas, definem qual acesso cada participante tem aos recursos disponíveis na rede. Por exemplo,

[7] Chave-valor é um paradigma que indexa dados ao uma chave de identificação.

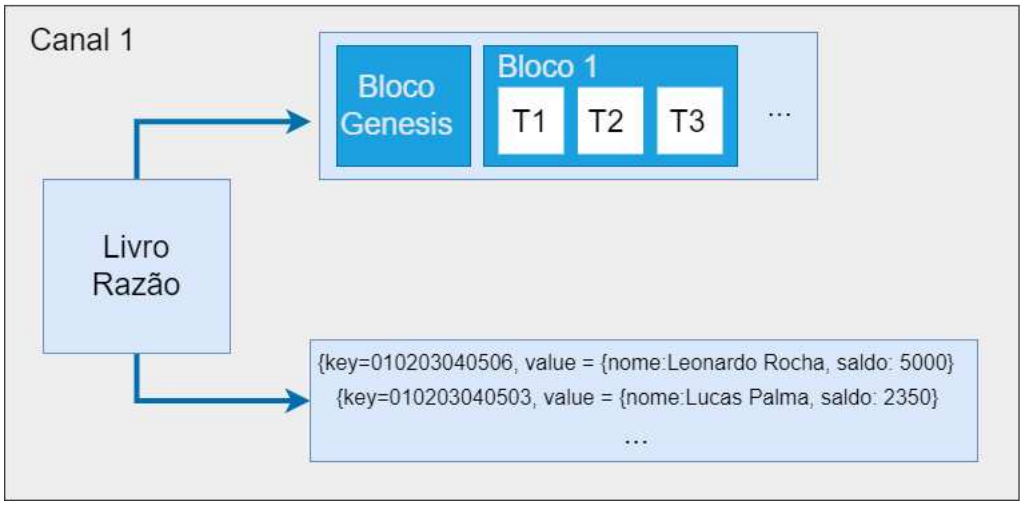


Figura 7 – Exemplo da estrutura do livro razão no Fabric.

uma política no Fabric estipula quantos e quais membros precisam entrar em um acordo para que uma alteração em um contrato seja feita, ou também para que uma alteração no canal seja aprovada.

O Fabric organiza sua política em diferentes níveis, onde cada um deles é responsável por diferentes componentes da rede. A documentação do Hyperledger Fabric (Hyperledger Fabric, 2020) determina, três níveis para suas políticas, sendo eles: *configurações do Canal do Sistema*, *configurações do Canal Aplicativo* e *configurações de Acesso a Recurso*

3.3.9 Serviço de ordenação

Diferentemente das *blockchains* vistas anteriormente, o Fabric mantém o consenso entre os diferentes nós na rede de outra maneira. Na Subseção 3.3.4 foram definidos três tipos: endossos, âncora e líder. Contudo, em sua organização ainda existe o par de ordenação. Na rede Fabric um ou mais pares de ordenação, participam ativamente do processo chamado: serviço de ordenação.

O serviço de ordenação junto a protocolos de consenso, como Gossip, é o que garante que os nós na rede Fabric, estejam em consistência. Os nós de ordenação na arquitetura do Fabric, fazem parte de uma organização, esta sendo definida nos arquivos de configurações conhecida por todos os consórcios participantes do canal.

Por organizar sua estrutura por algorítimos determinísticos, o Fabirc garante que todas as transações validadas por um par de endosso estão corretas. Dessa forma, cabe aos ordenadores definir a ordem correta de execução desta para que assim seja possível garantir um consenso entre os participantes da rede.

Para enviar uma transação na rede, um cliente utiliza o SDK, para uma proposta ser enviada. No SDK junto a proposta estão definidas configurações como: *Nome do canal*, *Nome da Chaincode para invocar*, *parâmetros de entrada* (MANEVICH; BARGER; TOCK, 2018). Com o nome do canal, será verificada as configurações do canal para saber quais pares são

responsáveis por endossar a transação solicitada. Caso a transação esteja bem estrutura, o par de endosso assina a transação usando o MSP. Ao obter um consenso entre os pares de endosso participantes da rede, uma proposta de transação é definida. Caso, a transação solicitada seja somente para leitura no livro razão, o serviço de ordenação não é necessário ser acionado, sendo assim, os pares de endosso solicitam o valor definido pela chave e retornam uma resposta de transação ao cliente. O SDK do Fabric para o aplicativo cliente irá confirmar as assinaturas e resposta da transação, confirmando que ela concorda com o esperado. Contudo, caso o cliente esteja solicitando uma transação que necessita fazer uma atualização no livro razão, será necessário então acionar o serviço de ordenação.

Quando acionado, o serviço de ordenação não possui a função de "validar" as transações, pois isto já foi feito e garantido pelos pares de endosso. Portanto, cabe a este serviço duas principais funções: (1) organizar as transações na ordem correta (já que as transações podem chegar ao serviço de ordenação em uma ordem diferente do que foram executadas) (2) atualizar o livro razão de maneira imutável. Nesta fase, ao receber as transações, o serviço de ordenação, as empacota em uma sequência correta e bem definida, as colocando em blocos. Os blocos, definidos pelo serviço, farão parte da blockchain, efetivando assim as transações.

No serviço de ordenação, o livro razão é atualizado, sendo este final e imutável e os novos blocos são passados para os pares que fazem parte do canal. Pares que estejam inativos no momento que a operação foi realizada, receberão estes blocos ao se conectarem ao serviço de ordem ou se comunicarem com outro par (Hyperledger Fabric, 2020). Isto, permite que o livro razão dos pares, se mantenha correto e em consenso.

4 TRABALHOS RELACIONADOS

Este capítulo, tem como principal objetivo realizar uma busca de trabalhos relacionados a proposta desenvolvida neste TCC, e a partir disso, ter uma base e referência para o desenvolvimento. Nas descrições realizadas neste capítulo, demonstram-se os pontos positivos e as fragilidades dos modelos relacionados. Ademais, busca-se realizar um estudo mediante as tecnologias utilizadas por outros trabalhos em comparativo com as que serão utilizadas no trabalho proposto.

Sendo assim, para uma seleção eficiente dos trabalhos relacionados, utilizou-se como base as métricas definidas por (KITCHENHAM et al., 2009). Além disso, definiram-se questões alvo da pesquisa, sendo elas:

1. Existem modelos de exemplos, ou propostas que abrangem o que é proposto neste trabalho? Como esses projetos de base incentivam e contribuem para o atual projeto?

2. De que maneira é possível integrar um dispositivo móvel a uma aplicação on *chain*?

3. De que maneira um estudante pode ingressar e participar da rede *blockchain*?

4. De que maneira aplicativos descentralizados (DApps) que utilizam de *blockchain*, podem trazer benefícios para o usuário?

Buscando responder a perguntas definidas, na Tabela 1 estabeleceu-se uma lista de palavras-chave e sinônimos, estas que representam os filtros para as buscas nas plataformas de busca.

Palavras-chave	**Sinônimos**
blockchain	*distributed ledger; bitcoin; ethereum; Hyperledger*
student	
degree certificates	*diploma; academic register; education; academic records;*
higher education	*government; educational institutions; education administration*
Dapp	*decentralized application; mobile; user interface*

Tabela 1 – Palavras-chave e sinônimos.

A partir da lista de palavras-chave e sinônimos definida e tendo como base os mecanismos virtuais voltados para área cientifica, como Google Scholar [1], ACM Digital Library

[1] https://scholar.google.com.br/

[2], Semantic Scholar [3] e Research Gate [4], foi aplicada diretamente na barra de pesquisa dos buscadores a seguinte *query*:

$$\begin{aligned}(&\text{"}blockchain\text{"} \ OR \ \text{"}distributed \ ledger\text{"} \ OR \ \text{"}bitcoin\text{"} \ OR \ \text{"}ethereum\text{"} \ OR \\ &\text{"}Hyperledger\text{"}) \ AND \\ &(\text{"}student\text{"}) \ AND \\ &(\text{"}diploma\text{"} \ OR \ \text{"}academic \ register\text{"} \ OR \ \text{"}education\text{"} \ OR \\ &\text{"}academic \ records\text{"} \ OR \ \text{"}degree \ certificates\text{"}) \ AND \\ &(\text{"}higher \ education\text{"} \ OR \ \text{"}government\text{"} \ OR \ \text{"}educational \ institutions\text{"} \ OR \\ &\text{"}education \ administration\text{"}) \ AND \\ &(\text{"}Dapp\text{"} \ OR \ \text{"}decentralized \ application\text{"} \ OR \ \text{"}user \ interface\text{"})\end{aligned} \tag{4.1}$$

Após as pesquisas iniciais, definiu-se três critérios de seleção para os trabalhos relacionados a proposta. O critério um diz respeito a leitura dos títulos dos trabalhos, eliminando aqueles não estão relacionados com a obra atual. Continuando, o segundo critério utilizado foi a leitura da introdução e resumo (*abstract*) dos trabalhos relacionados, eliminando aqueles que não estão compatíveis com esta proposta. Finalizando, o último critério de seleção foi leitura completa dos trabalhos, retirando aqueles que divergem do tema e/ou não respondem alguma das questões de pesquisa.

Para ser possível uma busca mais refinada, em algumas das bases científicas foi estabelecido o critério de seleção dos trinta primeiros resultados, filtrados por ordem de relevância. Dessa forma, evitou-se uma busca exaustiva de trabalhos. Ademais, para os resultados serem condizentes com o cenário atual foi atribuído o filtro de data. Esse filtro, definiu que somente as pesquisas realizadas no período de 2012 (incluindo) até 2022 (incluindo), fossem consideradas. Na tabela 9, destaca-se os resultados obtidos em cada uma das iterações de seleção.

Database	**Pesquisa Inicial**	**Seleção 1**	**Seleção 2**	**Seleção 3**
ACM Digital Library	97(30)	6	2	1
Google Scholar	56(30)	18	8	4
Semantic Scholar	100(30)	2	1	4
Research Gate	100 (30)	10	4	1
Total	**120**	**36**	**15**	**10**

Tabela 2 – Resultados da revisão de literatura.

[2] https://dl.acm.org/
[3] https://www.semanticscholar.org/
[4] https://www.researchgate.net/

4.1 SUMÁRIO

Nas próximas seções, será dado um breve resumo, descrevendo os trabalhos relacionados. Nesta descrição, abordam-se as ideias principais dos autores e os prós e contras encontrados nas propostas. A partir destes apontamentos, é possível ter uma base e um comparativo para o trabalho proposto neste Trabalho de Conclusão de Curso (TCC). Sendo assim, ao término deste capítulo, na seção 4.2, apresenta-se uma tabela visando-se ressaltar as diferenças mais importantes para uma proposta nesta pesquisa.

ANALYSIS OF BLOCKCHAIN TECHNOLOGY FOR HIGHER EDUCATION

No artigo (VIDAL; GOUVEIA; SOARES, 2019), os autores propõem um estudo mediante as tecnologias *blockchain* existentes, no que diz respeito a sua aplicabilidade na emissão e gerenciamento de certificados. No artigo, é descrito um modelo onde instituições armazenam transações emitindo um diploma para o estudante na *blockchain*. A partir disso, tanto empregadores como estudantes, podem verificar se o diploma é autêntico. O protótipo proposto utiliza como base a Blockcerts [5] Wallet, ela permite um estudante gerir os diplomas que possui direito. Por uma aplicação *mobile*, o estudante poderá comprovar a autenticidade e propriedade sobre seu diploma. Além disso, no portal de acesso, um estudante pode baixar os diplomas e interagir com os outros serviços da rede. Ademais, empresas podem verificar a autenticidade das informações passadas pelo estudante por verificações na blockchain.

No sistema proposto é transferida a responsabilidade e autoridade para o estudante. Assim, após receber o diploma gerado por uma instituição credenciada, somente ele poderá decidir com quem esse documento será compartilhado. Junto a isso, a proposta estabelece um modelo descentralizado, onde não existe a necessidade de uma entidade representativa para emissão de diplomas, ou seja, qualquer um com acesso à rede e com as informações corretas poderá se responsabilizar por este processo.

Contudo, apesar do DApp, junto a rede proposta pelos autores VIDAL; GOUVEIA; SOARES, permitir que o estudante seja o eixo principal de controle, existe um problema relacionado como esse irá se identificar na rede. A autenticação e autorização digital é feita utilizando criptografia de chaves públicas. Porém, apesar da segurança proporcionada pela utilização deste mecanismo, existe um ponto negativo. Caso um estudante perca sua chave privada, não existe uma maneira na proposta definida pelo autor de recuperará-la. Dessa forma, nesse cenário, um estudante perderia o acesso aos seus certificados. Portanto, se faz necessário buscar uma forma de resolver este problema.

[5] https://www.blockcerts.org/

A BLOCKCHAIN-BASED ARCHITECTURE FOR QUERY AND REGISTRATION OF STUDENT DEGREE CERTIFICATES

O Educ-Dapp (ABREU; COUTINHO; BEZERRA, 2020), é um protótipo de uma rede/aplicação descentralizada que utiliza da Ethereum como plataforma. Os autores propõem um modelo organizado em camadas, que representam as etapas necessárias para o gerenciamento de certificados de maneira distribuída. Ao emitir um certificado, é gerado uma versão eletrônica dele, contendo informações a respeito do diplomado. Após criptografado, é armazenado na rede o *hash* da versão eletrônica gerada. A partir disso, é criado um *QR-Code*, que está relacionado a consulta desta transação na rede. Esse código, poderá ser anexado a um certificado em papel, adicionando autenticidade e praticidade para verificação. Sendo assim, esse diploma poderá ser escaneado por um aplicativo para comprovar a veracidade. Assim como no modelo proposto por (VIDAL; GOUVEIA; SOARES, 2019), qualquer entidade com as informações válidas poderia participar do processo de emissão, removendo a ideia de uma figura central responsável por este papel.

Contudo, apesar da completude do trabalho proposto pelos autores, algumas questões não foram esclarecidas pela proposta, como, por exemplo:

(i) Como um estudante participa efetivamente da rede?

(ii) Como um estudante pode ingressar na rede?

(iii) Qual o custo para execução das transações na Ethereum?

DIGITAL CERTIFICATIONS IN MOROCCAN UNIVERSITIES: CONCEPTS, CHALLENGES, AND SOLUTIONS

O BCSC-DApp (MOHAMED et al., 2022) é uma proposta para reestruturação do sistema marroquino de emissão de certificados, onde as Instituições de Ensino Superior (HEI) estão unificadas logicamente. Instituições credenciadas podem emitir e armazenar certificados na rede *blockchain*. Já um estudante, ao criar uma conta no aplicativo do BCSC-DApp, poderá ver e baixar documentos digitais. Para isso, a autenticidade e controle de troca de informações entre estudantes, universidades e instituições é dado através de chaves privadas. Ou seja, ao emitir um certificado, uma instituição realiza uma transação para o estudante que possui a posse, utilizando como base sua chave pública. Dessa forma, somente o estudante que possui a chave privada poderá ter acesso ao certificado e controlar com quem ele deseja compartilhar. No BCSC-Dapp, diferentes instituições podem ser unificadas para o mesmo estudante, portanto, caso ele tenha diplomas gerado em entidades diferentes, não será necessário outras contas, isto permite que exista uma unificação do sistema de ensino.

Na proposta apresentada pelo autor MOHAMED et al., destaca-se a segurança, praticidade e descentralização do modelo apresentado. Contudo, no modelo apresentado o autor não

deixa claro alguns pontos no desenvolvimento da proposta, por exemplo, de que maneira o aplicativo desenvolvido se comunica com a rede Ethereum e quais tecnologias são utilizadas para isso. Além disso, o modelo não realiza uma descrição em relação aos custos as universidades precisaram pagar para utilização do sistema.

CVSS: A BLOCKCHAINIZED CERTIFICATE VERIFYING SUPPORT SYSTEM

CVSS (NGUYEN et al., 2018), é um modelo proposto que, diferentemente dos apresentados anteriormente, não visa mudar o processo de emissão de certificado, e sim unificar este processo a certificados digitais, portanto criando uma camada extra de segurança. Um emissor de um certificado, irá digitalizar um certificado. Esse certificado, irá conter informações como: (1) *Hash*, gerado a partir de todo conteúdo presente no certificado, para ser possível garantir a integridade dos certificados; (2) *Informação da blockchain*, representando o índex que será utilizado para obter o certificado na *blockchain*; (3) Informações do Proprietário, contendo dados como identificação pessoal, *e-mail*, número de identificação e outros; (4) Conteúdo do Certificado, apresentando informações a respeito da data de certificação, informações a respeito da criação e outros atributo e (5) *Snapshot* do Certificado [6], usado para comparação no processo de verificação. Ao emitir um certificado, uma instituição cria uma versão digital do mesmo, a partir disso, é gerado um índex na rede *blockchain* que poderá ser verificado. A identificação do certificado na blockchain, é enviado para o *e-mail* do proprietário, que poderá então verificar as informações digitalmente. Após finalizado o processo na *blockchain*, um *QR-Code* é gerado e será anexado ao documento físico, similar ao modelo proposto em (ABREU; COUTINHO; BEZERRA, 2020). Através do aplicativo *e-Certificate*, é possível validar um certificado físico através do QR-Code anexado, verificando se as informações são autenticas e condizentes com aquilo esperado.

O autor da proposta descreve relatórios a respeito do desempenho da execução dos certificados, demonstrando que, em média, são necessários cerca de 5 minutos para validação de 60 certificados. Junto a isto, são fundamentais dois custos: (1) US$19 para criação do contrato inteligente na Ethereum, (2) US$0.15 por cada certificado gerado.

Contudo, apesar dos benefícios propostos, acrescidos do baixo custo e desempenho satisfatório, o modelo pode apresentar alguns problemas, ou incompletudes, destaca-se:

(i) Problema de escabilidade relacionado a rede *blockchain* pública;

(ii) Não define um papel e uma função para o estudante, o tornando um ator secundário;

(iii) Devido a volatilidade de criptomoedas, possui uma taxa de emissão muito variável.

[6] Snapshot, é uma cópia instantânea de volume ou captura instantânea de volume é o estado de um sistema em um determinado ponto no tempo.

PRIVACY PROTECTED BLOCKCHAIN BASED ARCHITECTURE AND IMPLEMENTATION FOR SHARING OF STUDENTS' CREDENTIALS

No trabalho proposto, o autor (MISHRA et al., 2021) propõe uma arquitetura que fornece suporte a diferentes partes interessadas. A arquitetura possibilita, diferentes funcionalidades e atribuições na rede, dentre elas pode-se citar: (1) registro de usuários: o sistema permite instituições governamentais registrarem estudantes; (2) *login* e cadastro de usuários: após atribuída uma conta por instituições governamentais, o usuário deverá definir uma conta na *blockchain*; (3) inscrição de alunos essa funcionalidade permite uma instituição de cadastrar um aluno quando ele for admitido, utilizando como base o par de chaves públicas do estudante; (4) adição de credenciais: a escola gera credenciais para um aluno sempre que os requisitos forem atendidos (5) recuperação e visualização de credenciais: essa funcionalidade que está disponível somente para um estudante, permite que ele acesse as credenciais adicionadas a rede pelas partes interessadas; (6) entre outras funcionalidades.

Para ser possível aproveitar dos benefícios de redes *blockchain*, como segurança, imutabilidade e descentralização, sem que sofresse o impacto das limitações de armazenamento que existem neste tipo rede, o autor (MISHRA et al., 2021) propõe integração com um modelo *off-chain*. Devido ao custo e as limitações de espaço de armazenamento, redes *blockchain* podem ser um problema quando precisamos guardar informações mais complexas e arquivos mais pesados. Dessa forma, para contornar esse problema o autor adota a solução de integrar um banco de dados comum, permitindo obter os benefícios de dois modelos distintos, o centralizado e o descentralizado.

Apesar do modelo proposto trazer benefícios, existem alguns problemas atrelados a plataforma base do desenvolvimento o ser Ethereum. Apesar de a Ethereum ser completa no que diz respeito a execução de contratos, possibilitando as mais complexas possibilidades, ela está atrelada a uma criptomoeda (ETH), dessa forma, o custo pela emissão de certificados na rede sofre impacto em relação à grande volatilidade de moedas digitais.

USE OF BLOCKCHAIN TECHNOLOGY FOR ACADEMIC CERTIFICATION

No trabalho realizado pelos autores (JARAMILLO; PIEDRA, 2020), é proposto um modelo de governança acadêmica, utilizando, como princípio, três principais atores: (1) estudantes; (2) universidade; (3) e empregadores. O protótipo desenvolvido utiliza como base a plataforma Ethereum. Junto a isso, é utilizado *web3j* [7], sendo uma API que permite fornecer aos desenvolvedores acesso a funções e contratos da rede Ethereum. Além disso, para ser possível abstrair as complexidades do sistema para o usuário final, uma aplicação *front-end* utilizando de React, concede acesso às funcionalidades disponíveis.

O modelo estabelecido, define que um estudante poderá ter uma carteira digital utili-

[7] Web3 Java Ethereum Dapp API

zando Metamask, através disso um estudante poderá se comunicar com diferentes universidades. Dessa forma, o aluno pode compartilhar seus certificados, adicionar um certificado na rede e solicitar que este seja verificado. Além disso, uma universidade poderá validar os certificados e adicionar novos certificados no sistema. Já um empregador, poderá acessar o sistema para verificar a autenticidade de um certificado digital.

Contudo, apesar dos benefícios apresentados nessa aplicação, destacam-se algumas problemáticas, por exemplo, o alto custo para validação de um certificado, podendo variar de US$77 até US$160. Outro fator, é que esse modelo não pensa no estudante como uma figura participativa do processo de criação do certificado, por exemplo, não possibilita que um estudante envie uma atividade complementar que deseja que faça parte do seu certificado, ou seja, o estudante não participa ativamente do processo de validação.

EDUCTX: A BLOCKCHAIN-BASED HIGHER EDUCATION CREDIT PLATFORM

Edu CTX (TURKANOVIć et al., 2018), é uma proposta de rede *blockchain* e um aplicativo descentralizado que reestrutura todo modelo de emissão de certificados das HEI. Utilizando como base a plataforma Ark-Blockchain [8]. São definidas HEI, que participam do processo de emissão de certificados. Estas universidades podem então acessar a rede *blockchain* utilizando-se de uma API. Utilizando como base o Sistema Europeu de Transferência e Acumulação de Créditos (ECTS), nesta proposta, uma instituição transfere 0.1 crédito para o estudante que deseja registrar na rede, onde ela define informações básicas e aquelas que devem ser preenchidas pelo estudante. Para isso, as HEI utilizam como base a chave pública do estudante que deseja adicionar, dessa forma, somente ele terá acesso à informação através de sua chave privada, portanto, mesmo que os estudantes sejam passados em um canal público, a segurança estaria garantida. Após preencher as informações, o estudante realiza uma transação para a instituição que o transferiu créditos para ele anteriormente, para que ele possa ser efetivamente cadastrado na rede *blockchain*. Dessa forma, então, a carteira de um estudante é confirmada, possibilitando que ele possa receber certificados gerados. Para possibilitar que, tanto estudantes, como universidades se comuniquem com a rede, são utilizados *endpoints* [9], que se comunicam com uma API REST FULL, permitindo dessa forma a interação como usuário final.

Contudo, apesar do modelo estipular uma interação com o estudante, não são apresentadas muitas funcionalidades para ele, que possam ir além de visualizar e baixar um certificado. Além disso, destaca-se a necessidade de todas as operações estarem atreladas a um valor de crédito (criptomoeda), isto ocorre pela forma que a plataforma blockchain utilizada se organiza.

---

[8] https://ark.io/

[9] Um ponto de extremidade de comunicação é um tipo de nó de rede de comunicação. É uma interface exposta por uma parte em comunicação ou por um canal de comunicação.

DIUCERTS DAPP - A BLOCKCHAIN-BASED SOLUTION FOR VERIFICATION OF EDUCATIONAL CERTIFICATES

DIUcerts DApp (SHAWON et al., 2021), é uma proposta descentralizada para emissão, validação e gerenciamento de certificados, tanto para estudantes, como para universidades e empregadores. Assim como na proposta apresentada no trabalho "Analysis of Blockchain Technology for Higher Education", este modelo utiliza-se como base a plataforma Ethereum e a carteira digital Metamask para possibilitar seus serviços. A interface que abstrai a *blockchain* e facilita o uso das suas funcionalidades, foi desenvolvida através de React e Nodejs. Além disso, para comunicação com a *blockchain* foi utilizado o módulo *Web 3.js*.

Neste modelo, assim como em outros, subdivide-se a arquitetura em três principais atores, sendo eles, universidades, estudantes e empregadores. Na rede, cada um dos atores, terá contratos específicos, representandos as funcionalidades que ele poderá realizar. A ideia principal desse projeto, é permitir que uma instituição valide certificados e o vincule através do *hash* da transação e o *Certificate ID*. Após validado, a autoridade envia o certificado para o estudante que poderá ter acesso. Sendo assim, um empregador ao receber um certificado de um estudante, poderá verificar a autenticidade do mesmo na rede *blockchain*, através do Certificate ID fornecido. Caso ele seja inválido, uma mensagem de erro será retornada ao empregador.

Contudo, apesar do modelo apresentar funcionalidades para os três atores e possibilitar uma interface de fácil compreensão e de alta praticidade, foram encontrados algumas limitações: (1) a estrutura de um currículo é muito “moldada”, dessa forma, não permite que instituições diferentes possam abrir características únicas para seus certificados digitais, sendo necessário assim um padrão; (2) o autor, não aprofundou as funcionalidades que cada ator poderá ter e de que maneira são realizados os controles de segurança; (3) como o trabalho proposto utiliza-se como base a rede Ethereum, transações estão associadas a criptomoeda ETH, sendo assim, sofrem o impacto da volatilidade dos preços desta moeda digital; (4) e devido a como a implementação foi adotada, pode existir uma dificuldade para encontrar um contrato específico, problema este sendo identificado pelo próprio ator SHAWON et al.. Esse problema poderia ser contornado adotando uma alternativa com *QR-code* para identificação dos certificados.

EDUCATIONAL BLOCKCHAIN: A SECURE DEGREE ATTESTATION AND VERIFICATION TRACEABILITY ARCHITECTURE FOR HIGHER EDUCATION COMMISSION

HEDU-Ledger (KHAN et al., 2021) é um protótipo, que utiliza como base a plataforma Hyperledger Fabric. Na arquitetura de rede desenvolvida, as universidades são divididas em nós, que serão responsáveis pela validação e autenticação dos certificados. Como a rede utiliza a plataforma Hyperledger Fabric, permite que se adicione um grau de privacidade e uma maior escalabilidade, já que possibilita a utilização de protocolos de consenso customizados. Como se trata de uma rede privada, cabe as universidades aprovarem, em acordo, quais serão aquelas que irão ou não ingressar na rede. Além disso, o consenso na rede é garantido através de

uma comunicação e validação entre todos os nós da rede. Portanto, ao tentar inserir um novo certificado na *blockchain*, ele deverá ser aprovado pelas demais instituições no sistema.

Para ser possível facilitar a comunicação das universidades com a rede *blockchain*, uma API *REST FULL* foi desenvolvida. A API, permite se comunicar com os serviços oferecidos pelo Fabric na rede HEDU-Ledger, lá um participante poderá, por exemplo, enviar um certificado que será validado pelos outros participantes.

Contudo, apesar da proposta ser bastante explicativa no que diz respeito à arquitetura da rede e características do Hyperledger, a explicação de como a API interage com o sistema foi superficial. Outro fator, é que o trabalho apresentado não possui um foco na participação do estudante, sendo seu objetivo somente a validação de certificados digitais e armazenamento de maneira segura, utilizando blockchain no Fabric.

4.2 COMPARAÇÕES ENTRE AS PROPOSTAS

Em síntese, os trabalhos relacionados apresentam aspectos semelhantes em alguns quesitos, por exemplo, a forma que os estudantes são definidos na rede, de que maneira é realizada autenticação e controle de propriedade de certificados, a plataforma utilizada e outras características. Contudo, apesar desses trabalhos darem uma base e um guia para o desenvolvimento desta proposta, vários pontos divergem daquilo que se busca como resultado. Dessa forma, destaca-se, nos próximos parágrafos, as principais diferenças em relação às propostas utilizadas como referência e o trabalho proposto nesta pesquisa.

Nas propostas apresentadas nos trabalhos (KHAN et al., 2021), (NGUYEN et al., 2018), (MISHRA et al., 2021), (SHAWON et al., 2021) e (JARAMILLO; PIEDRA, 2020), o estudante é definido como um ator secundário na rede, ou seja, sua participação está somente atrelada a ser o proprietário para um certificado. Esses trabalhos, focam-se na emissão e validação dos certificados digitais das IES, dessa forma, não possibilitam uma participação mais ativa do diplomado na rede. Contudo, propõem-se neste TCC a possibilidade do estudante participar ativamente da rede. Na arquitetura, atribui-se uma organização representativa aos estudantes. Esta organização possui em seu canal um contrato inteligente, relativo às funcionalidades que estão disponíveis para um estudante participante. Um estudante poderá solicitar o registro de atividades complementares, como, por exemplo, cursos realizados, participações em eventos, estágios e entre outras atividades extracurriculares. Após armazenada, caberá a HEI, responsável pelo estudante, aprovar a atividade registrada, além disso, outras instituições serão responsáveis por endossar a transação. Ademais, além do cadastro de informações, um estudante terá acesso aos diplomas gerados por diferentes instituições que determinaram ele como proprietário do certificado, dessa forma, não será necessário que o aluno tenha diferentes carteiras digitais para cada instituição que faz parte. Outro diferencial do trabalho proposto neste TCC em relação àqueles que se utilizou como referência, está na plataforma *blockchain* utilizada. Por utilizar-se da plataforma Hyperledger Fabric, mesma plataforma utilizada na proposta (KHAN et al., 2021), não se utiliza uma criptomoeda como base para realização das transações.

Contudo, majoritariamente os trabalhos relacionados utilizam-se de plataformas que fazem isso com criptomoedas para troca de informações. Dessa forma, os trabalhos relacionados necessitam de um custo para realização das transações na rede, contundo, em grande maioria esses custos não são impactados diretamente para o estudante. Porém, os trabalhos (MOHAMED et al., 2022), (JARAMILLO; PIEDRA, 2020) e (TURKANOVIć et al., 2018) estabelecem custos para participação dos estudantes na rede e/ou para realização de transações por ele.

Ademais, a proposta apresentada neste trabalho estabelecerá a possibilidade da abstração das complexidades da utilização de uma rede *blockchain* por um dispositivo móvel. Isso, possibilita para um estudante acesso prático, eficiente e com poucas exigências de *hardware*. Contudo, majoritariamente os trabalhos relacionados, salva-se (VIDAL; GOUVEIA; SOARES, 2019) e (NGUYEN et al., 2018), desenvolveram aplicações que necessitam de computadores, ao exigirem de uma capacidade maior do que aquela fornecida por dispositivos moveis e/ou não apresentam compatibilidade com o mesmo, devido às ferramentas e plataformas utilizadas.

Junto a isso, percebe-se a carência de alguns trabalhos relacionados no que se diz respeito a realização de testes. Somente os trabalhos, (MISHRA et al., 2021), (NGUYEN et al., 2018), (KHAN et al., 2021), realizam experimentos em relação ao tempo de execução de transações na rede, sejam essas de leitura e/ou escrita. Sendo assim, no trabalho proposto nesta pesquisa, busca-se destacar o tempo de execução dos mais distintos processo, desde operações na rede *blockchain* até tempo de resposta para o estudante.

Finalmente, majoritariamente os trabalhos relacionados fazem a utilização de criptografia de chaves públicas para autenticação e autorização digital. Contudo, apesar da utilização de chaves públicas e privadas ser quase que unanimidade no processo de identificação de participantes em redes *blockchain*, sua utilização pode trazer problemas. Na criptografia de chaves públicas, não existe uma forma de recuperação caso, por exemplo, um estudante ou uma universidade perca o acesso a ela. Portanto, dar ao usuário mais leigo a responsabilidade de guardar suas chaves privadas pode ser um problema. Uma alternativa, por exemplo, é conseguir abstrair e facilitar esse processo de armazenamento para o participante da rede. Sendo assim, no trabalho proposto neste TCC, almeja-se estudar maneiras de contornar o problema citado, sem que interfira na confiabilidade e segurança proposta pelo modelo de chaves públicas e privadas.

A partir dos comparativos citados e tendo como bases o alvo final (estudante) e os objetivos deste trabalho, estabeleceram-se parâmetros de comparação entre as propostas. Os parâmetros estão relacionados a responder se o trabalho oferece ou não tal quesito, sendo eles:

(i) Qual o tipo de rede o trabalho relacionado foi desenvolvido?

(ii) Em que plataforma o trabalho foi desenvolvido?

(iii) O trabalho permite utilização em dispositivos móveis por um *app*?

(iv) O trabalho realiza experimentos em relação ao tempo de execução?

(v) O trabalho permite o estudante efetuar inserções na rede?

Figura 8 – Nome dos modelos apresentados.

Referências	Tipo
Analysis of Blockchain Technology for Higher Education	Pública
A Blockchain-based Architecture for Query and Registration of Student Degree Certificates	
Digital Certifications in Moroccan Universities: Concepts, Challenges, and Solutions	
CVSS: A Blockchainized Certificate Verifying Support System	
Privacy Protected Blockchain Based Architecture and Implementation for	
Use of blockchain technology for Academic Certification	
DIUcerts DApp - A Blockchain-Based Solution for Verification of Educational Certificates	
EduCTX: A Blockchain-Based Higher Education Credit Platform	Privada
Educational Blockchain: A Secure Degree Attestation and Verification Traceability Architecture for Higher Education Commission	

Figura 9 – Comparações entre os modelos.

Aplicativo Móvel	Teste de Tempo de Execução	Possibilita o estudante inserir dados na rede	Possibilita o estudante fazer leitura na rede	Custo em operações para o estudante	Interação com o estudante	Armazenamento offchain
✓	×	×	✓	×	✓	×
×	×	×	×	×	✓	×
×	×	×	✓	✓*	✓	×
×	✓	×	✓	×	✓	×
×	✓	×	✓	×	✓	✓
×	×	✓	✓	✓*	✓	×
×	×	×	✓	×	✓	×
×	×	×	✓	✓	✓	×
×	×	×	✓	×	✓	×

(vi) O trabalho permite o estudante efetuar leituras na rede?

(vii) O trabalho descreve os custos em relações aos custos das transações realizadas na rede?

(viii) O trabalho possibilita a interação do estudante com a rede?

(ix) O trabalho oferece utilização de um armazenamento *offcchain*?

5 PROPOSTA

Este capítulo apresenta uma proposta de integração entre estudantes e uma rede *blockchain* de gerenciamento de jornadas estudantis. O presente trabalho, utiliza como premissa primordial a rede em desenvolvimento pelo projeto Jornada, sendo atualmente mantido em uma parceria entre o MEC, LabSEC e Laboratório Bridge. Em suma, a rede base desenvolve uma arquitetura de rede *blockchain* que possibilita a emissão e controle de certificados digitais, de maneira descentralizada entre as diferentes Instituições de Ensino Superior. Contudo, a proposta em desenvolvimento não prevê o estudante como um ator primário na execução da rede. Dessa forma, foi vislumbrado a possibilidade de uma nova proposta que reformula algumas características e adiciona de forma mais completa e complexa a participação do estudante na rede *blockchain*. A rede que serve como base para este trabalho é descrita na região tracejada preta apresentada na Figura 10, onde sua arquitetura é composto por: (1) *n* organizações representado cada uma das universidades; (2) um conjunto de contratos inteligentes responsáveis principalmente pela emissão e controle de certificados; (3) um conjunto de aplicações para permitir acesso às funcionalidades da *blockchain*; (4) e um conjunto de API's que permite acessar as funcionalidades oferecidas pela Application.

Na proposta deste trabalho, busca-se desenvolver uma aplicação *mobile* em Flutter que por autenticação com o Governo Federal e comunicação com ferramentas de uma rede *blockchain* será possível a um estudante funcionalidades como: (1) visualizar os certificados emitidos para ele; (2) registro de atividades complementares; (3) unir todos os documentos oficiais de diferentes universidades em uma única carteira digital; (4) uma forma segura e prática de controle das suas atividades acadêmicas; (5) dentre outras funcionalidades. Sendo assim, foi necessária uma reformulação da arquitetura já existente, adicionando uma nova camada de controle na arquitetura da rede, e uma camada externa para comunicação da interface do usuário com os serviços oferecidos. Essas adaptações, podem ser observadas na Figura 10 e descritas da seguinte maneira:

(i) Uma nova organização para definir a participação dos estudantes.

(ii) Uma nova definição de contratos inteligentes em Go, destinados à participação dos estudantes e as funcionalidades que ele irá possuir na rede.

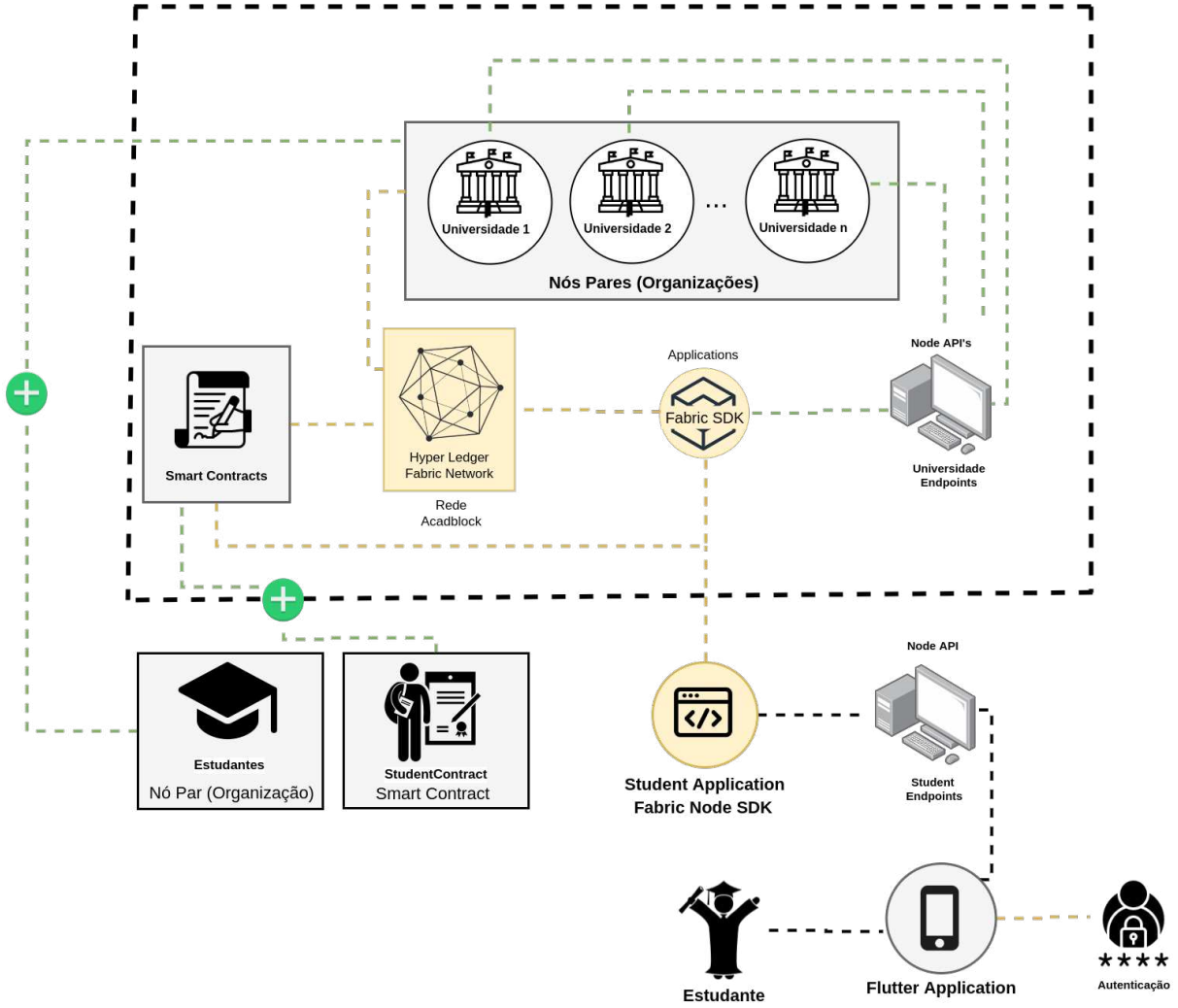
(iii) *Application*, utilizando o Fabric SDK para permitir comunicação com os serviços oferecidos pelo Fabric.

(iv) API para definir endpoints para comunicar o aplicativo com as funcionalidades da rede.

(v) Estruturação do sistema de autenticação que será utilizado.

(vi) Desenvolvimento da interface que se comunicará com a API e o usuário final.

Figura 10 – Motdelo proposto.



5.1 PREMISSAS DA PROPOSTA

Para o desenvolvimento desta proposta, é necessário um conjunto mínimo de dados preestabelecidos em uma rede. Esses dados, serão a base para apresentação de informações para o estudante. Sendo assim, para o foco deste desenvolvimento são cruciais duas estruturas: *currículo escolar* e *histórico escolar*.

Um currículo escolar é responsável por definir as regras e atividades que poderão ser realizadas por um estudante em sua jornada acadêmica. Portanto, para um currículo atender corretamente as demandas da proposta, este deve oferecer informações como:

- Dados do curso, como o nome e um código identificador;
- Dados da instituição, como nome e um código identificador;
- Carga horária total, podendo tanto ser em horas aulas como em horas relógio;
- Disciplinas obrigatórias oferecidas;

- Disciplinas optativas oferecidas;
- Atividades complementares oferecidas.

Um histórico escolar é a forma de um estudante provar e acompanhar sua trajetória no âmbito acadêmico, sendo, então, a estrutura mais importante para esta proposta. Para que os objetivos desta aplicação sejam atendidos em sua completude, espera-se que um histórico escolar ofereça:

- Informações do aluno, com atributos básicos como nome completo, data de nascimento e sexo, além de dados identificadores únicos;
- Código do currículo, para que seja possível validar e verificar a autenticidade de um histórico é necessário ter como referência o currículo regente que o define;
- Dados do curso;
- Dados da instituição;
- Data de ingresso do estudante no curso;
- Atividades realizadas, sendo este campo responsável por armazenar estágios realizados, atividades complementares, disciplinas obrigatórias e optativas;
- Situação do discente, atributo responsável por definir a condição do estudante no momento de emissão do histórico, como, por exemplo, se esse está formado, trancado, em intercâmbio e entre outras possibilidades;
- Dada e hora da emissão;
- Total da carga horária completa.

Devido ao Hyperledger Fabric possibilitar uma arquitetura modularizáveis e flexível, a proposta atual é capaz de se adaptar a diferentes tipos de rede *blockchain*. Ou seja, no desenvolvimento do aplicativo é possível comunicação com redes *blockchain* tanto públicas quanto privadas, quando forem definidas formas corretas de comunicação e acesso às organizações. Apesar da flexibilidade da proposta no que diz respeito ao Fabric, este trabalho não garante e não define maneiras de comunicação com outros tipos de redes *blockchain*, como por exemplo Ethereum. Ou seja, estipula-se como base para este projeto uma arquitetura Fabric.

Ademais, vislumbra-se como mínimo para o funcionamento da proposta a existência de um ou vários participantes na rede capazes de realizarem operações de escritas no livro-razão, para que, dessa forma, seja possível realizar a adição de históricos e currículos escolares, onde poderão ser acessados pelos estudantes. Além disso, é necessário que esta, ou outra entidade, consiga criar novos estudantes, para ser possível vincular as informações a um aluno.

5.2 REQUISITOS FUNCIONAIS

Nesta seção serão listados os requisitos que definem as funcionalidades oferecidas pela aplicação. Para este trabalho, estipula-se como requisitos funcionais:

- Permitir autenticação de um estudante;
- Comunicação de um aplicativo móvel com uma rede *blockchain*;
- Possibilidade do estudante cadastrar dados autodeclarados como por exemplo atividades complementares;
- Dar ao estudante a capacidade de visualizar seus dados acadêmicos;
- Possibilitar um estudante editar suas informações;
- Disponibilizar funcionalidade para baixar seus dados acadêmicos.

5.3 REQUISITOS NÃO FUNCIONAIS

Nesta seção serão listados os requisitos não funcionais relacionados ao uso e construção da aplicação em termos de desempenho, usabilidade, desenvolvimento e confiança:

- Utilização dos modelos do gov.br [1] para dispositivos móveis, visando uma interface limpa, padronizada e seguindo o modelo do projeto que dará base para o caso de uso em desenvolvimento.
- Utilização de Flutter para o *front end*, pois essa linguagem consegue manter uma estrutura visual entre diferentes plataformas (iOs e Android), sem que haja perdas em relação ao desempenho do desenvolvimento nativo;
- Utilização de Go para os contratos inteligentes, devido à plataforma que será utilizada para o desenvolvimento da rede (Hyperledger Fabric), ter Go como principal linguagem de desenvolvimento dos contratos.
- Utilização de Nodejs para a aplicação, devido a robusta documentação oferecida pelo Hypeledger Fabric que facilita o desenvolvimento utilizando o SDK.

[1] https://www.gov.br/ds/home

6 PROTÓTIPO

Neste capítulo será apresentado a aplicação que possibilitará demonstrar a proposta fundamentada neste TCC. Para isso, foi necessário tomar como base um projeto já em funcionamento: o Projeto Jornada, que será explicado em: Seção 6.1 e Seção 6.2. Contudo, tendo em vista que na proposta apresentada nessa pesquisa busca-se dar mais autonomia e transformar o estudante no ator principal, se faz necessário uma reformulação do modelo base. As adaptações necessárias e a estrutura de funcionamento da aplicação são explicados na Seção 6.3.

6.1 PROJETO JORNADA

O Projeto Jornada surge na ideia inovadora de trazer distribuição e descentralização para as IES brasileiras. Este projeto consiste em uma parceria entre o Ministério da Educação (MEC) e o LabSEC, onde através da tecnologia *blockchain* se visa propor um ecossistema de compartilhamento de dados que utilizam como base estruturas já definidas por outro projeto chamado Diploma Digital [1]. Estas estruturas são definidas por dois principais objetos: Currículo Escolar e *Histórico Escolar*. Dessa forma, o sistema em desenvolvimento capacita que instituições de ensino possam rastrear, validar e acompanhar, de maneira detalhada, informações a respeito do ensino superior. Além disso, o compartilhamento desses dados por *blockchain* inviabiliza possíveis fraudes.

Através das estruturas definidas e as *chaincodes* utilizadas no desenvolvimento do projeto, é possível acompanhar a jornada completa de um estudante durante seu período de graduação em uma Universidade Federal.

6.2 PREMISSAS OFERECIDAS PELO CASO DE USO.

No sistema desenvolvido pelo Projeto Jornada, IES são responsáveis pelo envio das informações que irão definir as regras de uma instituição em um curso específico e a participação de um estudante nesta entidade. Através da chaincode *Decree* é possível que instituições por meio de currículos escolares definam as regras como disciplinas, atividades complementares, cargas horárias e entre outras informações a respeito de um curso. Estes dados definem as configurações e regras de negócio que deverão ser respeitadas pelos históricos escolares que serão enviados. No sistema desenvolvido no Projeto Jornada é possível a comunicação de duas maneiras. A primeira sendo mais baixo nível, que possibilita a comunicação direta com a chaincode através da execução de uma transação diretamente na rede. A segunda possibilidade é a comunicação em alto nível através do *endpoint* disponibilizado pela API e enviando os dados. A comunicação direta com a chaincode não é recomendada, pois no nível mais baixo os erros que podem vir ocorrer não são simplificados para o usuário final. Já ao comunicar-se com API a instituição terá um suporte e um tratamento especial para os possíveis erros que possam ocorrer.

[1] http://portal.mec.gov.br/diplomadigital/

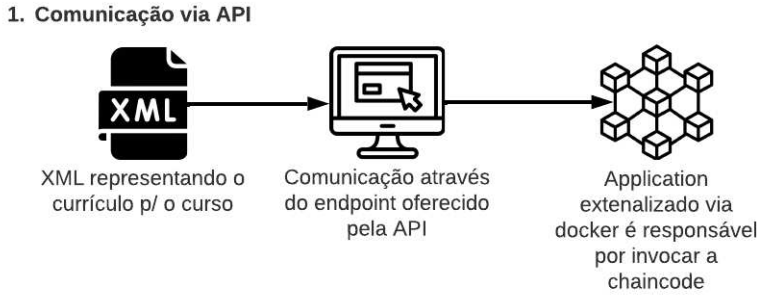


Figura 11 – Envio de XML pela API

Sendo assim, os currículos poderão ser enviados como duas estruturas, representados por XML via API, ou como JSON caso a comunicação seja feita na camada mais baixa. Na Figura 11, é possível observar o fluxo de execução do envio de currículo.

Ademais, para definir a jornada de um estudante no ensino superior instituições devem enviar os históricos escolares correspondentes. Através da chaincode *Academic Records*, é possível definir o estudante e suas participações em diferentes cursos. Os históricos escolares enviados precisam seguir as regras pre-definidas pelos currículos escolares. Assim como para a *Decree*, estes históricos podem ser enviados através da API Academic Records (com XML), ou via comunicação direta com a rede (com JSON).

Além de ambas chaincodes e APIs foi necessário a utilização de duas applications na execução do projeto jornada. As applications funcionam como um intermediador entre o usuário final e a rede *blockchain*. Sendo esta, responsável por criar uma nova transação através dos métodos disponibilizados pela biblioteca do Fabric.

Sendo assim, a execução do projeto jornada é dada da seguinte maneira: contêineres responsáveis pelo funcionamento da rede Fabric (contendo os pares, ordenadores e CAS), contêineres responsáveis pelas applications, onde estes estão externalizados através do seu endereço permitido que a API se comunique através da interface Docker e por último os contêineres responsáveis pelo funcionamento das APIs. Ressalta-se que para ser possível a comunicação entre os diferentes contêineres e camadas foi necessário a utilização de um conceito no Docker chamado de *bridge* ou ponte em português, que permite criar uma camada de isolamento na rede host possibilitando a comunicação entre os diferentes participantes.

Além de possibilitar o envio de currículos e históricos escolares, as chaincodes *Decree* e *Academic Records* já apresentam no seu desenvolvimento padrão outras funcionalidades, estas que estão voltadas principalmente para o processo de leitura dos dados. Na Figura 12, destacam-se as funcionalidades que podem ser invocadas por cada uma das chaincodes.

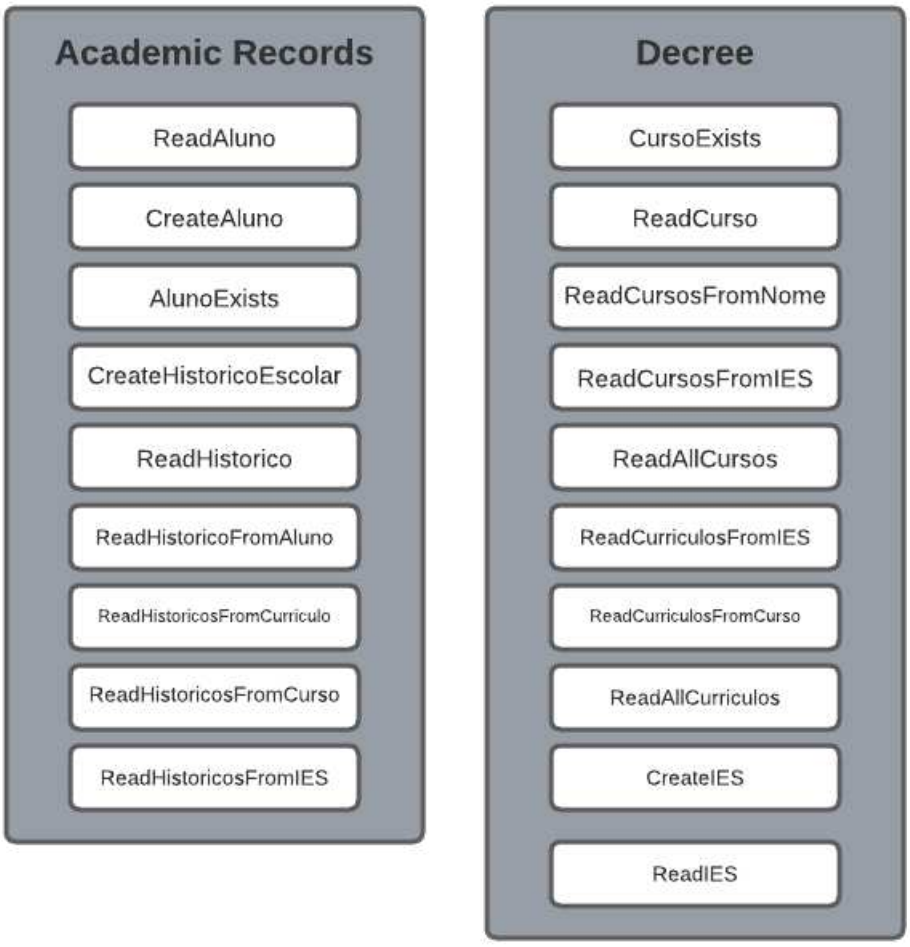


Figura 12 – Funcionalidades Chaincodes

6.3 ELABORAÇÃO DO APLICATIVO

6.3.1 Rede Blockchain

A estruturação da rede *blockchain* responsável por definir os processos internos da rede e armazenamento dos dados dos estudantes, históricos e currículos, foi desenvolvido a rede utilizando como base o *Hyperledger Fabric*. Assim como no Projeto Jornada, optou-se por utilizar a ferramenta minifabric para facilitar o desenvolvimento da rede. O minifabric possibilita dar suporte a uma rede Fabric em uma pequena máquina e expandi-la a tanto quanto seja necessária. O minifabric é responsável por construir a rede *blockchain* por uma série de roteiros diferentes. O grande benefício da utilização dessa ferramenta é a sua facilidade e o fato de utilizar Docker como base de toda sua execução, permitindo assim não exigir tanto da capacidade dos dispositivos. Para execução do aplicativo, fundamentou-se numa rede inicial contendo três chaincodes, sendo a *Academic Records* e *Decree* que foram baseadas no desenvolvimento já existente no Jornada, e a nova chaincode proposta *Student*. Além disso, cada uma das chaincodes possuem

respectivamente uma application responsável por fazer uma “ponte” entre a rede blockchain e a API. Ademais, foi utilizado a estrutura inicial das APIs *Academic Records* e *Decree*, além da adição de uma nova API nomeada como *Student*.

No novo modelo proposto optou-se por manter as tecnologias utilizados no projeto jornada. Ou seja, para as chaincodes utilizou-se Go e para as applications e APIs utilizou-se Node.js. Ambas as escolhas basearam-se devido o padrão adotado pelo próprio minifabric em seus exemplos de desenvolvimento [2]. Ademais, foi mantido a escolha de Docker como ferramenta para orquestração dos contêineres da rede, devido ao padrão já estabelecido pelo minifabric e a familiaridade com sua utilização.

6.3.1.1 Adaptações das Chaincodes

Devido ao fato desse trabalho trazer um novo foco para o estudante, foi necessário adaptar e alterar algumas funcionalidades já existentes na chaincode *Academic Records*, tendo em vista que ela está diretamente relacionada a um estudante, já que históricos escolares são responsáveis por representar a participação e jornada acadêmica de um estudante. Primeiramente, foi necessário reestruturar a lógica de identificação de um histórico escolar passando agora a utilizar um UUID (id único gerado aleatoriamente) de identificação para este, visando dar praticidade para obter e visualizar diferentes históricos. Ademais, existiam algumas dife-

[2] https://github.com/hyperledger-labs/minifabric

```
type AtividadeComplementar struct {
    Id                                string
    Codigo                            string
    DataInicio                        time.Time
    DataFim                           time.Time
    DataRegistro                      time.Time
    TipoAtividadeComplementar         string
    Descricao                         string
    CargaHorariaEmHoraRelogio         CargaHoraria
    DocentesResponsaveisPelaValidacao []Docente
    Certificado                       string
    Situacao                          SituacaoAtividade
}
type Estagio struct {
    Id                          string
    DataInicio                  time.Time
    DataFim                     time.Time
    Descricao                   string
    Concedente                  Concedente
    CargaHorariaEmHoraRelogio CargaHoraria
    DocentesOrientadores        []Docente
    Certificado                 string
    Situacao                    SituacaoEstagio
}
```

Figura 13 – Structs de Atividade e Estágio

```
fabric:
  cas:
  - "ca1.studentorg.acadblock.br"
  peers:
  - "peer1.studentorg.acadblock.br"
  - "peer2.studentorg.acadblock.br"
  orderers:
  - "orderer1.acadblock.br"
  settings:
    ca:
      FABRIC_LOGGING_SPEC: ERROR
    peer:
      FABRIC_LOGGING_SPEC: ERROR
    orderer:
      FABRIC_LOGGING_SPEC: ERROR
  netname: "mysite"
```

Figura 14 – Spec para configuração.

renças entre a forma que é o projeto jornada representa as estruturas complementares(estágios e atividades), e como é visualizado para este projeto. Portanto, para representar fielmente as estruturas foi necessário modificar e adicionar alguns campos, na Figura 13 é possível a visualizar nova composição da *struct* Atividade Complementares e Estágios.

Além disso, foi necessário a criação de novos métodos na chaincode *Academic Records* permitindo a interação do usuário com seus históricos escolares, essas funcionalidades adicionais serão explicadas nas próximas seções conforme a necessidade durante a explicação de cada uma das operações presentes no aplicativo. Contudo, apesar das mudanças citadas na *Academic Records*, foi possível manter o código original desenvolvido no Projeto Jornada para a *Decree*, isso se atribui ao modelo de negócio desenvolvido no projeto, onde esta chaincode está relacionado principalmente aos currículos escolares, ou seja, não apresentando nenhuma responsabilidade sobre o aluno.

6.3.1.2 Adaptações da API e Applications

As applications no desenvolvimento da rede possuem a funcionalidade de permitir que uma API que se comunica com o cliente possa invocar as funcionalidades da *blockchain*. Sendo assim, as applications e APIs precisam retratar todos os métodos que os participantes poderão ter acesso.

Dessa forma, foi necessário adaptar o código da API e *application* Academic Records para que esta apresente as novas funcionalidades desenvolvidas pela sua respectiva chaincode.

6.3.1.3 Funcionamento da Rede

As redes minifabric possuem seu funcionamento fundamentado na execução por contêineres Docker. Sendo assim, para possibilitar a execução da aplicação foi necessária uma sequência de etapas de preparação, sendo estas:

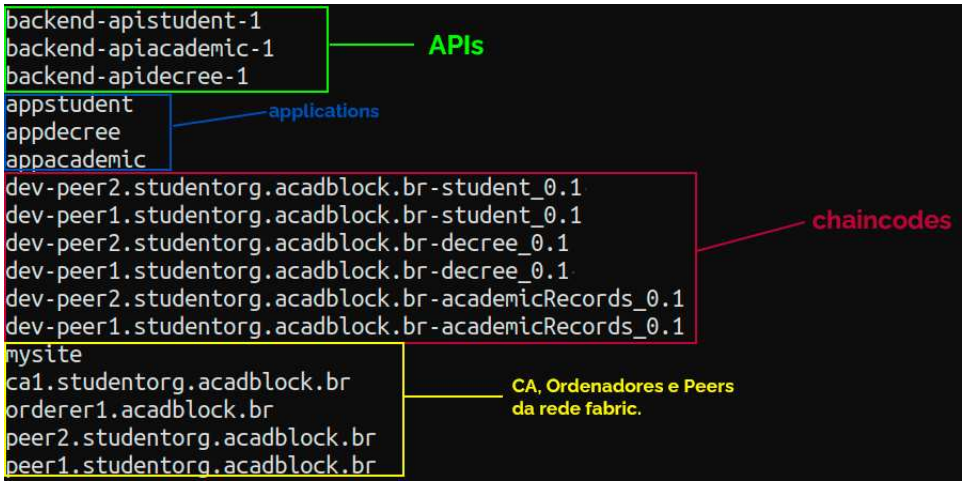


Figura 15 – Rede Docker.

1. Definir as configurações iniciais da rede através do arquivo de configuração *spec.yaml*, neste exemplo, definiu-se uma configuração inicial de 1 CA representando a organização *studetorg.acadblock.br*, 2 pares pertencentes a organização *studentorg.acadblock.br* responsáveis pela execução de operações na rede e 1 ordenador pertencente a organização *acadblock.br*. Na Figura 14, é demonstrado o arquivo de configuração que representa as configurações explicadas.

2. Subir a rede através do comando *minifab up*, sendo este responsável por gerar os certificados, criar o canal, ingressar os participantes e definir o bloco gênesis. Ao subir a rede, definiu-se que a chaincode *Academic Records* inicialmente instalada no canal.

3. Instalação das outras duas chaincodes necessárias, *Decree* e *Student**, através do comando *minifab ccup* responsável por instalar, aprovar e *commitar* uma nova definição de chaincode para o canal. Ressalta-se que em um ambiente maior com outras instituições e organizações não seria possível *commitar* automaticamente a chaincode, existindo a necessidade da aprovação da maioria das instituições participantes.

4. Execução das três applications desenvolvidas para comunicação com a API.

5. Execução da orquestração de contêineres através do Docker compose possibilitando que as APIs se comuniquem com sistemas externos, neste caso a aplicação Flutter.

Na Figura 15, é possível visualizar os contêineres necessário para execução de uma rede minifabric completa para uma organização.

6.3.2 Prototipação das Interfaces

Definiu-se como objetivo principal da construção do aplicativo a capacidade de atender estudantes que possuíssem o mínimo ou nenhum conhecimento sobre *blockchain*. Ou seja,

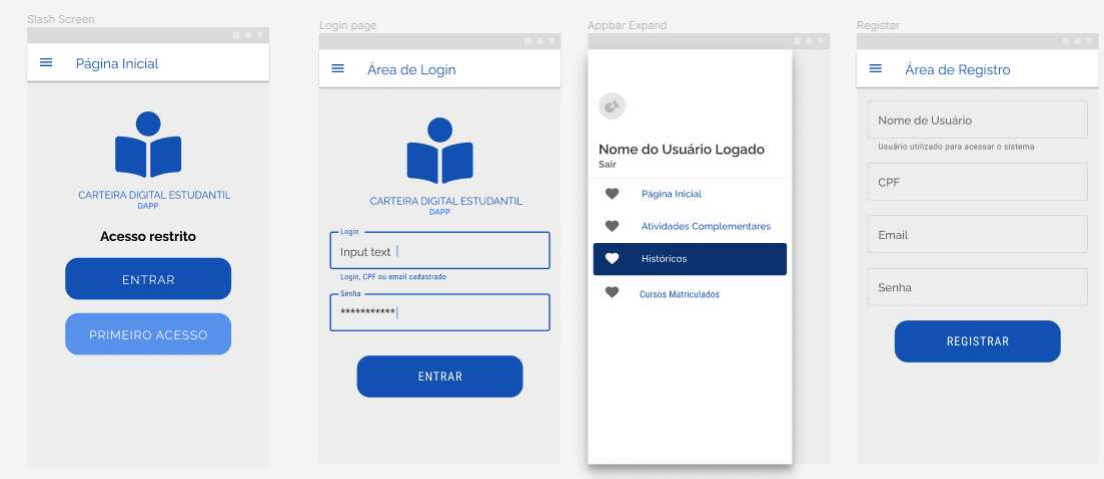


Figura 16 – Interface de Login/Registro

o estudante que optar por utilizar a ferramenta em desenvolvimento para acompanhar suas atividades acadêmicas, terá ao seu dispor uma interface rápida, prática e simples. Além disso, outro fator considerado na construção da aplicação foi atender a maioria dos dispositivos possíveis, sendo assim, busca-se desenvolver um aplicativo responsivo, adaptável e flexível. Ademais, a aplicação em desenvolvimento permitirá que dispositivos com distintas capacidades de software e hardware possam ter uma experiência gratificante.

Considerando os objetivos deste projeto e as demandas necessárias para interface gráfica, optou-se pela utilização de Figma [3] para prototipação das telas. Figma, é um editor de gráfico fácil e prático que permite prototipagem de projetos de design baseado principalmente no navegador web, oferecendo ferramentas offlines e aplicações adicionais para desktop tanto para Windows como para Linux. A rápida curva de aprendizagem necessária para adaptar-se a ferramenta e a compatibilidade desta com os materiais oferecidos pelo governo federal, foram as principais motivações para escolha como ferramenta de design.

Assim como o caso de uso utilizado como base, nesta proposta optou-se por utilizar o Padrão Digital do Governo [4]. O Padrão Digital, desenvolvido pelo governo, oferece materiais e modelos para web sites, aplicações moveis e guias de utilização e desenvolvimento. Além disso, o padrão define tipografias sofisticadas e paleta de cores atrativas para o usuário. Dessa forma, adiciona-se credibilidade e sofisticação para as interfaces desenvolvidas.

As telas desenvolvidas são dividas em três grandes grupos: (1) interface de login/registro (2) visualização (3) edição/adição. Como o objetivo é a simplicidade, optou-se por um design mais simples, que permite facilidade para o usuário alvo entender o que cada página oferece e que este consiga acesso aos recursos buscado com poucos cliques. Na Figura 22, demonstra-se a interface inicial para o usuário que ainda não efetuou login, ou seja, ainda não possui acesso os recursos oferecidos. Caso este, ainda não possua uma conta, ele poder se inscrever de maneira rápida e prática preenchendo corretamente os campos do formulário. Em

[3] www.figma.com/
[4] https://www.gov.br/ds/home

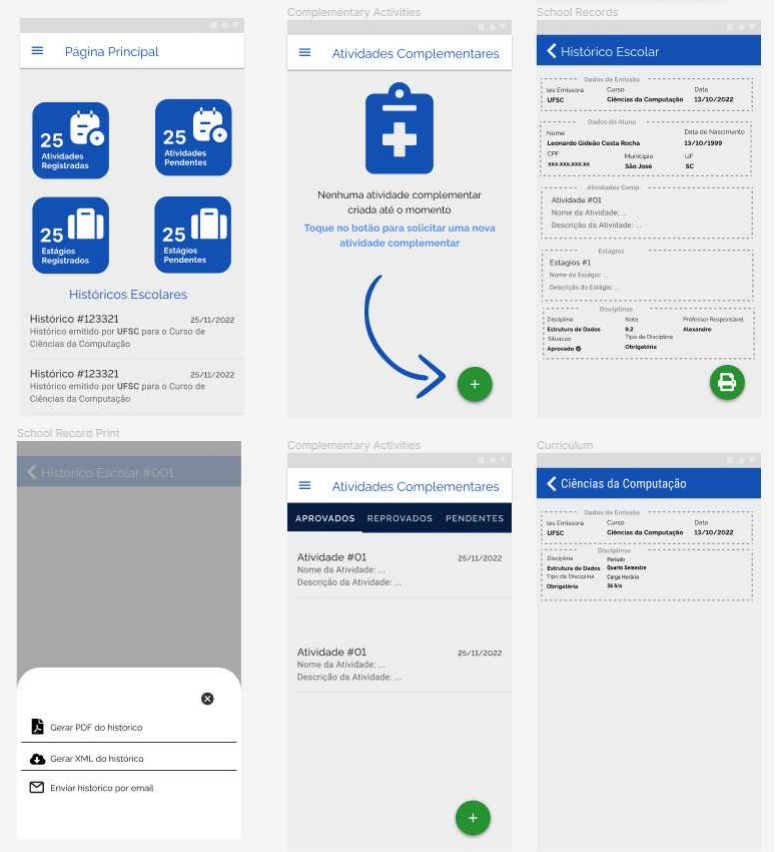


Figura 17 – Interface de Login/Registro

sequência, caso seu cadastro tenha sido feito corretamente, ele será transferido para página inicial que permite acessar todas as páginas disponíveis.

Na Figura 17, é demonstrado um resumo das interfaces de visualização para os diferentes dados existentes na rede. Na página inicial, o usuário poderá visualizar um resumo dos seus históricos escolares em uma lista, além disso, ele poderá acessar os estágios ou atividades complementares pendentes. Ao selecionar um histórico escolar, a interface exibirá uma tela demonstrando todos os dados úteis para o aluno, como informações a respeito de disciplinas, atividades complementares, estágios, curso, instituição e entre outras. Nesta mesma interface o usuário poderá optar por selecionar o botão posicionado no canto inferior direito, este irá gerar um modal [5] que disponibilizará funcionalidades como *Enviar o Histórico Escolar por e-mail*, *Gerar QR Code de Identificação do Histórico* e *Baixar o Histórico Escolar*. Além das interfaces disponibilizarem um sistema prático, elas são adaptadas em relação aos dados e informações que recebem. Na Figura 17 é possível ver a apresentação da mesma página de atividades complementares, com a diferença de uma ter dados para serem visualizados, já a outra não possui dados.

Na Figura 18, são demonstradas as telas referentes a atualizações em estados de dados ou inserção de novos dados na rede. Um estudante, poderá participar efetivamente da rede ao inserir dados que serão validados pelas instituições através destas interfaces, onde ele poderá:

[5] Caixa de diálogo estilizada que possibilita mostrar informações ou oferecer funcionalidades na interface

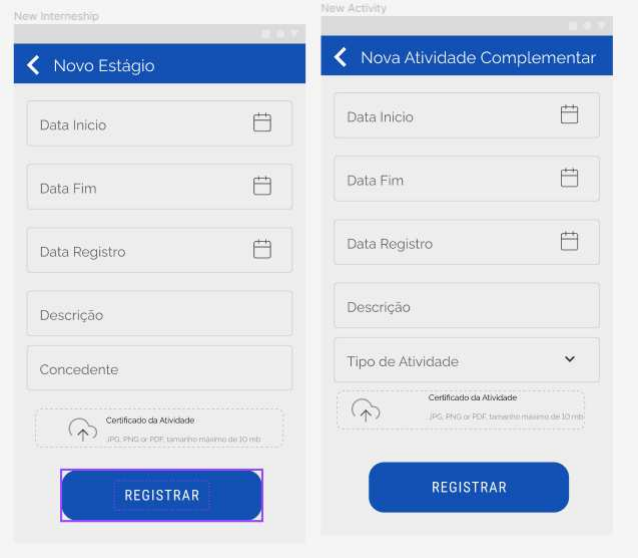


Figura 18 – Interface de Registro.

(1) adicionar uma nova atividade complementar; (2) adicionar um novo estágio.

6.3.3 Tecnologia para desenvolvimento das interfaces

Visando-se um desenvolvimento nativo, ou seja, voltado para uma plataforma específica, onde no caso dessa proposta é o Android, existiam algumas possibilidades como tecnologia/linguagem para desenvolvimento. Dentre as principais tecnologias para desenvolvimento Android pode-se citar: Ionic, React Native, Android Phonegap, Corona SDK e Flutter. Neste desenvolvimento, optou-se pela utilização do Flutter, que consiste em um kit de desenvolvimento de interface, de código aberto, criado pela Google e baseado na linguagem Dart (também desenvolvida pela Google). Flutter é um framework de compilação nativa, multiplataforma, que pode ser utilizado tanto para o desenvolvimento Web e mobile para sistemas operacionais como iOS e Android, ou seja, permitindo que através do conhecimento de uma única linguagem possa se efetuar interfaces para diferentes plataformas.

Ademais, além da utilização de Flutter para o desenvolvimento da interface, optou-se pela utilização do Android Studio como ambiente de desenvolvimento junto ao editor de código Visual Studio Code da Microsoft. As escolhas foram motivadas devido ao elevando número de usuários em ambas plataformas e pela experiência individual com ambas. Além disso, o ambiente de desenvolvimento Android Studio permite a utilização e testes em diferentes versões de sistemas operacionais e em diferentes dispositivos, oferecendo um sistema totalmente compatível as funcionalidades oferecidas a um telefone comum, destaca-se na Figura 19 o exemplo do desenvolvimento para construção da aplicação.

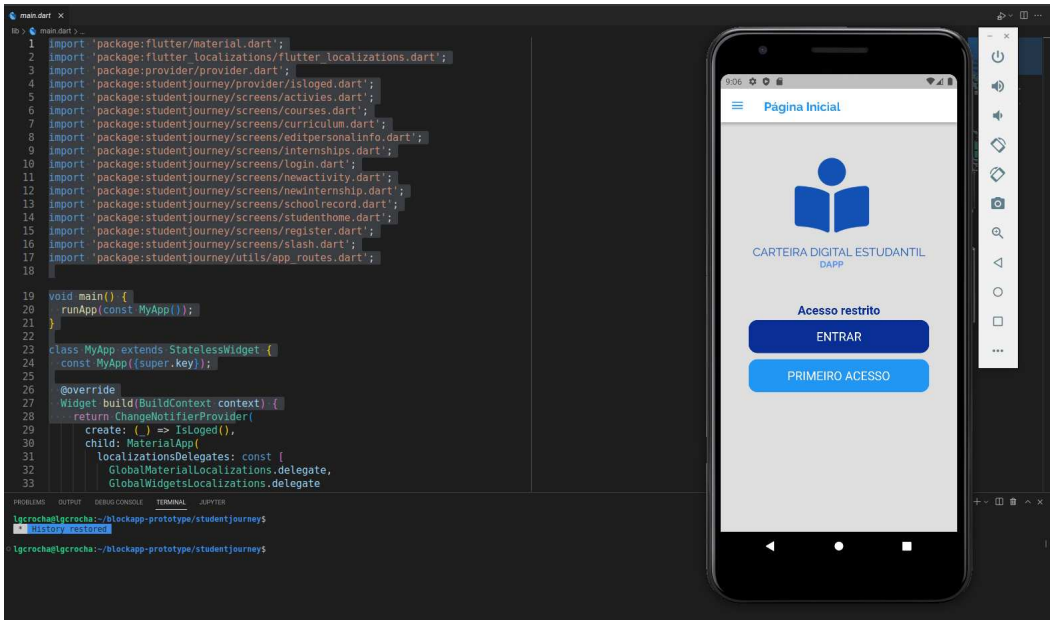


Figura 19 – Interface do Android Studio.

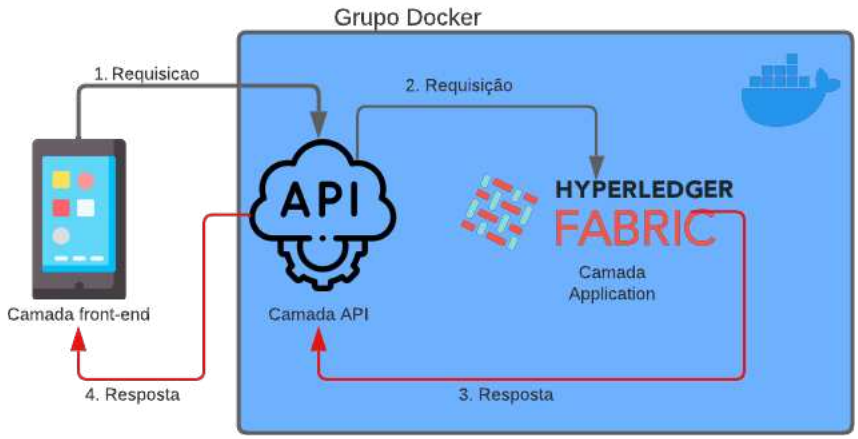


Figura 20 – Camadas do Desenvolvimento.

6.4 CAMADAS DO DESENVOLVIMENTO

A execução do aplicativo funciona através da troca de requisições e respostas entre diferentes camadas, estas que estão acessíveis através de endpoints de contêineres Docker. Dessa forma, na Subseção 6.4.1 até Subseção 6.4.3 será dado uma explicação detalhada do objetivo de cada uma das camadas. Contudo, resumidamente pode-se definir as camadas como: *application* sendo responsável por fazer uma ponte entre a rede *blockchain* e API possibilitando que transações Fabric sejam executadas. A camada de API sendo aquela que permite externalizar o sistema da rede *blockchain*, ou seja, permite que a aplicação se comunique com a interface móvel. Finalmente, o *front end* representa a interface apresentada para o usuário final e a sua forma de interagir. Na Figura 20, é possível observar um esquemático simples que demonstra essa lógica de comunicação entre camadas.

Para as camadas poderem se comunicar de maneira correta e consigam ter acesso efe-

tivo aos endpoints de cada uma delas, foi necessária mante-las em um mesmo grupo no Docker, unificando a comunicação em um único *host*.

6.4.1 Camada de Application

Applications permitem que através do Fabric SDK, seja possível execução de transações na rede, simplificando a execução e permitindo utilização de dados mais complexos como JSON, por exemplo. No Projeto Jornada duas *applications* já estavam previamente definidas, sendo elas, *appacademic* (Academic records) e *appdecree*. Para possibilitar a participação do estudante foi necessário adicionar mais uma *application* para comunicação a *appstudent*. Em seu funcionamento todas as *applications* agem como uma espécie de servidor aguardando por requisições, esse servidor é mantido por um contêiner no Docker onde o serviço é externalizado em uma porta específica da máquina *host*. Ao se comunicar com uma *application* a API solicitante deverá passar o parâmetro relativo à função que deseja chamar e os dados necessários para execução desta, ao término da execução uma resposta será devolvida ao cliente identificando um erro ou o resultado da operação. Dessa forma, a *application* é então a responsável por solicitar um serviço na rede *blockchain*, seja esse, de leitura, adição ou edição.

6.4.2 Camada de API

As APIs são responsáveis por executar as funcionalidades do aplicativo, ou seja, toda operação solicitada na interface gráfica precisará se comunicar com a camada de API respectiva. Além do usuário do aplicativo, instituições que desejam adicionar informações a rede, como por exemplo, um novo currículo precisarão se comunicar com a API para que esta trate os dados inseridos e os envie para as camadas inferiores. Neste projeto se faz necessário três diferentes APIs onde cada uma delas possui uma finalidade distinta, sendo estas:

- API Academic Records: possibilita que instituições controlem a participação de um estudante na rede e sua jornada no ensino superior através de históricos escolares, estes que poderão ser enviados por requisições como um arquivo *.XML, como pode ser observado na Figura 21. Em sua versão inicial esta API não tinha como objetivo permitir uma interação dinâmica através dos históricos, buscando somente o registro e verificação dos históricos, sendo impossível gerar novos dados a partir da rede. Dessa forma, para ser possível adicionar uma nova relação em que um estudante pudesse participar ativamente do processo de emissão de diploma, a API com a chaincode foi adaptada para uma instituição poder aprovar ou reprovar atividades complementares, isto se deu via adição de novos endpoints e comunicação com as camadas inferiores (chaincode Academic Records e application Academic Records).
- API Decree: possibilita que instituições definam as regras que precisarão ser seguidas por cada um dos históricos, ou seja, define de que maneira um histórico válido para um

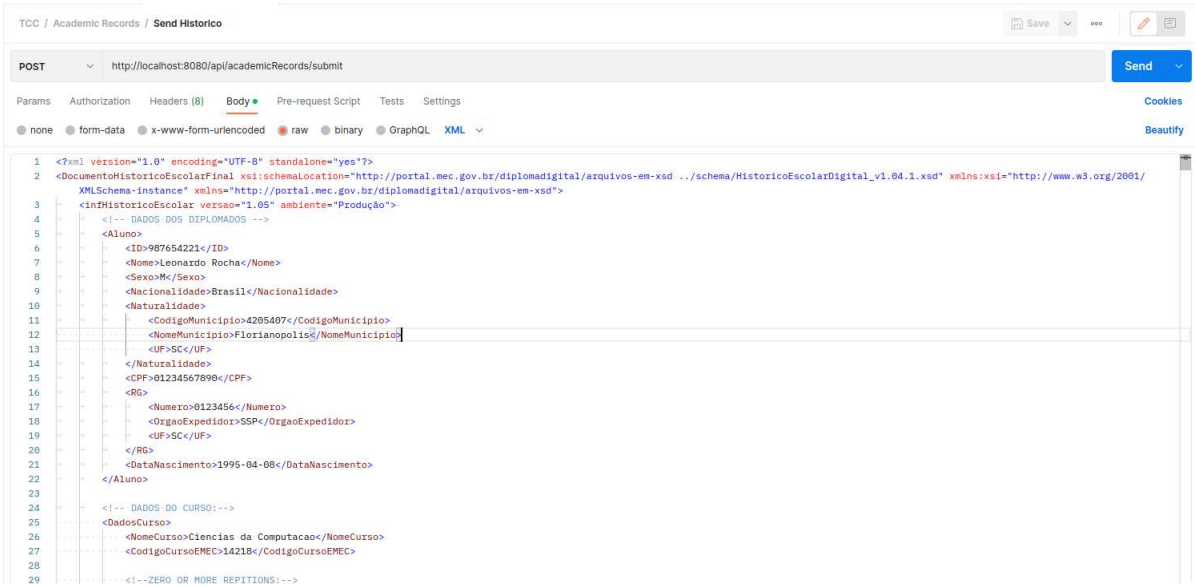


Figura 21 – Histórico XML.

estudante poderá ingressar na rede. Devido ao foco deste trabalho não estar nas regras de negócio do ensino superior, não foi necessário realizar nenhuma adaptação ao código original desta API. Dessa forma, seu objetivo na rede está somente em definir as regras que serão utilizadas pelas outras APIs. Ademais, assim como no fluxo das outras APIs, a *Decree* para realização de operações precisa enviar requisições para as camadas mais baixas (chaincode Decree e application Decree).

- API Student: esta API é a única que o aplicativo mobile terá acesso para execução de suas operações. Sendo assim, em sua lógica a API Student é responsável desde o controle de registro e autenticação do aplicativo até a comunicação com as outras APIs para ser possível retornar os dados para interface. Assim como as demais APIs, as operações da Student precisam ser comunicar com suas camadas inferiores para poder ter acesso à rede *blockchain*.

6.4.3 Camada de front end

A camada *front end* da aplicação possui toda sua lógica desenvolvida em Flutter, tendo como objetivo: controlar a autenticação do usuário, garantir que os dados sejam mostrados corretamente em cada uma das páginas e possibilitar a comunicação com os serviços das APIs.

Para adicionar praticidade e um controle de login foi utilizado a dependência *apiCacheManager*, que permite armazenar informações no próprio aplicativo, permitindo que usuário não tenha a necessidade de realizar login sempre que abrir o aplicativo. Ademais, o *front end* da aplicação se organiza através de navegação por rotas, onde cada uma das páginas está diretamente relacionada a uma rota e a um componente representando sua estrutura. Ou seja, ao navegar até uma página uma nova rota será adicionada a fila de rotas e esta irá carregar o componente principal, caso esse componente seja uma resposta necessária da API como, por

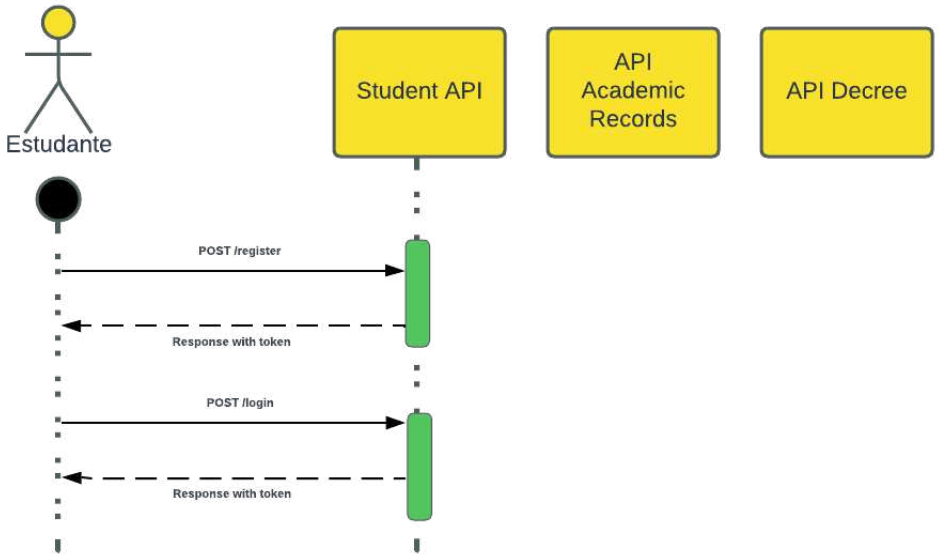


Figura 22 – Diagrama de sequência login.

exemplo, uma lista de históricos, será necessário adicionar uma espera para usuário através da utilização de *Future* (funções assíncronas) do Flutter.

6.5 EXECUÇÃO DO APLICATIVO

Nas próximas subseções são explicadas as principais funcionalidades presentes no aplicativo oferecido para o cliente, além de demonstrar o demonstrar funcionalidades secundárias presentes nas APIs de instituições que permitem atualizar os dados presentes na interface do estudante. Dessa forma, além da demonstração do fluxo de execução das APIs será explicado cada uma das etapas necessárias nas camadas inferiores para o resultado poder ser obtido corretamente.

6.5.1 Primeiro contato com o aplicativo

No seu primeiro contato com o aplicativo, o usuário precisará se registrar para acessar as funcionalidades do sistema. Da maneira que foi estruturado a chaincode um usuário poderá se registrar mesmo que não tenha históricos vinculados a sua conta. Como pode ser observado na Figura 22, ao registrar-se, caso o estudante tenha obtido uma conta válida ele receberá um token, este token é utilizado em todas as próximas operações que serão realizados na API Student, ele garante que somente um usuário autenticado poderá acessar as informações de um estudante. Esse controle por token é feito através do *JSON Web Token*, onde o token gerado apresenta os dados úteis do estudante que precisarão ser verificados, sendo estes: (1) nome de usuário, (2) CPF.

Ou seja, ao solicitar um registro via API o estudante irá solicitar o método *register*

do *AuthController* presente na API Student, passando como parâmetro os dados que foram preenchidos no formulário. A API Student por sua vez será responsável por se comunicar com a *application student*. Para que uma API se comunique corretamente com a application ela utilizará o endereço definido para o contêiner neste caso sendo *http://appstudent:8080* e além disso deverá ser definido os *app params*, parametrizando o nome da função chamada, o tipo e os parâmetros necessários para execução, como pode ser observado na Figura 23. Ao ser invocada a *application* irá utilizar o Fabric SDK para executar a transação que foi solicitada, sendo neste caso do tipo *submit* pois irá alterar a ledger da chaincode *Student*.

Ademais, caso o estudante já tenha um registro válido no sistema, ele poderá realizar o login na página inicial. A execução do login funciona de maneira análoga ao registrar tendo como diferença somente os métodos chamados. Sendo assim, a API Student será responsável por comunicar com o endpoint de login da *application* que irá executar uma transação do tipo *evaluate* na rede, caso os parâmetros passados estejam corretos será retornado as informações principais do estudante. Finalmente, após receber os dados da *application* a API irá armazená-los na cache com o token gerado, para não ser necessário realizar login em todas as entradas no aplicativo.

Ao realizar o login, a camada *front end* do aplicativo precisará comunicar-se com a API Student para obter os dados que serão apresentados na tela inicial. As funcionalidades responsáveis por carregar informações ou enviar dados para as APIs são tratadas como *services* no Flutter. Sendo assim, o *service fetchStudentData()* é responsável por comunicar-se com a API Student passando o token que será utilizado para verificar a autenticidade do estudante, por estarmos lidando com uma função assíncrona tendo em vista que não se sabe o tempo que a API levará para dar a resposta foi necessário utilizar a estratégia com *FutureBuilder* [6] na interface para carregar o conteúdo somente no momento que estiver pronto, enquanto este ainda está sendo buscado o aplicativo irá apresentar um carregamento. Na Figura 24, é possível observar a interface final para um usuário que já possui algumas informações registradas no sistema.

[6] https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html

```
const appParams = {
    op: 'write',
    type: 'registerStudent',
    nomeDeUsuario,
    CPF,
    email,
    senha: encryptedPassword
}
const { data } = await axios.post(`${apprun}/write`, {
    data: appParams
})
```

Figura 23 – Definição do App Params.

Figura 24 – Página inicial do estudante

6.5.2 Gerenciamento de Estágios e Atividades Complementares.

Como principal forma de interagir com a rede *blockchain* o estudante poderá enviar estágios e atividades complementares. Após um estudante ter realizado o envio, as instituições serão responsáveis por aprovarem ou reprovarem a atividade e/ou estágio. Na Figura 25, apresenta-se o modelo que representa o fluxo de execução para o processo de envio de uma atividade, o cadastro de um estágio é análogo, por isso a atividade será utilizada como exemplo para ambos. Inicialmente a API Student é responsável por formatar os dados que serão recebidos da interface do aplicativo. Vale ressaltar que as funcionalidades de adicionar novas informações para um estudante demandam o envio do token de autenticação. Após o token ter sido validado a API Student irá se comunicar com a *application* para a transação ser executada na rede *blockchain*, isto é feito através do *service addAtividade(ActivityRequest JSON)*. Nesse momento, o sistema passará a um estado de carregamento enquanto aguarda a execução da função. Ao término do processo a *application* retornará o resultado da operação, caso este seja

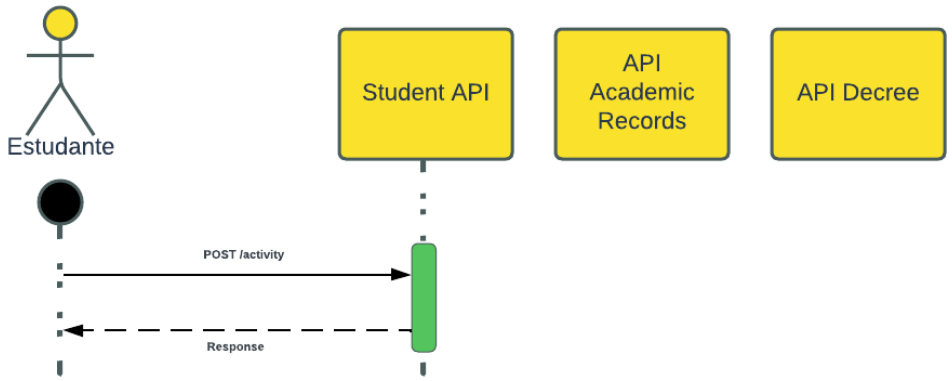


Figura 25 – Diagrama de Cadastro de Atividades.

válida a nova atividade já será adicionado a interface.

Atividades complementares e estágios que ainda não foram aprovados, ou seja, acabaram de ser adicionados por um estudante, não fazem parte de nenhum histórico escolar. Sendo assim, atividades e estágios recém-adicionados são colocados no ledger da chaincode *Student* utilizando como chave de identificação o UUID em sua estrutura. Portanto, para que uma atividade ou estágio faça parte de um histórico escolar, ou seja, esteja ativo no sistema, ele precisará da aprovação de uma instituição.

Além da possibilidade de enviar atividades e/ou estágios, o estudante poderá visualizar estas. Na Figura 26 é demonstrado o fluxo simples necessário para execução deste método na API Student. Contudo, algumas camadas precisam se comunicar ao fundo para que esse processo possa funcionar corretamente, ou seja, API Student precisa se comunicar com sua respectiva *application*. Sendo assim, após verificar se o token é válido, a API irá se comunicar com a *application* através do *service fetchAtividades()* definindo o parâmetro como sendo o CPF do estudante. Finalmente, a *application* irá executar uma transação na rede do tipo *evaluate* onde caberá a chaincode retornar a lista completa de atividades. Para que a lista de atividades seja retornada é necessário a comunicação de duas diferentes chaincodes, a *Student* responsável por armazenar as atividades que ainda não fazem parte de um histórico e a *Academic Records* que contém a lista de históricos de um estudante, ou seja, contém as atividades aprovadas. Portanto, ao invocar o método que retorna a lista de atividades na chaincode ele irá criar um *array* composto definido pelo seu ledger e em sequência irá executar o método *InvokeChaincode* do Fabric SDK que permite a comunicação entre chaincodes diferentes, podendo assim acessar o outro ledger.

Na Figura 27, é demonstrado a página que permite visualizar e a página que permite adicionar atividades complementares. A interface dos estágios é similar, mudando somente os parâmetros que poderão ser preenchidos pelo usuário.

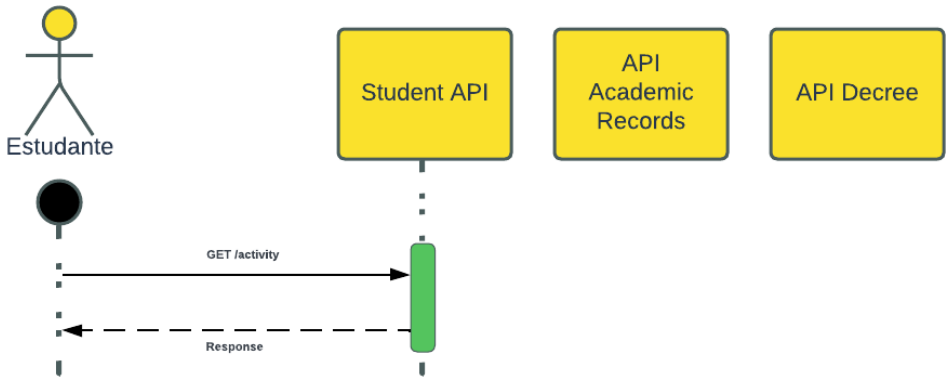


Figura 26 – Visualizar Atividades.

6.5.3 Visualizar cursos matriculados

A interface do aplicativo oferece uma página que possibilita o estudante visualizar os cursos atualmente matriculado ou os cursos que fez parte em algum momento. Para isso a camada de *front end* utiliza como base o *service fetchStudentData()* que utiliza como parâmetro as informações armazenada na cache e se comunica com API passando o token e o CPF do estudante. Após ter recebido uma requisição, a API irá verificar o token recebido e efetuar uma requisição GET para função respectiva na *application*. Como os cursos matriculados são obtidos a partir da lista de históricos escolares de um estudante, se faz necessário filtrar aqueles

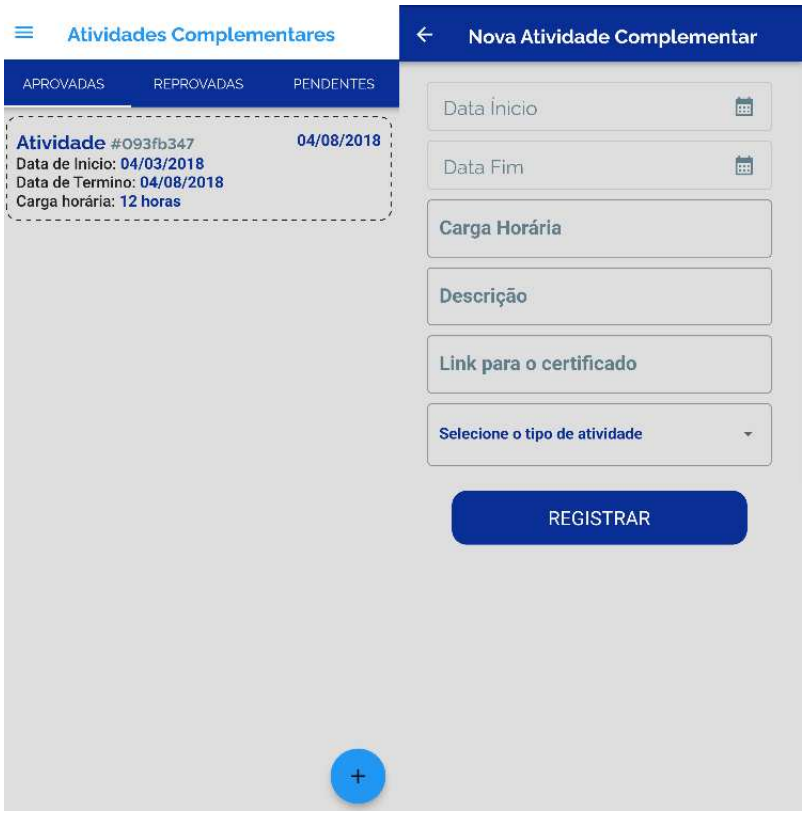


Figura 27 – Página de atividades complementares.

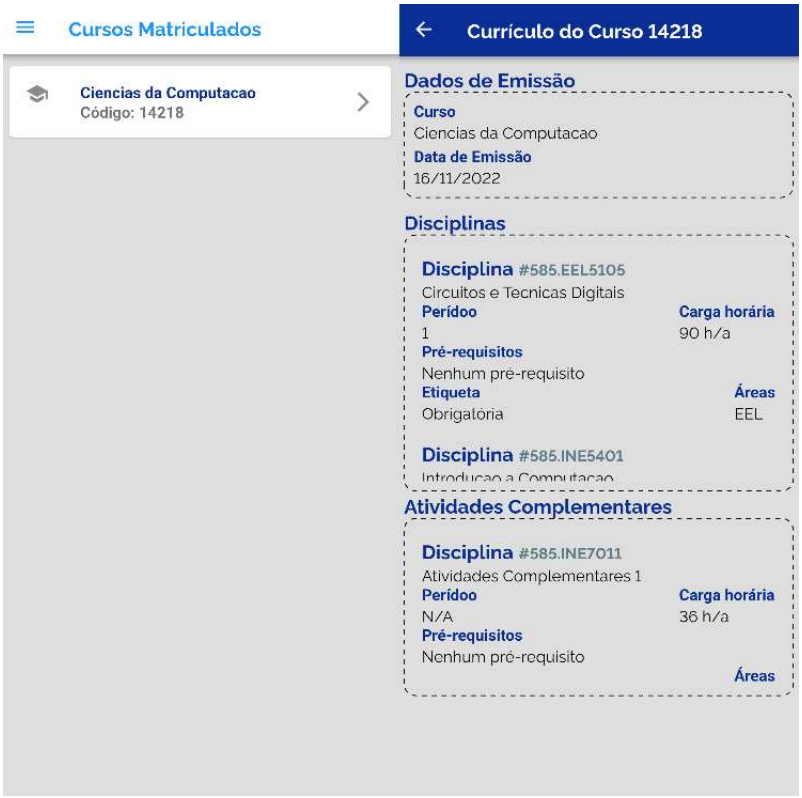


Figura 28 – Cursos matriculados.

que são repetidos, já que vários históricos escolares (representam períodos distintos) de um estudante podem ser enviados para o mesmo curso.

Ademais, a partir da lista de cursos oferecidos, a interface permite que um estudante clique em cada uma das opções de cursos e verifique o currículo escolar regente deste. Dessa forma, o estudante poderá acompanhar resumidamente as principais informações que definem a lógica de participação acadêmica. Na Figura 28, é possível observar a interface que mostra os cursos e a página que apresenta os detalhes de um currículo específico.

6.5.4 Visualizar históricos escolares

Na página inicial do estudante logado é apresentado a lista de seus históricos escolares, esta apresentação é feita de maneira resumida contendo informações a respeito do emissor e curso representado. Além disso, na interface, históricos emitidos por instituições a partir de XML são diferenciados de históricos gerados por um processo na rede, como por exemplo, uma atividade aprovada adicionada ao histórico, dessa forma, gerando um novo. Esta diferenciação é feita por um texto diferente e uma cor acinzentada para aqueles históricos gerados a partir da rede *blockchain*.

Caso deseje, um estudante poderá navegar pelas informações completas de um histórico escolar, na Figura 29 é possível observar as duas interfaces das páginas citadas. Para ser possível visualizar as informações totais de um histórico escolar, será necessário primeiramente realizar uma busca na API, já que não faz sentido ter o armazenamento de todos os dados na

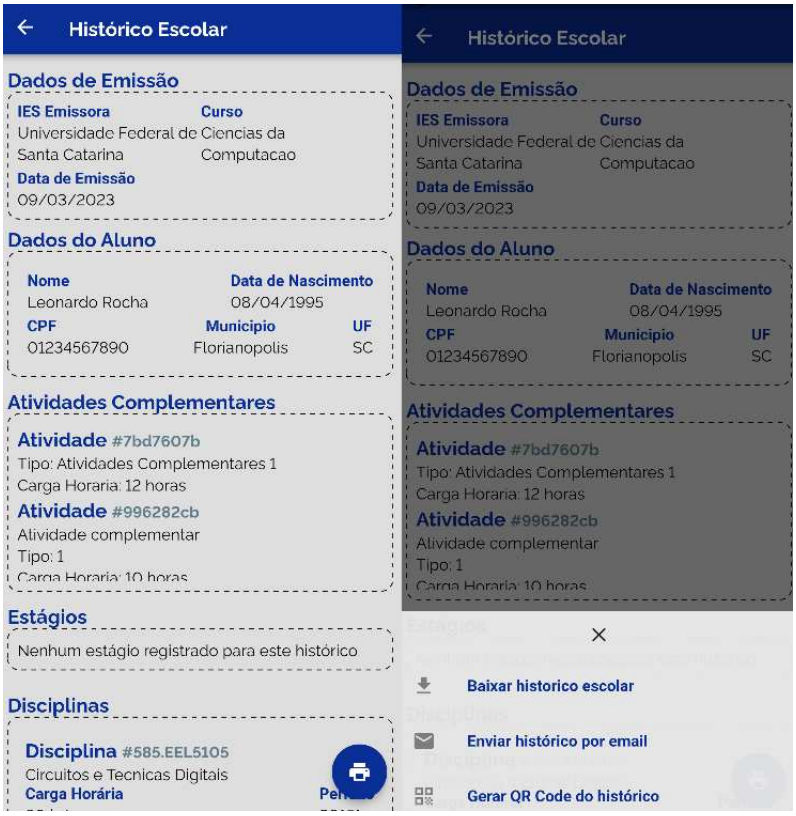


Figura 29 – Visualização de Históricos.

busca inicial. Na Figura 29, é demonstrada a página que apresenta as informações completas do estudante. Dessa forma, a camada *front end* solicita a comunicação com API *Student* através do endpoint *fetchSchoolRecord()* passando como parâmetro o código único do histórico, o CPF do estudante e o token de autenticação. Caso o token seja válido e o histórico encontrado, a informação será apresentada na interface, do contrário a página padrão de erros será chamada. Além de oferecer a visualização dos dados principais da formação de um estudante, a página de histórico escolar apresenta funcionalidades extras para o estudante interagir com a rede, estas que serão explicadas na Subseção 6.5.5.

6.5.5 Funcionalidades extras

Buscando oferecer uma maior dinâmica para a participação do estudante no aplicativo, através da interface é possível acessar funcionalidades extras oferecidas, sendo estas: enviar histórico por e-mail, baixar histórico escolar, gerar QR-code do histórico. Para realizar o envio de um histórico por e-mail utilizou-se a API EmailJS [7] em sua versão grátis, oferecendo 200 e-mails diários. Além disso, o envio utiliza como destinatário o e-mail utilizado no registro do estudante, e o conteúdo da mensagem é composto pela versão em JSON do histórico escolar. A função de baixar um histórico escolar, primeiramente solicita ao usuário uma permissão para acesso aos arquivos, assim que for atribuída a permissão o histórico será armazenado nos arquivos locais do aplicativo. O processo de gerar QR-code possibilitará integração com fun-

[7] https://www.emailjs.com/

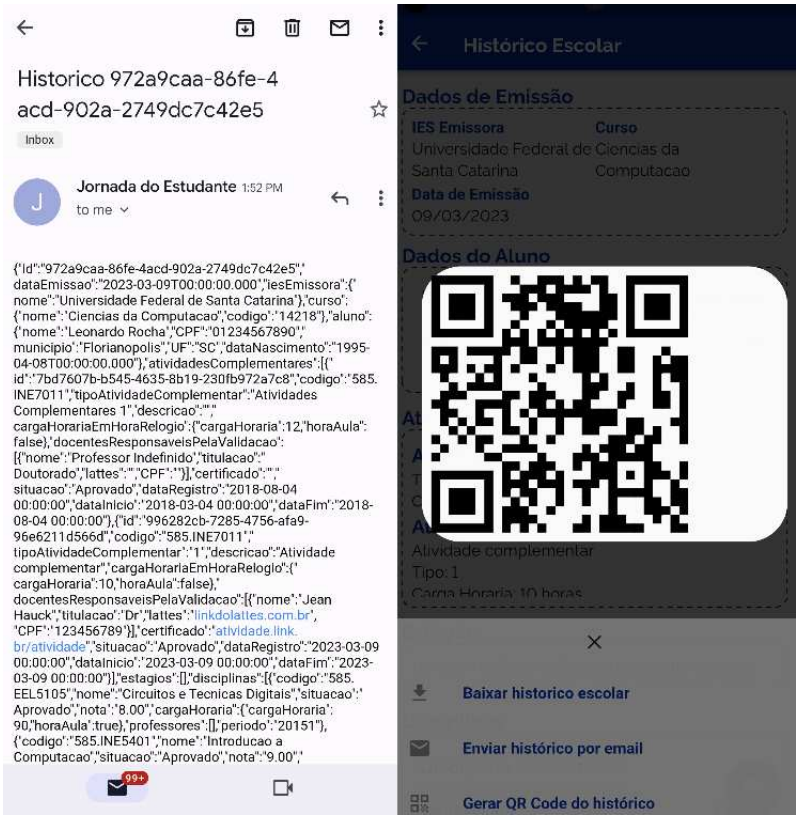


Figura 30 – Funcionalidades extras.

cionalidades futuras, onde instituições poderão verificar o QR-code atestando que o estudante realmente cursou sua graduação em uma instituição. Para a construção do QR-code apresentado para o cliente foi utilizado a biblioteca *qr-flutter* [8] oferecida pela linguagem. Na Figura 30, é demonstrado a utilização das funcionalidades de gerar QR-code e o exemplo de um recebimento de e-mail.

6.6 ADAPTAÇÕES NO PROJETO JORNADA

Para que a participação de um estudante fosse dinâmica e efetiva na rede foi necessário realizar adaptações nas funcionalidades já existentes no projeto jornada. As adaptações foram aplicadas unicamente na lógica da Academic Records, já que ela é responsável pela representação dos históricos escolares que definem a participação do estudante. Sendo assim, foi necessário adicionar funcionalidades como: visualizar a lista de atividades complementares pendentes, visualizar a lista de estágios pendentes, aprovar uma atividade/estágio e reprovar uma atividade/estágio.

O processo de visualizar as atividades e estágios pendentes necessita que a chaincode *Academic records* se comunique com a chaincode *Student* já que as informações estão salvas em seu livro-razão. Além da visualização, o processo de aprovar/reprovar atividades e estágios demanda da comunicação com a chaincode *Student*, o fluxo ocorre da seguinte maneira:

[8] https://pub.dev/packages/qrflutter

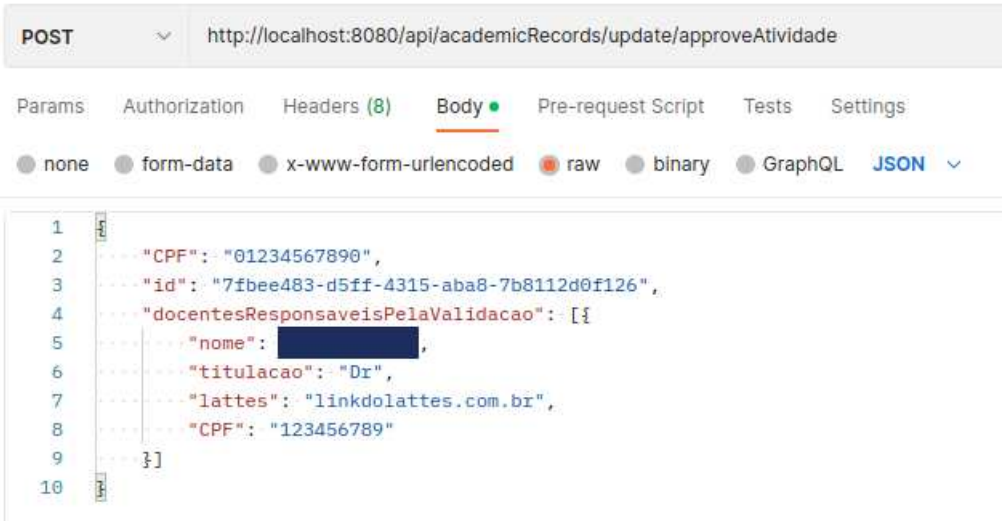


Figura 31 – Aprovar atividade.

1. A função de *approveAtividade* ou *approveEstagio* recebe como parâmetro o código único que identifica a estrutura, o CPF do aluno e a lista de professores responsáveis pelo processo. Na execução dessas funções na chaincode *Academic Records*, é necessário o processo de comunicar-se com o livro-razão da *Student*.

2. Após receber a lista de atividades/estágios pendentes na chaincode *Academic Records* é buscado o objeto correspondente através do atributo de identificação único. Como a chaincode *Academic Records* não tem acesso à lista de atividades/estágios pendentes diretamente, é necessário comunicar a chaincode *Student* que a atividade/estagio deve ser removido da lista de pendências, já que será marcado como aprovado. Ressalta-se que o Fabric trabalha com um conceito de *rollback*, sendo assim, caso alguma das operações sequências falhem, o processo que foi executado será desconsiderado.

3. Ao receber a confirmação que a atividade/estágio foi removido da lista de pendências, o objeto é retornado e adicionado ao último histórico escolar do estudante. Após adicionado, a chaincode *Academic Records* invoca a função responsável por criar um histórico escolar com a nova informação, para isso é necessário gerar um novo código de identificação para o histórico escolar e modificar o horário de emissão. Ademais, o histórico criado é sinalizado com uma flag indicando que o histórico foi gerado partindo de uma atualização.

Na Figura 31, é demonstrado o exemplo de requisição para solicitar a aprovação de uma atividade complementar executado pela API *Academic Records*.

7 EXPERIMENTOS

Nesta seção, será explicado os experimentos realizados para execução do protótipo definido neste projeto. Por uma rede *blockchain* com diversos dispositivos conectados, será demonstrado o funcionamento do gerenciamento descentralizado do Ensino Superior, além de comprovar a participação efetiva do estudante no processo. Para ser possível definir um critério de qualidade para o aplicativo desenvolvido, decidiu-se utilizar o sistema de escala padronizada *System Usability Scale* (SUS). O SUS foi o sistema de usabilidade proposto em 1986 por John Broke. Atualmente este tem sido amplamente utilizado em pesquisas de experiências de usuário (USABILITY, 2023).

O SUS constitui uma escala que mede o nível de agrado do usuário na utilização da aplicação por meio de dez perguntas, cinco delas positivas e cinco delas negativas, sendo estas: (1) Acho que gostaria de usar esse sistema com frequência; (2) Achei o sistema desnecessariamente complexo; (3) Achei o sistema fácil de usar; (4) Acho que precisaria do apoio de um técnico para poder utilizar este sistema; (5) Achei que as várias funções neste sistema estavam bem integradas; (6) Eu pensei que havia muita inconsistência neste sistema; (7)Imagino que a maioria das pessoas aprenderia usar esse sistema muito rapidamente; (8) Achei o sistema muito complicado/desajeitado de usar; (9) Eu me senti muito confiante utilizando o sistema (10) Eu precisava aprender muitas coisas antes de poder usar esse sistema. O usuário irá avaliar cada um dos itens com o nível de "Discordo totalmente" até "Concordo plenamente", após isso será calculada a nota final de usabilidade dada pelo usuário, obtida pelo cálculo das regras do SUS (CENTER, 2021).

Os resultados obtidos serão expressos em uma pontuação de 0 a 100, onde quanto maior a nota obtida melhor o resultado. Não existe uma definição teórica da nota que representa um bom resultado de usabilidade para o sistema, neste trabalho utilizaremos como base o definido por (KARLSSON; BERGVALL-KåREBORN; LIND, 2013), onde uma pontuação de 85 ou maior representa uma usabilidade excepcional e uma pontuação abaixo de 70 representa uma usabilidade inaceitável.

Segundo o MEC (MEC, 2021), a média etária dos estudantes que fazem parte das universidades é de vinte e um anos, onde em geral ingressam na instituição com a média de dezoito anos e saem dela com vinte e três anos. Dessa forma, a faixa etária dessa pesquisa será entre 18 a 24 anos, sendo estes participantes graduandos ou graduados em instituições de ensino superior.

John Brooke, o criador do SUS não define um número de participantes exato que caracteriza uma pesquisa eficiente em seu paper original SUS - A quick and dirty usability scale (BROOKE, 1995). Sendo assim, para este trabalho utiliza-se como base o critério definido na pesquisa *How Many Participants are Really Enough for Usability Studies?* (ALROOBAEA; MAYHEW, 2014), onde o autor ao realizar pesquisas com diversos usuários construiu uma tabela que determina a porcentagem de problemas que é possível encontrar na questão de usabilidade em relação ao número de participantes. Em uma amostragem de quinze participantes

em uma pesquisa já é possível descobrir 97.050% dos problemas de usabilidade do aplicativo. Dessa forma, tendo vista o objetivo deste trabalho busca-se alcançar entre dez a vinte usuários para participação do teste.

7.1 AMBIENTE DOS EXPERIMENTOS

Os testes utilizam-se como base a infraestrutura tanto física quanto virtual do laboratório LabSEC. Primeiramente, definiu-se um protótipo simples de rede Blockchain composta por uma organização participante e um ordenador, esta estrutura junto das APIs foram alocadas em uma máquina virtual criada a partir do servidor do laboratório. A máquina utilizada para os experimentos foi disponibilizada ao grupo BlockSEC responsável pelo desenvolvimento no Projeto Jornada. Além disso, para ser possível externalizar os serviços existentes na infra-estrutura foi necessário a liberação de portas. Este processo foi concebido pelo professor responsável e coordenador do laboratório Dr. Jean Everson Martina.

Ademais, tendo em vista que o aplicativo não foi oficialmente lançado em uma plataforma oficial como a *PlayStore*, é necessário a utilização de um *.apk* para sua utilização. Contudo, a instalação por meio de processo externo demanda permissões diferentes em cada tipo de dispositivo, sendo assim, disponibilizou-se um dispositivo Samsung Galaxy S20 FE Android na versão 13 para os usuários que preferirem. Contudo, participantes dispostos a baixar o .apk e conceder as permissões necessárias puderam utilizar ele em seu próprio dispositivo. Junto a isto, o acesso à rede é realizado por *ssh* no notebook pessoal do autor do trabalho.

Finalmente, definiu-se que para facilitar a participação dos estudantes que desejam fazer parte da pesquisa foi disponibilizado o ambiente do laboratório para realização da pesquisa. Sendo assim, caso necessário as pesquisas poderão ser realizadas na sala onde o projeto BlockSEC é mantido, localizada no prédio INE (Departamento de Informática e Estatística) na UFSC.

7.2 PROTOCOLO DE TESTES

Ademais, após realizado os experimentos o usuário deverá responder as dez questões do teste de SUS visando atribuir um nível de qualidade atingido pelo aplicativo. Este questionário é realizado por uma planilha criada no Google Sheets, onde o estudante participante poderá marcar um X no nível de concordância com cada uma das perguntas disponíveis. Na Figura 32 observa-se o modelo de pesquisa realizado com cada um dos participantes, onde é apresentado uma lista de perguntas e cabe a este relacionar ela com aquilo que melhor correspondeu. Ao término dessa pesquisa é utilizado o critério de SUS para calcular uma nota final, na Figura 33, demonstra-se uma pesquisa realizada com um dos estudantes e o resultado obtido conforme as opções selecionadas por este.

	A	B	C	D	E	F	G	H	I	J
1	Marque um X na opção que melhor descrever sua opinião de acordo com cada pergunta.					Discordo totalmente	Discordo	Neutro	Concordo	Concordo totalmente
2	Eu acho que gostaria de usar esse sistema com frequência.									
3	Eu achei o sistema desnecessariamente complexo.									
4	Eu achei o sistema fácil de usar.									
5	Eu acho que precisaria do apoio de um especialista para usar esse sistema.									
6	Eu achei as várias funções do sistema bem integradas.									
7	Eu achei que havia muita inconsistência no sistema.									
8	Eu acho que a maioria das pessoas aprenderia a usar esse sistema rapidamente.									
9	Eu achei o sistema muito trabalhoso de usar.									
10	Eu me senti confiante usando o sistema.									
11	Eu precisei aprender muitas coisas novas antes de conseguir usar o sistema.									

Figura 32 – Planilha para participação na pesquisa.

	A	B	C	D	E	F	G	H	I	J
1	Marque um X na opção que melhor descrever sua opinião de acordo com cada pergunta.					Discordo totalmente	Discordo	Neutro	Concordo	Concordo totalmente
2	Eu acho que gostaria de usar esse sistema com frequência.								x	
3	Eu achei o sistema desnecessariamente complexo.					x				
4	Eu achei o sistema fácil de usar.								x	
5	Eu acho que precisaria do apoio de um especialista para usar esse sistema.						x			
6	Eu achei as várias funções do sistema bem integradas.									x
7	Eu achei que havia muita inconsistência no sistema.					x				
8	Eu acho que a maioria das pessoas aprenderia a usar esse sistema rapidamente.								x	
9	Eu achei o sistema muito trabalhoso de usar.						x			
10	Eu me senti confiante usando o sistema.								x	
11	Eu precisei aprender muitas coisas novas antes de conseguir usar o sistema.					x				
12	Pontuação	87.5								

Figura 33 – Planilha para participação na pesquisa.

7.3 QUESTÕES ÉTICAS A RESPEITO DOS EXPERIMENTOS

Ressalta-se que a pesquisa realizada respeita os padrões éticos, e utiliza-se como base a pesquisa feita em (BELLOMY, 2018) onde se seguem os seguintes conceitos:

- Os participantes estarão totalmente conscientizados sobre o intuito do teste e o objetivo final da pesquisa realizada.

- As informações pessoais do usuário não serão compartilhadas, vendidas e/ou expostas. Ou seja, toda pesquisa terá como principal ponto o anonimato, onde a única informação final mostrada será o resultado obtido pela média de respostas dos participantes sem vincular este a qualquer um deles. Ademais, destaca-se que informações pessoais como CPF, e-mail, senha, atividades cadastradas entre outros, não precisam ter nenhuma relação com dados verdadeiros, portanto, caso o usuário não se sinta confortável ele pode preencher informações no aplicativo com dados fictícios. Dessa forma, necessita-se que somente as suas respostas no questionário final sejam verdadeiras.

- Qualquer resposta preenchida pelo participante, não será questionada e/ou criticada. Sendo assim, o estudante tem total liberdade para dar sua opinião em relação a cada um dos fatores.

- As informações preenchidas na rede serão totalmente apagadas após o término dos experimentos deste trabalho.

7.4 RESULTADO DOS EXPERIMENTOS

7.4.1 Experimentos de desempenho

Para esta seção busca-se produzir um teste de estresse na blockchain visando descobrir a capacidade de execução de transações desta. O Hyperledger Fabric, oferece em sua estrutura uma plataforma de benchmark para as *blockchains* existentes em sua estrutura, o Hyperledger Caliper [1]. O Hyperledger Caliper, permite que desenvolvedores executem testes de desempenho através da execução de cargas de trabalho realizada pelos *workers* (entidades responsáveis pela execução de operações na plataforma). Seu funcionamento acontece como uma estrutura cliente-servidor comum, onde o servidor aguarda por solicitações de operações simuladas por um aplicativo invocado através dos clientes (workers), simulando assim a atividade na camada mais baixa (rede blockchain). As transações paralelas executadas irão fazer com que os ordenadores precisem ordenar constantemente transações e os pares precisem participar frequentemente do processo de endosso. Assim como no Minifabric, o Caliper oferece uma estrutura modular que permite adaptar-se as necessidades do usuário que pode controlar os recursos de acordo com seu desejo, como por exemplo número de *workers*, tempo de execução e número de rounds.

Para a simulação dos testes realizados nesta etapa, foi definida uma rede geograficamente distribuída com duas instituições participantes: universidade1.acadblock.br (gerenciada na infraestrutura do LabSEC em um servidor) e universidade0.acadblock.br (mantida na máquina pessoal do autor). Além disso, no cenário definido, existe um ordenador responsável pela organização de todas as transações, denominado orderer1.acadblock, que também está configurado na infraestrutura do laboratório. É importante ressaltar que os testes foram executados na máquina do autor, que possui um processador AMD Ryzen 7 4800H com gráficos Radeon, com 16 núcleos de processamento (4300 MHz) e 32 GB de RAM. Dessa forma, a máquina do laboratório ficou encarregada apenas de realizar sua participação no processo de endosso e na comunicação entre os pares.

Ao término da execução dos testes são obtidas algumas métricas que definem os resultados obtidos e a capacidade da rede, descreve-se estas como sendo: *Succ* número de operações executadas com sucesso, *Fail* número de operações que obtiveram falham na execução, *Send Rate(s)* número de operações enviadas por segundo para rede, *Max Latency* tempo máximo de espera até o envio de uma nova transação em segundos, *Min. Latency(s)* tempo mínimo de espera até o envio de uma nova transação em segundos, *Avg. Latency(s)* tempo médio de espera para o envio de novas transações em segundos, *Throughput (TPS)* número de operações enviadas por segundo.

No teste executado, foi verificada a operação de registro de estudante, que demanda comunicação entre as chaincodes *Student* e *Academic Records*. Essa operação envolve tanto a

[1] https://www.hyperledger.org/use/caliper

leitura (para verificar se os dados já existem) quanto a escrita no ledger (para registrar o novo estudante). Além disso, é necessário verificar se o estudante possui algum histórico válido definido pela *Academic Records*. Nesse caso, é necessário que os pares aprovem as operações realizadas e o ordenador as organize corretamente para adição no ledger.

No código apresentado no Listing 7.1, demonstra-se a execução desse experimento, onde dados são gerados aleatoriamente (para evitar conflito de registro) e registrados na rede como um estudante. Na Figura 34, destacam-se os resultados obtidos ao término da execução dos testes: 46.798 transações executadas com sucesso, 0 operações com falhas, uma taxa média de 785 transações enviadas por segundo, um tempo de espera máximo de 0,05 segundos para o envio de uma transação, um tempo de espera mínimo de 0 segundos para o envio de uma transação e um tempo médio de espera de 0,01 segundos para o envio de novas transações. É importante ressaltar que essa execução se concentra apenas nas operações da camada blockchain, desconsiderando processos como a execução da API e a invocação das operações nas aplicações.

O Hyperledger Caliper avalia o processo de comunicação entre os participantes da rede e as transações, por isso, os processos mais demorados mencionados anteriormente não foram considerados nessa execução.

Listing 7.1 – Código de execução para experimento 1

```
async submitTransaction() {
        const functionCPF = () => {
            const cpf = Array.from({ length: 11 }, () => Math.floor(Math.
                ↪ random() * 10));
            return cpf.join('');
        }
        const randomizeCPF = functionCPF()
        const randomizeUsername = `user${randomizeCPF}`
        const randomizeEmail = `${randomizeCPF}@gmail.com`
        const randomizePassword = randomizeCPF
        const myArgs = {
            contractId: this.roundArguments.contractId,
            contractFunction: 'RegisterStudent',
            invokerIdentity: this.roundArguments.userID,
            contractArguments: [`${randomizeUsername}`, `${randomizeCPF}`,
                ↪ `${randomizeEmail}`, `${randomizePassword}`],
            readOnly: true
        }
        await this.sutAdapter.sendRequests(myArgs);
    }
```

Ademais, com o objetivo de verificar o atendimento das requisições dos serviços, foi realizado um segundo experimento focado na execução dos *endpoints* das APIs. Durante esse experimento, foi analisado o número e o tempo de requisições que o endpoint responsável por retornar a lista de atividades complementares conseguiu processar.

Para verificação do desempenho utilizou-se como base a biblioteca Autocannon [2] do JavaScript, utilizada para testar a performance de servidores HTTP, permitindo simular carga de trabalho através de solicitações em um modelo cliente/servidor, visando medir taxas de transferência e tempo para resposta. Assim como nos testes realizados no *benchmark* das chaincodes, este modelo também utiliza o conceito de *workers*, onde na configuração da conexão da biblioteca é possível definir a quantidade de trabalhadores para realizar as solicitações e o número de conexões simultâneas. Neste caso, utilizou-se um número de 8 *workers* e 8 *conexões simultâneas* todas realizando a mesma operação de leitura no endpoint *readAtividadesComplementaresFromAluno*.

Os resultados obtidos ao término do experimento, conforme demonstrado na Figura 35, revelam informações relevantes. O tempo médio de atendimento às requisições foi de 166 ms, enquanto o throughput médio alcançado foi de 659.5 requisições por segundo, com uma taxa de transferência média de 209kB por segundo. Esses resultados indicam que os experimentos apresentaram um desempenho satisfatório para o cenário da aplicação e sua demanda em um ambiente real. É importante mencionar que os experimentos foram realizados em uma máquina específica da infraestrutura do laboratório, que possui algumas limitações de hardware em comparação com a máquina utilizada anteriormente. Portanto, existe um potencial ainda maior a ser explorado. O hardware utilizado consistiu de um processador Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, com 2 núcleos de processamento, e uma memória virtual de 2GB.

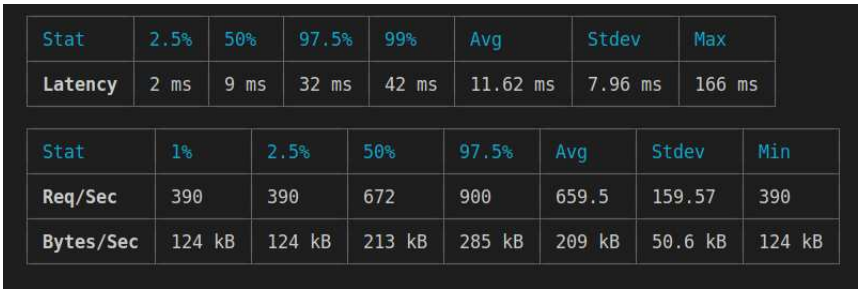
7.4.2 Experimentos de usabilidade

Para execução dos experimentos de usabilidade foi realizado pesquisas em um período de duas semanas, com 12 participantes. Onde após a utilização do aplicativo em um período de 5 a 10 minutos, os participantes respondiam um questionário com as seguintes perguntas: (1) Eu acho que gostaria de usar esse sistema com frequência; (2) Eu achei o sistema desnecessariamente complexo; (3) Eu achei o sistema fácil de usar; (4) Eu acho que precisaria do apoio de um especialista para usar esse sistema; (5) Eu achei várias funções do sistema bem

[2] https://github.com/mcollina/autocannon

```
2023.05.26-22:04:56.627 info  [caliper] [round-orchestrator]    Finished round 1 (student test) in 60.106 seconds
2023.05.26-22:04:56.627 info  [caliper] [monitor.js]    Stopping all monitors
2023.05.26-22:04:56.628 info  [caliper] [report-builder]        ### All test results ###
2023.05.26-22:04:56.630 info  [caliper] [report-builder]
+--------------+-------+------+-----------------+-----------------+-----------------+-----------------+------------------+
| Name         | Succ  | Fail | Send Rate (TPS) | Max Latency (s) | Min Latency (s) | Avg Latency (s) | Throughput (TPS) |
|--------------|-------|------|-----------------|-----------------|-----------------|-----------------|------------------|
| student test | 46790 | 0    | 785.1           | 0.05            | 0.00            | 0.01            | 785.0            |
+--------------+-------+------+-----------------+-----------------+-----------------+-----------------+------------------+
```

Figura 34 – Caliper teste cenário 1.



Stat	2.5%	50%	97.5%	99%	Avg	Stdev	Max
Latency	2 ms	9 ms	32 ms	42 ms	11.62 ms	7.96 ms	166 ms

Stat	1%	2.5%	50%	97.5%	Avg	Stdev	Min
Req/Sec	390	390	672	900	659.5	159.57	390
Bytes/Sec	124 kB	124 kB	213 kB	285 kB	209 kB	50.6 kB	124 kB

Figura 35 – Autocannon testes API.

integradas; (6) Eu achei que havia muita inconsistência no sistema; (7) Eu acho que a maioria das pessoas aprenderia a usar esse sistema rapidamente; (8) Eu achei o sistema muito trabalhoso de usar; (9) Eu me senti confiante usando o sistema; (10) Eu precisei aprender muitas coisas novas antes de conseguir usar o sistema. Após a pesquisa foi realizado os cálculos individuais de classificação SUS com as respostas de cada um dos participantes, através deles foi possível obter uma média de pontuação em 88.95, onde foi obtido uma nota máxima de 100 e uma nota mínima de 80. A pesquisa foi realizada no âmbito acadêmico com estudantes da graduação e do mestrado, onde dos pesquisados somente 6 deles conheciam previamente o tema do aplicativo ou são desenvolvedores, ao todo a pesquisa foi realizada entre estudantes de três universidades sendo estas FURG, UFPEL e UFSC. Dessa forma, através dos resultados obtidos da pesquisa foi possível concluir que o aplicativo atingiu aquilo que se esperava durante seu desenvolvimento alcançando um padrão excepcional segundo os critérios definidos pelo SUS.

8 CONCLUSÃO

O crescimento da tecnologia Blockchain fez com que as suas aplicações se tornassem possíveis nos mais distintos nichos, dentre eles o gerenciamento acadêmico. Os processos burocráticos do ensino superior são realizados entre diversas entidades e camadas responsáveis, sendo autorizados por uma entidade central o MEC. Contudo, os processos realizados são extensos, manuais e suscetíveis a falhas.

Sendo assim, a utilização da tecnologia Blockchain junto a contratos inteligentes possibilita a descentralização dos processos burocráticos e adição de segurança a estes devido à capacidade de auditoria das redes blockchains. Contudo, no modelo proposto pelo Projeto Jornada em desenvolvimento pelo LabSEC em uma TED com o MEC, as entidades principais do processo são as universidades e a sua comunicação distribuída. Dessa forma, o autor deste trabalho visualizou a possibilidade de tornar a participação do estudante efetiva como uma figura na rede distribuída, possibilitando a este a comprovação e autenticidade de das suas atividades realizadas, além de permitir que este participe ativamente do processo de emissão do seu histórico, através do envio de atividades e estágios.

Tendo em vista que o principal intuito do aplicativo foi fornecer praticidade em uma plataforma segura para o estudante, neste caso a Blockchain, sem que este precise estender as complexidades e abstrações da sua utilização, ter uma validação do usuário final se fez extremamente importante e necessário para conclusão deste trabalho. Para isto, foi utilizado o SUS como metodologia de validação do desenvolvimento, onde foram realizadas pesquisas com estudantes da graduação e mestrandos, pois estes representam o público alvo da pesquisa. Através desta pesquisa conclui-se que a proposta descrita nesse projeto atendeu aquilo que era esperado, ou seja, trazer praticidade e facilidade para estudantes do ensino superior via uma plataforma em contato com uma Blockchain.

Destaca-se que ao término dos experimentos do trabalho foi encontrado uma deficiência em relação a operação em um número maior de entidades distribuídas. Tendo em vista, que o processo de emissão de UUID é realizado em nível de chaincode e que todas os pares participantes precisam estar em consenso em relação a execução do código, já que o Fabric utiliza uma lógica determinística. Sendo assim, cada par irá gerar um código de UUID diferente causando um problema no consenso da rede. Dessa forma, para solucionar este problema é necessário adaptar o código removendo esta lógica do nível de chaincode e atribuindo esta responsabilidade as *application*.

Finalmente, com a proposta desenvolvida neste trabalho acredita-se ter dado uma visão de como a tecnologia Blockchain pode ser utilizada nos mais diversos âmbitos, utilizando-se as vantagens oferecidas pelo seu ecossistema em modelos de negócios já existente. Contudo, acredita-se que o trabalho atual precisaria de uma comprovação de efetividade maior em um cenário com participação de instituições reais, aplicando o modelo de negócio e verificando a sua capacidade de suprir as necessidades existentes em um complexo sistema de gerenciamento para o ensino superior. Além disso, em trabalhos futuros seria possível explorar os demais

processos no controle do ensino superior, tendo em vista que o grande foco neste TCC foi a utilização do mecanismo Blockchain para dar auditoria e credibilidade em dados emitidos anteriormente e vinculá-los a um sistema que se permitisse a participação do estudante. Sendo assim, um grande adicional seria vincular o processo de emissão de diplomas diretamente na Blockchain, onde o sistema educacional estaria vinculado a contratos inteligentes que iriam certificar de quando um estudante atingir os requisitos para conclusão de seu curso seu diploma fosse emitido de maneira automática e segura. A utilização de Blockchain possibilita que todo o ecossistema de emissão e controle de diplomas, históricos e currículos possa ser feito em uma única plataforma distribuída. Acredita-se também que futuros trabalhos e pesquisas na área podem explorar a maneira que os estudantes irão se autenticar seguramente, na proposta atual como este não foi o foco, utilizou-se uma chaincode especifica que verificava a senha por um hash criptográfico, contudo, ambientes reais e de larga escala demandam um tratamento especial para uma parte tão crítica do sistema.

REFERÊNCIAS

ABMES. **Concluir o Ensino Superior Triplica a renda, Mostra Ibge**. 2018. Acesso: 03-04-2022. Disponível em: https://abmes.org.br/noticias/detalhe/2746.

ABREU, A. W. S.; COUTINHO, E. F.; BEZERRA, C. I. M. A blockchain-based architecture for query and registration of student degree certificates. In: **Proceedings of the 14th Brazilian Symposium on Software Components, Architectures, and Reuse**. New York, NY, USA: Association for Computing Machinery, 2020. (SBCARS '20), p. 151–160. ISBN 9781450387545. Disponível em: https://doi.org/10.1145/3425269.3425285.

AGGARWAL, S.; KUMAR, N. Chapter twenty - attacks on blockchain. In: AGGARWAL, S.; KUMAR, N.; RAJ, P. (Ed.). **The Blockchain Technology for Secure and Smart Applications across Industry Verticals**. Elsevier, 2021, (Advances in Computers, v. 121). p. 399–410. Disponível em: https://www.sciencedirect.com/science/article/pii/S0065245820300759.

ALI, M. S. et al. Applications of blockchains in the internet of things: A comprehensive survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 21, n. 2, p. 1676–1717, 2018.

ALROOBAEA, R.; MAYHEW, P. How many participants are really enough for usability studies? In: . IEEE, 2014. Disponível em: https://encurtador.com.br/MOU06.

ANDROULAKI, E. et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: **Proceedings of the thirteenth EuroSys conference**. [S.l.: s.n.], 2018. p. 1–15.

BACKYARD, B. **Difference between anchor and leader Peer in fabric**. 2018. Acesso em: 23 de jan. 2022. Disponível em: https://blockchain-backyard.github.io/community-questions/posts/difference-between-anchor-and-leader-peer/.

BELLOMY, I. **Ethical Research in Usability Testing**. 2018. Acesso em: 23 de jan. 2022. Disponível em: https://www.bellomy.com/blog/ethical-research-usability-testing.

BIT2ME. **Como o bitcoin e outras criptomoedas podem ser armazenados**. [S.l.], 2020. Acesso em: 6 de março 2022. Disponível em: https://bityli.com/GOpnU. Acesso em: 6 de março 2022.

BRASILIA, R. Jornal de. **PF Investiga Grupo de Falsificação de Diplomas universitários**. 2022. Acesso em: 23 de jan. 2022. Disponível em: https:/jornaldebrasilia.com.br/noticias/brasil/pf-investiga-grupo-de-falsificacao-de-diplomas-universitarios.

BROOKE, J. Sus - a quick and dirty usability scale. **Usability Evaluation in Industry**, 1995.

BUTERIN, V. **The Power of Decentralization**. [S.l.], 2020. Acesso em: 23 de jan. 2022. Disponível em: https://proxet.com/blog/how-the-blockchain-is-changing-money-and-business/. Acesso em: 23 de jan. 2022.

BUTERIN, V. et al. Ethereum white paper. **GitHub repository**, v. 1, p. 22–23, 2013.

CASTRO, R. Q. d. **Falsificar diploma É crime antigo. Mas Blockchain Pode Ser a arma anti-fraude**. 2021. Disponível em: https://www.blocknews.com.br/opiniao/falsificar-diploma-e-crime-antigo-mas-blockchain-pode-ser-a-arma-anti-fraude/.

CEARA, G. **Alunos pagavam até R$ 3 mil por diploma falso emitido Por Faculdade no ceará**. 2021. Disponível em: https://g1.globo.com/ce/ceara/noticia/2021/05/11/alunos-pagavam-ate-r-3-mil-por-diploma-falso-emitido-por-faculdade-no-ceara.ghtml.

CENTER, U. E. **What Every Client Should Know About SUS Scores**. 2021. Bentley University. Disponível em: https://www.bentley.edu/centers/user-experience-center/what-every-client-should-know-about-sus-scores.

CROSBY, M. et al. Blockchain technology: Beyond bitcoin. **Applied Innovation**, v. 2, n. 6-10, p. 71, 2016.

DELTEC, B. **A proof of work explanation**. 2021. Disponível em: https://rb.gy/vz1mw.

DEV, J. A. Bitcoin mining acceleration and performance quantification. In: IEEE. **2014 IEEE 27th Canadian conference on electrical and computer engineering (CCECE)**. [S.l.], 2014. p. 1–6.

G1, T. **PF Cumpre Mandado contra Esquema Que tentou fraudar revalidação de diploma de medicina; Estudante Pagou R$ 100 mil**. 2022. Disponível em: https://g1.globo.com/to/tocantins/noticia/2022/02/23/policia-federal-cumpre-mandado-contra-esquema-que-tentou-fraudar-revalidacao-de-diploma-de-medicina.ghtml.

GAUR, N. et al. **Blockchain with Hyperledger Fabric: Build Decentralized Applications Using Hyperledger Fabric 2**. [S.l.]: Packt Publishing Ltd, 2020.

Hyperledger Fabric. **Hyperledger Fabric Documentation**. 2020. Disponível em: https://hyperledger-fabric.readthedocs.io/. Acesso em: 23 de jan. 2022.

IBM. **O que É o hyperledger fabric?** 2020. Disponível em: https://www.ibm.com/br-pt/topics/hyperledger.

IREDALE, G. **Top disadvantages of Blockchain technology**. 2022. Disponível em: https://101blockchains.com/disadvantages-of-blockchain/.

JARAMILLO, M. P.; PIEDRA, N. Use of blockchain technology for academic certification in higher education institutions. In: **2020 XV Conferencia Latinoamericana de Tecnologias de Aprendizaje (LACLO)**. [S.l.: s.n.], 2020. p. 1–8.

JELASITY, M. et al. Gossip-based peer sampling. **ACM Transactions on Computer Systems (TOCS)**, ACM New York, NY, USA, v. 25, n. 3, p. 8–es, 2007.

KARLSSON, H.; BERGVALL-KåREBORN, B.; LIND, T. The role of metaphors in user experience: A study of financial self-service technology. **International Journal of Human-Computer Interaction**, v. 29, n. 8, p. 514–527, 2013. Disponível em: https://encurtador.com.br/qHRZ1.

KHAN, A. A. et al. Educational blockchain: A secure degree attestation and verification traceability architecture for higher education commission. **Applied Sciences**, v. 11, p. 10917, 11 2021.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering–a systematic literature review. **Information and software technology**, Elsevier, v. 51, n. 1, p. 7–15, 2009.

MAKAROV, A. **Top 6 smart contract platforms: A deep dive**. 2021. Disponível em: https://www.itransition.com/blog/smart-contract-platforms.

MANEVICH, Y.; BARGER, A.; TOCK, Y. Service discovery for hyperledger fabric. In: **Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems**. [S.l.: s.n.], 2018. p. 226–229.

MEC. **Mulheres são maioria entre os universitários, revela o Censo**. 2021. Ministério da Educação. Disponível em: http://portal.mec.gov.br/ultimas-noticias/212-educacao-superior-1690610854/16227-mulheres-sao-maioria-entre-os-universitarios-revela-o-censo.

MENKE, F. Assinaturas digitais, certificados digitais, infra-estrutura de chaves públicas brasileira e a icp alemã. **Revista de Direito do Consumidor**, v. 12, n. 48, p. 17, 2003.

MISHRA, R. A. et al. Privacy protected blockchain based architecture and implementation for sharing of students' credentials. **Information Processing Management**, v. 58, n. 3, p. 102512, 2021. ISSN 0306-4573. Disponível em: https://www.sciencedirect.com/science/article/pii/S0306457321000212.

MOHAMED, L. et al. Digital certifications in moroccan universities: Concepts, challenges, and solutions. In: . [S.l.: s.n.], 2022. v. 201.

NABLE. **SHA-256 algorithm: N-able**. 2021. Disponível em: https://rb.gy/vz1mw.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. **Decentralized Business Review**, p. 21260, 2008.

NGUYEN, D.-H. et al. Cvss: A blockchainized certificate verifying support system. In: **Proceedings of the Ninth International Symposium on Information and Communication Technology**. New York, NY, USA: Association for Computing Machinery, 2018. (SoICT 2018), p. 436–442. ISBN 9781450365390. Disponível em: https://doi.org/10.1145/3287921.3287968.

PALMA, L. M. et al. Blockchain and smart contracts for higher education registry in brazil. **International Journal of Network Management**, Wiley Online Library, v. 29, n. 3, p. e2061, 2019.

PALMA, L. M. d. et al. Blockchain-based academic record system. 2020.

R7, R. **Estudo Mostra Aumento Na Busca Por Especialização na pandemia**. R7.com, 2021. Disponível em: https://noticias.r7.com/educacao/estudo-mostra-aumento-na-busca-por-especializacao-na-pandemia-03122021.

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

SECTIGO. **Public Keys and Private Keys in Public Key Cryptography**. [S.l.], 2020. Disponível em: https://bit.ly/3JBGrc5. Acesso em: 24 de março 2022.

SERGEENKOV, A. **Bitcoin mining difficulty: Everything you need to know**. CoinDesk, 2022. Disponível em: https://www.coindesk.com/learn/bitcoin-mining-difficulty-everything-you-need-to-know/#: :text=How

SHAWON, S. K. et al. Diucerts dapp: A blockchain-based solution for verification of educational certificates. In: **2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)**. [S.l.: s.n.], 2021. p. 1–10.

SOBTI, R.; GEETHA, G. Cryptographic hash functions: a review. **International Journal of Computer Science Issues (IJCSI)**, International Journal of Computer Science Issues (IJCSI), v. 9, n. 2, p. 461, 2012.

SSL.COM, E. d. S. **O que É um Certificado x.509?** 2021. Disponível em: https://www.ssl.com/pt/faqs/o-que-

STEINMETZ, R.; WEHRLE, K. **Peer-to-peer systems and applications**. [S.l.]: Springer, 2005. v. 3485.

STIFTER, N. et al. **Agreement with Satoshi – On the Formalization of Nakamoto Consensus**. 2018. Cryptology ePrint Archive, Report 2018/400. https://ia.cr/2018/400.

SUPERIOR, R. E. et al. **Taxa de brasileiros com ensino superior Chega a 34,3%**. 2019. Disponível em: https://revistaensinosuperior.com.br/ensino-superior-diploma/.

TURKANOVIć, M. et al. Eductx: A blockchain-based higher education credit platform. **IEEE Access**, v. 6, p. 5112–5127, 2018.

ULRICH, F. **Bitcoin: a moeda na era digital**. [S.l.]: LVM Editora, 2017.

USABILITY, M. **System Usability Scale (SUS)**. 2023. Disponível em: https://www.measuringux.com/sus/index.htm.

VIDAL, F.; GOUVEIA, F.; SOARES, C. Analysis of blockchain technology for higher education. In: **2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)**. [S.l.: s.n.], 2019. p. 28–33.

XU, X.; WEBER, I.; STAPLES, M. **Architecture for Blockchain Applications**. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2019. ISBN 3030030342.

APÊNDICE A – APÊNDICE

Blockchain Aplicada em uma Carteira Digital Acadêmica para Facilitar o Controle Curricular Estudantil

Leonardo Gideão Costa Rocha[1]

[1]Laboratório de Segurança em Computação (LabSEC)
Universidade Federal de Santa Catarina (UFSC) - Florianópolis/SC

leogcrocha@gmail..com

Resumo. *O surgimento e grande sucesso da tecnologia blockchain despertou o interesse da sua aplicação nas mais distintas áreas. Um cenário que possibilita trazer grandes benefícios através da utilização de blockchain é o acadêmico, onde, através da utilização desta tecnologia é possível trazer segurança, auditoria e descentralização para os processos burocráticos. Buscando trazer benefícios para o estudante de instituições de ensino superior, foi proposto um novo modelo que reestrutura uma aplicação que utiliza a blockchain desenvolvida pelo Projeto Jornada.*

1. Introdução

O termo *blockchain* teve repercussão no mundo em 2008, quando uma entidade anônima autointitulada de *Satoshi Nakamoto* [Nakamoto 2008], apresentou um sistema ponto-a-ponto (cada participante da rede funciona tanto como cliente, quanto como servidor, permitindo assim transmissão de informação sem uma entidade central) de transferências eletrônicas, utilizando como base uma moeda digital (criptomoeda), denominada Bitcoin.

Na rede Bitcoin, a confiança não é atribuída a uma entidade centralizada. Sua segurança é proporcionada por protocolos de consenso entre os participantes e incentivos. Em redes *blockchain*, todas as transações são armazenadas em blocos, estes que estão conectados em uma cadeia criptograficamente ligada, permitindo assim auditoria de todas as operações realizadas.

O sucesso da rede Bitcoin, despertou interesse de diversos pesquisadores e investidores, um deles foi Vitalik Buterin, fundador da Ethereum. Buterin, apresentou em 2013, em seu *whitepaper* [Buterin et al. 2013], um conceito que revolucionou as transações nas redes *blockchain*. Os *smart contracts*, ou contratos inteligentes, apresentados por Buterin, são códigos capazes de serem executados na rede *blockchain* que conseguem representar transações de diversas complexidades.

Contudo, apesar da capacidade de execução de complexas operações, a rede Ethereum não oferece um conceito importante buscado por empresas, negócios e governos, a privacidade. Sendo assim, o Hyperledger Fabric surge como a possibilidade de criar modelos de negócios privados e permissionados utilizando os benefícios da tecnologia Blockchain.

Atualmente o mercado de trabalho tem se tornado cada vez mais competitivo. Dessa forma, a procura por um diploma tem se tornado crescente. Contudo, vive-se uma desigualdade social e somente a menor parcela da população consegue ter acesso a uma educação de qualidade. Sendo assim, o esquema de falsificações tem se tornado um mercado chamativo e lucrativo para os criminosos.

Ademais, o cenário acadêmico de instituições de ensino superior é marcado pela alta burocracia, centralização (já que todas as instituições precisam ser vigidas pelo MEC) e processo manuais, oque causa um gargalo nas operações e processos demorados.

Nesse contexto, não só nó Brasil como em outros países, começa-se buscar soluções que utilizem *blockchain* e emissão de certificados digitais para que se possa combater os problemas de fraude.

Dessa forma, os autores [Palma et al. 2019] vislumbra-se a possibilidade de utilizar a tecnologia *blockchain* para combater fraudes e a alta burocracia. Em seu trabalho, os autores propõem um modelo distribuído de gerenciamento e emissão de certificados, e controle curricular. Neste modelo, a emissão de certificados é automatizada, onde um estudante ao terminar os créditos e requisitos necessários para conclusão do seu curso, terá um registro desta conquista armazenado na *blockchain*.

Contudo, apesar da completude do modelo apresentado pelos autores, no que diz respeito a jornada acadêmica de um estudante em graduação, seu foco está em ser um facilitador, desburocratizador e um combatente das fraudes voltado para as Instituições de Ensino Superior (IES). Assim, o modelo não propõe formas de trazer melhorias e utilizações diretas para o estudante. Neste cenário, o impacto final para o estudante só seria causado por uma melhoria no tempo de emissão de certificados. Portanto, neste trabalho, busca-se trazer benefícios e impactos mais diretos para o estudante de graduação, que agora será um participante direto da rede *blockchain*.

Neste contexto, é proposto um modelo onde um estudante fará parte da *blockchain* pre-estabelecida sendo desenvolvida pelo Laboratório de Segurança em Computação (LabSEC), onde ele poderá acessar suas informações acadêmicas como: (1) créditos realizados (2) atividades complementares realizadas; (3) certificados emitidos para ele. Ademais, o estudante poderá enviar informações para a *blockchain*, como por exemplo uma nova atividade concluída. Isto possibilitará, que o estudante consiga ter: (1) praticidade; (2) agilidade; (3) segurança; (4) uma forma autêntica e incontestável de provar e acompanhar a conclusão de suas atividades e seus certificados.

Para ser possível o ingresso do estudante na rede *blockchain* será necessária uma reformulação e um estudo a respeito de como a arquitetura da rede precisa se adaptar para possibilitar, atribuição de características específicas para o novo membro. Além disso, são necessários estudos a respeito de como este usuário irá interagir com a rede. Ademais, é necessário responder perguntas, como: que informações um estudante pode acessar; que informações um estudante pode adicionar na rede; de que maneira um estudante interage com a rede; quem irá permitir o seu ingresso; e como ele pode ser identificado.

Portanto, para este trabalho, será necessário todo um estudo de como possibilitar a integralização um sistema de contratos inteligentes de uma *blockchain* permissionada a um aplicativo. A partir disso, será possível trazer um impacto direto dos estudos realizados anteriormente. Dessa forma, as burocracias e trâmites que envolvem e atrasam os processos de certificação dos estudantes irão ser facilitadas.

2. Conceitos Fundamentais

2.1. Hash

É uma função matemática que dada uma entrada de qualquer tamanho é obtido um resultado de tamanho fixo e determinístico. Além disso, dentre outras propriedades o *Hash*

garante que dado uma saída é impossível chegar ao valor que gerou o resultado. Dessa forma, garantindo segurança e integridade de dados.

2.2. Blockchain

Blockchain pode ser vista, fundamentalmente, como um banco de dados distribuído para registro de transações. Estas operações ocorrem entre os participantes de uma rede armazenadas em um livro razão distribuído.

2.3. Livro razão

Livro razão é o registro de transações com finalidade de coleta de dados em ordem cronológica. Em redes como Hyperledger Fabric, este é dividido em duas partes uma contendo o registro de todas as transações a outra denominada *World State*, que contém os valores atuais de objetos específicos. Sendo assim, é possível por exemplo buscar um saldo de uma conta sem que seja necessário percorrer todas as transações.

2.4. Redes Públicas e Privadas

Em redes públicas qualquer participante pode ingressar sem a necessidade de uma aprovação (exemplo Bitcoin). Em redes privadas, participantes precisam ser aprovados antes de ingressarem na rede (exemplo Hyperledger Fabric).

2.5. Redes Permisionadas

Existe ainda mais um tipo de rede *blockchain*, sendo as redes de permissão que além da aprovação de um participante antes de ingressar na rede, estes precisam sempre estar identificados. Além disso, essas redes possibilitam que as entidades participantes tenham responsabilidades diferentes na rede. Um exemplo deste tipo de rede é o Hyperledger Fabric.

2.6. Contratos Inteligentes ou Chaincode

Permitem a programação e autoexecução de transações entre duas ou mais partes baseado em tecnologia *blockchain*, através desses contratos é possível representar as mais complexas operações sem a necessidade de um intermediário.

2.7. Protocolos de Consenso

São mecanismos em sistemas distribuídos que permitem que diferentes participantes cheguem a um acordo com o estado do sistema, mesmo que existam participantes inoperantes, inválidos ou maliciosos.

2.8. Hyperledger Fabric

Uma plataforma de blockchain de código aberto, capaz de suportar redes privadas e de permissão, trazendo flexibilidade, modularidade e privacidade para negócios mantendo as características de uma rede *blockchain* como imutabilidade, auditoria e descentralização.

3. Trabalhos Relacionados

Antes do início do desenvolvimento da proposta buscaram-se soluções de aplicações que se utilizam como base a tecnologia *blockchain* e permitissem algum tipo de participação do estudante diretamente com a rede.

Sendo assim, para uma seleção eficiente dos trabalhos relacionados, utilizou-se como base métricas definidas por [Kitchenham et al. 2009]. Além disso, definiram-se questões alvos da pesquisa, sendo estas:

Figura 1. Comparações entre os modelos.

Aplicativo Móvel	Teste de Tempo de Execução	Possibilita o estudante inserir dados na rede	Possibilita o estudante fazer leitura na rede	Custo em operações para o estudante	Interação com o estudante	Armazenamento offchain
✓	×	×	✓	×	✓	×
×	×	×	×	×	✓	×
×	×	×	✓	✓ *	✓	×
×	✓	×	✓	×	✓	×
×	✓	×	✓	×	✓	✓
×	×	✓	✓	✓ *	✓	×
×	×	×	✓	×	✓	×
×	×	×	✓	✓	✓	×
×	×	×	✓	×	✓	×

1. Existem modelos de exemplos, ou propostas que abrangem o que é proposto neste trabalho? Como esses projetos de base incentivam e contribuem para o atual projeto?
2. De que maneira é possível integrar um dispositivo móvel a uma aplicação on *chain*?
3. De que maneira um estudante pode ingressar e participar da rede *blockchain*?
4. De que maneira aplicativos descentralizados (DApps) que utilizam de *blockchain*, podem trazer benefícios para o usuário?

Ao término das buscas e verificação detalhada em cada um dos trabalhos, chegou-se ao resultado de dez propostas que atendem os requisitos e estão relacionadas a proposta em desenvolvida. Listam-se os trabalhos que contribuíram de alguma maneira para o desenvolvimento desta pesquisa.

- Analysis of Blockchain Technology for Higher Education [Vidal et al. 2019]
- A Blockchain-based Architecture for Query and Registration of Student Degree Certificates [Abreu et al. 2020]
- Digital Certifications in Moroccan Universities: Concepts, Challenges, and Solutions [Mohamed et al. 2022]
- CVSS: A Blockchainized Certificate Verifying Support System [Nguyen et al. 2018]
- Privacy Protected Blockchain Based Architecture and Implementation for Sharing of Students' Credentials [Mishra et al. 2021]
- Use of blockchain technology for Academic Certification [Jaramillo and Piedra 2020]
- EduCTX: A Blockchain-Based Higher Education Credit Platform [Turkanović et al. 2018]
- DIUcerts DApp - A Blockchain-Based Solution for Verification of Educational Certificates [Shawon et al. 2021]
- Educational Blockchain: A Secure Degree Attestation and Verification Traceability Architecture for Higher Education Commission [Ayub Khan et al. 2021]

Ao término da pesquisa nos trabalhos foi possível chegar a um comparativo entre cada um destes tendo em vistas as características importantes definidas para pesquisas, ou particularidades encontradas. Na Fígura 1, destaca-se os resultados obtidos para os trabalhos representando se cada um deles contêm ou não um fator.

Com os resultados obtidos, conclui-se que grande parte das aplicações desenvolvidas, não suprem a necessidade do usuário comum de menor poder computacional, que deseja por exemplo utilizar um dispositivo móvel simples para participação da rede *blockchain*. Ademais, grande parte dos trabalhos possibilitam a leitura, porém não permitem

que o estudante participe ativamente do processo de inserção de um novo dado na rede, deixando esta responsabilidade totalmente as instituições. Além disso, foi possível observar que por algumas redes *blockchain* estarem atreladas a uma criptomoeda para o incentivo do processo de consenso, foi necessário atribuir um custo para execução de operações para o estudante e/ou instituição. Ademais, destaca-se também que ao nível de código-fonte nenhum trabalho apresentou grandes contribuições, ou seja, para o desenvolvimento desta proposta a grande base foi o código-fonte oferecido pelo Projeto Jornada.

4. Proposta

Este trabalho propõe a integração entre estudantes e uma rede *blockchain* de gerenciamento de jornadas estudantis. O presente trabalho utiliza como premissa primordial a rede em desenvolvimento pelo Projeto Jornada, sendo atualmente mantido em parceria entre o MEC, LabSEC e Laboratório Bridge. Em suma, a rede base desenvolve a arquitetura de rede *blockchain* que possibilita a emissão e controle de certificados digitais de maneira descentralizada. Nesse modelo, IES realizam o envio de currículos e históricos escolares que representam as atividades realizadas pelos estudantes durante seu período acadêmico.

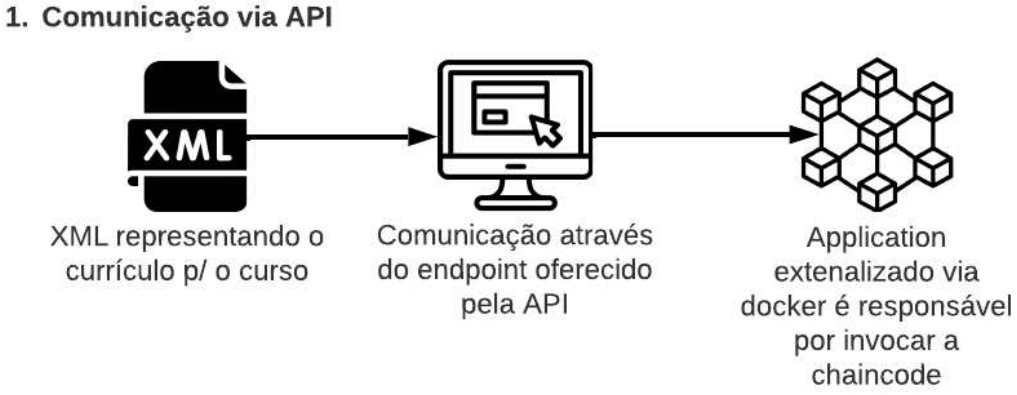
Para possibilitar que IES realizem operações na rede, o modelo base é composto por dois contratos inteligentes, *Academic Records* sendo responsável pela controle de históricos escolares, ou seja, representando toda a jornada do estudante e *Decree* que tem como finalidade definir as regras de modelo de negócio que precisarão ser seguidas pelo históricos enviados na rede, além disso, a *Decree* é responsável por deifnir as intituições e cursos presentes no rede. Para que IES consigam se comunicar com a rede de maneira mais simples, o modelo base propõe um esquema de *applications* e APIs que criam uma especie de ponte para comunicação entre os serviços, onde ao desejar enviar por exemplo um histórico para rede, uma IES deverá enviar uma requisição para API, que por sua vez irá enviar a operação para *application*, que por sua vez utilizará os serviços do *Fabric SDK* para enviar a operação para rede *blockchain*. Na Fígura 2, é possível observar um diagrama descrevendo este processo.

Sendo assim, tendo em vista que a *chaincode Academic Records*, é responsável por definir o objeto de um estudante na rede e construir a jornada acadêmica deste, foi necessário realização de adaptações nesta, visando a possibilidade de: (1) Adicionar atividades complementares; (2) Aprovar/reprovar atividades complementares; (3) Adicionar estágios; (4) Adicionar/reprovar estágios; (5) Emitir novos históricos de acordo com a adição de novos dados. Para chaincode *Decree* foi possível manter a lógica de negócio original.

Além disso, para possibilitar a comunicação e participação de um estudante na rede, foi necessário o desenvolvimento de três novos serviços: (1) chaincode *Student*; (2) *application Student*; (3) API *Student*. O modelo proposto é demonstrado na Fígura 3, onde descreve-se a estrutura em que o aplicativo móvel utiliza para se comunicar com a rede e oferecer as respostas para o usuário.

A chaincode *Student*, tem como objetivo permitir a um estudante a participação na rede. Sendo assim, quando este deseja por exemplo se registrar esse processo será feito e armazenado na *ledger*, desta chaincode no canal onde a rede foi definida. Ademais, quando um estudante deseja enviar por exemplo uma atividade complementar, ele precisará se comunicar com a chaincode *Student* e caberá a uma instituição que tenha permissão aos serviços da *Academic Records* aprovar esta atividade gerando assim um novo

Figura 2. Processo de comunicação de serviços.



histórico escolar, que teve a participação direta do estudante no processo de emissão.

Destaca-se que visando atribuir uma segurança maior a rede, somente é possível a um estudante se comunicar com os serviços oferecidos pela *Student*. Sendo assim, quando um estudante deseja por exemplo lê um histórico, este precisa enviar a operação pra *Student* e ela por sua vez irá solicitar os históricos vinculados ao estudante com um CPF específico. Essa arquitetura possibilita um controle maior de permissão para cada uma das operações.

5. Protótipo

Para execução do protótipo em desenvolvimento, partiu-se então da premissa da existência de uma infraestrutura para hospedar o servidor para execução da rede *blockchain*.

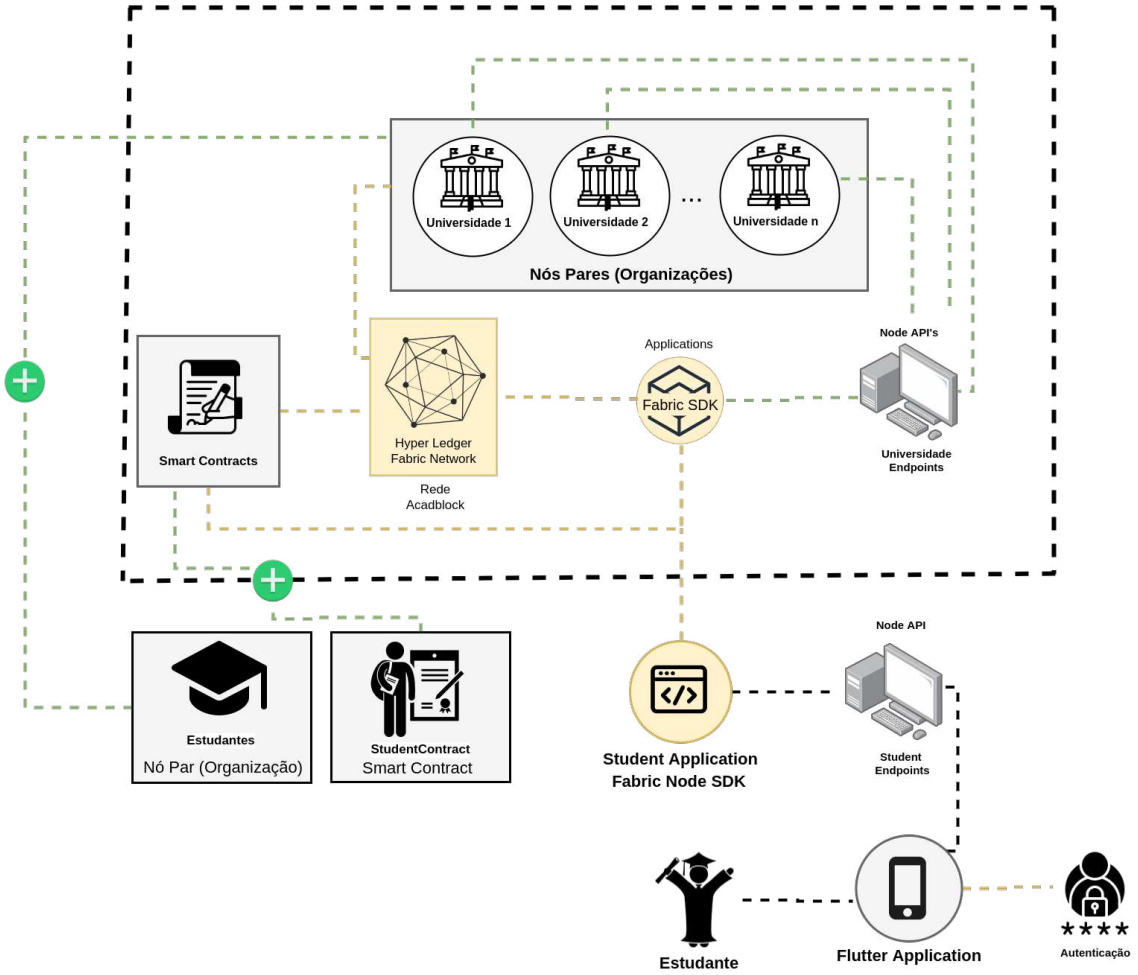
Para seu funcionamento toda rede, APIs e *applications* são mantidos através da execução por Contêineres Docker, permitindo assim que possa ser realizado em dispositivos com capacidades computacionais menores. Na Fígura 4, demonstra-se um esquematico simples de execução da rede.

Para os testes no ambiente, foi definida uma rede com uma instituição participante. Além disso, ofereceu-se a capacidade de qualquer dispotivio externo a infraestrutura do LabSEC pudesse se comunicar com os serviços da rede.

Na construção da interface gráfica do aplicativo segui-se o modelo apresentando pelo Governo Federal e utilizado pelo MEC e LabSEC no Projeto Jornada. A escolha se deu mediante ao fato de já ser um template válidado em diferentes métricas e desenvolvido por profissionais. Em sequência, utilizou-se o Figma para construção do protótipo do aplicativo que seria desenvolvido. Na Fígura 5, detaca-se as telas desenvolvidas na interfaces.

Ao término do desenvolvimento das interfaces foi feito a construção do aplicativo movél que seria oferecido através de um *.apk* para o usuário final. Para este desenvolvimento, optou-se pela utilização da linguagem Flutter devido a sua capacidade de ser

Figura 3. Modelo proposto.



multiplataforma, ou seja, utilizar o mesmo código nativo para Android, IoS e computadores possibilitando assim trabalhos futuros que busquem expandir o caso de uso.

6. Experimentos

Os experimentos do trabalho proposto foram divididos em dois grupos: (1) experimentos técnicos; (2) experimentos de usuabilidade. Experimentos técnicos tem como objetivo testar e medir a capacidade da rede, para isto, são executadas diversas operações na rede buscando causa um stress nesta, e verificar quantas operações podem ser atendidas. Ademais, tendo em vista que o projeto consiste no desenvolvimento de uma aplicação móvel que permite abstrair os conceitos de *blockchain* para o usuário e possibilitar que este faça parte ativamente de uma rede, se fez importante um teste de usuabilidade veirifacando a satisfação do mesmo durante o uso do aplicativo.

6.1. Experimentos técnicos

Os experimentos técnicos foram realizados em duas partes distintas, os testes para *blockchain* e testes para APIs.

Primeiramente para os testes de execução da *application* foram feito definindo uma rede distríbuida, onde um participante foi criado na infraestrutura do LabSEC e a

segunda instituição foi definida na máquina pessoal do autor. Visando testar ao máximo o desempenho da rede, optou-se pela verificação da operação de registro de estudantes. Esta operação, produz um stress máximo a rede já que precisa se comunicar com todas as chaincodes, onde primeiramente irá verificar se o estudante já existe no livro razão da *Student*, caso exista, precisará ser buscado os dados a respeito dos seus históricos escolar na *Academic Records*, e esta por sua vez precisará buscar informações a respeito dos currículos escolares do estudante utilizando a *Decree* para isso.

O Fabric oferece uma ferramenta nativa para execução de testes de desemepnho o Hyperledger Caliper. O Caliper é uma ferramenta de *benchmark* para *blockchains* Fabric, que possibilita execução de transações utilizando o SDK da própria plataforna.

Ao realizar os testes no computador pessoal do autor, com as caracteristicas de AMD Ryzen 7 4800H com gráficos Radeon, com 16 núcleos de processamento (4300 MHz) e 32 GB de RAM, observou-se um desempenho mais do que satisfatório no desempenho da rede, alcançado um total de 785 operações por segundo.

Visando verificar a capacidade das APIs de atenderem as solicitações recebidas dos dispositivos móveis, utilizou-se a biblioteca *Autocannon* do Node.js. Em seu funcionamento defini-se um número de trabalhadores e um tempo de execução para operações, durante este período será enviado o máximo de requisições possíveis visando medir e estressar a rede ao máximo. Tendo vista, que o objetivo desse teste não é verificar as operações na camada mais baixa (*blockchain*), optou-se por testar uma operação mais simples, neste caso, a leitura de atividades complementares.

Os testes em questão foram executados na máquina da infraestrutura do LabSEC, contendo a caracteristicas de Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, com 2 núcleos de processamento, e uma memória virtual de 2GB. Sendo assim, ao término da execução foi possível obter um média de 660 transações por segundo.

6.2. Experimentos usabilidade

Visando verificar a satisfação do usuário final, utilizou-se como base a métrica desenvolvida por John Brooke em 1986 [Usability 2023]. O SUS consiste em uma pesquisa onde são feitas dez perguntas para o usuário do aplicativo, após o término é calculado uma nota de usabilidade da aplicação de acordo com as respostas obtidas.

Os questionários foram realizados em um período de três semanas, sendo feito com doze estudantes de três instituições diferentes (UFPEL, UFSC e UFSC). No experimento, cada usuário utilizava o aplicativo livremente por um período de 5-10 minutos, e ao término respondia um questionário contendo as dez perguntas.

Finalmente, com o término do experimento foi possível obter uma nota média de 88.95%, onde levando em consideração a referência [Center 2021] seria um nível excelente de satisfação.

7. Conclusão

Ao término desta proposta foi possível concluir a possibilidade da participação de um estudante em uma rede *blockchain* funcional. Além disso, comprovou-se que não existe a necessidade de um entendimento sobre o assunto para que um usuário usufrua dos benefícios desta tecnologia. Foi possível perceber também a capacidade da comunicação de um dispositivo móvel com rede *blockchain*. Acredita-se que para trabalhos futuros seja possível verificar novos dispositivos, realização deu um teste mais robusto e expansão do modelo de negócio com mais instituições.

Figura 4. Grupo Docker.

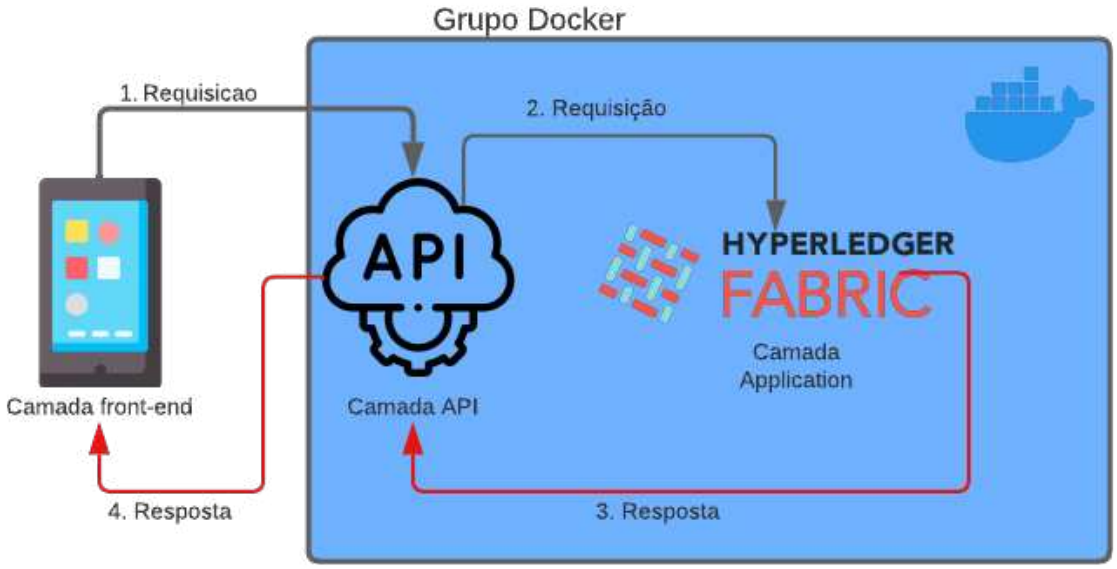
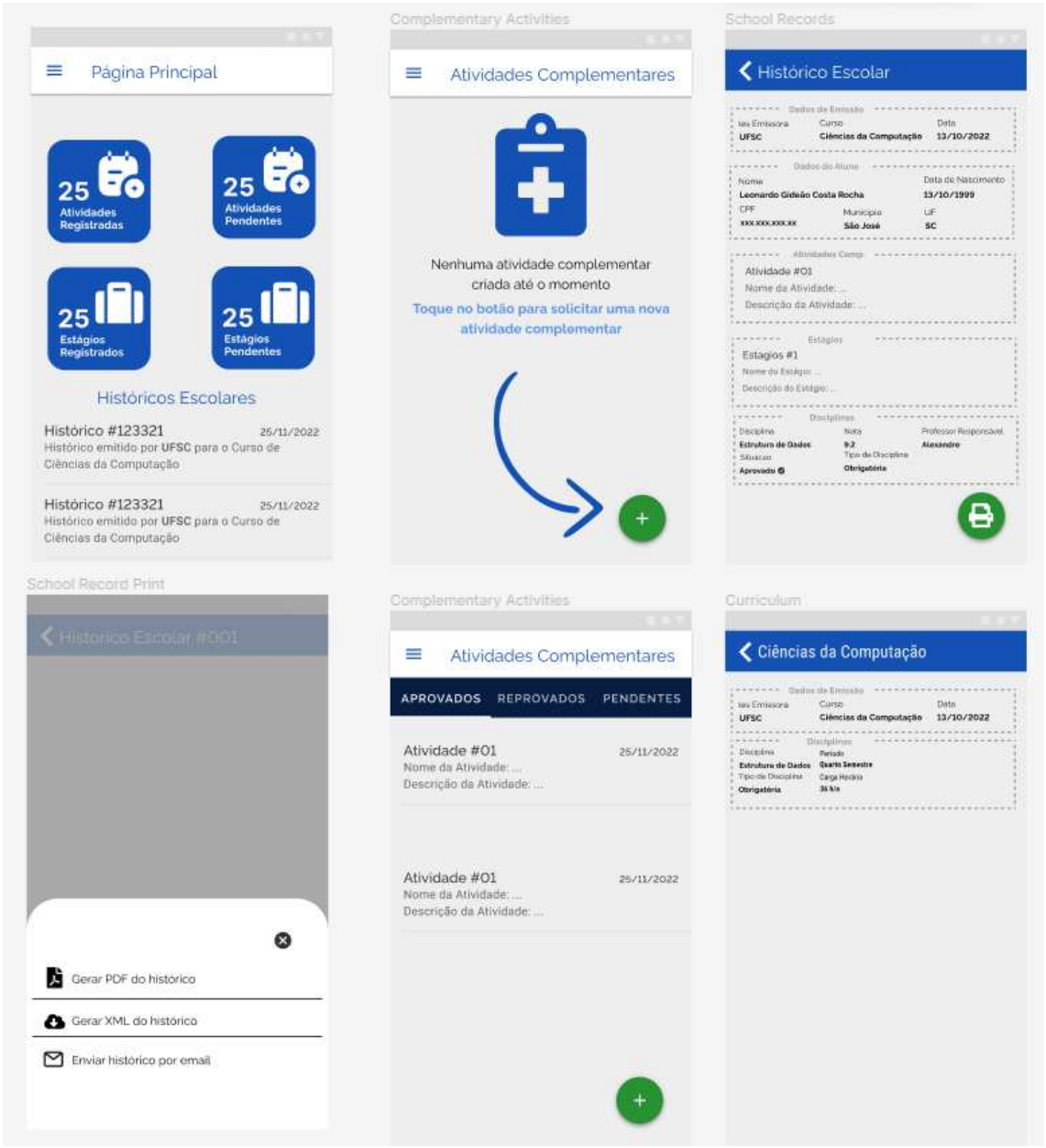


Figura 5. Interfaces do Aplicativo



Referências

Abreu, A. W. S., Coutinho, E. F., and Bezerra, C. I. M. (2020). A blockchain-based architecture for query and registration of student degree certificates. In *Proceedings of the 14th Brazilian Symposium on Software Components, Architectures, and Reuse*, SBCARS '20, page 151–160, New York, NY, USA. Association for Computing Machinery.

Ayub Khan, A., Laghari, A., Bourouis, S., Mamlouk, A., and Alshazly, H. (2021). Educational blockchain: A secure degree attestation and verification traceability architecture for higher education commission. *Applied Sciences*, 11:10917.

Buterin, V. et al. (2013). Ethereum white paper. *GitHub repository*, 1:22–23.

Center, U. E. (2021). What every client should know about sus scores. Bentley University.

Jaramillo, M. P. and Piedra, N. (2020). Use of blockchain technology for academic certification in higher education institutions. In *2020 XV Conferencia Latinoamericana de Tecnologias de Aprendizaje (LACLO)*, pages 1–8.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15.

Mishra, R. A., Kalla, A., Braeken, A., and Liyanage, M. (2021). Privacy protected blockchain based architecture and implementation for sharing of students' credentials. *Information Processing Management*, 58(3):102512.

Mohamed, L., Fartitchou, M., El Makkaoui, K., Ezzati, A., and El Allali, Z. (2022). Digital certifications in moroccan universities: Concepts, challenges, and solutions. volume 201.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260.

Nguyen, D.-H., Nguyen-Duc, D.-N., Huynh-Tuong, N., and Pham, H.-A. (2018). Cvss: A blockchainized certificate verifying support system. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, SoICT 2018, page 436–442, New York, NY, USA. Association for Computing Machinery.

Palma, L. M., Vigil, M. A., Pereira, F. L., and Martina, J. E. (2019). Blockchain and smart contracts for higher education registry in brazil. *International Journal of Network Management*, 29(3):e2061.

Shawon, S. K., Ahammad, H., Shetu, S. Z., Rahman, M., and Akhter Hossain, S. (2021). Diucerts dapp: A blockchain-based solution for verification of educational certificates. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–10.

Turkanović, M., Hölbl, M., Košič, K., Heričko, M., and Kamišalić, A. (2018). Eductx: A blockchain-based higher education credit platform. *IEEE Access*, 6:5112–5127.

Usability, M. (2023). System usability scale (sus).

Vidal, F., Gouveia, F., and Soares, C. (2019). Analysis of blockchain technology for higher education. In *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 28–33.