# Relato de Experiência do Processo de Implantação do Testbed para Gestão de Identidades Digitais Descentralizadas

**E. Bruno Evaristo C.**[1], **Ismael M. A. Ávila**[1], **Jeffson Celeiro** [1]
**Fiterlinge Sousa**[2], **Michelle Silva Wangham**[2,3]

[1]Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD)
Campinas – SP – Brasil

[2]Rede Nacional de Ensino e Pesquisa (RNP)

[3]Universidade do Vale do Itajaí (UNIVALI)

{elderb,avila_an,jcsousa}@cpqd.com.br,
{fiterlinge.sousa, michelle.wangham}@rnp.br

***Abstract.*** *The article presents the modeling and implementation of an experimentation environment (testbed) focused on decentralized digital identity. It was conceived in the scope of a research, development, and innovation (RDI) project for the evolution of GIdLab, a service for experimentation in identity management of the National Education and Research Network (RNP). The project was conducted by the digital identity teams of GIdLab and the Center for Research and Development in Telecommunications (CPQD) and covered topics such as the testing of verifiable identifiers between institutions, the mechanisms for records and audits through the blockchain of the Hyperledger Indy project and the development of a decentralized identity environment. The article presents the challenges encountered, and the lessons learned in this deployment of decentralized networks with such different architectural models.*

***Resumo.*** *O artigo apresenta a modelagem e a implantação de um ambiente para experimentação (testbed) focado em identidade digital descentralizada. Este foi concebido no âmbito de um projeto de pesquisa, desenvolvimento e inovação (PD&I) para a evolução do GIdLab, serviço para experimentação em gestão de identidade da Rede Nacional de Ensino e Pesquisa (RNP). O projeto foi conduzido pelos times de identidade digital do GIdLab e do Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPQD) e cobriu tópicos tais como os testes de identificadores verificáveis entre as instituições, os mecanismos de registros e auditorias, por meio da blockchain do projeto Hyperledger Indy, e o desenvolvimento de um ambiente de identidade descentralizada. O artigo apresenta os desafios encontrados e os aprendizados nessa implantação de redes descentralizadas com modelos de arquiteturas tão distintas.*

## 1. Introdução

O Comitê Técnico Blockchain (CT-Blockchain) foi criado pela Rede Nacional de Ensino e Pesquisa - RNP em 2020. Com caráter consultivo, ele busca fomentar o ecossistema e acompanhar os principais avanços técnico-científicos relacionados à tecnologia blockchain e propor à RNP uma visão de futuro no tema. Um dos eixos temáticos abordados

pelo CT-Blockchain engloba estudos sobre aplicações de Identidade Digital Descentralizada (IDD). A RNP conta ainda com o Comitê Técnico de Gestão de Identidade (CT-GID), cuja missão é prospectar soluções inovadoras de gestão de identidade, embasadas em pesquisas de médio e longo prazos para o sistema RNP, de modo a promover a cultura, conscientizar e incentivar o uso das identidades digitais no Brasil.

Adicionalmente, desde 2013 a RNP mantém o Gidlab[1], um serviço para experimentação que oferece consultoria especializada de PD&I em gestão de identidades com plataforma de experimentação disponibilizada sob medida. Em face do crescente número de aplicações de IDD que necessitam de acesso a redes blockchain, faz-se necessário promover ações de pesquisa, troca de conhecimentos e prospecção tecnológica relacionadas à oferta de um ambiente para experimentação apto a atender essa demanda, estendendo a infraestrutura atual do GIdLab.

O artigo relata e contextualiza as experiências decorrentes das atividades de pesquisa, desenvolvimento e de implantação de um ambiente (*testbed*) baseado na plataforma Hyperledger Indy [Dunphy 2022] para experimentação em gestão de IDD com blockchain. No protótipo de *testbed* desenvolvido, tem-se a execução de um *ledger* base, no qual é possível criar aplicações e provas de conceitos sobre essa base, essas atividades possibilitaram identificar e analisar plataformas e soluções que apoiam essas atividades, da concepção de um protótipo do ambiente para experimentação e sua avaliação em um caso de uso. O protótipo de IDD se constitui de uma rede formada por nós de duas instituições envolvidas no projeto (RNP e CPQD), sendo que cada instituição possui dois nós.

O restante do artigo está assim organizado: a Seção 2 discute o referencial teórico; a Seção 3 apresenta as etapas da pesquisa para construção do protótipo; a Seção 4 trata da implantação do protótipo do *testbed*, com serviços em execução; a Seção 5 descreve a prova de conceito e as formas de autenticação e uso no *testbed*; e, por fim, a Seção 6 apresenta a conclusão e os trabalhos futuros.

## 2. Referencial Teórico

Com o avanço da internet, de seus padrões, tecnologias e aplicações, a gestão de identidades digitais vem ganhando maior relevância e atingindo uma massa crítica. Um exemplo disso são iniciativas como a Sovrin[2], uma rede blockchain permissionada voltada para aplicações de identidade digital auto-soberana[Naik and Jenkins 2021], que abriu o código da sua rede, dando origem à Hyperledger Indy, um livro-razão (*ledger*) distribuído criado para identidades descentralizadas. Desde seu surgimento a Indy vem se desenvolvendo como um projeto completo e abarcando uma variedade de aplicações. No final de 2018, sua biblioteca de criptografia foi aceita como um projeto autônomo chamado Hyperledger Ursa. Em março de 2019, a sua parte referente a agentes também se tornou um projeto independente denominado Hyperledger Aries. Como resultado, o Hyperledger Indy original desdobrou-se em três projetos. Além dos projetos citados acima, existem iniciativas de padronização dos modelos de gerência relacionados a identidade digital, um exemplo, a fundação *Trust Over IP* [3], que trabalha na padronização de uma arquitetura

[1]https://www.rnp.br/servicos/testbeds/gidlab

[2]https://sovrin.org/

[3]https://trustoverip.org/

proposição de um ecossistema de confiança digital no qual as interconexões entre cada ecossistema de confiança digital são facilitadas por meio da pilha ToIP.

### 2.1. Hyperledger Indy

Hyperledger Indy[4] provê ferramentas, bibliotecas e componentes reutilizáveis para criar identidades digitais enraizadas em blockchains ou outros registros distribuídos de modo a serem interoperáveis em domínios administrativos, aplicativos e qualquer outro silo[Aggarwal and Kumar 2021]. A rede Hyperledger Indy é formada por dois projetos: *Indy-Node* e *Indy-Plenum*, O *Indy-Plenum* é um registro distribuído de propósito geral, no qual está implementado o algoritmo de consenso, enquanto o*Indy-Node* é uma especialização do *Indy-Plenum*, na qual são implementadas as transações específicas relacionadas às identidades. O *Indy-Node* implementa no *Indy-Plenum* um *plugin* que adiciona as transações específicas para a gestão de identidades, possibilitando a inserção de novos *plugins*.

Existem dois tipos de nós na implementação do Hyperledger Indy, validadores (*validators*) e observadores (*observers*), que assim se distinguem:

1. *Validator*: gerencia a escrita e é o nó que de fato participa do protocolo de consenso;
2. *Observer*: gerencia a leitura.

### 2.2. Criação do DID e registro do *trustee* e do *steward*

A Hyperledger Indy é uma rede permissionada pública. Isso significa que qualquer pessoa pode observar uma transação, mas somente alguns podem escrever transações no livro-razão. Isso inclui a criação de identificadores descentralizados (DIDs). Os *stewards* são aqueles aptos a escrever uma transação no livro-razão Hyperleder Indy mantido pela Sovrin. Podem ser *stewards* instâncias governamentais, sobretudo as de alto escalão. Sua tarefa é administrar e manter a rede Sovrin. Para se tornar um *steward* é necessário seguir um processo de integração definido pela Sovrin.

O papel do *trustee* é o de guardião de outros detentores/titulares de identidades, para caso seja necessário recuperar essas identidades. Por exemplo, o usuário `A` pode nomear o usuário `B` como seu *trustee* para que, no caso de o usuário `A` perder seu celular, a identidade ainda possa ser recuperada com a ajuda do usuário `B`. O usuário `B` pode ser um banco ou qualquer provedor confiável de serviços.

Alguns atores têm outros papéis, como os *trust anchors* (TA). Os TAs são o elo entre o usuário e os *stewards*. Os TAs podem ser bancos, universidades, hospitais, prestadores de serviços, seguradoras, etc. Os TAs são incorporados mediante aprovações dos *stewards*. Assim, o TA aceita uma solicitação do usuário e encaminha para os *stewards*, no caso de escrita no livro-razão [Alex Preukschat 2021]. Outras características importantes têm relação com alguns usuários iniciais, chamados de *endorsers*, que incluem *stewards* e *trustees*. Somente *stewards* e *trustees* podem criar novos *endorsers* e um *trustee* somente pode ser criado por outro *trustee*.

[4]https://www.hyperledger.org/use/hyperledger-indy

## 2.3. Hyperledger Ursa

A Hyperledger Ursa[5] é uma biblioteca criptográfica compartilhada e usada para evitar a duplicação de códigos criptográficos. Ela é um repositório opcional voltado à implementação e ao uso de criptografia[Aggarwal and Kumar 2021].

Está disponível no projeto Ursa uma abrangente biblioteca de assinaturas modulares e primitivas de chave simétrica, a fim de que os desenvolvedores possam inserir e excluir diferentes esquemas criptográficos, por meio da configuração, sem precisar modificar seu código. Além dessa biblioteca básica, a Ursa também inclui criptografia mais recente, tais como assinaturas agregadas, de limite e baseadas em pareamento. Além dessas assinaturas, também serão incluídas primitivas de conhecimento zero (*Zero Knowledge Protocol* - ZKP), incluindo *SNARKs*. O projeto Hyperledger Ursa identificou os seguintes benefícios:

- Evita a duplicação de solução de requisitos de segurança semelhantes em diferentes implementações de blockchain;
- As auditorias de segurança de operações criptográficas são mais simples de analisar quando o código é consolidado em um único local. Isso reduz os esforços de manutenção dessas bibliotecas e melhora os rastros de segurança para aqueles desenvolvedores que porventura tenham menos experiência com projetos de livro-razão distribuído;
- As revisões de especialistas ocorrem em todos os códigos criptográficos, a fim de reduzir a probabilidade de erros de segurança;
- A interoperabilidade entre plataformas que exigem verificação criptográfica melhora quando estas utilizam os mesmos protocolos de segurança;
- A modularidade de componentes comuns estabelece a estrutura para futuras plataformas de tecnologia de livro-razão distribuído modular que compartilham os mesmos componentes. Uma implementação de referência bem-sucedida de um componente comum, como segurança, cria oportunidades futuras;
- Novos projetos conseguem acelerar seu tempo de lançamento no mercado se um paradigma de segurança existente puder ser conectado, em vez de ser refeito em cada caso.

## 2.4. Hyperledger Aries

A biblioteca Aries[6] oferece uma camada de interface para criar, assinar e ler transações na blockchain. Ela oferece conteúdos que auxiliam na emissão e na troca de credenciais verificáveis, incluindo credenciais que utilizam as premissas de conhecimento zero, disponíveis na biblioteca Ursa. A Aries permite interações ponto-a-ponto (P2P) baseadas em identidades descentralizadas, e suporta diversos tipos de registros distribuídos. A exemplo da Hyperledger Indy e da Hyperledger Ursa, a Aries faz parte dos projetos Hyperledger voltados à implementação de sistemas de Identidade Auto-Soberana [Aggarwal and Kumar 2021].

[5]https://www.hyperledger.org/use/ursa
[6]https://www.hyperledger.org/use/aries

### 2.4.1. Aries agent

Um Agente é um trecho de software cujo propósito é interagir com outras entidades, seja por meio de identificadores descentralizados (*Decentralized Identifiers* - DIDs) e o gerenciador de chaves públicas (*Decentralized Key Management* - DKMS), seja por outros meios. Uma instância de um Agente Aries é composta por duas partes, o agente e o controlador.

- O Agente Aries é o responsável por gerir as funcionalidades principais do Aries, incluindo interação com outros agentes, gerenciamento de armazenamento seguro, além de troca de mensagens com o controlador.
- O controlador é quem provê a lógica de negócio, indicando como o agente deve responder a eventos. O Agente envia ao controlador notificações de evento, analisa a melhor resposta e envia mensagens administrativas ao Agente. A comunicação entre os dois é feita por meio de *webhooks* e de chamadas HTTP.

Assim, para desenvolver projetos com o Aries, em geral, basta programar o controlador, o qual proverá a lógica de negócios. Na arquitetura de uma rede que utiliza Indy e Aries, há diversos agentes. Os agentes que se encontram nas ”bordas”da rede são aqueles instalados em dispositivos móveis, tais como telefones celulares, *tablets*, computadores, etc. Há também agentes na ”nuvem”que atuam no roteamento das mensagens. Assim, como regra geral, enquanto os agentes da ”borda”representam pessoas e organizações, os agentes da nuvem têm principalmente como lógica de negócio o roteamento de mensagens entre os demais agentes, e mensagens enviadas entre dois agentes da borda são roteadas por agentes na nuvem.

1. Comunicação

A comunicação entre agentes ocorre por meio de um mecanismo de mensagem chamado DIDcomm (DID Communication) [Alex Preukschat 2021]. O DIDcomm envolve mensagens encriptadas fim-a-fim, as quais são geralmente roteadas por meio de agentes Aries intermediários. A criptografia é feita pela biblioteca Hyperledger Ursa.

O mecanismo usa uma instância do método *did:peer DID method*, que usa DIDs não publicados na blockchain, e que são somente utilizados privadamente entre os dois agentes que se comunicam [Davie et al. 2019]. Entre os protocolos padrões definidos, que configuram um conjunto de mensagens para realizar determinada tarefa, destacam-se:

- O protocolo ”estabelecer conexão”(*establish connection*), que permite a conexão de dois agentes por meio de um conjunto de mensagens: o convite, a requisição de conexão e a resposta de conexão;
- O protocolo ”emitir credencial”(*issue credential*), que permite a um agente emitir uma credencial para outro agente;
- O protocolo ”apresenta prova”(*present proof*), que permite que agentes façam requisições e recebam provas de outros agentes.

## 3. Etapas da pesquisa e da experimentação

Esta pesquisa é caracterizada como uma pesquisa aplicada, que contempla o desenvolvimento, implantação e avaliação de um protótipo de *testbed* para IDD. As seguintes

atividades foram desenvolvidas: levantamento e estudo das soluções tecnológicas que embasaram a construção do ambiente de IDD, a implantação de um protótipo do ambiente de gestão de identidade descentralizada baseada em blockchain e o seu uso em um estudo de caso.

O protótipo foi projetado considerando quatro nós, dois relacionados à RNP, e dois ao CPQD. A partir desse ambiente-base, a configuração da rede seguiu todos os pré-requisitos e orientações da documentação oficial do projeto Hyperledger Indy. A Figura 1 ilustra o cenário-base com a execução do *ledger* de comunicação entre as instituições.

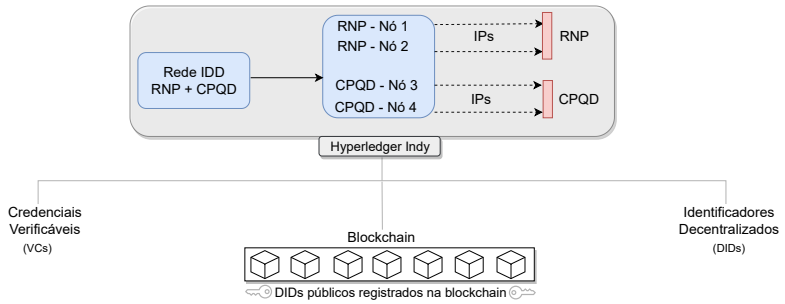



**Figura 1. Cenário de integração da rede IDD**

## 4. Implantação do protótipo do *testbed* IDD RNP/CPQD

Como passo inicial, foi necessária a configuração do ambiente seguindo a instalação de alguns pré-requisitos, como, por exemplo, instalação e configuração do *docker* e do *docker compose* com vistas ao manuseio de volumes da rede Indy. Em seguida, foi feito o ajuste para os nós poderem ser acessados via uma rede Indy node de nível de desenvolvimento, incluindo um navegador *legder*. Esse navegador permite que um usuário veja o *status* dos nós da rede e navegue, pesquise e filtre as transações na *ledger*, como mostrado na Figura 2.

No ambiente proposto, adotou-se o modelo de *Verifiable Organizations Network* (VON)[7]. Por meio do navegador da *ledger* da VON e do endereço *http://3.128.10.144:9000/* é possível verificar:

- O status dos nós do livro-razão (*ledger*);
- O status detalhado dos nós do livro-razão no formato JSON (clicando no link ”Status detalhado”);
- As três *ledgers* da rede Indy: Domain, Pool e Config (clicando nos respectivos *links*);
- As transações do arquivo *Genesis* para a instância *Indy Network* (em um Agente Indy, usando a URL *server/genesis* para obter o arquivo *Genesis* para inicializar o agente).

Ao usar a parte ”Autenticar um novo DID”da interface do usuário ou postar o JSON apropriado na *VON-Network API*, um novo DID pode ser adicionado à *ledger*. Um *Trust Anchor DID*, conhecido e publicado, é usado para gravar o novo DID na *ledger*. Ao clicar na opção *Domain*, é possível navegar por todas as transações criadas nesta instância

[7]https://digital.gov.bc.ca/digital-trust/

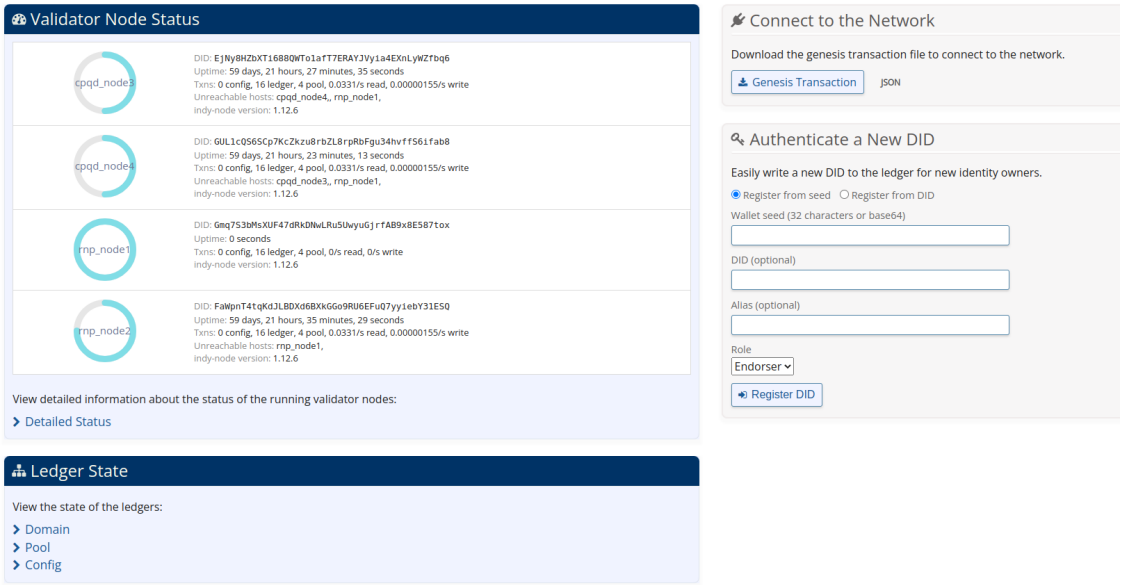



**Figura 2. Ledger da rede em execução, composta por nós da RNP e do CPQD**

da *ledger* [W3C 2022]. Além disso, é possível usar um filtro suspenso para ver apenas tipos específicos de transação da *ledger* (*nym* também conhecido como DID, *schema, CredDef*, etc.) e pesquisar *strings* no conteúdo das transações.

### 4.1. Adicionando um novo nó à ledger

Como as redes Indy são permissionadas, a comunidade de administradores que executam nós validadores precisa aprovar a inclusão de novos nós à rede. Esses nós validadores usam um protocolo bizantino tolerante a falhas. Nele, o primário nó atualiza a *ledger* e notifica os outros nós em busca por um consenso sobre essas alterações. Se os demais nós suspeitarem que o primário não está agindo corretamente, eles podem eleger um novo nó primário. Este protocolo é comprovadamente resiliente contra até um terço de nós maliciosos.

A Indy favorece essa abordagem porque o custo de liquidação de uma transação é muito menor do que caso ocorresse diretamente na blockchain. O protocolo Indy requer pelo menos quatro nós em consenso e funciona bem com 25 nós no *pool*[Indy 2022] de validadores antes que o desempenho comece a degradar. Para ler da rede, um cliente só precisa entrar em contato com um nó de consenso, pois cada resposta contém uma assinatura *Basic Life Support* (BLS) que prova que a resposta representa o valor mantido pela *ledger* em consenso. Uma informação de extrema relevância ligada à política de inserção no nó está ligada à gerência de acesso de cada nó. Por exemplo, como nos casos de um *trustee* adicionar um *steward*, se necessário (somente um *steward* pode adicionar um novo nó validador).

## 5. Prova de Conceito considerando uma aplicação de IDD

O protótipo contempla a criação de uma rede *distributed ledger technology* (DLT) especializada para a criação e gestão de IDDs, assim como o desenvolvimento de uma aplicação que execute os principais métodos e rotinas do *framework DLT (Hyperledger Indy)*, como,

por exemplo, emissão de DIDs entre aplicações (*Agents*), gerenciamento desses identificadores, autenticação de provas e a revogação de credenciais pelo seu emissor. De forma básica, foram criados três agentes por meio da Aries, o que possibilitou estabelecer o triângulo da confiança, conceito base quando se trata de identidade digital descentralizada.

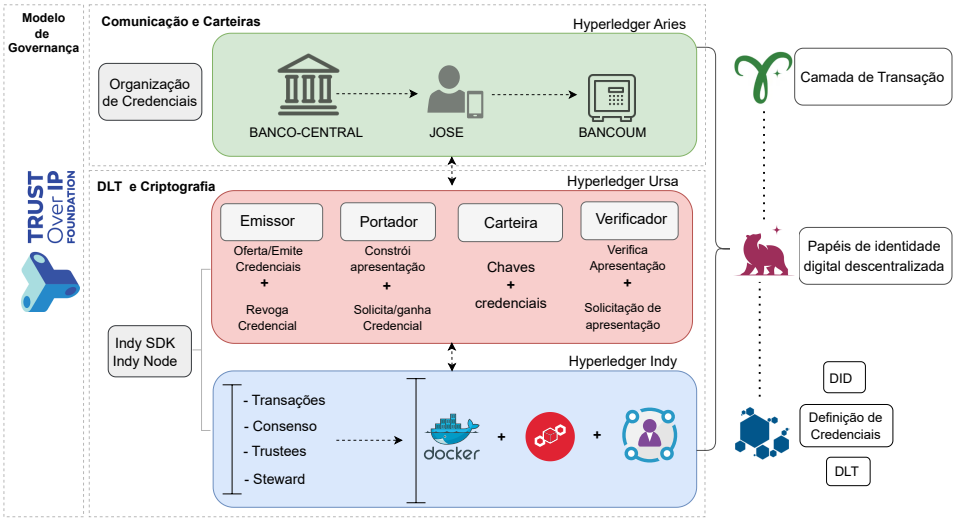



**Figura 3. Pilha de tecnologias em execução na Prova de Conceito (PoC)**

Assim, foi definida a pilha de tecnologias e conceitos adotados na PoC do *testbed*, cuja base foi executada com a definição de nós administradores e gerenciadores (*stewards* e *trustees*). A rede foi configurada e executada utilizando a rede Indy e *container* Docker. Em outra camada da pilha, a biblioteca Ursa provê a camada de segurança, possibilitando verificar os papéis e responsabilidades de cada ente. Por fim, a Aries provê uma camada de execução das transações na rede via *front-end*. De modo geral, todos os processos da PoC, inclusive de comunicação, podem ser definidos conforme a Figura 3, que mostra o triângulo da confiança, baseado no modelo de governança estabelecido pela fundação *Trust over IP*.

## 5.1. Agentes Criados

Conforme a Figura 4, foi utilizado a camada de execução, gerando um *sweguer*, onde é possível criar agentes e estabelecer processos de emissão, verificação e utilização de credenciais válidas. A comunicação é realizada através do protocolo de comunicação DIDcomm, onde as transações de verificação são realizadas via registros na blockchain. É Válido ressaltar que na blockchain não se registra nenhum dado sensível, apenas os schemas de informações que será atrelado a credencial do usuário.

- **BANCO-CENTRAL**: emite e valida as credenciais da rede;
- **JOSE**: solicita credencial válida, emitida pelo agente BANCO-CENTRAL;
- **BANCOUM**: verifica na blockchain a solicitação do agente JOSE, para confirmar se existe registro e se a credencial válida.

Para acessar a interface fornecida pela *aries cloud agent-python*, o *ACA-Py* fornece uma interface REST documentada pela OpenAPI para administrar o estado interno

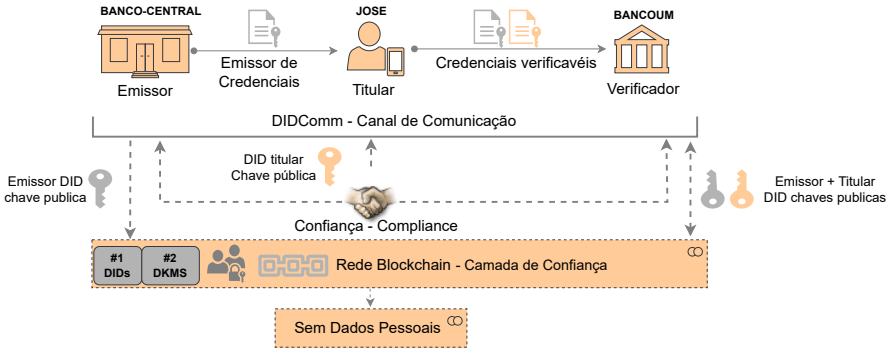



**Figura 4. Triângulo da confiança da PoC**

do agente e iniciar a comunicação com os agentes conectados. Ela pode ser acessada pelas seguintes portas: 8251, 8252 e 8253. Como meio de validação do processo, foi gerado um ambiente de simulação é composto por 13 passos, conforme descritos a seguir:

- Passo 1: Criar o aca-py de cada agente;
- Passo 2: Criar o *schema* utilizado para formatar as credenciais dos agentes;
- Passo 3: Criar convite e aceitar pelo BANCO-CENTRAL colocando um nome genérico (ALIAS) para se localizar;
- Passo 4: Receber o convite de JOSE;

**Passo 1**

**Criar o aca-py do agent BANCO-CENTRAL no terminal**

```
aca-py start --inbound-transport http 0.0.0.0 8151 \
--outbound-transport http --log-level debug \
--endpoint http://localhost:8151 \
--label BANCO-CENTRAL \
--seed 1ea7181731cac49278ebb4eb655ff58j \
--genesis-file ./pool_transactions_genesis \
--ledger-pool-name localindypool \
--wallet-key 123456 --wallet-name bancocentralwallet --wallet-type indy \
--admin 0.0.0.0 8251 --admin-insecure-mode \
--public-invites --auto-accept-invites --auto-accept-requests --auto-ping-connection --auto-respond-messages --auto-respond-credential-offer --auto-respond-presentation-request --auto-verify-presentation --auto-store-credential --auto-respond-presentation-proposal --auto-respond-credential-request --auto-respond-credential-proposal --debug-connections --debug-credentials --debug-presentations --auto-provision --log-file log
```

**Criar o aca-py do agent JOSE no terminal**

```
aca-py start --inbound-transport http 0.0.0.0 8152 \
--outbound-transport http --log-level debug \
--endpoint http://localhost:8152 \
--label JOSE \
--seed 2ea7181731cac53242ebb4eb655ff58j \
--genesis-file ./pool_transactions_genesis \
--ledger-pool-name localindypool \
--wallet-key 123456 --wallet-name josewallet --wallet-type indy \
--admin 0.0.0.0 8252 --admin-insecure-mode \
--public-invites --auto-accept-invites --auto-accept-requests --auto-ping-connection --auto-respond-messages --auto-respond-credential-offer --auto-respond-presentation-request --auto-verify-presentation --auto-store-credential --auto-respond-presentation-proposal --auto-respond-credential-request --auto-respond-credential-proposal --debug-connections --debug-credentials --debug-presentations --auto-provision --log-file log
```

**Criar o aca-py do agent BANCOUM no terminal**

```
aca-py start --inbound-transport http 0.0.0.0 8153 \
--outbound-transport http --log-level debug \
--endpoint http://localhost:8153 \
--label BANCOUM \
--seed 3ea7181731c9874242e123eb655ff58j \
--genesis-file ./pool_transactions_genesis \
--ledger-pool-name localindypool \
--wallet-key 123456 --wallet-name santanderwallet --wallet-type indy \
--admin 0.0.0.0 8253 --admin-insecure-mode \
--public-invites --auto-accept-invites --auto-accept-requests --auto-ping-connection --auto-respond-messages --auto-respond-credential-offer --auto-respond-presentation-request --auto-verify-presentation --auto-store-credential --auto-respond-presentation-proposal --auto-respond-credential-request --auto-respond-credential-proposal --debug-connections --debug-credentials --debug-presentations --auto-provision --log-file log
```

**PASSO 2**

ONDE: Swagger BANCO-CENTRAL (8251)
Abrir no navegador: http://localhost:8251
endpoint: /schemas

```
{
"schema_name": "enroll",
"schema_version": "0.1",
"attributes": [
"name",
"email",
"document"
]
}
```

**PASSO 3**

ONDE: Swagger JOSE (8252)
Abrir no navegador: http://localhost:8252
endpoint: /connections/create-invitation

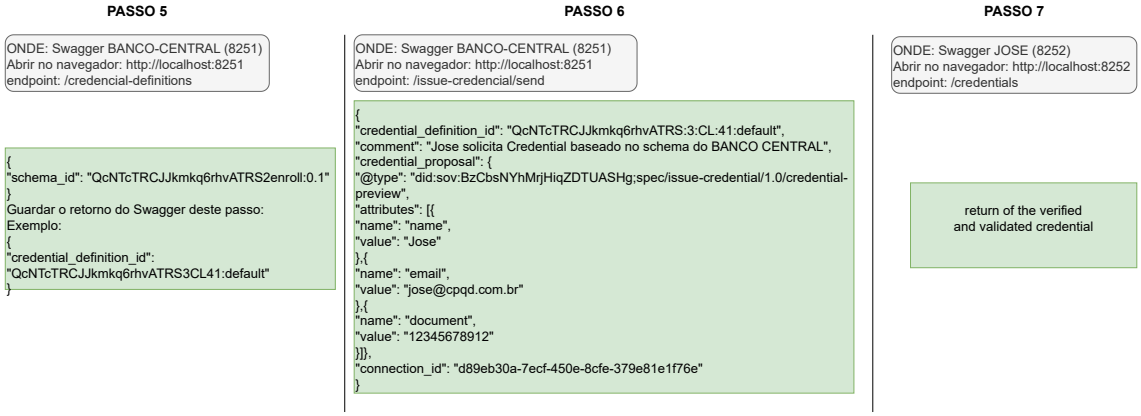
```
{
"@type": "did:sov:BzCbsNYhMrjHiqZDTUASHg;spec/connections/1.0/invitation",
"@id": "d0445279-967f-4bbd-9447-57d45f2b085e",
"label": "JOSE",
"recipientKeys": [
"AZDV2UqvkJSyVygwtYq2rqgpYRoSAVnj4cyz2EcH13pi"
],
"serviceEndpoint": "http://localhost:8152"
}
```

**PASSO 4**

ONDE: Swagger BANCO-CENTRAL (8251)
Abrir no navegador: http://localhost:8251
endpoint: /connections/receive-invitation

```
{
"routing_state": "none",
"created_at": "2019-10-24 19:23:48.587788Z",
"invitation_key": "AuYa3PAxYYubDrTuyY6iQGsAGnGx2rdR87HVMQ51fuzy",
"updated_at": "2019-10-24 19:23:48.628393Z",
"invitation_mode": "once",
"initiator": "external",
"state": "request",
"connection_id": "3d9036ab-80af-463c-9b95-6b3aa8b7eea3",
"request_id": "c84f60fe-93ec-4ea9-aeb9-30e58c18cccf",
"accept": "auto",
"their_label": "JOSE",
"my_did": "VipLPWeX6aTqb76a9E6KpZ"
}
```

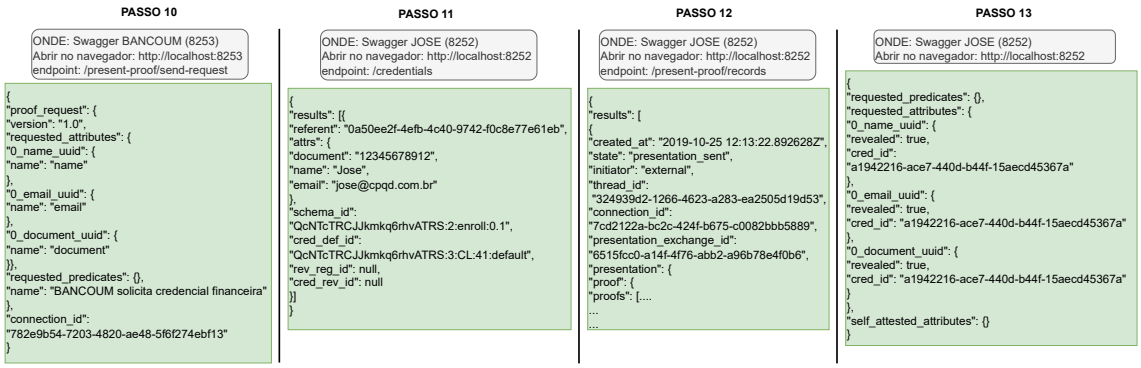
- Passo 5: Receber uma *Credencial-definition* com a chave de retorno do *schema*;
- Passo 6: JOSE solicita a credencial baseada no *schema* do BANCO-CENTRAL;
- Passo 7: JOSE confirma se sua solicitação foi aceita pelo BANCO-CENTRAL;





- Passo 8: BANCOUM cria convite e disponibiliza para o JOSE;
- Passo 9: JOSE aceita convite do BANCOUM para criar uma conexão;

- PASSO 10: BANCOUM solicita credencial financeira para o JOSE;
- Passo 11: JOSE verifica suas credenciais para copiar o valor da credencial;
- Passo 12: JOSE verifica sua credencial válida (emitida pelo BANCO-CENTRAL);
- Passo 13: JOSE envia a credencial válida para o BANCOUM.




## 5.2. Resumo das execuções

Segue um breve resumo de todos os passos executados e as atividades realizadas:

1. Foi realizado o processo de execução da rede blockchain chamada “Von Network”, na qual quatro nós chegaram ao consenso em relação às transações realizadas entre os agentes;
2. Criados três agentes entre os quais transações seriam realizadas;
3. Agente BANCO-CENTRAL: emitiu e validou as credenciais entre as entidades da rede;
4. Agente JOSE: desejava ter sua credencial validada pelo BANCO-CENTRAL para ser utilizada posteriormente junto a outros agentes;
5. Agente BANCOUM: enviou para o JOSE um convite que, uma vez aceito, possibilitou solicitar a prova da credencial para o schema criado pelo BANCO-CENTRAL;
6. Foi criado o schema que é um conjunto de atributos a serem posteriormente utilizados para realizar as transações de credencial entre os agentes criados na rede Blockchain;
7. JOSE criou um convite a partir da rede blockchain para que o BANCO-CENTRAL aceitasse e assim gerasse uma conexão entre estes agentes;
8. BANCO-CENTRAL aceitou o convite do JOSE e, deste momento em diante, os dois agentes possuem uma conexão ativa entre eles;
9. JOSE solicitou ao BANCO-CENTRAL a credencial com suas informações e baseada no *schema* criado pelo banco, com os atributos (”name”, “email”, ”document”);
10. Após enviar a credencial para o BANCO-CENTRAL, o JOSE conferiu sua credencial na rede blockchain;
11. A partir deste momento o JOSE passou a ter na rede blockchain uma credencial validada pelo BANCO-CENTRAL;
12. Como passou a ter uma credencial válida na rede, JOSE pode aceitar convites de outros agentes.
13. JOSE aceitou o convite de outro agente, chamado BANCOUM. O convite havia sido disponibilizado publicamente, de modo a poder ser aceito por usuários validados na rede blockchain;
14. Ao verificar que JOSE aceitou o convite, o BANCOUM solicitou a ele uma prova de suas informações na rede blockchain.
15. JOSE verificou sua credencial na rede blockchain para enviar a prova solicitada pelo BANCOUM;
16. JOSE enviou a prova solicitada de suas informações gravadas na rede blockchain para o BANCOUM utilizando somente o código de sua credencial.

O propósito da PoC apresentada foi avaliar o Testbed para IDD para analisar se este pode no futuro ser aplicado em cenários diversos (agricultura, financeiro, administrativo), além da possibilidade de criar uma infraestrutura que outros possam usar para realizar experimentos. Ate a escrita deste artigo, não foi encontrado na literatura outros *testbeds* de IDD, que possibilitem as experimentações realizadas sem ser necessário construir uma nova infraestrutura.

### 5.3. Dificuldades relacionadas ao protótipo

A seguir é detalhado alguns pontos de dificuldades na implantação do protótipo:

- Dificuldades em se encontrar cenários semelhantes ao proposto, logo julgamos ser uma nova iniciativa de testbes em IDDs;
- Documentação oficial pouco clara: o que trouxe dificuldades no entendimento do processo de instalação da rede, na configuração do ambiente e na gestão da documentação;
- Entendimento das diferenças entre uma rede de teste e a rede de produção para ambientes IDD, importante ressaltar que no protótipo foi desenvolvido um ambiente de produção;
- Dificuldades relacionadas a comunicação interna entre as instituições: o ambiente foi configurado em estruturas de redes distintas, enquanto a RNP oferecia serviços de computação em nuvem pública, no CPQD foi fornecido serviços locais de máquinas virtuais (VMs), nesse contexto foi necessário e fundamental verificar as políticas de consenso entre máquinas, como verificação de regras de firewall, disponibilização de IPs públicos e a configuração interna dos scripts de execução via docker, processo os quais foram necessários para estabelecer a comunicação entre os nós;
- Atenção ao tratamento de backups via docker: foi necessário a criação de volumes internos para melhor controle das informações; caso uma máquina seja desligada ou se desconecte por algum motivo, é mantido cópias dos dados nos outros nós.

## 6. Conclusão e Trabalhos Futuros

Este trabalho apresentou um relato da implantação de um *testbed* piloto, uma iniciativa da RNP em parceria com o CPQD. O projeto buscou e alcançou alguns objetivos, como o estudo das soluções tecnológicas para implantação de um ambiente e para gestão de identidades descentralizadas, a implantação e a configuração da rede de blockchain usada para gestão de identidades descentralizadas e, por fim, o desenvolvimento de um protótipo da solução de gestão de identidade descentralizada utilizado no ambiente proposto. O ambiente proposto mostrou-se ainda suficientemente genérico para abstrair a complexidade dos padrões e protocolos de comunicação envolvidos na solução.

Como trabalhos futuros, pretende-se a abertura do *testbed* piloto para a inserção de novos nós (instituições ou empresas) na rede, a fim de proporcionar resultados mais robustos via testes de desempenho e avaliações mais complexas do ambiente. Visa-se a definição do modelo de governança, atribuindo responsabilidades aos nós envolvidos na rede. Pretende-se também focar na especificação e nas melhorias relacionadas aos identificadores descentralizados e avançar no uso de Kubernetes na execução do *testbed*, com vistas a melhorar a gestão do ambiente. Por fim, buscar-se-á realizar mais testes como novas aplicações, de modo a gerar cada vez mais dados na rede.

## Referências

[Aggarwal and Kumar 2021] Aggarwal, S. and Kumar, N. (2021). Hyperledger. In *Advances in computers*, volume 121, pages 323–343. Elsevier.

[Alex Preukschat 2021] Alex Preukschat, D. R. (2021). *Self-Sovereign Identity: Decentralized digital identity and verifiable credentials*. Manning Publications, 1 edition.

[Davie et al. 2019] Davie, M., Gisolfi, D., Hardman, D., Jordan, J., O'Donnell, D., and Reed, D. (2019). The trust over ip stack. *IEEE Communications Standards Magazine*, 3(4):46–51.

[Dunphy 2022] Dunphy, P. (2022). A note on the blockchain trilemma for decentralized identity: Learning from experiments with hyperledger indy. *arXiv preprint arXiv:2204.05784*.

[Indy 2022] Indy, H. (2022). Hyperledger indy. https://github.com/lynnbendixsen/indy-node/blob/master/docs/source/ installation-and-configuration. Ultimo Acesso: 2023-03-15.

[Naik and Jenkins 2021] Naik, N. and Jenkins, P. (2021). Does sovrin network offer sovereign identity? In *2021 IEEE International Symposium on Systems Engineering (ISSE)*, pages 1–6. IEEE.

[W3C 2022] W3C (2022). Decentralized identifiers (dids). https://www.w3.org/TR/did-core/abstract. Ultimo Acesso: 2023-01-25.