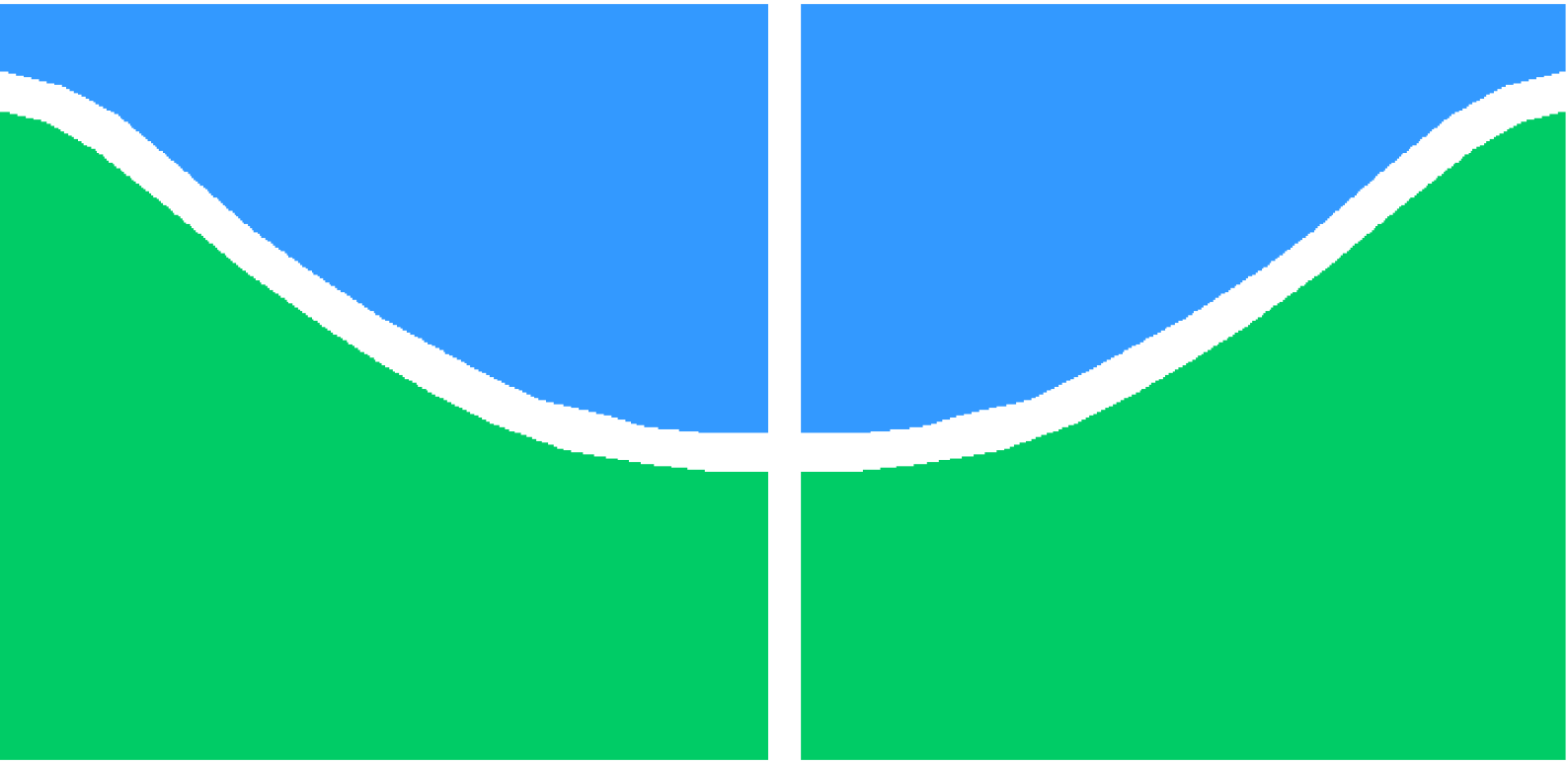

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Engenharia de Software

# Uma PoC sobre construção de aplicações com identidade auto soberana

Autores: Matheus Amaral Moreira
Nilo Mendonça de Brito Júnior

Orientador: Prof. Dr. Fernando William Cruz

Brasília, DF

2023

Matheus Amaral Moreira
Nilo Mendonça de Brito Júnior

# Uma PoC sobre construção de aplicações com identidade auto soberana

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Fernando William Cruz

Coorientador: Prof. Dr. Cláudio Gottschalg-Duque

Brasília, DF

2023

Matheus Amaral Moreira
Nilo Mendonça de Brito Júnior
Uma PoC sobre construção de aplicações com identidade auto soberana/ Matheus Amaral Moreira
Nilo Mendonça de Brito Júnior. – Brasília, DF, 2023-
78 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Fernando William Cruz

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2023.

1. Identidade autosoberana. 2. Blockchain. I. Prof. Dr. Fernando William Cruz. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Uma PoC sobre construção de aplicações com identidade auto soberana

CDU 02:141:005.6

Matheus Amaral Moreira
Nilo Mendonça de Brito Júnior

# Uma PoC sobre construção de aplicações com identidade auto soberana

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 21 de dezembro de 2023:

---

**Prof. Dr. Fernando William Cruz**
Orientador

---

**Prof. Dr. Cláudio Gottschalg-Duque**
Convidado 1

---

**Prof. Dr. Germana Menezes Da Nobrega**
Convidado 2

Brasília, DF
2023

# Resumo

A identidade auto soberana se caracteriza como uma abordagem inovadora que visa permitir aos indivíduos controle total sobre seus próprios dados de identidade, garantindo segurança, privacidade e descentralização. Esse modelo de desenvolvimento permite que os usuários mantenham seus dados pessoais em seu próprio dispositivo, utilizando criptografia e tecnologias de blockchain para garantir a autenticidade e a integridade dos dados. Sendo assim, o presente trabalho busca apresentar os conceitos fundamentais de identidade digital, assim como as limitações das soluções de identidade tradicionais, que geralmente dependem de autoridades centrais de confiança. Busca ainda por fim, demonstrar a implementação de uma aplicação segundo os modelos de SSI.

**Palavras-chaves**: Identidade auto soberana. Blockchain.

# Abstract

Self-sovereign identity is characterized as an innovative approach that aims to allow individuals total control over their identity data, guaranteeing security, privacy, and decentralization. This development model allows Users keep their data on their own devices, using encryption and blockchain technologies to ensure the authenticity and integrity of data. Thus, the present work seeks to present the fundamental concepts of digital identity, as well as the limitations of traditional identity solutions, which are usually dependent on central trusted authorities. Finally, it seeks to demonstrate the implementation of an application according to the SSI models.

**Key-words**: Blockchain. Self-sovereign identity.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| SSI | Identidade Auto-Soberana |
| LAI | Lei de Acesso à Informação |
| LGPD | Lei Geral de Proteção de Dados Pessoais |
| P2P | peer-to-peer |
| Tab | Tabela |
| Fig | Figura |
| PoW | Proof-of-Work |
| Pos | Proof-of-Stake |
| PBFT | Practical Byzantine Faul Tolerance |
| NIST | National Institute of Standards and Technology |
| MFA | Autenticação multifatorial |
| 2FA | Autenticação em duas etapas |
| OWASP | Open Web Application Security Project |
| HTTPS | Hyper Text Transfer Protocol Secure |
| HTTP | Hypertext Transfer Protocol |
| TCP | Protocolo de Controle de Transmissão |
| IP | Protocolo de Internet |
| OSI | Open Systems Interconnection |
| MAC | Media Access Control |
| FTP | File Transfer Protocol |
| SMTP | Simple Mail Transfer Protocol |
| CA | Certificate Authority |
| PKI | Infraestrutura de chave pública |

| | |
|---|---|
| PGP | Pretty Good Privacy |
| PoET | Proof of Elapsed Time |
| EVM | Máquina Virtual Ethereum |
| HLF | Hyperledger Fabric |
| DID | Decentralized Identifiers |
| DPKI | Decentralised Public Key Infrastructure |
| VC | Verifiable Credentials |
| VDR | Verifiable Data Registry |
| W3C | World Wide Web Consortium |
| SQL | Structured Query Language |
| JSON | JavaScript Object Notation |
| API | Application Programming Interface |
| ACID | Atomicidade, Consistência, Isolamento e Durabilidade |
| IEEE | Instituto de Engenheiros Eletricistas e Eletrônicos |

# Sumário

# 1 Introdução

Nos últimos anos, a evolução tecnológica tem revolucionado a forma como as transações financeiras e a gestão de identidade são realizadas. A ascensão das carteiras digitais representa uma mudança significativa no cenário financeiro e de segurança cibernética. Segundo o Fórum Econômico Mundial, “à medida que avançamos para a Quarta Revolução Industrial e mais transações são realizadas digitalmente, a representação digital da identidade de uma pessoa se torna cada vez mais importante; isso se aplica a humanos, dispositivos, entidades legais e além” (FORUM, 2021). Nesse sentido, uma carteira digital se apresenta como uma aplicação que permite aos usuários armazenar, gerenciar e utilizar informações sensíveis, como dados de pagamento e identidade, de forma digitalizada. A crescente adoção dessas carteiras traz consigo uma série de vantagens, incluindo a conveniência de transações sem dinheiro físico e a potencial melhoria da segurança em comparação com métodos tradicionais.

Um tópico central nessa discussão é o conceito emergente de *Self-Sovereign Identity* (SSI) ou identidade auto soberana, segundo Peixoto, "o termo self-sovereign identity representa uma abordagem à identidade digital que visa colocar o utilizador no centro do modelo, ou seja, o utilizador deve ser capaz de controlar totalmente a sua identidade, decidindo como, quando e com quem deseja partilhar as suas informações pessoais"(PEIXOTO, 2021). Isso contrasta com o paradigma convencional, no qual terceiros, como instituições financeiras e governamentais, detêm e gerenciam esses dados. A tecnologia SSI utiliza princípios criptográficos avançados, como assinaturas digitais e registros descentralizados, para permitir que os usuários compartilhem seletivamente suas informações pessoais, mantendo o controle sobre quem acessa esses dados e para quais finalidades.

Citando brevemente seu histórico, a identidade auto soberana surgiu como uma forma de superar limitações associadas a modelos tradicionais de gestão de identidades digitais. Se beneficiando principalmente de tecnologias como a *blockchain*, o conceito de SSI vem se popularizando de forma rápida. Sendo o projeto Sovrin um marco significativo em sua concretização, iniciado pela Sovrin Foundation, o projeto propôs uma infraestrutura baseada em *blockchain* específica para identidades, dessa forma oferecendo uma abordagem segura e transparente. A Sofrin Foundation buscou ainda estabelecer padrões e protocolos abertos, contribuindo para sua difusão.

Entre outros objetivos, o presente trabalho busca apresentar alguns conceitos relacionados ao SSI que são importantes de serem abordados para um melhor entendimento da tecnologia, entre eles a criptografia. Segundo Terada, "algoritmos criptográficos basicamente objetivam ’esconder’ informações sigilosas de qualquer pessoa desautorizada a

lê-las, isto é, de qualquer pessoa que não conheça a chamada chave secreta de criptografia"(TERADA, 2008).

Entre os diferentes modelos criptográficos existentes, os modelos de chave pública se destacam por oferecerem uma abordagem inovadora, em que diferente dos modelos de criptografia de chave simétrica, onde a mesma chave é utilizada para realizar a criptografia e descriptografia da informação, é utilizado um par de chaves. Sendo uma chave pública que é compartilhada livremente e uma chave privada, que é mantida apenas pelo dono. Dessa forma os modelos de criptografia de chave pública são frequentemente adotados em sistemas de *blockchain* e consequentemente de SSI.

Outro conceito importante que será abordado de forma mais profunda no decorrer do texto são as Autoridades Certificadoras (CAs). Segundo Soares, "a Autoridade Certificadora é o principal elemento de uma infraestrutura de chaves públicas. É ela a responsável por emitir certificados e LCRs, assim como manter informação sobre eles"(SOARES et al., 2017). O modelo utilizados pelas CAs se contrapõe totalmente à proposta do SSI, que visa disponibilizar um modelo descentralizado de controle de identidades.

Considerando os diversos avanços ocorridos no meio digital, surge a necessidade de avaliar a viabilidade e a segurança das carteiras digitais baseadas em SSI, bem como avaliar as principais diferenças entre essa nova proposta de sistema tendo em vista o modelo atual, que se baseia na utilização de autoridades centrais (Governos e CAs). Tendo isso em mente, o presente trabalho visa explorar esses tópicos, apresentando também uma prova de conceito que demonstra as funcionalidades e os desafios associados às carteiras digitais. A prova de conceito servirá como um laboratório controlado para analisar a interação entre SSI, as principais diferenças com os modelos atuais e os aspectos práticos da implementação dessas tecnologias.

Busca-se ainda descrever em detalhes a metodologia adotada para o desenvolvimento da prova de conceito, destacando os cenários de uso, as tecnologias empregadas e os resultados obtidos. Por fim, é feita a análise dos desafios encontrados durante o processo e as implicações mais amplas desses avanços no panorama das transações digitais e da gestão de identidade.

# 2 Identidade digital

## 2.1 O que é identidade no mundo real

Desde o princípio dos registros da civilização humana até os dias atuais, os documentos de identificação desempenham um papel fundamental na organização e na identificação das pessoas. O curso da história foi marcado por um contínuo desenvolvimento e evolução desses documentos, indo desde selos e insígnias até cartões e passaportes. Ao longo do tempo, a função de tais documentos se mostraram essencial para a manutenção da ordem social, proteção dos direitos individuais e facilitação de diversas interações e transações.

Servindo como pilares da sociedade, os documentos podem ser definidos como um meio utilizado pelas pessoas para provar que são realmente quem dizem ser, proporcionando uma maneira confiável de autenticação dos indivíduos. A Organização das Nações Unidas define a identidade jurídica como "[...] características básicas da identidade de um indivíduo, por exemplo: nome, sexo, local e data de nascimento, conferidos mediante registro e emissão de certidão por uma autoridade de registro civil autorizada após a ocorrência do nascimento".

Dessa forma a identidade se torna um elemento crucial para a participação dos indivíduos em atividades que vão desde acesso a serviços governamentais até a abertura de contas bancárias e a travessia de fronteiras internacionais.

## 2.2 O que é identidade digital

A identidade digital pode ser definida como uma representação única e reconhecível de uma pessoa ou entidade no espaço virtual, fornecendo um meio de fazer requisições de dados pessoais por meio de canais digitais. Sendo utilizada como uma forma de obter autenticação, autorização ou verificação de identidade em diversas atividades, como transações financeiras, acesso a serviços online, interações em redes sociais, etc.

Entre os principais pontos que justifica a necessidade de criar uma identidade digital se encontram a segurança, pois com o aumento da atividade online, surgiu a necessidade do desenvolvimento de métodos mais seguros para autenticação das pessoas e proteção de informações pessoais contra ameaças cibernéticas.

Outro ponto são as fraudes e roubos de identidade, uma vez que o comércio e as transações online requerem métodos mais eficazes de autenticação e verificação para facilitar trocas. Se destacando ainda a privacidade e o controle, pois cada vez mais dados

pessoais são coletados e compartilhados online, tornando-se necessário o desenvolvimento de maneiras de manter o controle sobre tais informações.

Atualmente existem diversos sistemas de identificação, sendo os três principais o sistema centralizado, o federado e o descentralizado. Eles utilizam de diferentes tipos de abordagem para gerenciar a autenticação e autorização de identidades, cada um com suas próprias características e vantagens, dependendo das necessidades e dos objetivos do sistema.

| | **Centralizada** | **Federada** | **Descentralizada** |
|---|---|---|---|
| **Definição** | Uma única organização estabelece e gerencia a identidade | Diferentes sistemas autônomos, cada um com sua própria âncora de confiança, estabelecer confiança uns com os outros | Várias entidades contribuem para uma identificação digital descentralizada; o utilizador controla o compartilhamento de dados pessoais |
| **Exemplos** | Cadernos eleitorais do governo, banco, plataforma de mídia social | Meta, Google | VCI, Identidade Digital Nacional do Governo do Butão (NDI) |
| **Forças** | – Pode ser construído para aplicações específicas ou para fins geral – A tecnologia é amplamente compreendida e implementável | – Pode permitir que os usuários acessem uma ampla gama de serviços – Pode ser conveniente para pessoa física, com potencial de reutilização | – Pode aumentar o controle do usuário, manter a privacidade e reduzir a quantidade de dados armazenados por intermediários – Pode melhorar a verificabilidade dos dados |
| **Limitações** | – Pode limitar o controle do usuário e criar risco de centralização – Pode não ser interoperável – Pode não ser interoperável com outras abordagens – Pode criar "honey pots" | – Pode limitar o controle do usuário – Pode não ser interoperável com outras abordagens – Pode criar "honey pots" | – Governança pode ser complexa – Aceitação e alinhamento sobre tecnologias subjacentes e padrões atualmente limitado – Alta complexidade técnica e alta demanda de pessoas |

Tabela 1 – Tipos de sistemas de identidade (FORUM, 2018)

Como podemos ver na Tab.1, os modelos possuem diferentes formas de centralização do controle, flexibilidade do compartilhamento de informações e autonomia dos usuários. O modelo centralizado se destaca por manter e gerir as informações de identidade por uma única autoridade central. Já no sistema federado, diferentes entidades colaboram para permitir que um único conjunto de credenciais de identidade seja usado em diferentes sistemas e serviços, sendo que a autenticação ocorre em um provedor de identidade, que emite tokens de identificação para serem usados em serviços federados. Já no sistema descentralizado o controle sobre as informações de identidade é distribuído entre os próprios indivíduos ou partes, sendo a identidade representada por meio de tecnologias como a *blockchain*, permitindo que os usuários controlem e compartilhem seletivamente suas informações com diferentes serviços.

## 2.3 Tecnologias subjacentes ao processo de identificação

Independente do sistema de identificação, o processo de estabelecer a identidade de uma pessoa, para posterior utilização em transações envolve alguns estágios, podendo esses estágios serem chamados de "ciclo de vida da identidade". Sendo ele vital para garantir a confiança entre as diferentes partes envolvidas no sistema.


Figura 1 – Ciclo de vida da identidade (ID4D, 2023)

Como é representado na Fig.1, o ciclo de vida da identidade possui diferentes etapas, iniciando quando um indivíduo registrar sua identidade pela primeira vez, tal registro envolve dois processos principais, a reivindicação de identidade que se caracteriza como a fase em que são capturados os atributos de uma pessoa que reivindica uma determinada identidade, como nome, data de nascimento, endereço, etc. E por último a prova de identidade, que ocorre após a pessoa fornecer os dados que então passarão por um processo de validação para confirmação de que tais dados são válidos.

Posteriormente ocorre a emissão das credenciais e/ou autenticadores. Para ser considerada uma identidade digital, as credenciais emitidas precisam armazenar dados eletronicamente ou serem utilizadas em ambientes digitais. Existe ainda a fase da autenticação que se caracteriza como o momento em que uma pessoa registrada e credenciada, se autenticar ou prova sua identidade para acessar determinado serviço, e as fases de autorização que é apenas uma validação de que tal indivíduo está apto a acessar determinado serviço e o gerenciamento das identidades realizado pelas organizações.

### 2.3.1 Modelo de confiança

Nas especificações que abordam as identidades digitais, é comum o modelo de confiança que demonstra a relação entre emissor, titular e verificador. Geralmente sendo representado na forma de um triângulo como demonstrado na Fig.2.

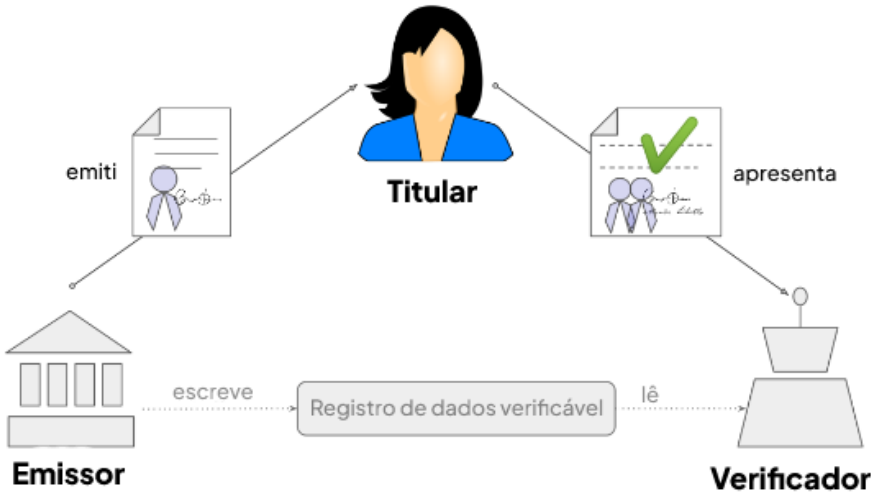


Figura 2 – Modelo de confiança (HANCOCK, 2020)

Os papéis de quem atua como verificador e emissor muitas vezes mudam a depender do contexto. Por exemplo, um servidor web (verificador) pode exigir a determinado visitante (titular) a verificação da sua identidade, da mesma forma em uma outra situação, um policial (verificador) pode exigir a determinado motorista (titular) a verificação da sua carteira de motorista. Esses casos demonstram que o verificador não necessariamente precisa ser uma pessoa, principalmente em ambientes digitais. Os emissores na grande maioria dos casos, são instituições que já existe um relacionamento estabelecido e que emitem algum tipo de documento, como diplomas ou certificados.

Quando se fala em identidade digital surge um termo muito importante, a credencial verificada, que simplificando, se define como uma declaração confiável entre um emissor, um titular e um verificador. Em novembro de 2019, o *World Wide Web Consortium* (W3C) publicou um importante padrão, o *Verified Credential Data Model*.

Esse padrão foi construído utilizando o formato do modelo de confiança de uma forma em que foram satisfeitos os princípios adotados na identidade descentralizada. Sendo constituída de três partes: um metadado de credencial, uma declaração e uma prova dessa declaração.

### 2.3.2 Princípios

Os princípios da identidade digital fornecem diretrizes e fundamentos para o desenvolvimento, implementação e uso de sistemas de identificação e autenticação digital. Tais princípios visam garantir a segurança, privacidade e confiabilidade, além de proporcionar interações online eficientes e confiáveis. Sendo eles:

- **Útil:** As identidades digitais devem ser úteis, possuindo entre suas características a portabilidade, a interoperabilidade, sendo aceitável e responsiva;
- **Inclusivo:** As identidades digitais devem ser inclusivas, sendo acessíveis e evitando discriminações;
- **Seguro:** As identidades digitais devem ser seguras, confiáveis e auditáveis;
- **Oferecer escolhas:** As identidades digitais devem disponibilizar escolhas, sendo centradas e gerenciadas pelos usuários e protegendo seus direitos;
- **Adequado para o propósito:** As identidades digitais devem se adequar ao seu propósito, de forma precisa, sustentável, aceitável e exclusiva;

# 3 Modelos de chave pública

## 3.1 O que é criptografia de chave pública

Considerando o contexto das comunicações digitais e do armazenamento de dados, a segurança é de vital importância. A criptografia, se define como um meio de transformar informações em um formato ininteligível para terceiros, sendo uma ferramenta essencial para proteger a confidencialidade e a integridade das informações sensíveis. Entre as várias técnicas criptográficas, a criptografia de chave pública, também conhecida como criptografia assimétrica, revolucionou a maneira como os dados são protegidos e autenticados em ambientes digitais.

A criptografia de chave pública opera com um par de chaves distintas, composto por uma chave pública e uma chave privada. A chave pública é compartilhada livremente, enquanto a chave privada é mantida em segredo. O processo de criptografia envolve a codificação de informações com a chave pública, enquanto a decodificação só é possível com a chave privada correspondente. Esse método permite que qualquer pessoa criptografe mensagens que só podem ser decifradas pelo detentor da chave privada correspondente.

A criptografia de chave pública tem várias aplicações significativas na segurança digital. Uma das mais notáveis é a autenticação de identidade e a troca segura de chaves para estabelecer conexões seguras em redes, como no protocolo HTTPS. Além disso, é a base da infraestrutura de chaves públicas (PKI), que sustenta a emissão de certificados digitais utilizados em assinaturas digitais, autenticação em serviços online e garantia de integridade de documentos.

Entre as principais vantagens da criptografia de chave pública se destaca a segurança aprimorada proporcionada pela separação das chaves de criptografia e decodificação. Isso supera uma das limitações da criptografia de chave simétrica, que exige a troca segura de chaves entre as partes. Entretanto, como uma das principais desvantagens se encontra o elevado custo computacional, exigindo operações matemáticas complexas, que acabam aumentando a carga de trabalho em sistemas com recursos limitados.

## 3.2 O que são as CAs

Uma Autoridade Certificadora (CA), também conhecida como *Certificate Authority*, é uma entidade de confiança encarregada de emitir e gerenciar certificados digitais. Tais certificados são utilizados para verificar a autenticidade de identidades digitais em um ambiente de comunicação criptografada, como a Internet.

A principal função de uma CA é verificar a identidade de uma entidade, seja ela uma organização, pessoa ou dispositivo, além de emitir um certificado digital que vincule essa identidade a uma chave criptográfica pública. Os certificados emitidos contém informações como o nome da CA que realizou a emissão, o nome da entidade e também sua chave pública.

Quando um usuário ou dispositivo deseja estabelecer uma comunicação segura com outra parte, ele pode utilizar a infraestrutura de chave pública (PKI) para verificar a autenticidade do certificado digital. A PKI é descrita como um conjunto de tecnologias, políticas e práticas que visam garantir a segurança da comunicação por meio de chaves criptográficas.

Ao confiar em uma CA reconhecida e confiável, é possível validar a autenticidade do certificado digital, determinar se a chave pública pertence à entidade desejada e estabelecer uma conexão segura. Podendo uma comunicação ser considerada suspeita nos casos em que a entidade não possua um certificado confiável ou válido, ou ainda, no caso de haver problemas de autenticidade.

O papel das autoridades de certificação em garantir a confiança e a integridade das comunicações é extremamente importante para a infraestrutura de segurança da Internet, promovendo a confiança e integridade das comunicações. Além da emissão de certificados, as CAs podem ainda desempenhar outras funções, como a revogação de certificados comprometidos, a renovação de certificados expirados e a realização de auditorias de segurança.

As Autoridades Certificadoras podem ser públicas ou privadas. As públicas são amplamente reconhecidas, e seus certificados são confiáveis pelos navegadores e sistemas operacionais mais utilizados. Já as privadas são utilizadas internamente por organizações ou em ambientes restritos, onde a confiança é estabelecida internamente.

### 3.2.1 Pretty Good Privacy

"O PGP é uma das soluções mais populares e amplamente utilizadas para criptografia de e-mails e arquivos. Ele oferece recursos como criptografia de chave pública, assinatura digital e compactação de dados, proporcionando segurança e privacidade nas comunicações eletrônicas."(ZIMMERMANN et al., ).

O *Pretty Good Privacy* (PGP) é um software de criptografia de dados que tem como objetivo garantir a confidencialidade, autenticidade e integridade das comunicações eletrônicas. Desenvolvido por Phil Zimmermann em 1991, ele é uma das soluções mais populares e amplamente utilizadas para criptografar e-mails e arquivos.

O PGP utiliza o sistema de criptografia assimétrica, sendo assim cada usuário possui um par de chaves composto por uma chave privada e uma chave pública. A chave

privada é mantida em segredo e é usada para descriptografar mensagens recebidas e assinar digitalmente arquivos ou e-mails. Já a chave pública é compartilhada com outros usuários e utilizada para criptografar mensagens destinadas ao proprietário da chave.

Ao enviar uma mensagem criptografada, o remetente utiliza a chave pública do destinatário para criptografá-la. Somente o destinatário, que possui a chave privada correspondente, é capaz de descriptografar e ler a mensagem, garantindo assim a confidencialidade dos dados transmitidos.

Além da criptografia de mensagens, o PGP também oferece suporte para assinatura digital. Através da assinatura digital, o remetente pode comprovar a autenticidade e a integridade dos dados enviados. A assinatura digital é gerada utilizando a chave privada do remetente e pode ser verificada pelo destinatário utilizando a chave pública correspondente.

O PGP disponibiliza ainda recursos adicionais, como compactação de dados para reduzir o tamanho dos arquivos criptografados, bem como a proteção de arquivos e diretórios por meio de criptografia.

A segurança comprovada e a ampla adoção do PGP são os principais fatores responsáveis pelo seu sucesso e popularidade. No entanto, a troca de chaves públicas entre os usuários pode ser complicada em comunicações em grande escala. Para facilitar esse processo, foram desenvolvidas soluções como servidores de chaves públicas ou PGP *key servers* e o conceito de círculo de confiança ou *web of trust*.

Na *web of trust*, os usuários desempenham o papel de validadores das chaves públicas uns dos outros. Cada usuário possui sua própria chave pública e assume a responsabilidade de assinar digitalmente as chaves públicas de outras pessoas para confirmar sua autenticidade. Essas assinaturas digitais são conhecidas como certificações ou endossos e demonstram a confiança do usuário na identidade associada à chave pública.

## 3.3 Chaves públicas controladas por uma *blockchain*

### 3.3.1 *Blockchain*

*Blockchain* pode ser considerada uma tecnologia emergente e disruptiva, que oferece uma maneira segura e confiável para realização de diferentes transações entre participantes que tenham ou não confiança entre si. Pode ser considerada disruptiva, pois cria uma entidade de confiança descentralizada, podendo dessa forma substituir uma terceira parte de confiança, como cartórios, governos, bancos, etc. Possuindo um enorme potencial para a criação de diferentes aplicações e em diferentes áreas do conhecimento como artes, finanças, saúde, entre outras.

A *blockchain* opera como uma máquina de estados replicada para a manutenção consistente de um estado geral compartilhado por diferentes conjuntos de pares distribuídos em uma rede P2P. Todos os nós existentes nessa rede mantém uma cópia dos registros das transações efetuadas, funcionando como uma espécie de livro-razão distribuído, sendo imutável, estando sempre disponível e podendo ser auditado (ZHENG et al., 2017).

A tecnologia *blockchain* se apresenta como o resultado de uma combinação de diferentes técnicas provenientes da teoria dos jogos, da criptografia e da computação distribuída, agregando elementos eficientes e capazes de implementar um sistema de acordo em escala global. Na origem da Blockchain, está presente o protocolo Bitcoin, apresentado por Satoshi Nakamoto (NAKAMOTO, 2008), que entrou em operação no ano de 2009. Tal artigo propõe uma rede P2P em que transações com moeda digital, propostas por clientes, são recebidas por mineradores (um tipo especial de servidor), que por sua vez, decidirão, por meio de um protocolo de consenso byzantino, baseado em uma série de desafios criptográficos, a ordem na qual serão armazenadas e realizadas numa cadeia de blocos, sendo essa última replicada em cada servidor.

Um dos grandes feitos do *blockchain*, foi a eliminação da terceira parte de confiança, até então indispensável nas transações financeiras. Trazendo dessa forma, uma ruptura nas transações de negócios, uma vez que introduziu os mecanismos de criptomoedas e incentivos digitais em diferentes níveis e relações.

### 3.3.2 Propriedades da *blockchain*

Essas são as principais propriedades presente na *blockchain* que colaboram de modo inovador para o desenvolvimento de sistemas e aplicações:

- **Descentralização:** Os sistemas e aplicações desenvolvidos em *blockchain* são executados de forma distribuída, por meio do estabelecimento de confiança entre pares;
- **Integridade e Disponibilidade:** Todos os dados na *blockchain* são replicados nos nós que compõem a rede, permitindo alta disponibilidade e consistência;
- **Auditabilidade e transparência:** Às transações adicionadas ao livro-razão são públicas, permitindo sua verificação e elevando a auditabilidade;
- **Irrefutabilidade e imutabilidade:** As transações adicionadas no livro-razão são imutáveis, atualizações são feitas apenas gerando novas transações;
- **Anonimidade e Privacidade:** Na *blockchain* cada usuário é responsável por sua própria chave e os nós armazenam somente uma parte criptografada dos dados;

- **Desintermediação:** A *blockchain* permite a integração entre diferentes sistemas de forma eficiente e direta, sendo considerado assim, um conector de sistemas muitas vezes complexos e permitindo ainda a subtração de intermediários;

### 3.3.3 Tipos de *blockchain*

Existem três tipos de redes *blockchain*, as públicas, consórcio e privadas (ZHENG et al., 2017). Cada uma possui diferentes características em relação a que pode ler, escrever e acessar os dados da *blockchain*. As do tipo pública, permite a visualização dos dados por qualquer pessoa, além de permitir a qualquer pessoa participar e contribuir tanto para o consenso quanto para mudanças no software central (ZHENG et al., 2017), sendo muito utilizada por criptomoedas. As redes *blockchain* do tipo consórcio é considerada parcialmente centralizada, tendo uma quantidade limitada de grupos selecionados aos quais é permitida a visualização e participação no protocolo de consenso, já as redes *blockchain* do tipo privada, a rede é distribuída mas geralmente centralizada, permitindo apenas que alguns nós selecionados participem da rede, geralmente esses nós são gerenciados por uma autoridade central (ZHENG et al., 2017).

| **Propriedade** | **Pública** | **Consórcio** | **Privada** |
|---|---|---|---|
| **Determinação de consenso** | Todos os mineradores | Conjunto de nós selecionado | Uma organização |
| **Permissão de leitura** | Público | Público ou restrito | Público ou restrito |
| **Imutabilidade** | Quase impossível | Pode ser adulterado | Pode ser adulterado |
| **Eficiência** | Baixo | Alto | Alto |
| **Centralizado** | Não | Parcial | Sim |
| **Processo de consenso** | sem permissão | com permissão | com permissão |

Tabela 2 – Tipos de *blockchain* (ZHENG et al., 2017)

### 3.3.4 Mecanismos de consenso

Um dos principais componentes da *blockchain* é o modo como são aceitas as entradas no livro-razão distribuído, feitas por meio de um mecanismo de consenso que realiza a validação dos dados de entrada. Existem diversos mecanismos mas os principais estão descritos na (Tab. 3) a seguir.

| **Propriedade** | **PoW** | **PoS** | **PBFT** |
|---|---|---|---|
| **Gerenciamento de nós** | Aberta | Aberta | Permissionada |
| **Consumo de energia** | Alto | Médio | Baixo |
| **Poder tolerado do adversário** | <25% de poder computacional | <51% de participação | <33.3% de réplicas com defeito |
| **Exemplo** | Bitcoin | Peercoin | Hyperledger Fabric |

Tabela 3 – Comparação entre mecanismos de consenso (HASSELGREN et al., 2020)

- ***Proof-of-Work*** é o protocolo de consenso utilizado pelo Bitcoin, se caracteriza pelos mineradores competirem pela resolução de problemas computacionais utilizando

força bruta, basicamente o objetivo é encontrar um *hash* para determinado bloco. O primeiro minerador que calcula o valor do *hash* que valida as transações é premiado. A grande desvantagem desse método é o alto consumo de energia quando aplicado em grandes *blockchains*.

- ***Proof-of-Stake*** é o método em que a seleção de um nó de aprovação é determinada que cada nó tem na *blockchain*, no caso das moedas digitais, a aposta é representada pelo saldo da moeda em questão. Uma das desvantagens desse método é o fato dele dar vantagem pros usuários com maiores saldos.

- ***Practical Byzantine Fault Tolerance*** se baseia em um protocolo de acordo bizantino, em que todos os nós precisam ser conhecidos pela rede, o que acaba por limitar seu uso a *blockchains* públicas. São estabelecidas três fases: pré-preparo, preparo e comprometimento. Cada um dos nós precisa de dois terços dos votos dos demais nós para passar pelas três fases.

### 3.3.5 Chaves controladas por *blockchain*

Como citado anteriormente, uma Autoridade Certificadora é uma entidade confiável que emite certificados digitais para verificar a autenticidade e a integridade de informações eletrônicas. Desempenhando um papel central na autenticação em sistemas que utilizam o protocolo SSL/TLS, como em sites seguros. Sendo elas responsáveis por emitir certificados para servidores e outras entidades.

Suponha que um atacante consiga comprometer os sistemas da CA. Como demonstrado na Fig.3, isso poderia permitir que o atacante emita certificados falsos, enganando os usuários ao acreditar que estão se conectando a sites legítimos, quando na verdade estão se comunicando com um site controlado pelo atacante (ataque de *man-in-the-middle*).

No PGP, que de forma resumida se caracteriza como um sistema de criptografia de código aberto que oferece autenticação e privacidade em comunicações eletrônicas,

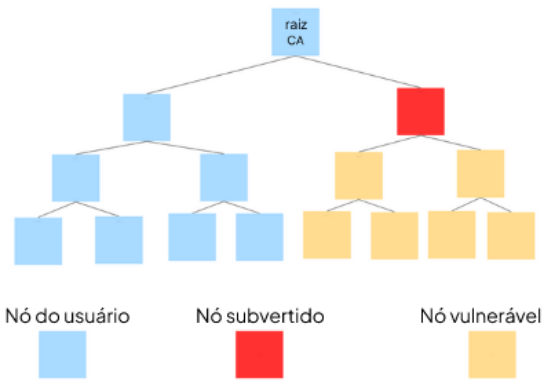


Figura 3 – Autoridade certificadora

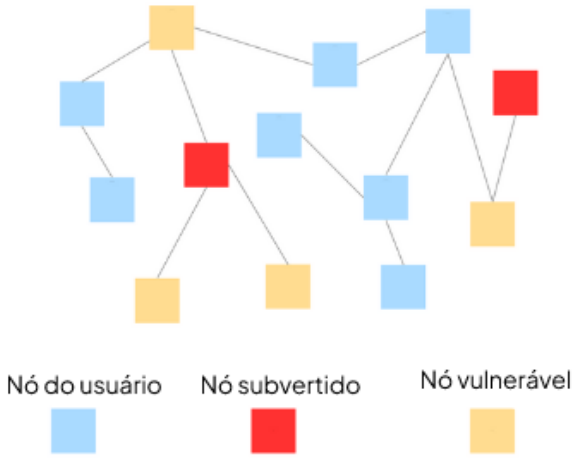


Figura 4 – *Pretty Good Privacy*

utilizando de um sistema de chaves assimétricas, que permite aos usuários criarem pares de chaves (pública e privada) para assinar digitalmente e criptografar mensagens. Como demonstrado na Fig.4 se um atacante obtiver acesso à chave privada de um usuário, poderá descriptografar suas mensagens criptografadas e, possivelmente, se passar por esse usuário ao assinar mensagens falsas. Porém diferente das CAs a cadeia de nós inferiores não é comprometida.

Por último, em um sistema DPKI baseado em *blockchain* que utiliza uma rede distribuída e imutável para armazenar chaves públicas e outros certificados digitais, eliminando dessa forma, a necessidade de uma única CA centralizada o que acaba por aumentar a confiabilidade por meio da descentralização, em casos de ataque como demonstrado na Fig.5 a rede se mantém confiável, não afetando os nós inferiores nem adjacentes como no caso do PGP.

Porém, mesmo que a natureza distribuída do *blockchain* torne os ataques mais difíceis, ainda é possível sofrer ataques, como um ataque de 51% em que um grupo mali-

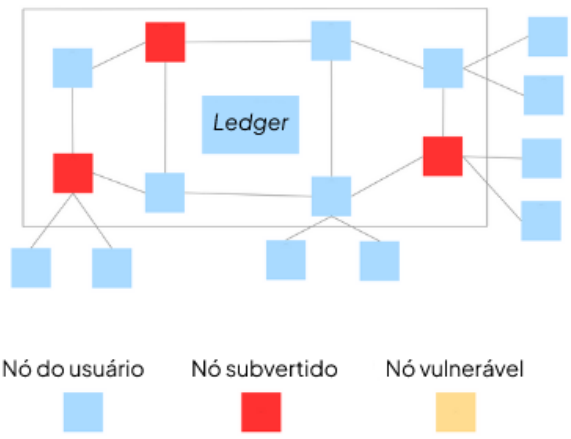


Figura 5 – DPKI com *blockchain*

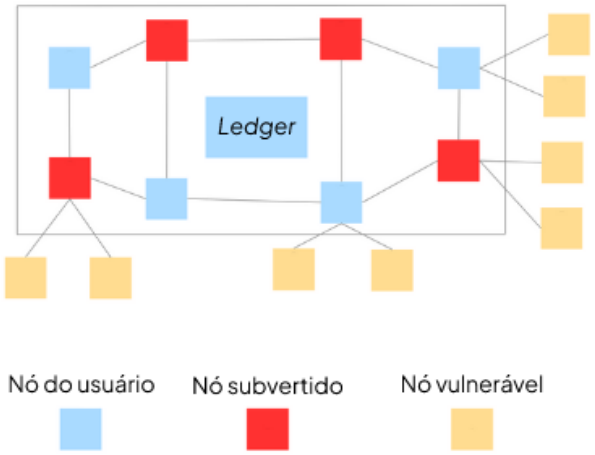


Figura 6 – Ataque a DPKI com *blockchain*

cioso controla a maioria do poder de computação da rede *blockchain*. Isso permitiria que o grupo adulterasse as informações de certificados ou chaves públicas no *blockchain* como demonstrado na Fig.6, se trata de um ataque difícil de ser realizado pelo alta exigência de recursos computacionais.

# 4 Identidade auto soberana

Levando em consideração a crescente digitalização das interações humanas, surgiu a necessidade do gerenciamento e proteção das identidades online. Dessa forma, surgiu o conceito de *Self-Sovereign Identity* (SSI) ou identidade auto soberana. Como citado anteriormente a SSI se baseia no princípio fundamental de que os indivíduos devem ter controle total sobre as suas informações de identidade, permitindo a verificação de forma segura e descentralizada, respeitando a liberdade e privacidade dos usuários.

Entre os principais componentes da SSI estão os identificadores descentralizados (são identificadores únicos, associados a usuários, organizações ou indivíduos), as credenciais verificáveis (são estruturas de dados que representam informações de identificação digitalmente), as carteiras digitais (são aplicativos onde os usuários armazenam seus DIDs e suas credenciais verificáveis de forma segura) e a *blockchain* (usada para registrar os DIDs e as credenciais verificáveis).

## 4.1 Modelos de credenciais

Um modelo de credencial se define como um conjunto de métodos e procedimentos para verificar a identidade e os privilégios de determinado usuário, sistema ou aplicativo em um sistema de computação. Envolvendo armazenar e verificar informações de autenticação, como nomes de usuários, senhas e chaves, para permitir o acesso a recursos específicos.

O funcionamento ocorre de maneira simples, quando um usuário ou aplicativo tenta acessar um sistema, ele fornece informações de identificação (credenciais). O sistema então verifica essas credenciais em relação aos dados armazenados e, se a autenticação for bem-sucedida, concede acesso aos recursos autorizados.

Entre os principais elementos que constituem os modelos de credenciais estão:

- **Nome de usuário e senha:** Método comumente utilizado, onde o usuário do sistema fornece um nome de usuário e uma senha para autenticação;
- **Token de Autenticação:** Método onde uma chave temporária é gerada após autenticação bem-sucedida, normalmente usada em aplicativos da web e APIs;
- **OAuth:** Protocolo para autorização e acesso a recursos em nome do usuário, sem compartilhar senhas;

- **OpenID Connect:** Estende o OAuth, permite autenticação única (SSO) em vários aplicativos;
- **SAML (Security Assertion Markup Language):** Protocolo para SSO, autenticação baseada em informações de um provedor de identidade;
- **Biometria:** Usa características únicas do corpo (impressões digitais, reconhecimento facial) para autenticação;
- **Chaves de API:** Chaves exclusivas para autenticação em serviços web e APIs;
- **Certificados digitais:** Usados para autenticação em redes seguras, baseados em criptografia.

A SSI por sua vez utiliza um modelo de credencial descentralizado, colocando o indivíduo no controle de suas próprias credenciais digitais, sendo elas emitidas como dados criptograficamente seguros e auto soberanos. Tais dados são armazenados em carteiras digitais, controladas pelo usuário e são compartilhadas de forma seletiva, sem a necessidade de uma autoridade central para realizar a validação dos dados.

## 4.2 *Self-sovereign identity*

No que diz respeito ao seu histórico, o conceito em si surgiu na década de 2010, quando houve uma crescente preocupação com a concentração de dados pessoais por parte das grandes corporações e entidades governamentais. Nesse cenário, iniciou-se a busca por alternativas para colocar o controle de volta nas mãos dos indivíduos. Sendo em 2016, a publicação de um *white paper* por Christopher Allen e Drummond Reed, estabelecendo os princípios básicos da SSI, onde é enfatizado a importância da interoperabilidade, portabilidade e consentimento do usuário.

No coração da identidade auto soberana está a tecnologia *blockchain*, que cria um *ledge* imutável e descentralizado de informações de identidade. Ao usar *blockchain*, os dados de identidade podem ser mantidos de forma segura e verificável, dessa forma, minimiza os riscos associados a violações de segurança centralizadas. A SSI também faz uso de criptografia de chave pública para garantir a autenticidade e integridade dos dados, permitindo que os usuários tenham chaves de criptografia exclusivas para controlar o acesso às suas informações.

As SSIs de forma geral são projetadas para suportar diferentes formatos de dados, permitindo a inclusão de diferentes tipos de informações. Isso inclui não apenas dados textuais, mas também dados multimídia, números, booleanos e outros formatos. Dessa forma sua utilização envolve diferentes casos de uso. Por exemplo, em finanças, a SSI pode efetivamente implementar a verificação de identidade para abertura de contas bancárias e

empréstimos. No caso de áreas como a saúde, os pacientes podem compartilhar informações médicas específicas com profissionais de saúde sem revelar seu histórico médico completo, além de diversas outras aplicações.

### 4.2.1 Credenciais Verificáveis

Para entender o modelo de dados de credenciais verificáveis, antes é importante citar os principais modelos de dados utilizados no tempo atual.

No contexto anterior ao surgimento da internet, temos um modelo de dados descentralizado baseado em credenciais físicas em papel como por exemplo: carteiras de identidade, carteiras de habilitação, certidão de nascimento, entre outros documentos em papel. Esse modelo é por natureza descentralizado pois o controle dos dados está na mão do usuário final que carrega um pedaço de papel com a certificação de uma terceira parte que arbitra como entidade neutra de confiança de todas as partes que no caso seria uma entidade do governo. Fazendo um contraponto, a versão centralizada desse modelo consiste em uma estrutura de hierarquia, onde o árbitro dessa comunicação também seria o detentor dos papéis de autorização como por exemplo: modelos rígidos de escolas onde para os alunos se locomoverem nos interiores durante o horário de aula precisam de um crachá de autorização de posse do professores. Nesse modelo, se qualquer usuário quiser fazer uma ação nova, antes ele precisa pedir a credencial física para uma das entidades prestadoras do serviço.

A partir desse exemplo centralizado é possível procurar por fragilidades no modelo. Para isso, pode-se usar o simples método de imaginar que uma das partes está subvertida por algum motivo. Uma parte subvertida é aquela que busca fazer algo malicioso (fora do protocolo) por alguma motivação que pode ser suborno, falsificação, ou até mesmo ausência. Se a parte subvertida for um aluno, ou até vários alunos, o modelo se mantém funcionando normalmente, pois o professor poderá barrar suspeitas de tentativas maliciosas. Porém, basta subverter o único professor da sala, que a sala toda estará comprometida.

Os modelos digitais de dados não diferem muito dos modelos físicos apontados. O modelo mais adotado para identidades digitais descentralizadas funciona escaneando o documento físico em uma imagem digital que por sua vez também será armazenada em algum banco de dados centralizado eventualmente. Quase todos os modelos adotados no ambiente digital são modelos centralizados, onde todos os dados de identificação do usuário são armazenados no banco de dados de posse do prestador de serviço.

O processo de autenticação de quase todos os modelos é feito pela comparação entre a palavra-chave armazenada no banco de dados centralizado e a da palavra-chave que esteja armazenada na memória humana do usuário. Isso foi eficiente por muito tempo,

porém a capacidade de armazenamento humana não é determinística, cada usuário tem uma memória diferente e que busca sempre padrões conhecidos. Isso acaba gerando o problema de senhas pequenas e frágeis, ou grandes e facilmente esquecíveis.

Com a evolução das tecnologias de subversão de usuário, hoje senhas fracas são quebradas em pouco tempo. Com isso, vem sendo criadas diversas alternativas para combater esses problemas de senhas e com isso é importante destacar um modelo OpenID Connect bastante adotado que soluciona esse problema. O modelo de dados OpenID Connect implementa um triângulo de confiança onde os dados do usuário ficam na posse de entidades confiáveis e sempre que o usuário precisar autenticar no prestador de serviço, ele envia um identificador de consentimento para a entidade confiável e o prestador de serviço, autorizando a entidade confiável de compartilhar os dados pessoais do usuário com o prestador de serviço. Uma observação desse modelo é que a entidade confiável é definida por número de usuários, sendo a implementação mais comum confiar a autenticação por meio de uma *BigTech* como Google ou Facebook.

Essa solução de ter uma entidade confiável funciona bem para a segurança nos dias atuais, porém ainda é uma solução centralizada onde todos os dados do usuário estão em posse de um árbitro e toda iteração que o usuário faz sempre terá a entidade confiável envolvida. A fragilidade de ter todo o controle dos dados e acessos nas mãos de um árbitro já foi apontada anteriormente no modelo físico centralizado de dados. O modelo protege a tentativa de subversão por meio de hacking da palavra-chave, porém se o alvo da subversão for o árbitro, todos os usuários estarão com seus dados pessoais comprometidos. E vale lembrar que a subversão pode acontecer principalmente por fatores humanos, independente da qualidade da tecnologia criptográfica utilizada.

Com isso, surge o modelo de dados das Credenciais Verificáveis propondo um modelo que assim como o OpenID Connect implementa um triângulo de confiança, porém com as seguintes diferenças: o controle dos dados pessoais na mão do usuário, entidades confiáveis apenas assinam credenciais que validam que o usuário tem posse dos dados e registram a chave-prova para validar essa credencial em um *ledger*, os prestadores de serviço apenas checam se a prova de posse apresentada pelo usuário condiz com chave-prova registrada no *ledger*. Observações importantes desse modelo é que só o usuário armazena dados, só as entidades confiáveis têm a permissão de escrever no *ledger* de registros e só os prestadores de serviço têm a permissão de ler os registros do *ledger*.

### 4.2.2 Identificadores descentralizados

Identificador Descentralizado (DID) é um conceito fundamental no campo da Identidade Auto Soberana e sistemas de identidade digital baseados em *blockchain* e tecnologias descentralizadas. O DID é um método globalmente exclusivo e durável para criar, resolver e gerenciar identidades digitais independente de qualquer intermediário centrali-

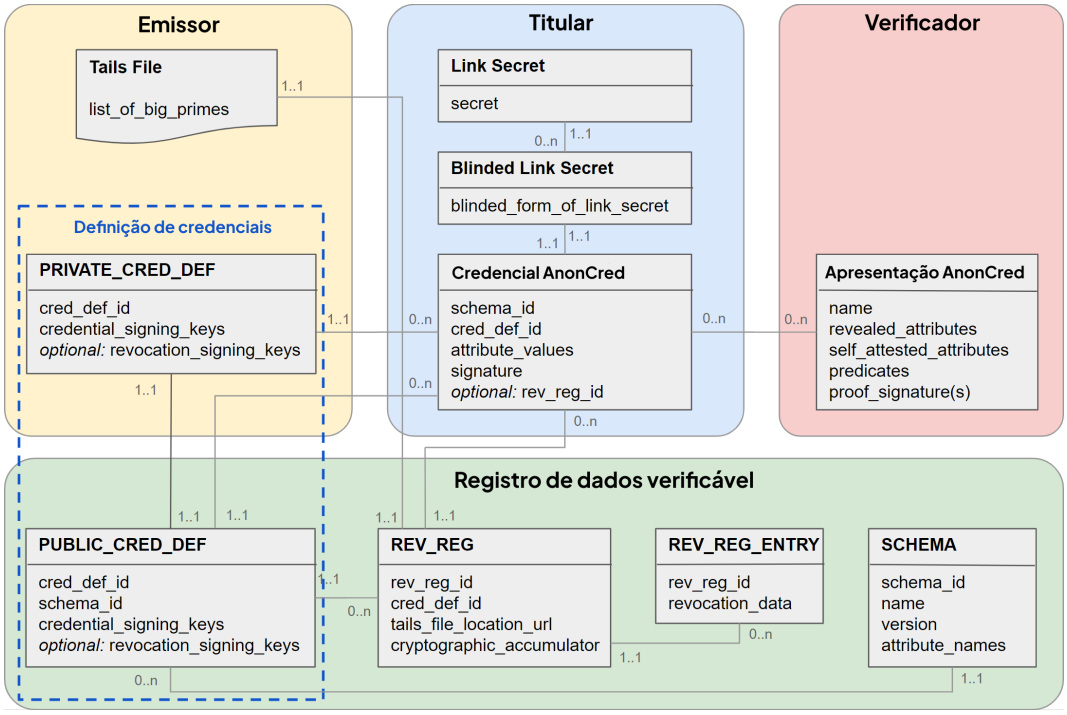


Figura 7 – Especificação do AnonCreds, (CURRAN ARTUR PHILIPP, 2023)

zado. Em outras palavras, é um método que permite que indivíduos e entidades assumam o controle total de suas identidades digitais sem depender de entidades externas para verificar e autenticar suas identidades.

Basicamente eles consistem de uma sequência única de caracteres e servem como um identificador único e permanente, semelhante a uma URL. No entanto, a diferença fundamental é que os DIDs são projetados para serem endereçados em sistemas descentralizados e distribuídos, como *blockchains* e redes *peer-to-peer*, em vez de depender de servidores centralizados.

Podem ainda ser usados para representar qualquer tipo de entidade, como indivíduos, organizações, dispositivos e até coisas físicas ou abstratas. São criados de forma a permitir que o próprio titular da identidade tenha o controle exclusivo sobre quem pode visualizar e verificar os dados associados a esse DID. Isso é obtido por meio de chaves criptográficas, assinaturas digitais e ledges descentralizados.

Por fim, eles também são projetados para serem independentes de qualquer sistema de registro específico, permitindo que as entidades usem diferentes *blockchains* e sistemas de armazenamento de dados para gerenciar suas identidades de acordo com suas preferências e necessidades.

### 4.2.3 Carteira digital

A carteira digital trata-se de uma solução eletrônica, que permite o armazenamento de dados financeiros e pessoais, para que possam ser utilizados, posteriormente, em operações financeiras e comerciais (GONÇALVES et al., 2022). Uma carteira digital no contexto de SSI se caracteriza como uma aplicação que permite a um indivíduo ou entidade gerenciar suas informações de identidade de maneira segura, privada e descentralizada. Estas carteiras digitais desempenham um papel essencial na implementação prática do conceito SSI, dando aos utilizadores o controle total sobre os seus dados pessoais, permitindo aos proprietários controlar como será feita a comunicação com as demais partes que requerem as informações. Sendo assim, as carteiras digitais são capazes de realizar várias funções básicas, entre elas as citadas abaixo:

- **Armazenamento de dados criptografados:** as carteiras digitais armazenam as identidades dos usuários e outros dados pessoais em um formato criptografado e seguro. Isso garante que os dados sejam mantidos em sigilo e não possam ser acessados por terceiros não autorizados.
- **Geração e gerenciamento de chaves:** A carteira digital é responsável por gerar e armazenar a chave do usuário. Essas chaves são usadas para criar assinaturas digitais e provar a autenticidade das informações compartilhadas.
- **Compartilhamento seletivo de informações:** Uma das principais características das carteiras digitais no contexto do SSI é a capacidade de compartilhar informações de identidade de maneira seletiva e controlada. Os usuários podem decidir quais informações desejam compartilhar com cada interação sem expor todos os detalhes.
- **Interoperabilidade com diferentes redes e protocolos:** As carteiras digitais são projetadas para interoperar com diferentes redes e protocolos de identidade descentralizados, permitindo que os usuários se conectem a vários serviços e sistemas.
- **Verificação de Credenciais:** As carteiras digitais podem receber e armazenar credenciais digitais, como diplomas, certificados e outros documentos autenticados. Essas credenciais podem ser compartilhadas conforme necessário para comprovar qualificações e histórico educacional.

### 4.2.4 Contratos inteligentes

*Smart Contracts* são contratos digitais construídos em um código de computador e armazenados na *blockchain*, auto executáveis, de caráter descentralizado, e que prezam pela praticidade, redução de custos e pelo anonimato (DIVINO, 2018). Sendo eles projetados para automatizar e facilitar a execução de acordos digitais sem que seja necessário a presença de intermediários. Eles contêm lógica de programação que define as

condições, termos e ações a serem realizadas quando certas condições pré-definidas são atendidas. Sua execução ocorre automaticamente, garantindo transparência, segurança e imutabilidade nas transações.

No contexto da sua utilização junto à SSI, cria-se um ecossistema onde as identidades digitais podem interagir diretamente com os contratos inteligentes de forma segura e transparente. Tal combinação torna possível uma série de aplicações, como transações financeiras, acordos contratuais e verificações de identidade, tudo de maneira descentralizada e eficiente.

A utilização da SSI em conjunto com os contratos inteligentes oferece maior segurança e privacidade em transações digitais, uma vez que as identidades são controladas pelos próprios usuários e as execuções contratuais são automatizadas e verificáveis na *blockchain*. Isso reduz a dependência de intermediários e centralizações, o que por sua vez, acaba gerando uma abordagem mais eficaz e transparente para a gestão de identidades e contratos digitais.

### 4.2.5 *Verifies*

Os *verifies* ou verificadores são entidades ou serviços que solicitam e verificam informações de identidade dos usuários, desempenhando o papel de verificar a autenticidade das informações apresentadas por um titular de identidade. Os verificadores podem ser instituições, empresas, organizações ou qualquer entidade que precise confirmar detalhes de identidade de um usuário para certo propósito.

No momento em que um usuário deseja compartilhar suas informações de identidade, ele faz a apresentação de credenciais verificáveis a um verificador específico, esse por sua vez, valida as credenciais apresentadas pelo titular da identidade, confirmando assim a autenticidade das informações.

O uso de verificadores no contexto de SSI contribui para a descentralização e segurança do ecossistema de identidade digital, garantindo que as informações pessoais sejam compartilhadas apenas quando necessário e com o consentimento do titular da identidade. Essa abordagem visa superar os desafios associados à centralização de dados de identidade, como violações de privacidade e riscos de segurança.

### 4.2.6 *Zero-Knowledge Proofs*

Provas de zero conhecimento, ou *Zero Knowledge Proof* (ZKP) em inglês, é um protocolo criptográfico onde um usuário é capaz de provar a outro usuário, através de computação polinomial, que ele conhece o valor de X, sem revelar qualquer outra informação além do fato que ele conhece X (HARIKRISHNAN; LAKSHMY, 2019). Em

resumo, significa que uma parte pode provar que ela conhece um determinado dado, sem revelar o próprio dado.

O conceito fundamental por trás das *Zero-Knowledge Proofs* é permitir a verificação de informações sem a necessidade de compartilhar essas informações diretamente. Sendo útil em situações em que a privacidade e a segurança dos dados são essenciais. Existem diversos tipos de *Zero-Knowledge Proofs*, sendo eles aplicados em diversos contextos. Entre os mais conhecidos estão o *Proof of Knowledge of a Secret* ou Prova do Conhecimento de um Segredo em que possibilita uma parte provar que conhece um segredo específico sem revelar o próprio segredo, o *Proof of Knowledge of a Statement* ou Prova do Conhecimento de uma Afirmação em que possibilita que uma parte prove ter conhecimento de certa solução para determinada afirmação ou equação, sem revelar como chegou a essa solução e por último a *Zero-Knowledge Proof of Identity* ou Prova de Identidade com Conhecimento Zero em que uma parte pode provar sua identidade sem revelar informações adicionais sobre si mesma.

#### 4.2.6.1 Anoncreds

Sendo uma das aplicações práticas do conceito de *Zero-Knowledge Proof*, o termo *anoncreds* se define como uma abreviação de *anonymous credentials* ou credenciais anônimas. Trata-se de um conceito e uma tecnologia que se refere à capacidade de provar algum atributo sobre determinado indivíduo sem revelar sua identidade em si. Em outras palavras, as credenciais anônimas permitem que alguém prove que possui certas qualidades ou características sem revelar quem eles são.

O objetivo principal das credenciais anônimas é fornecer um meio de compartilhar informações de maneira seletiva, mantendo a privacidade dos indivíduos. Isso é particularmente relevante em cenários em que é necessário verificar a autenticidade de certas informações sem que seja revelado a identidade do detentor dessas informações.

Um exemplo prático de uso de credenciais anônimas é na verificação de idade para acessar conteúdo ou serviços online restritos a uma faixa etária específica. Ao invés de revelar a data de nascimento completa, uma credencial anônima pode ser usada para provar que o usuário tem idade suficiente sem expor informações pessoais adicionais.

## 4.3 Limitações

Entre as principais limitações da SSI está a sua adoção de forma generalizada, enfrentando obstáculos significativos de interoperabilidade. Para que esse modelo de identidade seja eficaz, é vital um amplo apoio e cooperação entre diversas partes interessadas, como organizações governamentais, empresas, instituições financeiras e provedores de ser-

viços. A falta de coordenação e padronização nesse sentido pode dificultar a aceitação e consequentemente sua eficácia.

Outro desafio é a questão da acessibilidade e inclusão. Segundo Chaves, "é fundamental compreender que o acesso à web deve ser concebido por meio de uma abordagem ampla. As percepções sensoriais de cada indivíduo devem ser respeitadas, garantindo-lhes o direito a interagir com a web"(CHAVES et al., 2019). A implementação da SSI muitas vezes pressupõe o acesso à tecnologia, como smartphones e conexão à internet. Isso pode excluir segmentos da sociedade que não têm acesso fácil a esses recursos, criando disparidades digitais e limitando sua aplicabilidade em contextos mais amplos.

Por fim, existem algumas questões em discussão entre os interessados na área, como a problemática referente à capacidade de armazenamento de dados por meio de carteiras digitais. Tendo em vista que os registros são mantidos localmente nos dispositivos dos usuários, depreende-se que tais dispositivos estão sujeitos a limitações inerentes ao espaço de armazenamento disponível. Essa circunstância a longo prazo pode causar problemas. Outra discussão importante é a questão da segurança, uma vez que os dados ficam armazenados localmente, em casos de roubo ou furto dos aparelhos a instituição de mecanismos eficazes para a inacessibilidade dos dados se revela crucial, sendo necessário a adoção de alternativas mais eficazes além dos recursos já empregados, como a criptografia.

## 4.4 Tecnologias semelhantes

Com a criação das redes *blockchain* diversas tecnologias se tornaram possíveis, entre elas uma que se assemelha bastante com a SSI que é o Solid/Pods, sendo uma iniciativa liderada por Sir Tim Berners-Lee, o inventor da *World Wide Web*, que tem por objetivo reinventar a maneira como os dados são armazenados e gerenciados na internet. Busca-se uma abordagem descentralizada, conhecida como Solid (*Social Linked Data*) e propõe que os usuários tenham maior controle sobre seus próprios dados, armazenando-os em repositórios chamados "Pods"(*Personal Online Data Stores*). Os Pods são estruturados em torno do conceito de dados interoperáveis, onde as informações podem ser compartilhadas de forma seletiva entre aplicativos e serviços, proporcionando uma maior autonomia aos usuários sobre sua presença digital.

Outra tecnologia que vem se destacando são as Redes Sociais Descentralizadas (DSN) que se caracterizam por serem plataformas que buscam transformar a arquitetura tradicional das redes sociais, dessa forma, assim como as demais tecnologias com base em *blockchain*, oferecem maior controle sobre interações online e dados pessoais aos usuários.

As semelhanças entre essas duas tecnologias e a SSI residem na abordagem de dar poder aos usuários sobre suas informações digitais. Sendo um reflexo da necessidade de superar a centralização excessiva de dados e identidades na internet, oferecendo aos

usuários maior autonomia e controle. Dessa forma, as Redes Sociais Descentralizadas e os Solid/Pods proporcionam um ambiente mais democrático, onde as interações online são governadas pelos próprios usuários, reduzindo a dependência de plataformas centralizadas e mitigando preocupações relacionadas à privacidade e segurança dos dados.

No que diz respeito às principais diferenças, elas residem nas áreas específicas de foco e na implementação prática. O Solid/Pods se concentra em redes sociais e no gerenciamento descentralizado de dados. Enquanto as DSN visam descentralizar as interações sociais online. Por outro lado, a SSI é uma abordagem mais ampla que se estende além do gerenciamento de dados para abranger a identidade digital.

# 5 Plataformas de identidade auto soberana

Apesar das particularidades pertencentes às diferentes redes de *blockchains*, o conceito de SSI pode ser implementado na grande maioria, por se tratar de um modelo recente existem poucas redes desenvolvidas exclusivamente para a utilização em carteiras digitais. Entre as redes *blockchains* popularmente conhecidas e adequadas para o desenvolvimento de identidades auto soberanas pesquisadas para o desenvolvimento do projeto estão:

- **Ethereum:** Sendo uma das *blockchains* mais populares e amplamente adotadas para o desenvolvimento de aplicativos descentralizados. Suporta contratos inteligentes e oferece recursos avançados de privacidade e segurança. Possui ainda uma grande comunidade de desenvolvedores e uma ampla gama de bibliotecas e ferramentas disponíveis.
- **Sovrin:** É uma rede *blockchain* construída em cima do Hyperledger Indy. Ele oferece uma infraestrutura escalável e segura para identidades auto soberanas. A rede Sovrin é projetada para facilitar a troca de informações de identidade de forma confiável entre diferentes organizações e sistemas.
- **R3 Corda:** Foi projetada para atender às necessidades de aplicações empresariais e oferece recursos avançados de privacidade e segurança. A Corda suporta a criação de contratos legais digitais e facilita a troca segura de informações entre as partes.
- **Stellar:** Embora não seja exclusivamente voltada para identidades auto soberanas, a *blockchain* Stellar é conhecida por suas capacidades de transações rápidas e de baixo custo. Ela oferece recursos como contratos inteligentes e gateways de confiança que podem ser usados para construir sistemas de identidade auto soberana eficientes.
- **Hyperledger Indy:** É uma *blockchain* específica para identidades auto soberanas. Foi projetado para fornecer recursos e protocolos específicos para o gerenciamento de identidades digitais e permite a criação de sistemas de identidade altamente confiáveis e interoperáveis. O Hyperledger Indy oferece suporte a recursos como prova de conhecimento zero (zero-knowledge proofs) e criptografia avançada.

Dentre todas essas opções avaliadas, optou-se pela utilização do Hyperledger Indy, pelos seguintes pontos:

- **Arquitetura distribuída e descentralizada:** O Hyperledger é baseado em tecnologia de *blockchain*, o que significa que oferece uma arquitetura distribuída e descentralizada. Isso é fundamental para a SSI, pois permite que os usuários tenham

controle total sobre suas identidades, eliminando a necessidade de intermediários centralizados.

- **Privacidade e segurança:** O Hyperledger oferece recursos avançados de privacidade e segurança que são essenciais para aplicações de identidade auto soberana. Ele utiliza criptografia robusta e assinaturas digitais para proteger as informações pessoais e garantir a autenticidade dos dados. Além disso, o modelo de permissões granulares do Hyperledger permite controlar quem tem acesso a quais dados, garantindo a privacidade dos usuários.

- ***Smart contracts*:** O Hyperledger suporta *smart contracts*, também conhecidos como *chaincode*. Os *smart contracts* são contratos digitais programáveis que permitem a execução automática de ações com base em condições predefinidas. Na SSI, os *smart contracts* podem ser usados para implementar regras de governança e gerenciar interações entre identidades. Isso permite que os usuários tenham controle sobre suas próprias informações e determinam como elas são compartilhadas com outras partes.

- **Escalabilidade:** O Hyperledger oferece soluções escaláveis para o desenvolvimento de aplicações de identidade auto soberana. Através do uso de consenso distribuído e mecanismos de governança flexíveis, o Hyperledger pode lidar com um grande número de transações e participantes na rede, sem comprometer o desempenho.

- **Comunidade e suporte:** O Hyperledger é um projeto de código aberto mantido pela Linux Foundation, o que significa que possui uma comunidade ativa e engajada de desenvolvedores e especialistas. Isso proporciona acesso a recursos, documentação e suporte contínuo, tornando mais fácil para os desenvolvedores construírem e aprimorarem aplicações de identidade auto soberana usando o Hyperledger.

## 5.1 Hyperledger Indy

Para ter uma visão mais completa a respeito do Hyperledger Indy, antes, é necessário conhecer um pouco a respeito dos principais elementos que compõem este espaço:

- **Linux Foundation:** Fundada em 2000 é a líder mundial no que diz respeito ao desenvolvimento de tecnologia aberta, é ainda bastante estimada pela comunidade de desenvolvedores. A Linux Foundation realiza, dentre outras práticas, o fomento de parcerias que abordam os maiores desafios do mundo através da computação de código aberto. Faz também uma série de investimentos em diversos projetos, sendo responsável além disso por construir um ecossistema que abriu caminho para as tecnologias *blockchain*;

- **Hyperledger:** se caracteriza como um projeto de código aberto que saiu da Linux Foundation, sendo criado para ajudar no avanço das tecnologias *blockchain* entre setores. É um projeto global aberto para colaboração e envolve líderes de vários setores;

- **Códigos e padrões abertos:** Como observado anteriormente, o movimento de computação aberta lançou as bases para o desenvolvimento da *blockchain* e consequentemente do Hyperledger. O termo *open source* na computação se caracteriza como um modelo de licenciamento de software, no qual o usuário possui direitos sobre o código, podendo usá-lo, distribuí-lo e até mesmo aprimorá-lo. Entre as principais vantagens de aplicações comerciais de código aberto é a grande flexibilidade disponibilizada por meio do código-fonte aberto, a aderência a padrões e a modularização dos componentes. Dessa forma permite que diferentes organizações adaptem a tecnologia para alcançar um grau elevado de usabilidade com pouco esforço.

### 5.1.1 Estruturas do Hyperledger

O Hyperledger é composto por cinco *frameworks blockchain*, como segue:

- **Hyperledger Fabric:** Fornecido pela IBM, é projetado para ser uma base para o desenvolvimento de aplicativos ou soluções com uma arquitetura modular. Permite componentes *plug-and-play*, como serviços de consenso e associação e aproveita os contêineres para hospedar contratos chamados chaincode que compõem a lógica de aplicação do sistema.

- **Hyperledger Iroha:** Foi projetado para o desenvolvimento mobile, se baseia no Hyperledger Fabric e teve contribuições de Soramitsu, Hitachi, Dados NTT e Colu. Apresenta um design C++ orientado a domínios como um novo algoritmo de consenso tolerante a falhas bizantino que se baseia em cadeia;

- **Hyperledger Sawtooth:** Sendo uma contribuição da Intel, inclui um novo algoritmo de consenso chamado *Proof of Elapsed Time* (PoET), que busca alcançar consenso distribuído da forma mais eficientemente possível. Tem potencial em muitas áreas, com suporte para implantações com e sem permissão e reconhecimento de diversos requisitos.

- **Hyperledger Burrow:** Surgiu inicialmente como contribuição da Monax e Intel, se caracteriza por ser uma *blockchain* modular construída pelo cliente para a especificação da Máquina Virtual Ethereum (EVM).

- **Hyperledger Indy:** Contribuído inicialmente pela Fundação Sovrin, Indy é um projeto Hyperledger feito para suportar identidade independente, fornece ainda ferramentas, bibliotecas e recursos reutilizáveis.

### 5.1.2 Indy

O Hyperledger Indy é um *framework* que propõe o modelo da Hyperledger Foundation para o desenvolvimento de SSI. Ele engloba todos os componentes que se entendem necessários para um sistema SSI, sendo eles: *Digital Wallet* para descentralizar o controle dos dados nos usuários finais de uma maneira moderna que não precise de papel físico; Uma DPKI (*Decentralised Public Key Infrastructure*) composta por DIDs (*Decentralized Identifiers*) em uma estrutura de VCs (*Verifiable Credentials*) armazenadas em uma VDR (*Verifiable Data Registry*) que armazena e compartilha todas as credenciais em *blockchain*.

#### 5.1.2.1 Características

O Hyperledger Indy é uma plataforma de código aberto projetada para fornecer uma infraestrutura segura e descentralizada para identidade digital. Suas principais características incluem:

- **Autonomia:** O Hyperledger Indy possibilita aos indivíduos o controle total sobre sua identidade digital, permitindo-lhes criar, gerenciar e controlar suas próprias identidades independente de autoridades centrais ou intermediárias.
- **Privacidade:** A plataforma enfatiza a privacidade dos usuários, possibilitando que eles compartilhem somente as informações essenciais em transações e interações específicas, sem expor dados pessoais desnecessários.
- **Portabilidade:** A infraestrutura do Hyperledger Indy foi desenvolvida para facilitar a portabilidade das identidades digitais, possibilitando que os usuários as utilizem em diversos aplicativos, organizações e ecossistemas sem a necessidade de criar identidades separadas.
- **Segurança:** A plataforma emprega criptografia avançada e técnicas de assinatura digital para assegurar a segurança das identidades e das transações. Adicionalmente, o Hyperledger Indy adota uma arquitetura descentralizada, distribuída e resiliente a ataques para garantir a proteção dos dados.
- **Interoperabilidade:** O Hyperledger Indy é construído com base nos padrões abertos do W3C (*World Wide Web Consortium*), o que simplifica a interoperabilidade com outras soluções e sistemas de identidade digital já existentes.

#### 5.1.2.2 Arquitetura

A arquitetura do Hyperledger Indy é baseada em uma rede descentralizada e distribuída que permite a criação, emissão, controle e verificação de identidades digitais. Na Fig.8 é demonstrado um diagrama com os principais componentes da arquitetura.

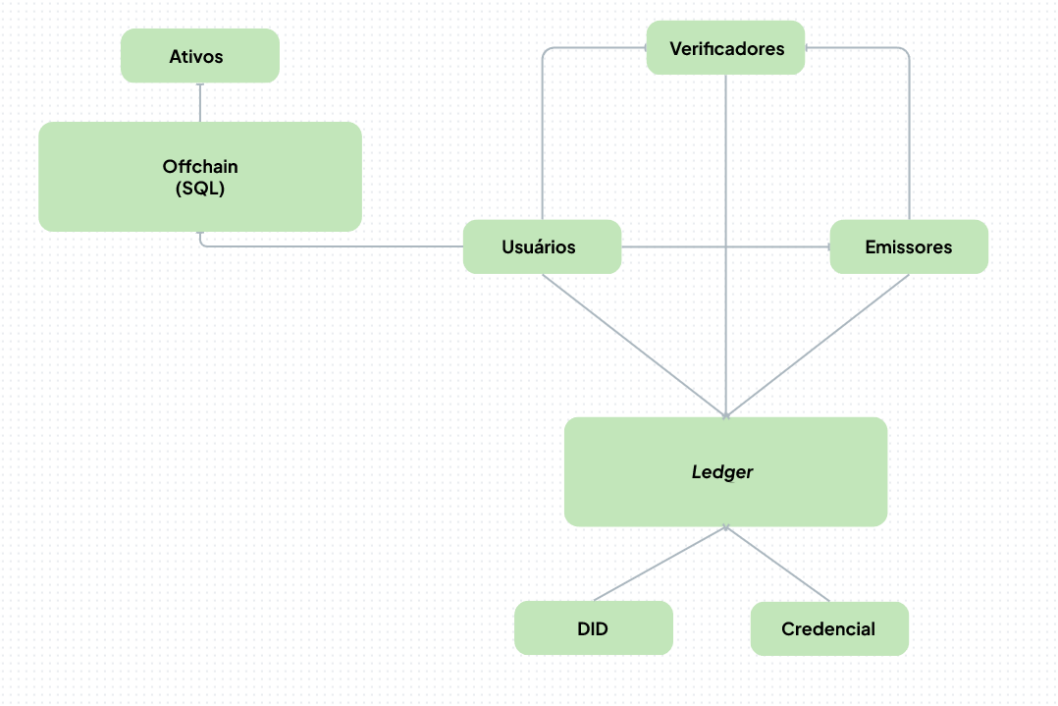


Figura 8 – Arquitetura do Hyperledger Indy

- **Identidades Digitais:** Os usuários do Hyperledger Indy são representados por identidades digitais. Cada identidade é vinculada a um agente, que atua como um intermediário para interagir com a rede. Os agentes podem ser executados em dispositivos pessoais, servidores ou outros ambientes. Essas identidades são únicas e podem conter informações pessoais, como nome, endereço, histórico educacional, entre outros. Esses dados sensíveis são armazenados em um banco de dados *offchain*, fora do *ledger*.

- ***ledger*:** O Hyperledger Indy utiliza um *ledger* distribuído, chamado de *ledger* de identidade, ele é responsável por registrar todas as transações relacionadas às identidades digitais. O *ledger* é compartilhado entre os participantes da rede, garantindo a transparência e a integridade dos dados.

- ***Smart Contracts*:** O Hyperledger Indy usa *smart contracts* específicos chamados *credential contracts* ou contratos de credencial. Esses contratos definem as regras para emissão e verificação de credenciais digitais. Eles garantem que apenas informações relevantes e verificáveis sejam incluídas nas credenciais.

- **Emissores:** Os emissores são entidades confiáveis que emitem credenciais digitais para os usuários. Elas podem ser organizações governamentais, instituições educacionais, empresas, etc. As emissoras interagem com os agentes para emitir as credenciais relevantes.

- **Provas:** O Hyperledger Indy permite a criação de provas criptográficas que permitem que os usuários compartilhem informações verificáveis sem revelar detalhes sensíveis. As provas são geradas usando credenciais e contratos de prova, permitindo que os usuários comprovem certas afirmações sobre si mesmos sem revelar todos os detalhes subjacentes.

- ***Zero-Knowledge Proofs*:** O Hyperledger Indy suporta o uso de provas de conhecimento zero, uma técnica criptográfica que permite que uma parte prove o conhecimento de um fato ou declaração, sem revelar a informação em si. Isso ajuda a preservar a privacidade dos usuários durante as transações.

- ***Anoncreds*:** A implementação do *anoncreds* no Hyperledger Indy permite que os usuários tenham controle sobre suas identidades, proteja sua privacidade e compartilhem informações verificáveis de maneira confiável. Sendo uma parte crucial do *framework*, funciona através de um sistema de criptografia e protocolos específicos. Possui três componentes principais:

  - **Emissão de Credencial:** Uma entidade emissora emite uma credencial anônima para um usuário, contendo atributos específicos.
  - **Prova Seletiva:** O usuário pode, posteriormente, apresentar essa credencial a uma parte que deseja verificar a autenticidade de seus atributos. O usuário pode escolher quais atributos compartilhar sem revelar sua identidade completa.
  - **Verificação de Prova:** A parte verificadora pode usar a credencial anônima para verificar se os atributos alegados pelo usuário são autênticos, sem saber a identidade do usuário.

#### 5.1.2.3 LibIndy Crypto

LibIndy Crypto é uma biblioteca de criptografia utilizada no ecossistema da plataforma Hyperledger Indy, sendo responsável por disponibilizar funcionalidade criptográficas essenciais, como a geração e gerenciamento de chaves criptográficas, assinaturas digitais e criptografia de dados. É usada para garantir a segurança e a privacidade das transações e interações em sistemas de identidade digital, possibilitando que os usuários mantenham o controle de suas informações pessoais de forma segura e descentralizada.

Existem outras bibliotecas e *frameworks* que podem ser utilizados em conjunto ou como alternativas à LibIndy Crypto, dependendo dos requisitos específicos do projeto. Como por exemplo, o Hyperledger Ursa que fornece uma biblioteca modular de criptografia para uso em diversos projetos Hyperledger. Entretanto optou-se pelo uso da LibIndy Crypto por ser o modelo mais atual em desenvolvimento pela Hyperledger.

#### 5.1.2.4 DID no Hyperledger Indy

No modelo do Hyperledger Indy, fazemos uso de *blockchain* para o registro de dados verificáveis, o que dá propriedades importantes de dependência forte, imutabilidade e rastreabilidade, mas não tem foco algum na busca e indexação dos dados. Para solucionar esse problema de indexação, temos o DID que é um sistema de identificação único e universal. Em outras palavras, um DID é tem um formato único, que serve como um URI para referenciar algum dado.

Comparando com o modelo SQL, no modelo SQL temos tabelas e cada elemento de uma tabela precisa de um identificador que não se repete naquela tabela, mas que pode se repetir no banco de dados como um todo. Por exemplo, podemos ter duas tabelas Carros com os IDs 1, 2, 3 e Pessoas com os IDs 1, 2, 4. Ambos têm os mesmos identificadores, mas cada um em seu contexto particular. O caso contrário é o identificador global, onde usando o mesmo exemplo, para Carros nós teríamos os DIDs 1, 4, 6 e para Pessoas os DIDs 2, 3, 5.

Com isso, o DID é uma solução para identificação e indexação no *ledger*, possibilitando a criação de fôrmas para as credencias (definições das credenciais), que seria equivalente as tabelas, e a criação de credenciais digitais de fato, que seriam equivalentes a uma elemento de uma tabela. No Indy fazemos uso e gerenciamento dos DIDs no *ledger* e nas carteiras por meio do componente did do LibIndy serve para criar uma interface que viabilize as carteiras digitais gerenciar os DIDs verificáveis.

#### 5.1.2.5 LibIndy Ledger

O Indy faz uma implementação de SSI com *blockchain*, então conta com uma arquitetura de computação distribuída e por isso precisa ter *ledgers* (log de transações) para manter o consenso entre todos os nós pareados. A arquitetura do Indy conta com 3 *ledgers* por padrão:

- **Pool ledger:** transações relacionadas à configuração do *pool*/rede (listando todos os nós, suas chaves e endereços)
- **Config ledger:** transações para configuração do *pool* mais transações relacionadas à atualização do *pool*
- **Domain ledger:** todas as transações específicas do domínio principal e do aplicativo (incluindo transações do NYM para DID)

Tendo isso em mente, o LibIndy Ledger é o componente que tem a responsabilidade de possibilitar interfaces das entidades com os *ledgers*. Um exemplo muito comum desse componente são as funções para estruturar requisições específicas de transações em um

JSON, a partir desse JSON com a requisição da transação, o próximo passo é assinar e submeter a requisição dessa transação para ser autenticada por consenso e finalmente incorporada em um dos *ledgers*.

Assim como no caso da LibIndy Crypto, existem bibliotecas alternativas que podem ser utilizadas em substituição a LibIndy Ledger, tais alternativas necessitam ser avaliadas de acordo com os requisitos do projeto. Nesse projeto em específico optou-se pela utilização da LibIndy Ledger por ela ser o projeto mais recente e desenvolvido especificamente para a utilização em SSIs.

#### 5.1.2.6 LibIndy Pool

O LibIndy Pool é um componente do Hyperledger Indy que permite a criação e manutenção de um conjunto de nós (servidores) que armazenam registros de transações e eventos relacionados a sistemas de identidade digital. Sendo esses registros compartilhados e mantidos de forma confiável, dessa maneira possibilitando que diferentes partes interajam e validem transações e atualizações na rede de forma segura e descentralizada.

A LibIndy Pool é utilizada diretamente em conjunto com a LibIndy Ledger e assim como ela, também possui bibliotecas alternativas. Possuindo motivação da sua escolha também igual.

#### 5.1.2.7 *Wallet*

A função principal da *Wallet* é armazenar de forma segura informações sensíveis, como chaves criptográficas, credenciais e outros dados relacionados à identidade do usuário. Sendo um instrumento essencial para controlar e proteger as informações de identidade digital nos sistemas baseados no Hyperledger Indy, garantindo dessa forma a privacidade e a segurança das informações pessoais dos usuários.

### 5.1.3 Aplicação de exemplo

Na documentação oficial do Hyperledge Indy, é disponibilizado um exemplo de aplicação, do qual foram feitos testes com o objetivo de verificar a consistência da ferramenta, uma vez que por ter seu lançamento recente e está na versão 1.0 poderia conter bugs que impedissem a implementação da solução. A aplicação é escrita utilizando a linguagem Python, sendo as bibliotecas e demais recursos necessários para rodar o ambiente disponibilizados em um container docker que utiliza uma imagem do Ubuntu 16.04.

No exemplo da aplicação existem cinco atores, a Alice, o Thrift Bank, a Acme Corp, a Faber College e o Governo. Sendo a Alice o usuário que interage com as organizações ao longo da história.

Entre os papéis de cada ator está o do Governo, que possui o objetivo de definir os padrões para os esquemas de credenciais. No exemplo são utilizados dois esquemas: o primeiro chamado transcrição e o segundo certificado de trabalho.

A problemática gira em torno da Alice que está se candidatando a um emprego na Acme Corp e precisa comprovar sua formação educacional, para isso ela apresenta a transcrição, que é a credencial gerada pelo Faber College para garantir que ela se formou na instituição. Após a sua contratação pela Acme Corp ela deseja solicitar um empréstimo no Thrift Bank com o intuito de realizar a compra de um carro. O banco então exige a comprovação de emprego e identidade, a Alice então apresenta o certificado de trabalho emitido pela Acme Corp, seguindo os esquema de certificado de trabalho emitido pelo Governo.

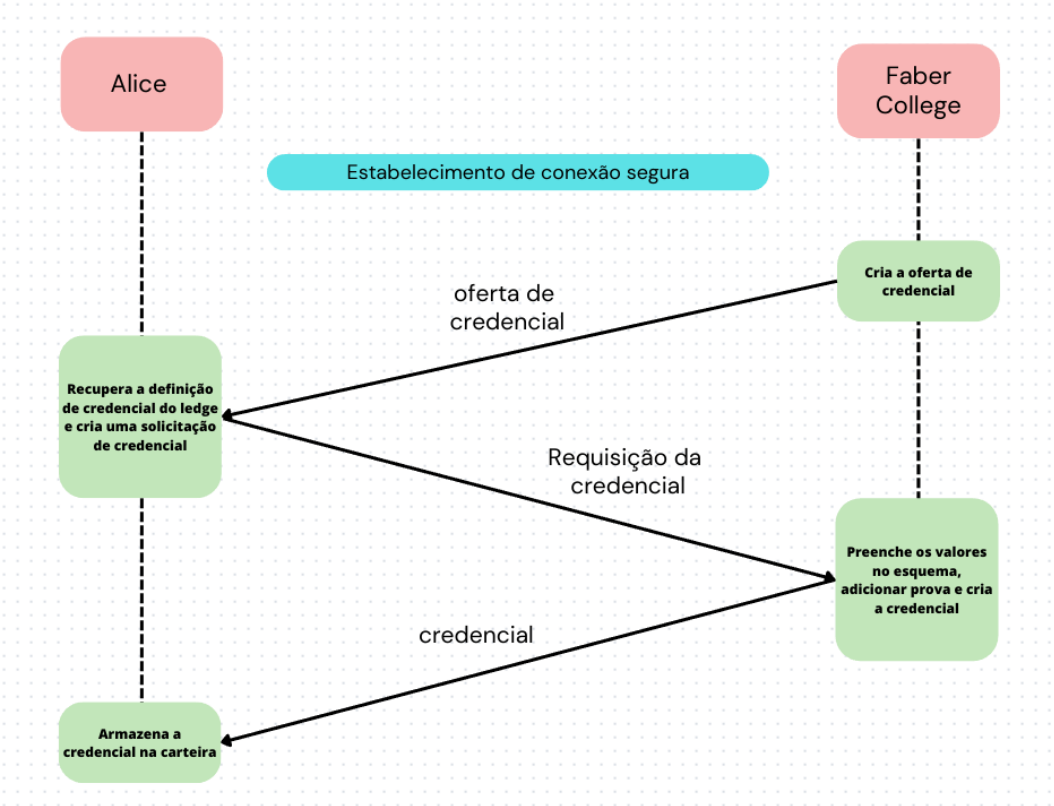


Figura 9 – Obtenção de credencial

Na Fig.9 tem um diagrama que detalha o funcionamento do processo em que Alice obtém uma credencial do Faber College, podendo ser dividido em etapas:

- **Passo 1:** O primeiro passo é a construção de uma conexão entre Faber College e Alice.

- **Passo 2:** O Faber College cria e envia uma Oferta de Credencial para Alice.

- **Passo 3:** Alice recupera a "Definição de credencial de transcrição da Faber" do registro, e então cria uma solicitação de credencial e a envia para o Faber College.
- **Passo 4:** Faber College cria a credencial para Alice. Sendo que dentro da Credencial contém os valores dos itens listados na "Definição da Credencial Faber Transcript", mais a prova necessária que Alice pode usar posteriormente quando solicitado pela Acme.
- **Passo 5:** Alice agora recebe a credencial e a guarda em sua carteira.

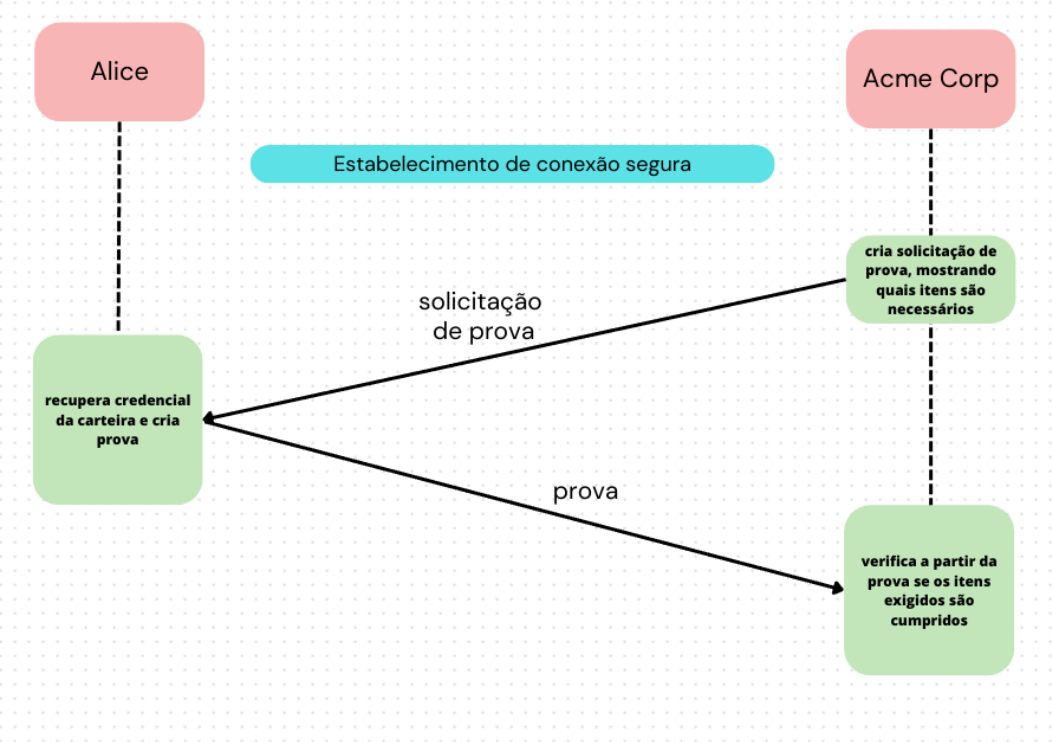


Figura 10 – Solicitação de prova

Já na Fig.10 temos o exemplo do funcionamento da solicitação de uma prova de nível educacional de Alice por parte da Acme Corp, entre os passos estão:

- **Passo 1:** Primeiro é construída uma conexão entre Acme Corp e Alice;
- **Passo 2:** A Acme Corp cria uma solicitação de prova, que lista os itens e a condição necessária. Neste caso, a Acme exige comprovação de graduação, status e ssn do Faber College, e se a média é superior a 4.
- **Passo 3:** Alice recebe essa solicitação de Prova e cria uma Prova com base na credencial que obtém do Faber College. A Prova contém informações de modo que o requisito da Solicitação de Comprovação da Acme possa ser atendido.

- **Passo 4:** A Acme Corp recebe a Prova de Alice. Dentro da Prova, a Acme Corp vê as informações e condições exigidas e verifica se elas são provenientes do Faber College.

- **Passo 5:** Por último a Acme Corp aceita esta Prova.

```
INFO:__main__:==============================
INFO:__main__:=== Getting Trust Anchor credentials for Faber, Acme, Thrift and Government  ==
INFO:__main__:------------------------------
INFO:__main__:"Sovrin Steward" -> Create wallet
INFO:__main__:"Sovrin Steward" -> Create and store in Wallet DID from seed
INFO:__main__:==============================
INFO:__main__:== Getting Trust Anchor credentials - Government Onboarding  ==
INFO:__main__:------------------------------
INFO:__main__:"Sovrin Steward" -> Create and store in Wallet "Sovrin Steward Government" DID
INFO:__main__:"Sovrin Steward" -> Send Nym to Ledger for "Sovrin Steward Government" DID
INFO:__main__:"Sovrin Steward" -> Send connection request to Government with "Sovrin Steward Government" DID and nonce
INFO:__main__:"Government" -> Create wallet
INFO:__main__:"Government" -> Create and store in Wallet "Government Sovrin Steward" DID
INFO:__main__:"Government" -> Get key for did from "Sovrin Steward" connection request
INFO:__main__:"Government" -> Anoncrypt connection response for "Sovrin Steward" with "Government Sovrin Steward" DID, verkey and nonce
INFO:__main__:"Government" -> Send anoncrypted connection response to "Sovrin Steward"
INFO:__main__:"Sovrin Steward" -> Anondecrypt connection response from "Government"
INFO:__main__:"Sovrin Steward" -> Authenticates "Government" by comparision of Nonce
INFO:__main__:"Sovrin Steward" -> Send Nym to Ledger for "Government Sovrin Steward" DID
INFO:__main__:==============================
INFO:__main__:== Getting Trust Anchor credentials - Government getting Verinym  ==
INFO:__main__:------------------------------
INFO:__main__:"Government" -> Create and store in Wallet "Government" new DID
INFO:__main__:"Government" -> Authcrypt "Government DID info" for "Sovrin Steward"
INFO:__main__:"Government" -> Send authcrypted "Government DID info" to Sovrin Steward
INFO:__main__:"Sovrin Steward" -> Authdecrypted "Government DID info" from Government
INFO:__main__:"Sovrin Steward" -> Authenticate Government by comparision of Verkeys
INFO:__main__:"Sovrin Steward" -> Send Nym to Ledger for "Government DID" with TRUST_ANCHOR Role
INFO:__main__:==============================
```

Figura 11 – Output da aplicação

```
INFO:__main__:==============================
INFO:__main__:== Getting Transcript with Faber - Getting Transcript Credential ==
INFO:__main__:------------------------------
INFO:__main__:"Faber" -> Create "Transcript" Credential Offer for Alice
INFO:__main__:"Faber" -> Get key for Alice did
INFO:__main__:"Faber" -> Authcrypt "Transcript" Credential Offer for Alice
INFO:__main__:"Faber" -> Send authcrypted "Transcript" Credential Offer to Alice
INFO:__main__:"Alice" -> Authdecrypted "Transcript" Credential Offer from Faber
INFO:__main__:"Alice" -> Create and store "Alice" Master Secret in Wallet
INFO:__main__:"Alice" -> Get "Faber Transcript" Credential Definition from Ledger
INFO:__main__:"Alice" -> Create "Transcript" Credential Request for Faber
INFO:__main__:"Alice" -> Authcrypt "Transcript" Credential Request for Faber
INFO:__main__:"Alice" -> Send authcrypted "Transcript" Credential Request to Faber
INFO:__main__:"Faber" -> Authdecrypt "Transcript" Credential Request from Alice
INFO:__main__:"Faber" -> Create "Transcript" Credential for Alice
INFO:__main__:"Faber" -> Authcrypt "Transcript" Credential for Alice
INFO:__main__:"Faber" -> Send authcrypted "Transcript" Credential to Alice
INFO:__main__:"Alice" -> Authdecrypted "Transcript" Credential from Faber
INFO:__main__:"Alice" -> Store "Transcript" Credential from Faber
INFO:__main__:==============================
```

Figura 12 – Output da aplicação

Nas Fig.11 e Fig.12 são demonstrados os logs da execução do algoritmo em um terminal linux, onde são registradas as conclusões das etapas apresentadas anteriormente.

#### 5.1.3.1 Diferenças em relação ao modelo tradicional

Como podemos ver no exemplo citado acima, uma das principais diferenças de uma aplicação que utiliza como base a identidade auto soberana, está na forma como os certificados são emitidos e armazenados. Uma vez que na SSI os certificados são emitidos e controlados pelos próprios indivíduos. Cada usuário mantém seus certificados em

uma carteira digital, proporcionando propriedade direta e controle sobre suas credenciais. Já no modelo tradicional, baseado em CAs, a emissão e controle dos certificados são centralizados. A autoridade certificadora emite e gerencia os certificados em nome dos usuários.

Quanto ao armazenamento, na SSI os certificados são distribuídos em uma rede *peer-to-peer*. Não existindo uma entidade central que detém o controle. No modelo tradicional existe a dependência de uma autoridade certificadora central para emissão, validação e revogação de certificados. Esta autoridade é um ponto único de controle.

Outro ponto perceptível é que a SSI permite uma verificação seletiva, onde os usuários compartilham apenas as informações necessárias para uma finalidade específica, aumentando a privacidade. Além do fato da autenticação poder ser realizada sem a necessidade de uma autoridade central. No modelo com utilização de CAs, a verificação e autenticação são geralmente realizadas por meio dos certificados digitais emitidos pela autoridade certificadora. A validação depende da confiança na autoridade central.

Outro ponto que vale ser mencionado é a questão da interoperabilidade, uma vez que a SSI favorece seu uso entre diferentes sistemas, através de padrões baseados em protocolos abertos, que possibilitam a implementação em várias plataformas. Os modelos baseados em CAs possuem grande fragilidade nesse quesito, especialmente quando diferentes autoridades certificadoras utilizam padrões ou tecnologias distintas.

### 5.1.4 Justificativa do uso de SSI

Como foi citado ao longo do texto, a SSI possui algumas características marcantes, que possibilita aos seus usuários maior autonomia sobre suas próprias identidades digitais. O que contrasta com os modelos centralizados, onde as autoridades certificadoras detêm o controle. Outro ponto positivo em relação ao uso de SSI, está na segurança. Tendo em vista que modelos centralizados são suscetíveis a ataques, uma vez que um único ponto de falha pode comprometer todo o sistema. Diferente do modelo utilizado pela identidade auto soberana, que realiza a distribuição das informações de identidade em uma rede descentralizada, eliminando o risco associado a servidores centralizados. Essa descentralização reduz significativamente a atratividade para ataques maliciosos, uma vez que não existe um único ponto vulnerável para explorar.

Outro ponto em que a SSI se destaca, é na questão de eficiência e agilidade nas transações, uma vez que todo o processo de verificação de identidade é simplificado, eliminando a necessidade de intermediários. Em modelos centralizados, a validação por autoridades certificadoras muitas vezes envolve procedimentos burocráticos e demorados. Com SSI, a verificação pode ser realizada de forma mais rápida e eficiente, melhorando a experiência do usuário e acelerando as transações.

Por fim, existe ainda o fato da SSI promover maior interoperabilidade entre diferentes sistemas e serviços, permitindo que as identidades sejam verificadas de maneira eficiente em vários contextos. Em contrapartida, modelos centralizados muitas vezes enfrentam desafios de interoperabilidade e podem resultar em custos adicionais para a integração de sistemas heterogêneos.

Diante das vantagens apresentadas, se torna evidente que sistemas de identidade descentralizadas representam uma evolução significativa em relação aos modelos tradicionais. A autonomia do indivíduo, a redução de riscos de segurança, a eficiência nas transações e a interoperabilidade destacam-se como elementos-chave que posicionam as SSI como uma solução mais robusta e adaptável para os desafios contemporâneos relacionados à identidade digital.

# 6 Prova de conceito

A área da saúde enfrenta desafios significativos quando se trata do gerenciamento de informações e dados sensíveis dos pacientes. A falta de interoperabilidade entre sistemas, a preocupação com a privacidade dos dados e a necessidade de autenticação segura são apenas algumas das questões que precisam ser abordadas para melhorar a eficiência e a segurança do setor.

Nesse contexto, o uso da SSI em uma aplicação de saúde pode trazer inúmeras vantagens e resolver muitos dos problemas existentes. A identidade auto soberana permite que os pacientes tenham controle total sobre seus próprios dados de saúde, garantindo privacidade, segurança e confiança na troca de informações.

Ao desenvolver uma aplicação de saúde utilizando SSI, é possível criar um ecossistema em que os pacientes possam armazenar suas informações médicas de forma criptografada. Esses dados podem incluir histórico médico, resultados de exames, alergias, medicações prescritas e outros detalhes relevantes.

Com seu uso, os pacientes têm a liberdade de compartilhar seletivamente suas informações de saúde com profissionais médicos e instituições de saúde, evitando a necessidade de preencher formulários repetidamente e permitindo uma colaboração mais eficiente entre os diferentes prestadores de cuidados.

Além disso, a verificação descentralizada permite a autenticação segura dos pacientes e dos profissionais de saúde. Em vez de depender de identidades centralizadas, os usuários podem utilizar suas próprias credenciais criptográficas para comprovar sua identidade de forma confiável.

Podendo ainda ajudar a mitigar problemas de segurança, uma vez que os dados de saúde não estão concentrados em um único sistema suscetível a ataques. A criptografia e a descentralização oferecidas pelo SSI proporcionam um nível adicional de proteção aos dados confidenciais do paciente.

No entanto, é importante considerar os desafios associados à implementação de SSI em uma aplicação de saúde. Questões como padronização, conformidade com regulamentações de privacidade e a necessidade de conscientização e aceitação por parte dos pacientes e profissionais de saúde podem representar obstáculos que precisam ser superados.

Em suma, uma aplicação de saúde desenvolvida com uso de SSI tem o potencial de revolucionar a forma como os dados de saúde são gerenciados, oferecendo segurança, privacidade e controle aos pacientes. Ao capacitar os indivíduos a controlar suas próprias

identidades digitais e informações médicas, a SSI pode melhorar a colaboração entre os prestadores de cuidados, facilitar o compartilhamento seguro de dados e aprimorar a experiência do paciente no setor da saúde.

Tendo em vista toda essa problemática em torno da área, a construção da aplicação nesse contexto se torna uma excelente forma de trabalhar nos diferentes conceitos relacionados a SSI, assim como ajudar a desmistificar as dificuldades do seu uso, visando como objetivo ajudar programadores que possuam interesse em construir aplicações utilizando essa tecnologia.

## 6.1 Tecnologias utilizadas

### 6.1.1 Kotlin

Kotlin é uma linguagem de programação executada sobre a máquina virtual Java. A linguagem é desenvolvida pela empresa JetBrains conhecida pela criação do IDE InteliJIDEA, amplamente usado para desenvolvimento de aplicações Java (MOSKALA; WOJDA, 2017). O Kotlin é amplamente utilizado no desenvolvimento de aplicativos Android, além de encontrar aplicações em várias outras áreas de desenvolvimento de software, disponibilizando uma alternativa à linguagem Java.

### 6.1.2 React

O React.js é um biblioteca JavaScript para desenvolvimento de IUs através da utilização de componentes. O React.js criado por Jordan Walke, então funcionário do Facebook, e foi publicado como ferramenta Open Source em 2013, na JSConf US (Tom Occhino, Jordan Walke, 2013). O React.js permite que os desenvolvedores construam interfaces responsivas e eficientes, atualizando apenas as partes necessárias da página quando os dados mudam. Com uma vasta comunidade de desenvolvedores e um ecossistema rico em ferramentas e bibliotecas, o React.js é bastante popular para o desenvolvimento de aplicações web.

### 6.1.3 Python

Python é uma linguagem de altíssimo nível orientada a objetos, de tipagem dinâmica e forte, interpretada e interativa (BORGES, 2014). Sendo considerada versátil e podendo ser utilizada em diversas aplicações, desde desenvolvimento web e científico até automação de tarefas e inteligência artificial. Possui uma sintaxe intuitiva e uma extensa biblioteca, tornando-se uma das linguagens mais populares. Sendo utilizado no *backend* da aplicação para a criação da API.

### 6.1.4 PostgreSQL

O PostgreSQL é um Sistema Gerenciador de Banco de Dados (SGBD) Relacional, utilizado para armazenar informações de informática em todas as áreas de negócios existentes, bem como administrar o acesso a estas informações (MILANI, 2008). Com sua flexibilidade, desempenho e suporte ativo da comunidade, o PostgreSQL é amplamente utilizado em diversas aplicações. Foi utilizado como o banco de dados *offchain* do projeto.

### 6.1.5 Docker

Docker é uma plataforma aberta, criada com o objetivo de facilitar o desenvolvimento, a implantação e a execução de aplicações em ambientes isolados. Foi desenhada especialmente para disponibilizar uma aplicação da forma mais rápida possível (GOMES, 2019). O Docker permite aos desenvolvedores empacotar seus aplicativos em contêineres consistentes e portáteis, garantindo que eles funcionem de maneira confiável em diferentes ambientes, desde o desenvolvimento até a produção. Facilitando a implantação e o gerenciamento de aplicativos. Durante o desenvolvimento do projeto foram criados três containers de micro serviços para a execução da aplicação, um para o *frontend*, um para o *backend* e outro para o banco de dados *offchain*.

## 6.2 Aplicação no modelo convencional

Provavelmente para o desenvolvimento de uma aplicação com os mesmo requisitos e que não pudesse utilizar tecnologias descentralizadas, optaria pela abordagem de utilização de identidade federada junto de uma arquitetura baseada em microserviços. Possibilitando ao usuário não precisar criar diferentes contas em diferentes estabelecimentos, obtendo um resultado semelhante ao diminuir a sobrecarga de gerenciamento de contas e senhas, ao mesmo tempo em que garante um processo eficaz de autenticação.

Seguindo o modelo adotado, a aplicação seria dividida em dois grandes blocos de serviços, o *frontend* e o *backend*. Sendo o *frontend* responsável pela interface que terá a interação direta com o usuário, fornecendo telas de login, cadastro e exibição do histórico de consultas e exames realizados pelo usuário.

Já o *backend* fica responsável pela lógica de negócios, processamento de dados e integração com possíveis serviços externos, recebendo os dados do *frontend*, fornecendo e validando o token de acesso, além de disponibilizar os dados que seriam salvos em um banco de dados, tudo por meio de uma API.

## 6.3 Adaptando a aplicação ao contexto de identidade auto soberana

### 6.3.1 Arquitetura

No desenvolvimento da aplicação, como demonstrado na Fig. 13 foi utilizado o conceito de microserviços, que se trata de um estilo de arquitetura de software em que a aplicação é dividida em componentes independentes e autônomos, conhecidos como microserviços. Cada um deles é responsável por uma única função específica e efetua a comunicação com os demais por meio de APIs (interfaces de programação de aplicativos).

Diferente da abordagem monolítica, em que um aplicativo é desenvolvido como um bloco único de código, os microserviços permitem que cada componente seja desenvolvido e implantado de forma independente. Essa abordagem traz diversos benefícios, como a capacidade de escalar cada microserviço individualmente, melhorar a manutenção e a evolução do sistema, além de facilitar a adoção de tecnologias diferentes para cada componente.

Cada microserviço pode ser desenvolvido usando a tecnologia mais adequada para sua função específica, o que permite escolher a linguagem de programação, o banco de dados e as ferramentas mais adequadas para cada caso. No projeto em questão foram utilizados três microserviços, o *frontend*, o *backend* e o banco de dados, sendo ainda adicionados em containers Docker.

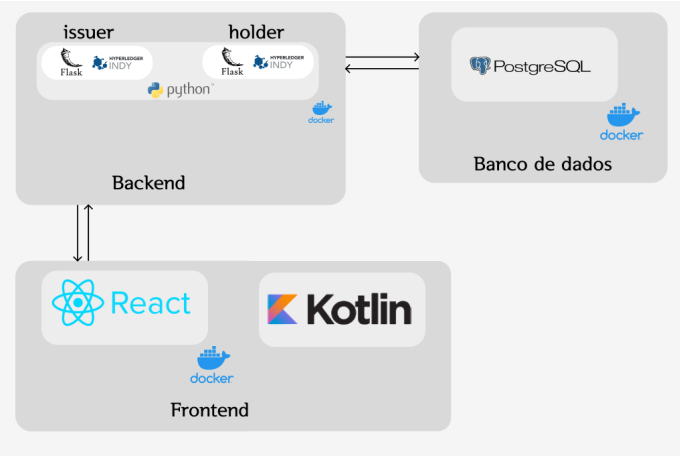


Figura 13 – Arquitetura da aplicação

#### 6.3.1.1 *Frontend*

O *frontend web* da aplicação tem como objetivo fornecer uma interface amigável e intuitiva para os as instituições de saúde que farão uso do sistema, permitindo que eles interajam com a aplicação e acessem seus recursos de forma eficiente e segura.

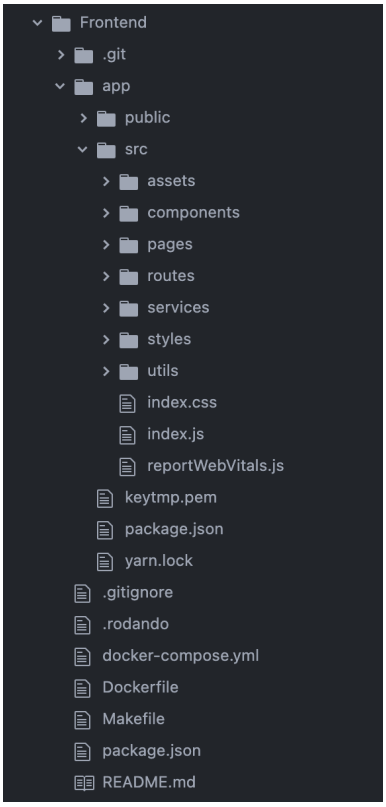


Figura 14 – Diretório do *frontend web* da aplicação

Utiliza o ReactJS como *framework* de desenvolvimento, sendo criados componentes reutilizáveis além de ter sido construído utilizando uma interface responsiva, adaptável a diferentes dispositivos e tamanhos de tela.

Para garantir a segurança dos dados pessoais e a privacidade dos usuários, o *frontend* da aplicação utiliza criptografia e protocolos de segurança para a transmissão e armazenamento de informações sensíveis.

No que diz respeito à estruturação dos diretórios, é demonstrado na Fig. 14 o modelo que adotamos. Na pasta principal tem-se os arquivos responsáveis pela configuração do container Docker, o arquivo contendo a lista de bibliotecas externas utilizadas e os arquivos referentes ao Git. No diretório "app/src", estão os arquivos principais da aplicação, sendo que no diretório "*assets*"estão contidos os ativos utilizados na aplicação, como imagens e fontes. No diretório "*components*", estão contidos os componentes utilizados para compor as páginas do aplicativo. No diretório "*pages*", se encontram as páginas disponíveis na aplicação. Em "*routes*"se encontram os arquivos referentes às rotas da aplicação, basicamente contém dois arquivos, o primeiro com a indexação das rotas e o segundo responsável por realizar a configuração de rotas protegidas, que são rotas que só podem ser efetuadas após a realização do login. No diretório "*services*"podem ser encontrados os arquivos com as configurações para comunicação entre a aplicação e serviços externos como o *backend*. Em "*utils*"se encontram arquivos com trechos de códigos que são

utilizados de formas repetidas. Por fim, no diretório "*styles*", estão contidos os arquivos com as folhas de estilo (CSS).

#### 6.3.1.2 Aplicativo móvel

A aplicação Android foi desenvolvida em Kotlin utilizando o Indy SDK, ou Hyperledger Indy Software Development Kit, que se caracteriza como uma biblioteca de desenvolvimento de software open-source projetada para facilitar a criação de sistemas de identidade descentralizada e agentes de confiança em ambientes distribuídos. Faz parte do projeto Hyperledger e é particularmente conhecido por fornecer ferramentas e protocolos para a criação de identidades digitais auto soberanas e seguras, sem depender de uma autoridade central.

O Indy SDK também oferece alguns recursos no que diz respeito à gestão de identidades digitais, incluindo a capacidade de criar e armazenar DIDs, emitir credenciais verificáveis e realizar transações de forma segura e descentralizada. Utiliza uma arquitetura modular e foi projetado para permitir interoperabilidade, possibilitando a integração com outras tecnologias e blockchains. Na Fig. 15 é demonstrado os diretórios de arquivos utilizados na aplicação.

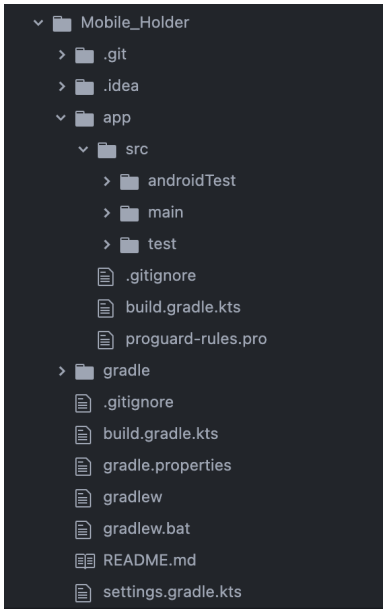


Figura 15 – Diretório do aplicativo móvel

#### 6.3.1.3 *Backend*

O *backend* da aplicação foi desenvolvido em Python e utiliza dois conjuntos de bibliotecas chave: o conjunto de bibliotecas do Hyperledger Indy e o Flask. O Hyperledger Indy como já citado é um *framework* de *blockchain* voltado para identidade auto soberana, enquanto o Flask é um *framework* leve de desenvolvimento web em Python.

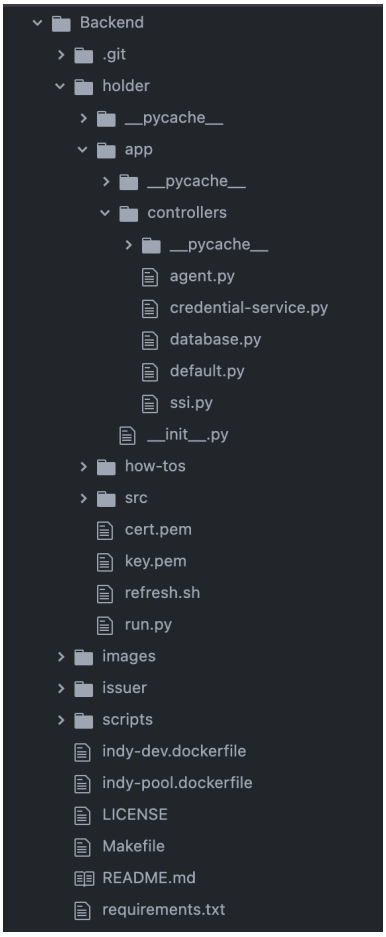


Figura 16 – Diretório do *backend* da aplicação

A aplicação tem como objetivo fornecer um sistema seguro e descentralizado para o gerenciamento de identidades e registros médicos dos pacientes. O Hyperledger Indy é utilizado para criar uma rede *blockchain* privada, na qual os usuários podem manter o controle sobre suas próprias identidades e dados de saúde.

O Flask é usado para desenvolver a camada de interface web da aplicação, permitindo que os usuários acessem e interajam com os recursos disponíveis. O *framework* Flask é conhecido por sua simplicidade e flexibilidade, tornando-o uma escolha adequada para o desenvolvimento ágil de uma API.

A aplicação disponibiliza algumas funcionalidades importantes para a área da saúde. Os usuários podem criar e gerenciar suas identidades digitais auto soberana, armazenando suas informações pessoais e de saúde de forma criptografada no Hyperledger Indy. Eles podem controlar o acesso a essas informações, compartilhando seletivamente com médicos e instituições de saúde conforme necessário.

Além disso, a aplicação permite que os usuários visualizem e atualizem seus registros médicos, sendo esses registros armazenados na rede *blockchain* privada garantindo a integridade e a segurança dos dados.

O Flask é responsável por fornecer as rotas e os endpoints necessários para a interação com a aplicação. Por exemplo, os usuários podem se registrar, fazer login e acessar suas informações por meio das rotas criadas com Flask. O *framework* também facilita a implementação de autenticação e autorização para proteger os dados e garantir que apenas os usuários autorizados possam acessá-los.

Na Fig. 16 é demonstrada a estruturação dos diretórios do *backend* da aplicação, sendo adicionados no diretório principal os arquivos de configuração dos containers Docker, do Git, além do arquivo com a descrição das bibliotecas utilizadas. No diretório "*holder*"são encontrados os arquivos responsáveis pelos *scripts* dos usuários.

#### 6.3.1.4 Banco de dados *offchain*

Enquanto a SSI utiliza a *blockchain* para armazenar registros e transações imutáveis, um banco de dados *offchain* complementa essa abordagem, fornecendo uma solução que visa realizar o armazenamento de demais dados relacionados à aplicação.

Sendo o PostgreSQL um sistema de gerenciamento de banco de dados relacional amplamente utilizado, conhecido por sua confiabilidade, escalabilidade e recursos avançados. Oferecendo ainda suporte a recursos como ACID (Atomicidade, Consistência, Isolamento e Durabilidade), consultas complexas e recursos de segurança robustos.

Nesse contexto, foi escolhido para ser utilizado como o banco de dados *offchain*. A principal importância de um banco de dados *offchain* em aplicações SSI, é o armazenamento de alguns dados sensíveis, uma vez que que na *blockchain* em si são armazenados apenas os *hashes* dos dados, logo os dados reais são armazenados de forma criptografada em um banco de dados fora do *ledger*.

Outra vantagem do seu uso é a capacidade de realizar *backups* regulares dos dados, garantindo a recuperação em caso de falhas ou perdas. Esses *backups* podem ser automatizados e configurados de acordo com as políticas de retenção de dados da aplicação.

Na Fig. 17 é demonstrada a organização dos diretórios do banco de dados. Sendo que no diretório principal, assim como nos demais, são encontrados os arquivos de confi-

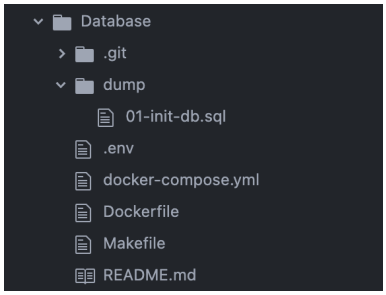


Figura 17 – Diretório do banco de dados da aplicação

guração do container Docker e do Git, já no diretório "*dump*", se encontra o arquivos com as configurações do banco dados e também alguns dados para o carregamento inicial do banco.

### 6.3.2 Fluxo de uso do sistema

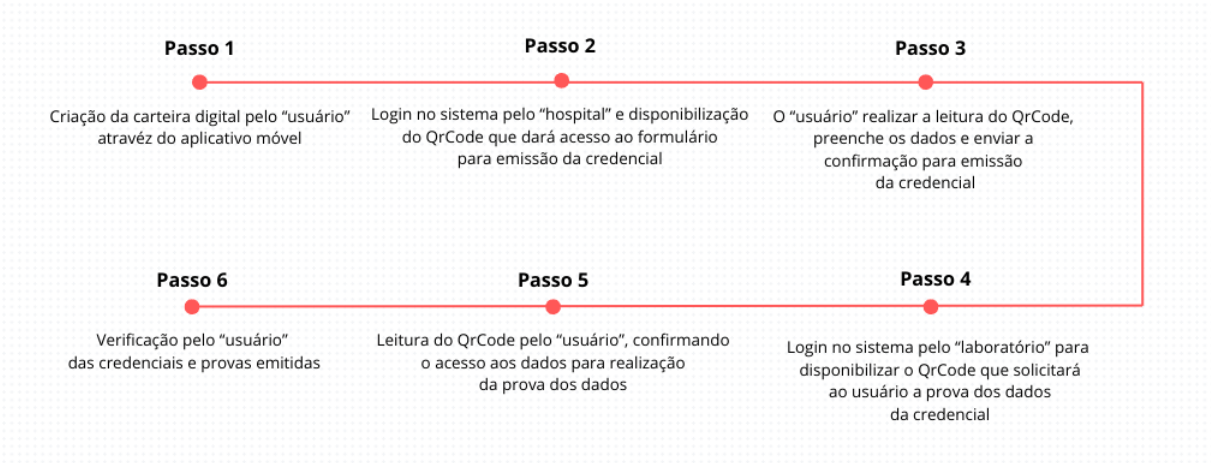


Figura 18 – Etapas da jornada do usuário

A Fig. 18 demonstra os principais passos percorridos pelos usuários do sistema para obtenção/emissão das credenciais e fornecimento das provas. O sistema conta com três perfis de usuários, sendo que o "usuário"descrito na figura representa os pacientes, ou seja, o cidadão comum que fará uso do aplicativo para acessar os serviços médicos. O outro tipo de usuário são os "hospitais"que representa grandes empresas da área de saúde, fornecendo o atendimento aos pacientes e necessitando de dados como fichas cadastrais. No contexto da SSI, os "hospitais"se classificam como emissores e sua principal função é emitir credenciais para os "usuários". Por fim, o último tipo de usuário do sistema são os "laboratórios", que no contexto da SSI, se classificam como verificadores, sendo eles que farão uso das informações das credenciais, no contexto da aplicação, eles representam subdepartamentos de hospitais que necessitam verificar a veracidade das informações apresentadas pelos pacientes.

O primeiro passo do fluxo é feito pelo "usuário"no momento em que ele inicia o uso da aplicação móvel, efetuando seu registro (demonstrado na Fig. 19) e consequentemente criando sua carteira digital na *blockchain*. Os dados cadastrados por esse usuário são salvos tanto localmente de forma criptografada, usando uma criptografia de chave simétrica que implementa o algoritmo AES e utiliza como cifra a própria senha do usuário, quanto no banco de dados *offchain*, além de terem seus identificadores descentralizados adicionados no *ledger*.

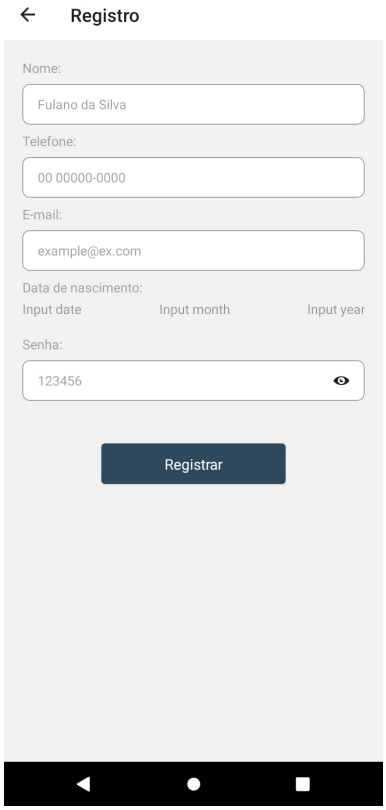


Figura 19 – Tela de registro da aplicação

```
logger.info("=============================")
logger.info("== Empresa Credential Definition Setup ==")
logger.info("-----------------------------")

logger.info("\"Empresa\" -> Get from Ledger \"Ficha-cadastral\" Schema")
(_, ficha_cadastral_schema) = await get_schema(pool_handle, empresa_did, ficha_cadastral_schema_id)

logger.info("\"Empresa\" -> Create and store in Wallet \"Empresa Ficha-cadastral\" Credential Definition")
(empresa_ficha_cadastral_cred_def_id, empresa_ficha_cadastral_cred_def_json) = \
    await anoncreds.issuer_create_and_store_credential_def(empresa_wallet, empresa_did, ficha_cadastral_schema,
                                                           'TAG1', 'CL', '{"support_revocation": false}')

logger.info("\"Empresa\" -> Send \"Empresa Ficha-cadastral\" Credential Definition to Ledger")
await send_cred_def(pool_handle, empresa_wallet, empresa_did, empresa_ficha_cadastral_cred_def_json)
```

Figura 20 – Adição do esquema na *blockchain*

Após o processo de criação da carteira digital pelo usuário, é a vez dos "hospitais"no passo 2, eles já dispõem de um esquema de credencial previamente adicionado na *blockchain*, sendo demonstrado pelo trecho de código representado na Fig. 20, tal esquema solicita dados que segue os padrões disponibilizados pela *Fast Healthcare Interoperability Resources* (FHIR), um padrão desenvolvido pela *Health Level Seven* (HL7) na qual são citados os dados essenciais em fichas médicas de pacientes dos sistemas de saúde. Nesse momento se inicia o passo 3 e também o processo de obtenção da credencial por parte do usuário, onde por meio do QrCode gerado no passo anterior o "usuário"obtém acesso ao formulário para envio das informações para emissão da credencial por parte do "hospital". A tela onde fica disponibilizado o QrCode pode ser observada na Fig. 21.


Figura 21 – Tela da aplicação com QrCode para emissão da credencial

Após a leitura do QrCode é aberto o formulário para o usuário, exibindo quais dos seus dados serão utilizados para a emissão da credencial, o usuário então confirma a emissão. A resposta retorna a empresa e por fim a credencial é emitida na *blockchain* e salva na carteira digital do usuário. Na Fig. 22 é possível observar a tela de confirmação dos dados pelo "usuário"para a emissão da credencial. Já na Fig. 23 é possível ver a demonstração do JSON utilizado para emissão da credencial na *blockchain*. Um ponto importante a ser esclarecido é que uma credencial utilizada por um determinado hospital ou instituição de saúde pode também ser utilizada por outras instituições desde que elas adotem os mesmos modelos de dados.


Figura 22 – Tela de leitura do QrCode

Outro ponto importante da SSI são as provas, onde após um usuário já ter obtido uma credencial, certos estabelecimentos podem solicitar a comprovação desse vínculo. Tal

```
schema_id = await steward.new_schema('RegistroPaciente',
                ['name', 'phone', 'gender', \
                'dateOfBirth', 'address', 'maritalStatus', \
                'multipleBirth', 'contactRelationship', 'contactName', \
                'contactPhone', 'contactAddress', 'contactGender', \
                'languages', 'preferredLanguage', 'generalPractitioner',])

prover = Holder()
await prover.create(pool_handle, json.dumps({"id": "prover_wallet"}), json.dumps({"key": "prover_wallet_key"}))

cred_def_id = await issuer.new_cred_def(schema_id)
cred_offer_json = await issuer.new_cred_offer(cred_def_id)

(cred_req_json, cred_req_metadata_json) = await prover.offer_to_cred_request(cred_offer_json, cred_def_id)

cred_values_json = json.dumps({
    'name': {'raw': 'matheus', 'encoded': '12345'}, 'phone': {'raw': '61912341234', 'encoded': '12345'}, 'gender': {'raw':
    'dateOfBirth': {'raw': '01011999', 'encoded': '12345'}, 'address':{'raw': 'Brasilia', 'encoded': '12345'}, 'maritalSta
    'multipleBirth': {'raw': '0', 'encoded': '12345'}, 'contactRelationship': {'raw': 'a', 'encoded': '12345'}, 'contactNa
    'contactPhone': {'raw': '61901011010', 'encoded': '12345'}, 'contactAddress': {'raw': 'Brasilia', 'encoded': '12345'},
    'languages': {'raw': 'pt', 'encoded': '12345'}, 'preferredLanguage': {'raw': 'pt', 'encoded': '12345'}, 'generalPracti
})
```

Figura 23 – Código com estrutura para adição da credencial na *blockchain*

confirmação é obtida através das provas.

O passo 4 e 5 trata exatamente desse ponto, onde na aba "provas"da aplicação web, demonstrado na Fig. 24 é possível selecionar uma credencial e indicar quais campos são exigidos para determinado processo, é então gerado um QrCode que direciona o usuário a um formulário que contém apenas os dados solicitados, ficando ele dispensado de apresentar todos os seus dados. Nesse ponto também entra a questão das provas de conhecimento zero, onde por exemplo na idade não é obrigatório exigir a idade atual do usuário, é possível solicitar apenas a verificação se ela é superior ou inferior a certo valor. Por fim, confirmando a solicitação, os dados são provados e então a empresa é notificada da veracidade dos dados. Por fim, no último passo é demonstrado a possibilidade do "usuário"verificar uma lista com suas credenciais e histórico de provas emitidas.


Figura 24 – Tela para seleção dos dados exigidos pela prova

Na Fig. 25 é demonstrado de forma rápida o fluxo de uso da aplicação. Como citado anteriormente, pode-se observar que a jornada do "usuário"se inicia na autenticação. Sendo disponibilizada a opção de solicitar uma nova credencial ou provar uma credencial já existente. No fluxo do aplicativo *web*, que no contexto da aplicação é voltado para utilização pelas instituições de saúde, a jornada também se inicia com o login na plataforma, logo após são disponibilizadas as opções de emitir a credencial para algum usuário ou solicitar provas a depender do perfil logado.

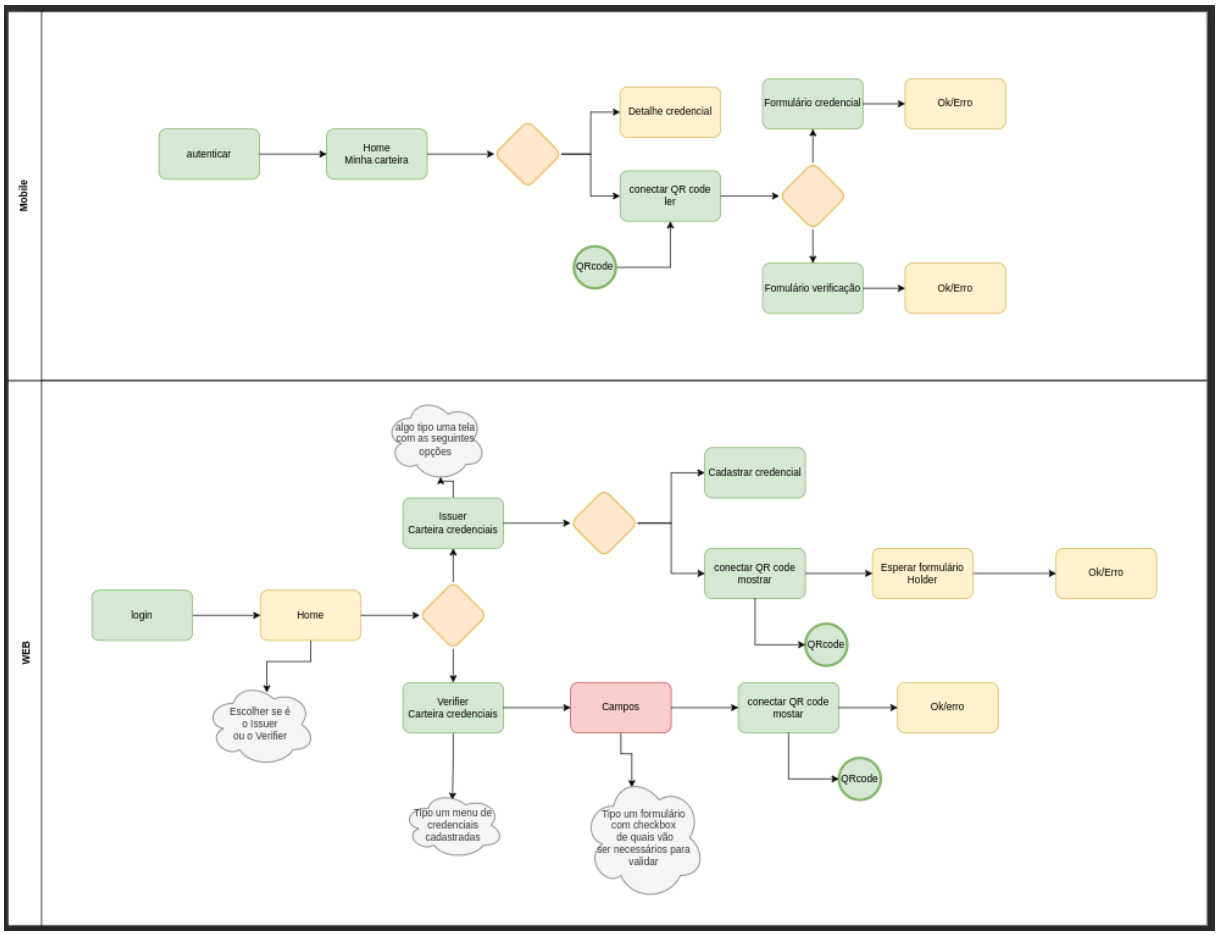


Figura 25 – Fluxo de uso da aplicação

Nas figuras 26 e 27, são demonstrados diagramas das requisições em relação ao tempo do conteúdo explicado anteriormente, sendo que a Fig. 26 representa o processo de emissão de uma nova credencial e a Fig. 27 a solicitação de prova pelas empresas.

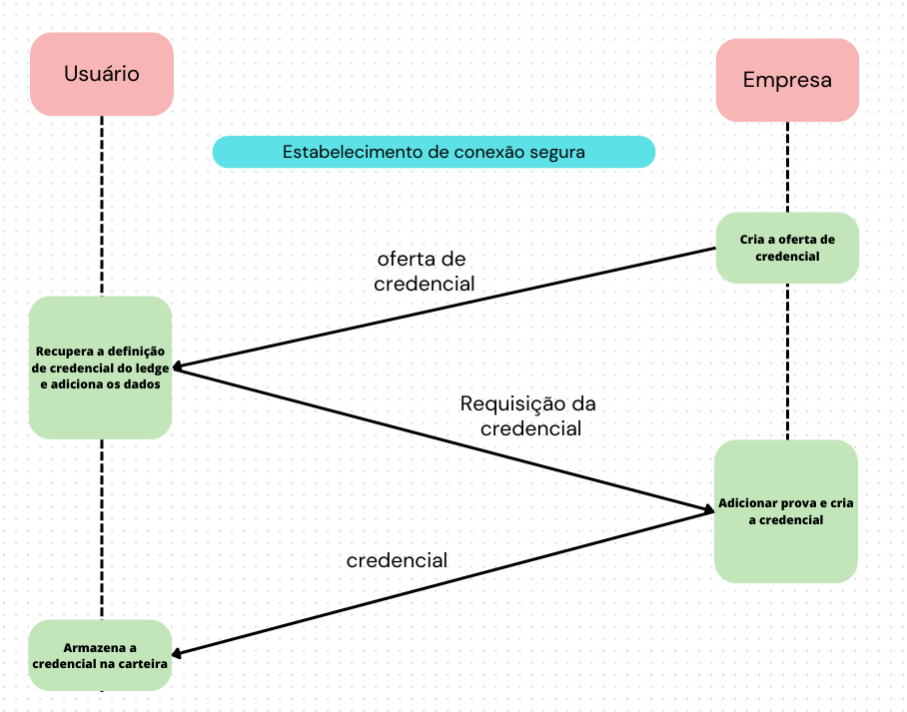


Figura 26 – Diagrama de emissão de credencial

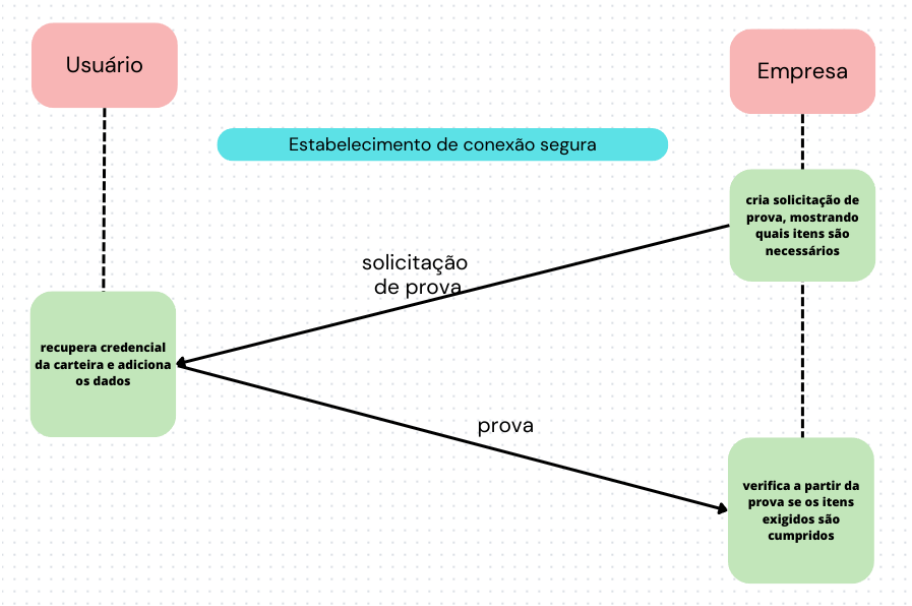


Figura 27 – Diagrama de solicitação de prova

## 6.4 Desafios encontrados

No contexto atual, as principais dificuldades encontradas durante a execução do trabalho estão relacionadas à escassez de documentação disponível para as principais tecnologias. O Hyperledger Indy, por exemplo, apresenta documentação apenas sobre a

arquitetura de solução utilizada em cada um de seus módulos, além de um guia para casos de uso simples de sistemas de SSI. Contudo, este guia se limita a tópicos básicos, como iniciar o *pool de ledgers*, realizar o *onboarding* de um emissor no *ledger*, adicionar uma definição de credencial (cred-def) à rede, solicitar uma credencial ao emissor conforme a definição estabelecida, criar uma credencial verificável, elaborar uma solicitação de verificação de credenciais e como responder à verificação com os dados solicitados.

Embora esses exemplos proporcionem uma compreensão geral, são apenas esboços simplificados, uma vez que o guia condensa todo o código em um único bloco, sem abordar aspectos essenciais, como o uso da biblioteca de criptografia do Indy para garantir a comunicação entre pares, a separação de códigos em agentes com comunicação entre eles, exemplos aplicáveis a tecnologias móveis e métodos de armazenamento e recuperação de dados da carteira digital. A ausência dessas descrições implica que, para desenvolver uma aplicação SSI que efetivamente utilize a arquitetura do Hyperledger Indy, é necessário consultar o código-fonte do Indy SDK, que, felizmente, está bem documentado em termos de entradas e saídas esperadas para cada função.

A dificuldade subsequente reside no desenvolvimento de uma comunicação de *onboarding* mais robusta do que o exemplo fornecido, que não utiliza criptografia. Nossa solução para este problema envolveu a incorporação de QrCode em cada agente, buscando garantir que ambas as máquinas estejam no mesmo espaço físico durante a conexão, transferindo assim parte da responsabilidade de autenticação da comunicação para os participantes humanos. Isso se justifica pela necessidade de criar credenciais dentro do contexto específico de uma instituição, como um hospital, e o uso do padrão adotado por essa instituição.

Adicionalmente, o gerenciamento seguro de chaves e DIDs representou um ponto crítico, exigindo uma implementação robusta para armazenar e proteger as chaves criptográficas nos dispositivos móveis. A atenção dedicada à integração com o *backend* também foi crucial, exigindo uma configuração sólida para a troca segura de dados via API, além de considerações sobre privacidade para garantir que as informações do usuário sejam armazenadas e compartilhadas de maneira consentida e segura.

Outra significativa complicação enfrentada, foi durante o processo de integração com o Indy SDK para o desenvolvimento da aplicação Android. A versão atual do Hyperledger Indy apresenta desafios na geração do kit de desenvolvimento para a plataforma Android. Para contornar esse obstáculo, foram realizadas tentativas diversas, explorando iniciativas que utilizavam versões anteriores do Indy SDK para Android e outras que empregavam o Hyperledger Aries. A solução foi alcançada ao adotar a versão que incorporava o Indy SDK para Java em conjunto com alguns pacotes de Java Native Access (JNA).

Após essa adaptação, as principais funcionalidades do Indy SDK foram efetivamente implementadas, contudo, as funções relacionadas ao acesso aos *ledgers* não estavam

operacionais, o que era esperado, dada a restrição de acesso exclusivo aos *Issuers* e *Verifiers*. Nesse contexto, foi necessário transferir para o *Issuer* as responsabilidades associadas à leitura do *ledger* que estavam sendo gerenciadas pela implementação do *Holder*, como a leitura de um DID ou dos *schemas*. Com todas as devidas adaptações e configurações nas dependências realizadas, obteve-se, finalmente, um Agente funcional em um dispositivo móvel pronto para utilização.

Subsequentemente, foram enfrentados desafios significativos relacionados ao desenvolvimento de uma aplicação Android nativa, demandando uma considerável curva de aprendizado para desenvolvedores não familiarizados com a plataforma Android, além da complexidade intrínseca às tecnologias de *blockchain* e identidade auto soberana. Foi necessário empreender estudos aprofundados sobre diversas tecnologias subjacentes, incluindo os conceitos de DIDs, verificadores, provas de credenciais, conexões, a arquitetura do Hyperledger AnonCreds, PlenumBFS, a arquitetura do Hyperledger Indy, e o modelo de dados de credenciais verificáveis do W3C.

Outro desafio substancial durante o desenvolvimento residiu na disparidade entre os formatos utilizados nos módulos de criptografia do Indy, no módulo de acesso ao *ledger* do Indy e no protocolo de comunicação. Esta discrepância demandou considerável esforço para alinhar de maneira segura e integrada a comunicação entre os Agentes com o *ledger*.

Por fim, a questão de UI/UX apresentou desafios adicionais, dada a limitada experiência, requerendo esforços consideráveis na criação de uma experiência de usuário minimamente intuitiva e eficiente, tanto para o aplicativo Android quanto para a aplicação web.

# 7 Sugestões futuras

O propósito desta seção consiste em delinear uma visão prospectiva com base no trabalho realizado, buscando consolidar este esforço como uma contribuição significativa para o entendimento global da SSI no contexto da engenharia de software. Com esse propósito, serão categorizadas as áreas de evolução em três tipos distintos: evoluções comportamentais, estruturais e processuais.

Na perspectiva da evolução comportamental, o presente sistema foi concebido para realizar a prova de conceito da identidade auto soberana, com foco no caso de uso na área da saúde. Para a continuidade, é proposto o registro de novas normas de dados em diferentes áreas, além da saúde, mediante a incorporação de instituições com o papel de *Verifier* (Endorser). Isso permitirá a solicitação de verificações de credenciais do *Holder*. Além disso, é essencial cadastrar novas definições de credenciais (cred-def) alinhadas às novas normas, expandindo assim a abrangência do sistema para além da norma HL7 inicialmente considerada.

No que tange à evolução estrutural futura, propõe-se a inclusão de gerenciadores de *Agents* no software. A estrutura atual separa as responsabilidades dos agentes em classes, mas realiza requisições diretas ao *ledger*, carecendo de uma instância mediadora para processar os dados das requisições e desacoplar o código, facilitando a generalização dos geradores de mensagem. O Hyperledger Aries, em fase de desenvolvimento, é destacado como uma tecnologia promissora para superar essas dificuldades de comunicação entre dados e mediação entre agentes. A utilização de um gerenciador de agentes como o Hyperledger Aries proporcionará não apenas a desvinculação da lógica dos dados, mas também uma maior interoperabilidade com outros padrões de SSI, incluindo o DIDComm v2, OpenID4VC, W3C JSON-LD, e outras iniciativas voltadas ao desenvolvimento de uma norma para o modelo W3C-VC.

As evoluções no processo de software emergem como aspectos cruciais no atual ecossistema de SSI. Dada a fase de desenvolvimento predominante das ferramentas existentes, a comunidade concentra seus esforços na criação e teste dessas ferramentas, negligenciando, em certa medida, o desenvolvimento do processo para construção de produtos finais utilizando essas ferramentas. Nesse sentido, é imperativo fortalecer as ferramentas operacionais, incorporando mais princípios de Engenharia de Software ao processo. Comparativamente, ecossistemas *blockchain* como o Ethereum oferecem uma ampla variedade de ferramentas operacionais para elucidar e padronizar *Smart Contracts*, redes auxiliares para testes, redes de homologação, documentação e ferramentas para automatizar validações e verificações. No contexto atual da SSI, a documentação oficial é escassa,

limitando-se a uma imagem docker para simular a rede, o que implica em testes manuais. Outros exemplos, como a GoLedger, apresentam bibliotecas para facilitar o desenvolvimento, proporcionando uma arquitetura bem definida e estrutura de testes unitários e E2E integrados, além de *scripts* integrados para *build*, empacotamento e *deploy*. No específico ecossistema de SSI, a *Trust Over Ip Foundation* tem iniciativas voltadas para agregar Engenharia de Software no processo de desenvolvimento.

# 8 Conclusão

Neste estudo, foi realizada uma prova de conceito para avaliar a viabilidade da utilização de identidades auto soberanas no contexto da segurança de dados e autenticação. Através da implementação prática, identificamos algumas dificuldades importantes no desenvolvimento e adoção dessas identidades.

Como citado anteriormente uma das principais dificuldades encontradas foi a complexidade no próprio desenvolvimento, tendo em vista que por se tratar de uma tecnologia nova, os *frameworks* muitas vezes possuem certas instabilidades e alguns recursos limitados. Além disso, questões relacionadas à interoperabilidade entre diferentes soluções de identidade auto soberana também se mostraram desafiadoras.

Apesar desses obstáculos, nossa prova de conceito demonstrou claramente o potencial dessas identidades para melhorar a segurança e privacidade dos usuários. A capacidade de permitir que os indivíduos controlem suas próprias informações de identidade pode reduzir significativamente os riscos associados a violações de dados e identidade.

Além desses obstáculos, o desenvolvimento do trabalho proporcionou ainda uma visão mais ampla a respeito das diferenças e dos avanços que a utilização de identidade auto soberana traz em relação a modelos já existentes, que utilizam em suas bases as CAs e modelos mais avançados como o PGP.

Em conclusão, apesar dos obstáculos técnicos e de adoção, as identidades auto soberanas representam uma abordagem promissora para aprimorar a segurança da identidade digital. Investir em pesquisa contínua e colaboração entre os setores público e privado pode levar a avanços significativos nessa área, contribuindo para um ambiente digital mais seguro e confiável.

# Referências


BORGES, L. E. *Python para desenvolvedores: aborda Python 3.3*. [S.l.]: Novatec Editora, 2014. Citado na página 58.

CHAVES, J. et al. Acessibilidade e identidade digital: um estudo de caso acerca da adequação do portal institucional do instituto federal de educação, ciência e tecnologia catarinense. In: SBC. *Anais do VI Encontro Nacional de Computação dos Institutos Federais*. [S.l.], 2019. Citado na página 41.

CURRAN ARTUR PHILIPP, H. Y. S. C. V. M. J. S. *AnonCreds Specification*. 2023. Disponível em: <https://hyperledger.github.io/anoncreds-spec/>. Citado 2 vezes nas páginas 9 e 37.

DIVINO, S. B. S. Smart contracts: conceitos, limitações, aplicabilidade e desafios. *Revista Jurídica Luso-Brasileira*, p. 2776, 2018. Citado na página 38.

FORUM, W. E. *Identity in a Digital World: A New Chapter in the Social Contract*. 2018. Disponível em: <https://www3.weforum.org/docs/WEF_INSIGHT_REPORT_Digital%20Identity.pdf>. Citado 2 vezes nas páginas 11 e 20.

FORUM, W. E. Digital identity on the threshold of a digital identity revolution. 2021. Citado na página 17.

GOMES, R. Docker para desenvolvedores. *Leanpub, Salvador, Bahia,*, 2019. Citado na página 59.

GONÇALVES, L. A. da S. et al. Carteiras digitais: o futuro dos pagamentos móveis. *Revista Ibero-Americana de Humanidades, Ciências e Educação*, v. 8, n. 1, p. 377–393, 2022. Citado na página 38.

HANCOCK, A. *Digital Identification Must Be Designed for Privacy and Equity, Electronic Frontier Foundation*. 2020. Disponível em: <https://www.eff.org/deeplinks/2020/08/digital-identification-must-be-designed-privacy-and-equity-10>. Citado 2 vezes nas páginas 9 e 22.

HARIKRISHNAN, M.; LAKSHMY, K. Secure digital service payments using zero knowledge proof in distributed network. In: IEEE. *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. [S.l.], 2019. p. 307–312. Citado na página 39.

HASSELGREN, A. et al. Blockchain in healthcare and health sciences—a scoping review. *International Journal of Medical Informatics*, Elsevier, v. 134, p. 104040, 2020. Citado 2 vezes nas páginas 11 e 29.

ID4D, T. W. B. *Practitioner's Guide: Identity Lifecycle*. 2023. Disponível em: <https://id4d.worldbank.org/guide/identity-lifecycle>. Citado 2 vezes nas páginas 9 e 21.

MILANI, A. *PostgreSQL-Guia do Programador*. [S.l.]: Novatec Editora, 2008. Citado na página 59.

MOSKALA, M.; WOJDA, I. *Android Development with Kotlin*. [S.l.]: Packt Publishing Ltd, 2017. Citado na página 58.

NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, p. 21260, 2008. Citado na página 28.

PEIXOTO, R. J. M. *DIDs, Claims, Credentials e Blockchains (Self-sovereign Identity)*. Tese (Doutorado), 2021. Citado na página 17.

SOARES, P. H. L. et al. *Alternativa para emissão de certificados por autoridade certificadora online*. Tese (Doutorado) — Florianópolis, SC., 2017. Citado na página 18.

TERADA, R. *Segurança de dados: criptografia em rede de computador*. [S.l.]: Editora Blucher, 2008. Citado na página 18.

Tom Occhino, Jordan Walke. *JS Apps at Facebook*. 2013. Disponível em: <https://youtu.be/GW0rj4sNH2w>. Citado na página 58.

ZHENG, Z. et al. An overview of blockchain technology: Architecture, consensus, and future trends. In: IEEE. *2017 IEEE international congress on big data (BigData congress)*. [S.l.], 2017. p. 557–564. Citado 3 vezes nas páginas 11, 28 e 29.

ZIMMERMANN, P. et al. Advanced praise for pgp: Pretty good privacy. Citado na página 26.