



**UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA**

Autorizada pelo Decreto Federal nº 77.496 de 27/04/76  
Recredenciamento pelo Decreto nº 17.228 de 25/11/2016



**PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO**  
COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

## **XXVIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2024**

### **Virtualização de Uma Rede Hyperledger Fabric em Ambiente de Emulação Fogbed**

**Andreza Nobrega Carvalho<sup>1</sup>; Antonio Augusto Teixeira Ribeiro Coutinho<sup>2</sup>**;

1. Bolsista PROBIC, Graduanda em Engenharia de Computação, Universidade Estadual de Feira de Santana,

e-mail: [ancarvalho@ecomp.uefs.br](mailto:ancarvalho@ecomp.uefs.br)

2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana,

e-mail: [augusto@ecomp.uefs.br](mailto:augusto@ecomp.uefs.br)

**PALAVRAS-CHAVE:** Internet das Coisas; Blockchain; Hyperledger Fabric; Fogbed.

### **INTRODUÇÃO**

A crescente complexidade das aplicações IoT e as limitações associadas à computação em nuvem, como alta latência, largura de banda restrita e vulnerabilidades na segurança dos dados, têm impulsionado a adoção de paradigmas de Computação em Borda/Névoa (Edge/Fog Computing). Esses paradigmas trazem a capacidade de processamento mais próxima da fonte dos dados, reduzindo a dependência de centros de dados remotos e melhorando a eficiência das operações (AAZAM et al., 2014; SATYANARAYANAN, 2017). Em paralelo, tecnologias de livro razão distribuído (*Distributed Ledger Technologies*, DLT), como Blockchain, têm ganhado destaque por sua capacidade de garantir a integridade e a segurança dos dados em redes descentralizadas. Sua utilização elimina a necessidade de intermediários confiáveis tradicionais (IORG et al., 2018) e permite operações transparentes entre os envolvidos.

Este projeto explora a integração dessas duas abordagens, focando na adaptação do ambiente de emulação *Fogbed* para suportar o *Hyperledger Fabric*, uma plataforma de *blockchain* permissionada e de código aberto. A escolha do *Hyperledger Fabric* se deve à sua arquitetura modular e flexível, que permite a personalização de componentes essenciais, como o consenso e a gestão de identidades, além de suportar contratos inteligentes em várias linguagens de programação (HYPERLEDGER, 2024). Essa integração visa criar um ambiente experimental que simula aplicações em cenários de computação em névoa, permitindo a realização de testes e experimentos de maneira eficiente e com custo reduzido.

O estudo é fundamentado em uma revisão abrangente da literatura e em experimentos práticos que envolvem a virtualização do *Hyperledger Fabric* no *Fogbed*, um emulador projetado para facilitar a prototipagem e teste de sistemas de computação em névoa (fog computing) e tecnologias de livro-razão distribuído (DLT/blockchain) (Coutinho et al., 2018), bem como a implementação de interfaces para comunicação entre componentes. A pesquisa procura contribuir para o desenvolvimento de soluções IoT mais seguras e eficientes, explorando as sinergias entre a computação em névoa e as DLTs.

## METODOLOGIA

Para alcançar os objetivos do estudo, adotou-se uma abordagem metodológica que combinou uma revisão bibliográfica detalhada com experimentação prática. Inicialmente, foi realizada uma revisão minuciosa das ferramentas e tecnologias relevantes, focando na integração de sistemas de Computação em Borda/Névoa com tecnologias de livro-razão distribuído (DLTs), com ênfase no *Hyperledger Fabric* (HYPERLEDGER, 2024). A revisão incluiu a análise de literaturas recentes, estudos de caso e especificações técnicas para compreender melhor as interações e desafios na integração dessas tecnologias.

Seguiram-se testes práticos para identificar e resolver desafios específicos relacionados à integração dessas tecnologias em um ambiente de emulação (DAMASCENO; COUTINHO, 2021). O foco foi em resolver problemas relacionados à implementação e ao funcionamento do *Hyperledger Fabric* no contexto do *Fogbed*.

No contexto do estudo, desenvolveu-se uma rede *Hyperledger Fabric* no ambiente emulado com o *Fogbed*. Inicialmente, foram utilizados materiais criptográficos pré-gerados a partir do material disponível no repositório *fabric-samples* (HYPERLEDGER, 2024). No entanto, as limitações impostas pelo *Fogbed*, que não suportava a nomenclatura padrão do *Hyperledger Fabric*, exigiram a adaptação do plugin e a renomeação dos componentes, seguidos de novos testes.

Os testes abordaram ações críticas para a operação da rede *Hyperledger Fabric*, como a criação de canais e a adesão de peers a esses canais. Essas simulações refletiram cenários reais de interação entre os componentes da rede, permitindo avaliar a comunicação entre os nós, a adesão dos peers aos canais e a execução de transações. Os resultados mostraram que a rede funcionou adequadamente fora do ambiente, bem como indicaram a necessidade de aprimoramentos na arquitetura do plugin para maior eficiência e flexibilidade.

A investigação aprofundada resultou na adoção do *SDK Python* do *Hyperledger* (HYPERLEDGER, 2021), uma biblioteca que permite interações programáticas com redes *Hyperledger Fabric*, facilitando a criação de clientes que podem realizar operações na blockchain de maneira automatizada. O *SDK* proporciona maior flexibilidade e modularidade ao processo, permitindo que desenvolvedores criem scripts para tarefas como criação de canais, instalação de *chaincodes* e envio de transações. Portanto, os testes com o *SDK Python* incluíram a simulação dessas operações.

## RESULTADOS E DISCUSSÃO

A partir dos testes realizados, foram geradas novas imagens Docker configuradas especificamente com os requisitos do Containernet (CONTAINERNET, 2024), e instaladas para a versão 1.4 do *Hyperledger Fabric*. A escolha dessa versão baseou-se em sua compatibilidade com o *SDK Python*, o que facilitou a integração entre o ambiente de emulação e a rede blockchain (HYPERLEDGER, 2024).

A validação da solução ocorreu por meio de testes dentro e fora do ambiente *Fogbed*. Durante os testes, foram executadas mais uma vez ações críticas para o funcionamento de uma rede *Hyperledger Fabric*, como a criação de canais e a adesão de peers a esses canais. Essas simulações refletiram cenários reais de interação entre os componentes da rede, permitindo avaliar a integridade da comunicação entre os nós, a adesão correta dos *peers* aos canais e a execução de transações. Os resultados demonstraram que a rede

funcionou de maneira adequada no ambiente emulado, embora ainda tenha sido identificada a necessidade de aprimoramentos na arquitetura do *plugin*, visando maior eficiência e flexibilidade.

Por fim, os testes confirmaram que a integração entre o *Hyperledger Fabric* e o *Fogbed* foi bem-sucedida, com a versão 1.4 do Hyperledger Fabric fornecendo uma base estável para as operações. No entanto, para aprimorar ainda mais a flexibilidade e as funcionalidades da rede, está planejada a migração do SDK Python para o Fabric Gateway (HYPERLEDGER, 2021). Essa migração permitirá o uso de funcionalidades mais recentes do Hyperledger Fabric, como uma interface simplificada e otimizada para a submissão de transações e consultas, tornando o sistema mais eficiente e fácil de operar.

## CONSIDERAÇÕES FINAIS

O estudo resultou na criação de uma arquitetura básica de *plugin* para o *Hyperledger Fabric* (HYPERLEDGER, 2024), adequada para operar em ambientes de emulação como o *Fogbed* (DAMASCENO; COUTINHO, 2021). A arquitetura demonstrou ser eficaz para os objetivos propostos, embora seja evidente a necessidade de melhorias contínuas. Entre os próximos passos, está a migração para o Fabric Gateway, o que deve trazer benefícios adicionais em termos de flexibilidade e atualização contínua da solução. Além disso, está prevista a implementação de uma funcionalidade que permitirá a realização de testes sem a necessidade de criação prévia de arquivos de configuração, simplificando o processo de experimentação e validação de cenários diversos.

O projeto continuará sendo desenvolvido no contexto de um trabalho de conclusão de curso, com o objetivo de validar a arquitetura proposta para aplicações de Internet das Coisas (IoT). Espera-se que os resultados contribuam para a pesquisa e o desenvolvimento de soluções na interseção entre Computação em Borda/Névoa e DLTs, oferecendo uma base para futuras investigações e aplicações práticas.

## REFERÊNCIAS

- AAZAM, M.; KHAN, I.; ALSAFFAR, A. A.; HUH, E.-N. Cloud of things: Integrating internet of things and cloud computing and the issues involved. In: **APPLIED SCIENCES AND TECHNOLOGY (IBCAST), 2014 11th International Bhurban Conference on, 2014, Bhurban**. IEEE, 2014. p. 414-419.
- CONTAINERNET. Container requirements. Disponível em: <<https://github.com/containernet/containernet/wiki#container-requirements>>. Acesso em: 25 fev. 2024.
- COUTINHO, A.; DAMASCENO, U. D. C.; MASCARENHAS, E. D. S.; SANTOS, A. C. D. S.; SILVA, J. E. B. T. da; GREVE, F. **Rapid-prototyping of integrated Edge/Fog and DLT/Blockchain systems with Fogbed**. In: IEEE International Conference on Communications (ICC 2023), Rome, Italy, 2023. Nova York: IEEE, 2023. p. 622-627. DOI: 10.1109/ICC45041.2023.10279234.
- DAMASCENO, U. C.; COUTINHO, A. **Implementação da Arquitetura SOFT-IoT em Ambiente de Emulação Fogbed**. Apresentação de Trabalho: XIX Seminário de Iniciação Científica (SEMIC), Feira de Santana: Universidade Estadual de Feira de Santana, 2021.

HYPERLEDGER. Hyperledger Fabric. Disponível em: <<https://www.hyperledger.org/projects/fabric>>. Acesso em: 25. fev. 2024

HYPERLEDGER. Umbra. Disponível em: <<https://labs.hyperledger.org/labs/archived/umbra.html>>. Acesso em: 02 out. 2024.

HYPERLEDGER. Fabric SDK Python. Disponível em: <<https://github.com/hyperledger/fabric-sdk-py/>>. Acesso em: 02 out. 2024.

HYPERLEDGER. Fabric Gateway. Disponível em: <<https://github.com/hyperledger/fabric-gateway>>. Acesso em: 02 out. 2024.

IORG, M.; FELDMAN, L.; BARTON, R.; MARTIN, M. J.; GOREN, N. S.; MAHMOUDI, C. Fog computing conceptual model. In: **NIST Special Publication (NIST SP) - 500-325**. Gaithersburg: NIST, 2018.

SATYANARAYANAN, M. **The emergence of edge computing**. Computer, v. 50, n. 1, p. 30-39, 2017.