Avaliação de Arquitetura e Performance de Contratos para Identidade Digital Descentralizada em Redes Blockchain Permissionadas

Jeffson Celeiro Sousa[1,2], **Bruno Evaristo**[1,2],
Antonio Mateus de Sousa[1], **Ismael Ávila**[1]

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

[2]Universidade Federal do Pará (UFPA)
Belém – PA – Brasil

{jcsousa, elderb, amateus, avila_an}@cpqd.com.br

Resumo. *Este trabalho propõe uma avaliação arquitetural e de desempenho de contratos inteligentes voltados à gestão de identidades digitais descentralizadas em redes blockchain permissionadas. A análise compara abordagens baseadas em Hyperledger Indy e Hyperledger Besu, considerando operações fundamentais como criação de identidades, esquemas de credenciais e controle de revogação. Métricas como tempo de resposta, vazão, uso de recursos e escalabilidade são analisadas para subsidiar decisões sobre infraestrutura segura e eficiente para identidade digital no contexto da governança pública digital. Os resultados evidenciam ganhos de desempenho superiores na abordagem baseada em Besu, além de maior previsibilidade no uso de recursos, reforçando seu potencial para aplicações sustentáveis em larga escala. Códigos, contratos e planos de teste estão disponíveis publicamente para reprodutibilidade.*

Abstract. *This paper proposes an architectural and performance evaluation of smart contracts aimed at managing decentralized digital identities in permissioned blockchain networks. The analysis compares approaches based on Hyperledger Indy and Hyperledger Besu, considering fundamental operations such as identity creation, credential schemes, and revocation control. Metrics such as response time, throughput, resource usage, and scalability are analyzed to support decisions on secure and efficient infrastructure for digital identity in the context of digital public governance. The results show superior performance gains in the Besu-based approach, in addition to greater predictability in resource usage, reinforcing its potential for sustainable large-scale applications. Codes, contracts, and test plans are publicly available for reproducibility.*

1. Introdução

A crescente demanda por serviços digitais seguros e interoperáveis tem impulsionado a adoção de modelos de governança baseados em identidades digitais descentralizadas (IDD), também conhecidas como Self-Sovereign Identity (SSI). Diferentemente dos sistemas tradicionais centralizados, a SSI permite que indivíduos e organizações controlem suas credenciais e identidades, promovendo autonomia, portabilidade e segurança [Allen 2016, W3C 2022].

Plataformas como o Hyperledger Indy foram pioneiras na implementação desses conceitos, estabelecendo um ecossistema de identidades baseado em ledgers permissionados, operações especializadas e protocolo de consenso bizantino tolerante a falhas [Hyperledger Foundation 2019]. No entanto, limitações de desempenho, interoperabilidade e escalabilidade levaram à exploração de alternativas mais flexíveis, como o Indy Besu, que transpõe a lógica de identidades para contratos inteligentes executados em redes Ethereum permissionadas, utilizando o Hyperledger Besu como infraestrutura base [Community 2024a].

Neste artigo, propomos uma análise comparativa entre essas duas abordagens, com ênfase na eficiência arquitetural, escalabilidade computacional e aderência a padrões abertos. Através de experimentos reprodutíveis em ambientes controlados, avaliamos o desempenho de funções críticas do ciclo de vida de DIDs, como criação, atualização, emissão e revogação de credenciais. Utilizamos a ferramenta Hyperledger Caliper para mensuração de métricas como latência, vazão, uso de CPU e memória.

Ao inserir essa análise no contexto de governança digital, este trabalho contribui para o debate sobre arquiteturas sustentáveis e seguras para identidades digitais em larga escala. Os resultados obtidos demonstram o potencial do modelo baseado em Besu para aplicações governamentais, institucionais e reguladas, apoiando decisões arquiteturais com base em evidências quantitativas.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados à proposta. A Seção 3 apresenta a descrição da proposta. A Seção 4 apresenta a metodologia de avaliação do trabalho. A Seção 5 apresenta os resultados. E, por fim, a Seção 6 apresenta a conclusão e trabalhos futuros acerca deste estudo.

2. Trabalhos Relacionados

Diversos estudos recentes têm se dedicado à avaliação de desempenho de plataformas blockchain, incluindo Ethereum, Hyperledger Fabric e Hyperledger Indy, refletindo sua adoção crescente em aplicações empresariais e institucionais. Uma das ferramentas mais utilizadas nesses estudos é o Hyperledger Caliper, que oferece suporte padronizado e reprodutível para benchmarking de redes permissionadas e públicas.

Kaushal e Kumar [Kaushal e Kumar 2024] utilizaram o Hyperledger Caliper para avaliar uma rede Hyperledger Fabric simulando um sistema de monitoração remota de pacientes (RPM). A rede incluía três organizações com autoridades certificadoras e nós ordenadores. O estudo mediu latência e vazão em operações de leitura e gravação, e concluiu que a Fabric apresentou bom desempenho sob diferentes taxas de transação, com variações mínimas.

Kshirsagar e Pachghare [Kshirsagar e Pachghare 2022] propuseram um novo algoritmo de consenso, denominado Proof of Scope, e o compararam com Raft e PoW-Ethash. Os resultados mostraram ganhos de até 38% na latência e 22% na vazão. O trabalho evidencia como a escolha do consenso pode influenciar drasticamente o desempenho da rede.

Melo et al. [Melo et al. 2024] aplicaram Stochastic Petri Nets para modelar o desempenho de redes Hyperledger Fabric. O estudo demonstrou que parâmetros como tamanho do bloco e políticas de endosso afetam diretamente a latência e a vazão. A validação

experimental mostrou alta confiabilidade, reforçando o valor de modelos analíticos para prever o desempenho.

Choi e Hong [Choi e Won-Ki Hong 2021] compararam uma rede privada Ethereum com a testnet Ropsten. Através de Caliper, observaram que redes privadas possuem menor latência e maior throughput para transações simples, mesmo com limites de gás semelhantes. A pesquisa destaca como o ambiente de execução influencia diretamente os resultados.

Bastos et al. [Bastos et al. 2024] apresentaram o MinIndy, uma ferramenta de automação para implantação e gestão de redes Hyperledger Indy. Com o uso de Ansible e Docker, a ferramenta facilita a adoção da SSI em ambientes controlados. As avaliações mostraram que o desempenho é equivalente ao de implantações manuais, com benefícios operacionais e reprodutibilidade aprimorada.

No ecossistema da Hyperledger Besu, Fan et al. [Fan et al. 2022] realizaram uma análise detalhada utilizando o Caliper para avaliar algoritmos como PoA, IBFT 2.0 e QBFT. Os resultados indicam que o QBFT escala até 14 validadores sem degradação perceptível e que o tempo e o tamanho dos blocos são parâmetros críticos para o desempenho.

Mostarda et al. [Mostarda et al. 2023] propuseram uma ferramenta personalizada de benchmarking para redes Besu, superando as limitações do Caliper em cenários multi-organizacionais. A ferramenta revelou anomalias operacionais, como blocos vazios recorrentes, permitindo uma avaliação mais precisa da contribuição de cada nó validador.

A Tabela 1 apresenta um resumo dos trabalhos analisados com foco em blockchain, reprodutibilidade e aplicabilidade a identidades descentralizadas:

Tabela 1. Resumo dos trabalhos relacionados e suas características principais.

Trabalho	Blockchain	Caliper	Consenso Avaliado	Modelo Reproduzível	Avaliação DID
[Kaushal e Kumar 2024]	Fabric	✓	Solo	✗	✗
[Kshirsagar e Pachghare 2022]	Fabric	✓	Proof of Scope	✗	✗
[Melo et al. 2024]	Fabric	✓	Solo	✓	✗
[Choi e Won-Ki Hong 2021]	Ethereum	✓	PoW	✗	✗
[Bastos et al. 2024]	Indy	✗	RBFT	✓	✓
[Fan et al. 2022]	Besu	✓	QBFT, IBFT 2.0, PoA	✗	✗
[Mostarda et al. 2023]	Besu	✓	QBFT	✓	✗
Proposta	Besu	✓	QBFT	✓	✓

Apesar das contribuições anteriores, poucos trabalhos abordam diretamente a avaliação de desempenho de contratos inteligentes voltados à identidade descentralizada, com reprodutibilidade e foco na eficiência arquitetural. Nosso trabalho busca preencher essa lacuna, oferecendo um estudo comparativo entre Hyperledger Indy e Indy Besu, com base em evidências experimentais e alinhamento aos princípios da governança digital.

3. Proposta de arquitetura e modelo de avaliação

Esta seção descreve a arquitetura proposta para a realização dos experimentos de avaliação de desempenho de contratos inteligentes de identidade descentralizada (DID) nas plataformas *Hyperledger Indy* e *Indy Besu*. O objetivo é permitir uma comparação sistemática entre as duas abordagens, considerando aspectos como desempenho, complexidade de operação e escalabilidade, bem como sua aplicabilidade a contextos de governança digital pública e institucional.

3.1. Hyperledger Besu

A *Hyperledger Besu* é um cliente Ethereum de código aberto desenvolvido em Java e mantida como um projeto graduado pela Hyperledger Foundation desde 2020 [Besu 2023b]. Ela é compatível com a Ethereum Virtual Machine (EVM) e permite a operação em redes públicas e privadas, com suporte completo a contratos inteligentes escritos em Solidity. No contexto de redes permissionadas, a Besu oferece suporte a algoritmos de consenso como *Clique* (Proof of Authority), *QBFT* e *IBFT 2.0*, viabilizando governança controlada sobre a validação de blocos, o que é essencial em contextos regulados, consórcios interinstitucionais e serviços públicos digitais [Besu 2023a].

Uma das principais vantagens da Besu é sua arquitetura modular, que permite a integração de *plugins*, gerenciamento de permissões por conta, canais privados de transação, e APIs RPC compatíveis com ferramentas Ethereum existentes [Besu 2023c]. Ela também oferece ferramentas de monitoramento nativas via Prometheus e métricas detalhadas para análise de desempenho. Além disso, sua compatibilidade com bibliotecas como `ethers.js` e *frameworks* como Hardhat facilita o desenvolvimento, teste e automação de contratos inteligentes.

No contexto de identidade descentralizada, a Besu serve como base para a iniciativa *Indy Besu* [Community 2024b], que implementa contratos para operações DID como `createDid`, `updateDid` e `createCredentialDefinition`. Essa abordagem permite a execução de identidades digitais descentralizadas em um ambiente Ethereum permissionado, utilizando endereços como identificadores e mantendo compatibilidade com métodos estabelecidos como `did:ethr` e extensões como `did:indy:besu`. Essa flexibilidade reduz a necessidade de infraestrutura dedicada, permitindo que identidades descentralizadas operem sobre redes já existentes, o que contribui para racionalização de recursos computacionais.

3.2. Hyperledger Indy

A *Hyperledger Indy* é um *framework* especializado de identidade digital descentralizada, com foco exclusivo na gestão de DIDs e credenciais verificáveis. Lançado originalmente pela Sovrin Foundation e posteriormente incorporado à Hyperledger como projeto graduado, a Indy provê uma *ledger* permissionada pública baseada no protocolo de consenso RBFT (Redundant Byzantine Fault Tolerance), desenvolvido para suportar confiança distribuída entre nós validadores [Indy 2022].

A arquitetura da Indy é composta por dois componentes principais: a *Indy Ledger*, responsável por armazenar as transações de identidade como `NYM` (para criação de DIDs), `SCHEMA` (para definição de atributos), `CRED_DEF` (para definição de credenciais) e registros de revogação [Indy 2023]; e a *Indy SDK*, um conjunto de bibliotecas escritas em Rust com *wrappers* para diversas linguagens, utilizado por agentes Aries para interagir com a *ledger*, gerir carteiras e estabelecer conexões *peer-to-peer* com segurança criptográfica.

Apesar de sua maturidade e adoção em ambientes como a rede Sovrin, a Hyperledger Indy tem limitações de desempenho, interoperabilidade com padrões modernos (como `did:ethr`) e dificuldades na evolução da base de código devido à sua arquitetura monolítica [Community 2023], o que impõe maior sobrecarga computacional e custo de manutenção, especialmente em ambientes de larga escala. A iniciativa *Indy Besu* surge como uma proposta complementar que busca migrar a lógica de identidade da Indy para

contratos inteligentes em Ethereum, preservando a semântica das transações originais e facilitando a integração com ferramentas amplamente utilizadas no ecossistema Ethereum. Essa abordagem também melhora o desempenho da rede e reduz os requisitos computacionais dos nós.

Dessa forma, é oportuno comparar as funções utilizadas nas duas tecnologias a fim de avaliar o desempenho da Indy Besu em termos das operações correspondentes na Hyperledger Indy, conforme especificadas na Indy DID Method Specification. A Tabela 2 apresenta essa comparação:

Tabela 2. Comparação entre funções da Indy-Besu e operações da Hyperledger Indy.

Função na Indy-Besu	Operação na Indy	Descrição
`createDid`	`NYM`	Cria um DID e sua chave pública na rede.
`updateDid`	`ATTRIB`	Atualiza chaves ou metadados de um DID existente.
`createSchema`	`SCHEMA`	Registra um esquema de atributos para credenciais.
`createCredential Definition`	`CRED_DEF`	Cria uma definição de credencial baseada em um esquema.
`createRevocation Registry`	`REVOC_REG_DEF`	Cria um registro de revogação para controle de credenciais.
`createOrUpdate Entry`	`REVOC_REG_ENTRY`	Insere ou atualiza o status de revogação de credenciais.

4. Avaliação

Nesta seção, descrevemos a metodologia adotada, a configuração experimental dos ambientes avaliados e os parâmetros de execução utilizados nos testes. Os experimentos foram conduzidos de forma controlada, variando a taxa de envio de transações — definida como o número total de requisições de transações enviadas por segundo por todos os *workers* (req/s) — de 20 até 120 req/s, com incremento de 10 e duração fixa de 10 segundos por rodada. Essa configuração foi definida diretamente nos arquivos YAML utilizados pela ferramenta de *benchmark*.

Todos os artefatos utilizados neste estudo — incluindo códigos-fonte, arquivos de configuração, contratos inteligentes, módulos de carga, logs e resultados experimentais — estão disponíveis publicamente em nossos repositórios reproduzíveis[1] e [2].

Para efeito de comparação, avaliamos o desempenho de operações relacionadas à gestão de Identidades Descentralizadas (DID) em dois contextos distintos:

- **Cenário 1 — Hyperledger Indy:** Implantada uma rede blockchain baseada no Hyperledger Indy em um servidor dedicado com Ubuntu 20.04, 32 GB de RAM e 12 núcleos de CPU. Esses nós foram configurados como validadores RBFT numa rede permissionada.
- **Cenário 2 — Hyperledger Besu:** Implantada uma rede blockchain baseada no Hyperledger Besu em um servidor dedicado com Ubuntu 22.04, 32 GB de RAM e 12 núcleos de CPU. A rede foi configurada com quatro nós validadores utilizando

[1] `https://github.com/jeffsonsousa/evaluation-contracts-indy-besu`
[2] `https://github.com/jeffsonsousa/performance-test-indy`

o consenso QBFT e 2 *bootnodes*, seguindo as recomendações da documentação oficial para ambientes de produção.

4.1. Ferramenta de avaliação de desempenho

Para garantir que as medições de desempenho refletissem a lógica dos contratos inteligentes e não fossem afetadas por limitações operacionais de consumo de gás, todas as redes Besu foram configuradas como *gasless*. Especificamente, definiu-se:

- `gasLimit` = `0x1fffffffffff`;
- `contractSizeLimit` = 2147483647;
- `min-gasprice` = 0;
- `difficulty` = `0x1`;
- `txPoolLimit` = 4096 (padrão).

A ferramenta utilizada para execução dos testes foi o *Hyperledger Caliper*[3] na versão 0.5.0, utilizando o Ethereum SDK v1.4. O Caliper permitiu a orquestração de chamadas aos contratos inteligentes implementados em Solidity para simular operações como `createDid`, `createSchema`, entre outras.

Ademais, configurou-se a monitoração com Docker para coletar métricas de uso de CPU e memória dos contêineres dos nós da rede Besu durante cada rodada de teste. Os resultados foram processados automaticamente e armazenados em relatórios HTML e CSV, que foram utilizados para gerar os gráficos e análises apresentados neste trabalho.

4.2. Cenários de teste

A Figura 1 apresenta a arquitetura utilizada para a execução do cenário de avaliação de desempenho dos contratos inteligentes de identidade digital descentralizada implantados na rede Hyperledger Besu. O processo é conduzido com o auxílio da ferramenta *Hyperledger Caliper*, que permite simular múltiplos clientes (*workers*) enviando transações para o ledger por meio de chamadas aos contratos Solidity responsáveis pelas funções de identidade. A seguir, é detalhada cada etapa numerada da figura.

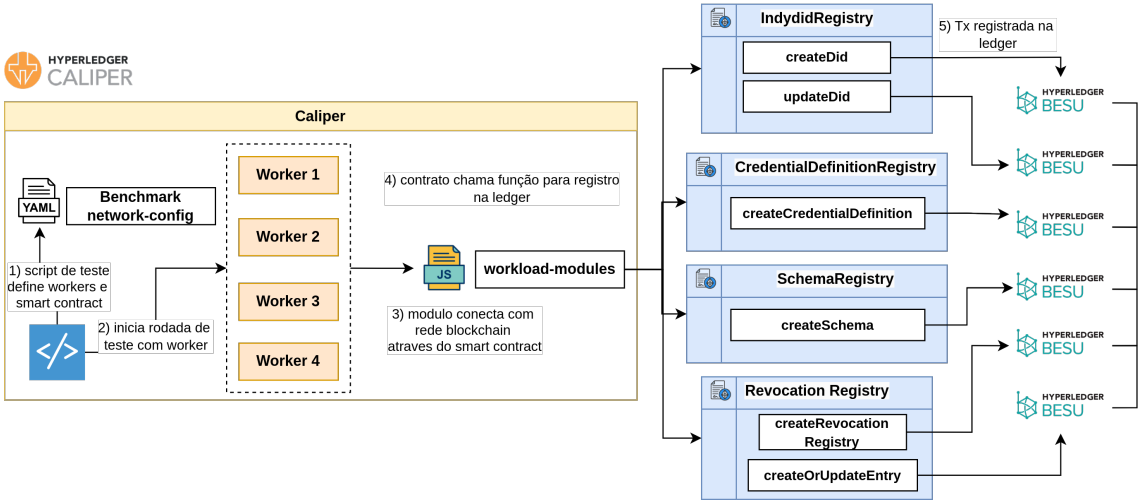

Figura 1. Arquitetura de avaliação de performance dos contratos de identidade na rede Hyperledger Besu com Hyperledger Caliper.

[3] `https://www.lfdecentralizedtrust.org/projects/caliper`

Na primeira etapa, o *benchmark* é configurado por meio de arquivos no formato YAML, os quais especificam: a topologia da rede blockchain (nós, validadores, consenso), os contratos inteligentes a serem avaliados, os módulos de carga (*workload-modules*) e o número de *workers* que irão simular clientes concorrentes. Cada *worker* será responsável por executar chamadas independentes e paralelas aos contratos definidos, simulando um ambiente de uso realista.

Em seguida, com os arquivos de configuração prontos, um orquestrador simples foi criado para gerenciar no Caliper as instâncias dos *workers* conforme especificado, ativando as rotinas de teste definidas no plano de carga. Cada *worker* executa chamadas específicas às funções do contrato inteligente.

No passo 3, os módulos de carga, escritos em JavaScript, encapsulam a lógica de envio de transações à rede. Cada *worker* carrega seu módulo, que realiza a conexão com a rede Besu, invocando diretamente funções específicas dos contratos inteligentes. No passo 4, cada módulo conecta-se à rede Besu e chama funções dos contratos, de acordo com o tipo de operação avaliada.

Após a execução da função, a transação é propagada e registrada no *ledger* da rede Besu. Com o uso do mecanismo de consenso *QBFT (Quorum Byzantine Fault Tolerance)*, os blocos contendo as transações são validados e adicionados à blockchain. A escolha por mecanismos como QBFT ou RBFT reflete diretamente o modelo de governança adotado pela rede, influenciando critérios como controle de acesso, confiabilidade e auditabilidade — fatores centrais em iniciativas públicas ou reguladas.

4.3. Métricas de coleta

A análise destas métricas permite não apenas avaliar o desempenho bruto das plataformas, mas também sua viabilidade em termos de escalabilidade sustentável, custo computacional e adequação a estratégias de governança digital baseadas em consórcios ou serviços públicos interoperáveis. O desempenho da *blockchain* é medido pelas métricas vazão e tempo de resposta da transação.

Vazão da transação (Vt): refere-se ao número de transações cujo resultado a blockchain pode processar e registrar na *ledger* num dado segundo. Equação 1:

$$Vt = \frac{TotalValidTransactions}{TotalTime(s)} \tag{1}$$

Tempo de resposta da transação (Trt): refere-se ao tempo que uma transação leva do momento em que é chamada pelo cliente até ser salva na *ledger*. Equação 2:

$$Trt = ConfirmationTime * Threshold - SendTime \tag{2}$$

5. Resultados

Com base nos gráficos de vazão e latência, as Figuras 2 e 3 mostram que as operações de escrita na rede Besu, como `createDid`, `createSchema` e `create Credential Definition`, atingiram picos de vazão de até 37,5 TPS, 58 TPS e 49 TPS, respectivamente. Já as Figuras 4 e 5 mostram que, embora a rede Indy apresente registros de throughput aparentemente elevados — chegando a **106,23 TPS** para `createSchema` e

128,67 TPS para `createdid` — essa métrica deve ser interpretada com cautela. A análise cruzada com o Gráfico de Falhas (Figura 6) revela que esses cenários estão associados a taxas elevadas de erro e timeout nas transações.

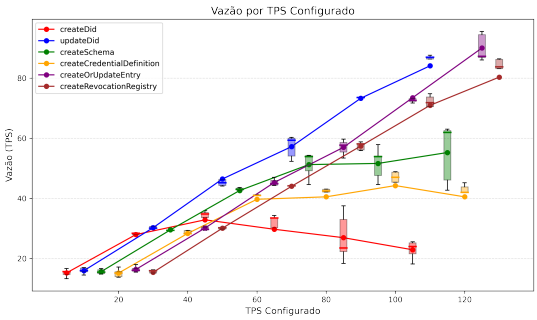

Figura 2. Comparação de vazão entre funções na rede Besu.

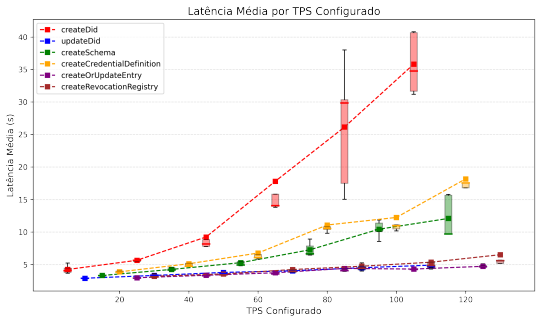

Figura 3. Comparação de latência entre funções na rede Besu.

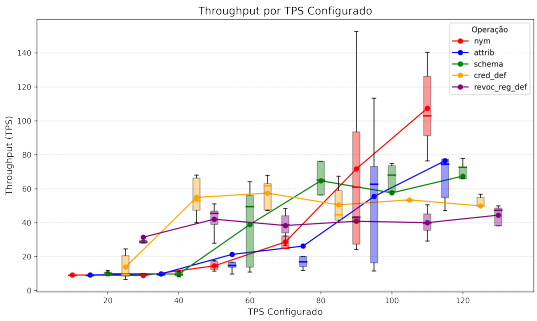

Figura 4. Comparação de vazão entre funções na rede Indy.

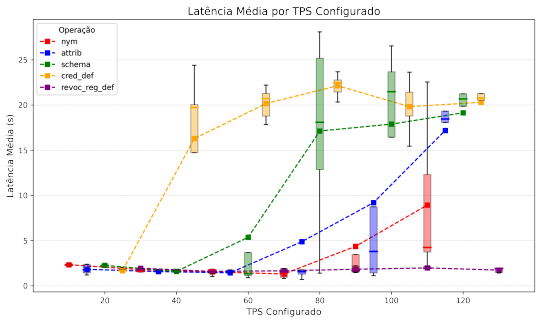

Figura 5. Comparação de latência entre funções na rede Indy.

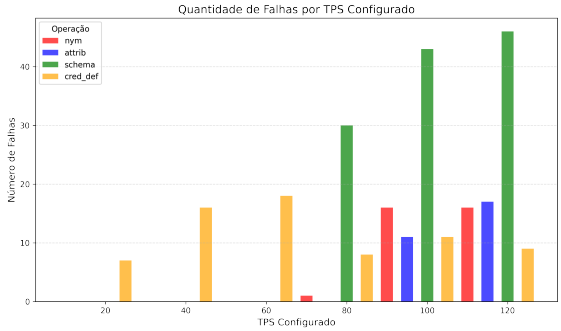

Figura 6. Taxa de falhas nas transações da rede Hyperledger Indy sob diferentes níveis de carga.

Na prática, o aumento do throughput na rede Indy não reflete maior capacidade de processamento, mas é um efeito colateral das falhas que aliviam a fila de transações, reduzindo a carga dos validadores e gerando um TPS médio artificialmente elevado — um fenômeno estatístico, não operacional. Esse comportamento revela uma fragilidade da arquitetura baseada em RBFT, que não escala de forma linear. À medida que a carga cresce, a rede sofre colapsos parciais, com aumento severo de latência (superior a 50 segundos em alguns casos) e taxas crescentes de falhas, especialmente nas operações de escrita como `createSchema` e `createCredentialDefinition`.

Observa-se que, até cerca de 60 TPS, a Indy opera de forma estável, com latência entre 2 e 5 segundos, mas acima desse ponto há degradação exponencial, com saturação dos validadores e colapso de desempenho. Em contraste, a Besu mantém equilíbrio entre throughput, latência e carga mesmo sob 120 TPS, alcançando até 63 TPS com latência média inferior a 10 segundos e praticamente sem falhas. Esse desempenho resulta diretamente do uso do consenso QBFT e da arquitetura baseada em contratos inteligentes, mais eficiente em ambientes permissionados.

Portanto, a análise não deve se limitar ao TPS bruto, mas considerar a sustentabilidade operacional do sistema. Nesse aspecto, a Besu demonstra **maior robustez, previsibilidade e escalabilidade prática** frente à arquitetura da Indy.

A Tabela 3 sintetiza essa comparação no cenário de 60 TPS — limite operacional da Indy sem falhas críticas. A rede Besu apresenta desempenho superior de **95,73%** em `createDid`, **118,61%** em `createSchema` e **104,58%** em `createCredentialDefinition`. Esse ganho decorre da eficiência dos contratos inteligentes no ambiente EVM, que permite maior paralelismo e menor overhead de consenso. Mesmo antes da Indy colapsar, a Besu já se mostra claramente mais escalável e adequada para aplicações com alta demanda transacional.

Tabela 3. Comparação percentual de desempenho (throughput) entre a rede Indy e a rede Besu antes do ponto crítico de falhas (60 TPS).

Operação	Throughput Indy (60 TPS)	Throughput Besu (60 TPS)	Superioridade do Besu (%)
`createDid`	17.76 TPS	34.8 TPS	95.73%
`createSchema`	19.56 TPS	42.7 TPS	118.61%
`createCredentialDefinition`	20.14 TPS	41.2 TPS	104.58%

As operações de leitura, como `getDid` e `getSchema`, apresentaram latência inferior a 5 segundos em ambas as redes, com desempenho estável mesmo sob aumento de carga, evidenciando que o gargalo está concentrado nas operações de escrita e consenso.

As Figuras 7 e 8 mostram que, na rede Besu, o uso de CPU e memória é bem distribuído entre os nós, mantendo latência controlada mesmo com aumento gradual da carga. As funções — `createDid`, `createSchema` e `createCredentialDefinition` — são as que mais demandam CPU, com um ponto de inflexão entre 40 e 60 TPS, quando a rede atinge sua vazão máxima.

A partir desse ponto, o consumo de CPU por nó se estabiliza próximo a **100%**, indicando saturação da capacidade de processamento e reforçando a necessidade de balanceamento ou otimização dos contratos em ambientes produtivos.

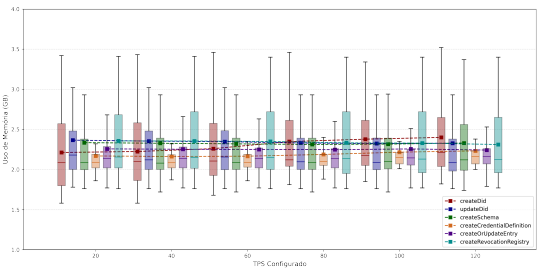

Figura 7. Média de Memória — para todas as operações da Rede Besu.

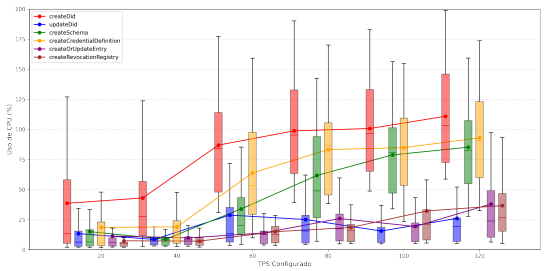

Figura 8. Média de CPU — para todas as operações da Rede Besu.

As funções `updateDid`, `createOrUpdateEntry` e `createRevocation Registry` demandam significativamente menos CPU, não ultrapassando 50% mesmo nas maiores cargas, o que indica menor complexidade computacional e maior previsibilidade operacional.

O consumo de memória foi estável para todas as funções, variando entre **2 e 3 GB por nó**, demonstrando eficiência no gerenciamento de *heap* e *threads* no Hyperledger Besu, garantindo escalabilidade previsível em ambientes com múltiplos validadores.

Na rede Indy, observaram-se picos elevados de latência, especialmente em operações como `createSchema` e `createCredentialDefinition`, reflexo de sua arquitetura monolítica e do alto overhead do protocolo RBFT. Mesmo com estabilidade no uso de memória, o consumo de CPU na Indy é menos previsível e sofre degradação acentuada com o aumento da carga, tornando-a menos adequada para cenários que exigem eficiência energética e escalabilidade controlada.

A sustentabilidade dessas soluções envolve tanto custos computacionais quanto financeiros. No caso da Indy, sua operação exige infraestrutura dedicada, com custos fixos elevados e maior impacto ambiental devido à necessidade de validadores ativos constantemente. Por outro lado, soluções como Indy-Besu, quando operam em redes públicas EVM (Ethereum, Polygon, Arbitrum), eliminam custos de infraestrutura própria, adotando um modelo transacional baseado no consumo de gás, que reflete diretamente o esforço computacional de cada operação. A Tabela 4 apresenta esse custo para operações de identidade digital descentralizada.

Tabela 4. Estimativa de consumo de gás e custos das funções na Indy-Besu em redes públicas baseadas em EVM.

Operação	Gás Consumido	Custo (Ethereum)	Custo (Polygon)
`createDid`	$\sim$ 682,145	$\sim$ \$40.93	$\sim$ \$0.014
`updateDid`	$\sim$ 157,655	$\sim$ \$9.44	$\sim$ \$0.003
`createCredential Definition`	$\sim$ 650,091	$\sim$ \$39.04	$\sim$ \$0.013
`createSchema`	$\sim$ 480,572	$\sim$ \$28.85	$\sim$ \$0.010
`createOrUpdate EntryRevocation`	$\sim$ 205,038	$\sim$ \$12.31	$\sim$ \$0.004
`createRevocation Registry`	$\sim$ 294,060	$\sim$ \$17.65	$\sim$ \$0.006

Dados da Ethereum Foundation, Polygon, Etherscan e benchmarks como ENS, OpenSea e Polygon ID mostram que a operação `createDid` na Indy-Besu consome cerca de 682 mil unidades de gás, valor semelhante a registros de domínio na ENS (600k–800k). Na mainnet Ethereum, com gás a 20 Gwei e ETH a \$3.000, essa operação custa aproximadamente \$40,93. Em redes Layer 2 como Polygon, com gás a 30 Gwei e MATIC a \$0,70, o custo cai para apenas \$0,014 — uma redução superior a 99,96% [Foundation 2024].

Diante disso, o modelo EVM não só supera a Indy em escalabilidade, como também oferece vantagens econômicas e energéticas significativas, eliminando custos fixos de infraestrutura e tornando as operações mais sustentáveis.

Tabela 5. Exemplos de consumo de gás em aplicações populares baseadas em EVM.

Plataforma	Operação	Gás Médio	Custo (Polygon)
Ethereum Name Service (ENS)	Registro de domínio `.eth`	600k–800k	$0.012
Uniswap	Swap de tokens	120k–170k	$0.002
OpenSea	Mint ou transferência de NFT	150k–300k	$0.003–$0.006
Polygon ID	Criação de credenciais de identidade	500k–700k	$0.010–$0.014
Verite (Circle)	Emissão de Verifiable Credential (VC) e DID	∼500k	$0.010

6. Conclusão e Trabalhos Futuros

Os resultados confirmam que a arquitetura *Indy Besu* supera a *Hyperledger Indy* em escalabilidade, controle, transparência, modularidade e sustentabilidade, especialmente nas operações de escrita e sob alta concorrência. Enquanto a Indy opera de forma estável até 60 TPS, acima disso enfrenta falhas críticas que comprometem taxa de sucesso, latência e throughput.

Além do desempenho, a *Indy Besu* se destaca pela aderência a padrões abertos, alta observabilidade e flexibilidade, podendo operar em redes públicas EVM como Ethereum e Polygon, o que reduz custos de infraestrutura e favorece a adoção em larga escala.

A ausência de métricas de CPU e memória na Indy reflete uma limitação estrutural da arquitetura, que carece de observabilidade nativa — um contraste direto com a robustez da Besu. Para trabalhos futuros, sugere-se avaliar cenários multi-organizacionais, resiliência a falhas bizantinas, custos energéticos, além de aprofundar segurança, privacidade e governança, visando consolidar infraestruturas de identidade digital pública sustentáveis e interoperáveis.

7. Agradecimentos

Os autores agradecem o apoio dado a este trabalho, pelo MCTI-Ministério da Ciência, Tecnologia e Inovação, com recursos financeiros do FUNTTEL e administrados pela FINEP, no âmbito especificamente dos projetos AERF - Ações Estratégicas para Redes Futuras, Contrato 01.22.0471.00, Referência 1508/22 e TECSEG - Desenvolvimento de tecnologias e metodologia de avaliação e investigação de segurança para redes e aplicações de governo digital, Contrato 01.21.0163.01, Referência 1196/21.

Referências

Allen, C. (2016). The path to self-sovereign identity. *Life with Alacrity Blog*.

Bastos, M., Veloso, A., Sousa, J., Evaristo, B., Abreu, D., and Abelém, A. (2024). Minindy: Automating the deployment and management of hyperledger indy networks. In *11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*.

Besu, H. (2023a). Consensus protocols overview. `https://besu.hyperledger.org/how-to/configure/consensus/`.

Besu, H. (2023b). Hyperledger besu documentation. https://besu.hyperledger.org.

Besu, H. (2023c). Privacy and permissioning features. https://besu.hyperledger.org/HowTo/Use-Privacy/Privacy-Overview/.

Choi, W. and Won-Ki Hong, J. (2021). Performance evaluation of ethereum private and testnet networks using hyperledger caliper. In *22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*.

Community, H. (2023). Challenges and limitations in hyperledger indy. https://wiki.hyperledger.org/display/indy.

Community, H. (2024a). Indy-besu: An experimental vdr implementation for self-sovereign identity. https://github.com/hyperledger/indy-besu.

Community, H. (2024b). Indy besu experimental project. https://github.com/hyperledger/indy-besu.

Fan, C., Lin, C., Khazaei, H., and Musilek, P. (2022). Performance analysis of hyperledger besu in private blockchain. In *2022 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pages 64–73.

Foundation, E. (2024). Ethereum Gas Fees Explained. Acesso em: maio 2025.

Hyperledger Foundation (2019). Hyperledger indy. https://www.hyperledger.org/use/hyperledger-indy.

Indy, H. (2022). Hyperledger indy documentation. https://hyperledger-indy.readthedocs.io.

Indy, H. (2023). Indy vdr specification. https://hyperledger.github.io/indy-did-method/.

Kaushal, R. K. and Kumar, N. (2024). Exploring hyperledger caliper benchmarking tool to measure the performance of blockchain based solutions. In *11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*.

Kshirsagar, A. and Pachghare, V. (2022). Performance evaluation of proof of scope consensus mechanisms on hyperledger. In *IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*.

Melo, C., Gonçalves, G., Silva, A. S., and Soares, A. (2024). Performance modeling and evaluation of hyperledger fabric: An analysis based on transaction flow and endorsement policies. In *IEEE Symposium on Computers and Communications (ISCC)*.

Mostarda, L., Pinna, A., Sestili, D., and Tonelli, R. (2023). Performance analysis of a besu permissioned blockchain. In *Advanced Information Networking and Applications*, pages 279–291.

W3C (2022). Verifiable credentials data model 1.1. https://www.w3.org/TR/vc-data-model/. W3C Recommendation.