

8-9-2021

## **Investigando o Efeito do Deploy na Ethereum em Repositórios Open Source de Contratos Inteligentes: Uma Proposta Metodológica**

Saori Costa  
*State University of Ceará, saoricosta@gmail.com*

Allyson Allex Araújo  
*Federal University of Ceará, allysson.araujo@crateus.ufc.br*

Jerffeson Souza  
*State University of Ceará, jerffeson.souza@uece.br*

Follow this and additional works at: <https://aisel.aisnet.org/isla2021>

---

### **Recommended Citation**

Costa, Saori; Araújo, Allyson Allex; and Souza, Jerffeson, "Investigando o Efeito do Deploy na Ethereum em Repositórios Open Source de Contratos Inteligentes: Uma Proposta Metodológica" (2021). *ISLA 2021 Proceedings*. 13.

<https://aisel.aisnet.org/isla2021/13>

This material is brought to you by the Latin America (ISLA) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ISLA 2021 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Investigando o Efeito do *Deploy* na Ethereum em Repositórios *Open Source* de Contratos Inteligentes: Uma Proposta Metodológica

*Artigo em Desenvolvimento*

**Saori Costa**

Aluna de Mestrado, PPGCC,  
Universidade Estadual do Ceará  
saoricosta@gmail.com

**Allysson Alex Araújo**

Professor, GESID, Universidade  
Federal do Ceará (*Campus* de Crateús)  
allysson.araujo@crateus.ufc.br

**Jerffeson Souza**

Professor, PPGCC, Universidade Estadual do Ceará  
jerffeson.souza@uece.br

## Abstract

Blockchain and smart contracts (SCs) have been demonstrated as valuable enablers for organizations to innovate on its information systems. In this sense, one can see an adhesion to the open source development by those involved in such ecosystem, especially following Ethereum platform and GitHub usage. However, despite the presence of previous research on SCs maintenance, we may notice a research gap concerning 1) the effect that the *deploy* on Ethereum implies on open source projects in collaborative terms and 2) about the comprehension of what changed in these software repositories after this process. As an ongoing research, this paper aims at presenting an exploratory research proposal drawing on Mining Software Repositories to better understand the effect in open source repositories regarding the Ethereum's SC *deploy* process. Therefore, we contribute by discussing a methodological framing that aims to advance the understanding of the open source SC development process.

## Keywords

Blockchain, smart contracts, mining software repositories, collaborative aspects, software evolution.

## Resumo

A adoção de *blockchain* e contratos inteligentes (CIs) tem se demonstrado uma opção valiosa para organizações inovarem em seus sistemas de informação. Nesse sentido, presencia-se uma adesão ao desenvolvimento *open source* por parte dos envolvidos em tal ecossistema, especialmente com base na plataforma Ethereum e no uso do GitHub. Porém, apesar da presença de pesquisas prévias sobre manutenção de CIs, constata-se uma lacuna de pesquisa quanto 1) o efeito que o *deploy* na Ethereum exerce sobre os projetos *open source* em termos colaborativos e 2) na compreensão sobre o que mudou nesses repositórios após tal processo. Como uma pesquisa em andamento, este artigo objetiva apresentar uma proposta metodológica exploratória pautada em Mineração de Repositórios de Software para compreender o efeito do *deploy* na Ethereum em repositórios *open source* de CIs. Assim, contribui-se ao discutir um enquadramento metodológico o qual busca avançar no entendimento do desenvolvimento *open source* de CIs.

## Palavras-chave

*Blockchain*, contratos inteligentes, mineração de repositórios de software, aspectos colaborativos, evolução de software.

## Introdução

O *blockchain*, que teve seu uso difundido após a criação da criptomoeda *Bitcoin*, é o mais recente de uma série de tecnologias digitais que tem contribuído para mudanças estruturais na forma como as atuais instituições se relacionam (BODÓ *et al.*, 2018). Em suma, o *blockchain* funciona como um livro-razão distribuído, onde o registro, a verificação, o armazenamento, a manutenção e a transmissão de dados são baseados numa arquitetura distribuída protegida por criptografia, cuja governança e confiança mútua entre os nós da rede é estabelecida de forma descentralizada através de algoritmos de consenso (YUAN; WANG, 2018; BECK, *et al.*, 2017). Em contraste aos bancos de dados relacionais, os dados registrados em *blockchain* não podem ser modificados ou excluídos (KASSAB *et al.*, 2019). Em decorrência de tal imutabilidade de dados e do uso de mecanismos de criptografia, o *blockchain* proporciona benefícios especiais como integridade, resiliência e transparência dos dados, tornando-o uma opção valiosa para organizações inovarem em seus sistemas de informação (VIRIYASITAVAT; HOONSOPON, 2019).

A adoção de *blockchain* tem expandido seus horizontes para além das criptomoedas, especialmente devido à ascensão da plataforma Ethereum e, consequentemente, com o amadurecimento de contratos inteligentes (CIs) no suporte ao desenvolvimento de aplicações descentralizadas (em inglês, *decentralized applications* ou *dApps*). A Ethereum viabiliza a execução de *dApps* baseados em *blockchain*, tendo sido projetada para ser flexível e adaptável como uma poderosa infraestrutura global compartilhada (SAJANA *et al.*, 2018). Dessa forma, a Ethereum proporciona uma plataforma de programação *Turing-complete* composta por um ambiente de tempo de execução chamado *Ethereum Virtual Machine* (EVM) a qual executa os *bytecodes* de CIs (HEGEDŰS, 2019). Tais contratos, em forma de código-fontes auto-verificáveis e resistentes às adulterações, especificam as obrigações a serem cumpridas de forma automática, possibilitando, assim, um relacionamento ímpar entre as entidades que compõem a rede (ZUMKELLER, 2018). Diante de tais características, o desenvolvimento e manutenção de CIs lida com particularidades distintas como, por exemplo, a natureza inalterável dos dados, a garantia da simplicidade para redução de custos operacionais em criptomoedas e transparência de acesso ao código-fonte do contrato inteligente (CHAKRABORTY *et al.*, 2018).

Ademais, pode-se perceber em tal domínio de soluções a adesão, por parte dos envolvidos nesse ecossistema, às práticas de desenvolvimento *open source*, inclusive com a presença de projetos de alto valor de mercado como BNB<sup>1</sup> (\$44 bilhões) e USD Coin<sup>2</sup> (\$25 bilhões) (ETHERSCAN, 2021). Nesse sentido, tem-se a formação de um ambiente colaborativo para desenvolvimento de CIs através do uso de plataformas sociais para hospedagem de código-fonte, como o GitHub, em especial. Em decorrência do caráter público e vasto de repositórios, torna-se possível o acesso a informações valiosas às quais são passíveis de serem submetidas a técnicas de mineração de dados visando a extração de *insights* acerca do processo de desenvolvimento (HASSAN, 2008). Nesse cenário, é válido destacar a presença de trabalhos prévios (TIKHOMIROV *et al.*, 2018; PINNA *et al.*, 2019; OLIVA *et al.*, 2020; AJIENKA *et al.*, 2020) com foco em desafios relativos ao desenvolvimento e manutenção de CIs à luz da Mineração de Repositório de Software (MRS), no entanto, constata-se, em especial, uma lacuna quanto 1) a análise do efeito que o processo de *deploy* na Ethereum exerce sobre os projetos *open source* em termos colaborativos e 2) na compreensão sobre o que mudou nos repositórios após tal processo. Tais perspectivas se justificam como particularmente relevantes tendo em vista o impacto dos aspectos colaborativos e o engajamento da comunidade para o sucesso de projetos *open source* (SUBRAMANIAM *et al.*, 2009), bem como a importância da análise de elementos relacionados à evolução de software sob o contexto das particularidades de desenvolvimento de CIs esclarecidas anteriormente.

Diante da lacuna de pesquisa contextualizada, este trabalho objetiva apresentar uma proposta metodológica pautada em Mineração de Repositório de Software para investigar o efeito do processo de *deploy* de contratos inteligentes *open source* na Ethereum. Como fruto de uma pesquisa em andamento, este artigo contribui ao discutir um enquadramento metodológico exploratório fundamentado em uma questão de pesquisa a qual busca avançar na compreensão de aspectos colaborativos e de evolução de software relacionados ao processo de desenvolvimento *open source* de CIs.

---

<sup>1</sup> <https://github.com/binance-chain/>

<sup>2</sup> <https://github.com/centrehq>

## Trabalhos Relacionados

Pinna *et al.* (2019) realizaram um amplo estudo experimental de CIs implantados na Ethereum com objetivo de analisar e caracterizar a interação entre CIs e *blockchain*, em termos de medidas de software, da versão do compilador EVM, das práticas dos desenvolvedores, dos recursos da linguagem de e programação *solidity*, do ambiente *blockchain* e das principais características de contratos desenvolvidos. Por sua vez, Oliva *et al.* (2020) conduziram um estudo exploratório cuja finalidade foi obter uma ampla compreensão sobre os contratos que são implantados na Ethereum, com foco na frequência que os contratos são utilizados, o que fazem e quão complexo são. Finalmente, Ajienska *et al.* (2020) investigaram os CIs de 11 projetos baseados na Ethereum e hospedados no GitHub, tendo como objetivo avaliar o gás (unidade que mede a quantidade de esforço computacional necessário para realizar as transações na Ethereum) necessário para *deploy*. Adicionalmente, para cada um dos contratos investigados, foi extraído um conjunto de métricas orientadas a objeto para avaliar características estruturais. Especificamente, analisou-se se há uma correlação entre as métricas orientadas a objetos de um contrato inteligente e a quantidade de gás utilizado. Os trabalhos mencionados apresentam uma rica compreensão sobre os aspectos relacionados ao processo de desenvolvimento de CIs da plataforma Ethereum. Assim, constata-se uma lacuna teórica no que tange 1) a análise do efeito que o processo de *deploy* na Ethereum exerce sobre os projetos *open source* em termos colaborativos e 2) na compreensão sobre o que efetivamente mudou nos repositórios após tal processo.

## Procedimentos Metodológicos

A presente proposta caracteriza-se como uma pesquisa exploratória cujo objetivo visa possibilitar maior familiaridade com o problema, com vistas a torná-lo mais explícito (GIL, 2002). Pautado a partir de um experimento computacional baseado em Mineração de Repositório de Software (MRS), este trabalho se propõe a investigar quali-quantitativamente o impacto que o 1) processo de *deploy* na Ethereum exerce sobre os projetos em termos de colaboração *open source* (com foco no número de contribuidores, *commits* e quantidades de *issues*) e 2) na compreensão sobre o que mudou quanto à evolução de software em tais repositórios. Justifica-se o uso de MRS devido à capacidade de obter, em larga escala, evidências empíricas sobre os dados dos repositórios e, conseqüentemente, revelar *insights* pertinentes sobre características evolutivas de software (KAGDI *et al.*, 2007). Os procedimentos metodológicos que orientam esta pesquisa são organizados em três etapas, conforme Figura 1.

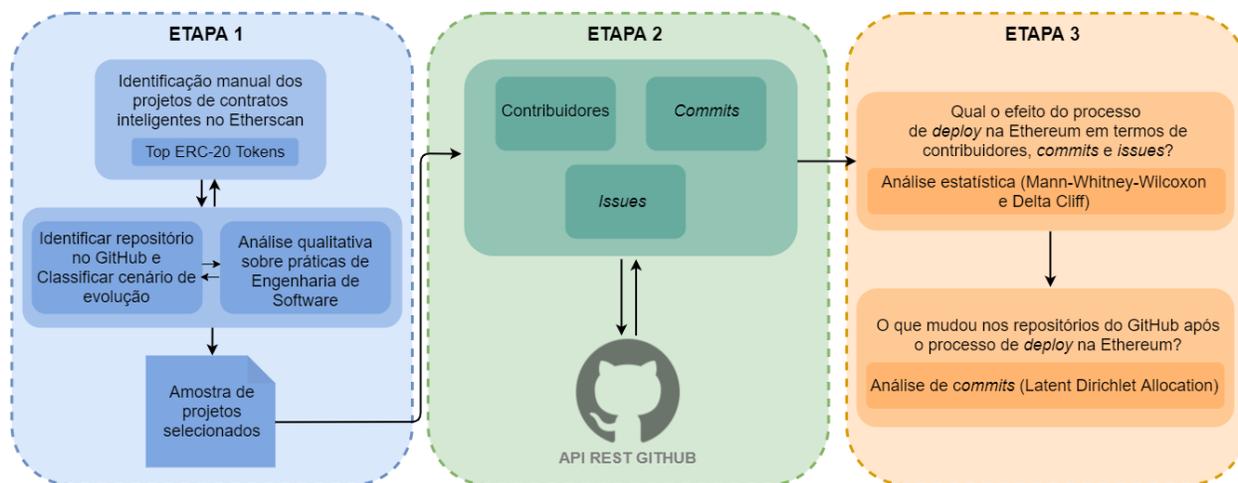


Figura 1 - Procedimentos Metodológicos

A **Etapa 1** (tracejada com cor azul), inicia-se com a identificação manual de projetos que estão entre aqueles com maior valor de mercado no Etherscan de acordo com o *ranking ERC20 Top Tokens*. Além de prover tal *ranking* com os projetos mais valorizados, o Etherscan funciona como um *block explorer* o qual viabiliza acesso ao código-fonte dos CIs implantados na Ethereum, bem como a data de *deploy*. Para cada

projeto do referido *ranking*, realizar-se-á um procedimento manual baseado em dois passos visando a curadoria de 384 repositórios (o qual reflete uma amostra estatisticamente significativa considerando a população atual de projetos listados no *ranking ERC20 Top Tokens* do Etherscan):

- 1) Identificar o repositório (quando disponível) do projeto no GitHub e classificá-lo com base em três cenários de evolução: a) a data de *deploy* disponível no Etherscan é posterior ao último *commit* no GitHub; b) a data de *deploy* no Etherscan está dentro do intervalo no GitHub e c) a data de *deploy* no Etherscan é anterior ao primeiro *commit* no GitHub. No contexto deste trabalho, somente os projetos classificados no cenário b serão analisados, haja vista a necessidade de investigar os dados antes e após a data de *deploy*;
- 2) Analisar qualitativamente se os repositórios refletem características de um projeto com práticas de engenharia de software, ou seja, não é utilizado apenas para exposição de código-fonte. Será adaptado e explorado o *framework* proposto por Munaiah *et al.* (2017) o qual avalia diferentes dimensões (arquitetura, comunidade, etc) para curadoria de repositórios *open source*.

Na **Etapa 2** (tracejada com cor verde), após a consolidação da amostra de repositórios disponíveis no GitHub, o processo de mineração de dados ocorrerá com a finalidade de obter os dados sobre os contribuidores, *commits* e *issues* para cada repositório da amostra. Para isso, *scripts* em python serão implementados para concretizar a coleta dos dados a partir da API REST do GitHub.

Finalmente, na **Etapa 3** (tracejada com cor laranja), têm-se as análises dos dados obtidos previamente. Para análise do efeito colaborativo do *deploy* na Ethereum sobre os repositórios *open source*, será comparado, através do teste não-paramétrico Mann-Whitney-Wilcoxon, a diferença estatística entre as métricas avaliadas (quantidade de contribuidores, quantidade de *commits* e quantidade de *issues*) antes e após a data de *deploy* na Ethereum. A partir desses resultados, será possível evidenciar o impacto que o processo de *deploy* exerceu sobre os repositórios. Em seguida, de forma complementar, será investigado efetivamente o que mudou em relação à evolução de software dos projetos analisados. Nesse sentido, será empregada a técnica de *Topic Modelling* denominada Latent Dirichlet Allocation (LDA) com o objetivo de identificar, a partir dos *logs* de *commits* dos repositórios, os tópicos que denotam as mudanças concretizadas. Conforme discutido por Pratt *et al.* (2011), o LDA é uma alternativa bem-vinda para classificação de *logs* de *commits*, pois permite descobrir o conteúdo atual de uma grande coleção de documentos de forma não-supervisionada. Considerando tais análises, tem-se a possibilidade de avançar na compreensão de aspectos colaborativos e de evolução de software relacionados ao processo de desenvolvimento *open source* de CIs implantados na Ethereum.

## Considerações Finais

O processo de desenvolvimento de CIs reflete particularidades diferentes do escopo de softwares tradicionais. Nesse sentido, a exploração de práticas *open source* tem se mostrado frequente no domínio de CIs, especialmente através do uso do GitHub. Diante desse contexto, este trabalho contribui ao discutir uma proposta metodológica baseada em Mineração em Repositório de Software para análise de aspectos colaborativos e de evolução de CIs. Almeja-se, como resultado esperado da presente pesquisa em andamento, avançar no entendimento do desenvolvimento *open source* de CIs. Como trabalhos futuros, pretende-se realizar a implementação da presente proposta, incluindo a curadoria dos repositórios e análise dos dados.

## Referências

- AJIENKA, N.; VANGORP, P.; CAPILUPPI, A. An empirical analysis of source code metrics and smart contract resource consumption. *Journal of Software: Evolution and Process*, Wiley Online Library, v. 32, n. 10, 2020.
- BECK, R.; AVITAL, M.; ROSSI, M.; THATCHER, J. B. *Blockchain technology in business and information systems research*. [S.l.]: Springer, 2017.
- BODÓ, B.; GERVAIS, D.; QUINTAIS, J. P. *Blockchain and smart contracts: the missing link in copyright licensing?* *International Journal of Law and Information Technology*, Oxford University Press, 2018.

- CHAKRABORTY, P.; SHAHRIYAR, R.; IQBAL, A.; BOSU, A. Understanding the software development practices of blockchain projects: a survey. In: ACM. Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. [S.l.], 2018. p. 28.
- CHEN, J.; XIA, X.; LO, D.; GRUNDY, J.; YANG, X. Maintaining smart contracts on Ethereum: Issues, techniques, and future challenges. arXiv preprint arXiv:2007.00286, 2020.
- ETHERSCAN. Ranking ERC20 Top Tokens. 2021. Disponível em: <<https://etherscan.io/tokens>>. Acesso em: 02/07/2021.
- GIL, A. C. Como classificar as pesquisas. Como elaborar projetos de pesquisa, Atlas São Paulo, 2002.
- HASSAN, A. E. The road ahead for mining software repositories. In: IEEE. 2008 Frontiers of Software Maintenance. [S.l.], 2008. p. 48–57.
- HEGEDŰS, P. Towards analyzing the complexity landscape of solidity based ethereum smart contracts. Technologies, Multidisciplinary Digital Publishing Institute, v. 7, n. 1, p. 6, 2019.
- KAGDI, H.; COLLARD, M. L.; MALETIC, J. I. A survey and taxonomy of approaches for mining software repositories in the context of software evolution. Journal of software maintenance and evolution: Research and practice, Wiley Online Library, v. 19, n. 2, p.77–131, 2007.
- KASSAB, M.; DEFRANCO, J.; MALAS, T.; DESTEFANIS, G.; NETO, V. V. G. Investigating quality requirements for blockchain-based healthcare systems. In: IEEE. 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). [S.l.], 2019. p. 52–55.
- KOCH, S. Software evolution in open source projects—a large-scale investigation. Journal of Software Maintenance and Evolution: Research and Practice, Wiley Online Library, v. 19, n. 6, p. 361–382, 2007.
- MOHANTA, B. K.; PANDA, S. S.; JENA, D. An overview of smart contract and use cases in blockchain technology. In: IEEE. 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT). [S.l.], 2018. p. 1–4.
- MUNAIHAH, N.; KROH, S.; CABREY, C.; NAGAPPAN, M. Curating github for engineered software projects. Empirical Software Engineering, Springer, v. 22, n. 6, p. 3219–3253, 2017.
- OLIVA, G. A.; HASSAN, A. E.; JIANG, Z. M. J. An exploratory study of smart contracts in the ethereum blockchain platform. Empirical Software Engineering, Springer, p. 1–41, 2020.
- PINNA, A.; IBBA, S.; BARALLA, G.; TONELLI, R.; MARCHESI, M. A massive analysis of ethereum smart contracts empirical study and code metrics. IEEE Access, IEEE, v. 7, p.78194–78213, 2019.
- PRATT, L. J.; MACLEAN, A. C.; KNUTSON, C. D.; RINGGER, E. K. Cliff walls: An analysis of monolithic commits using latent dirichlet allocation. In: SPRINGER. IFIP International Conference on Open Source Systems, 2011.
- SAJANA, P.; SINDHU, M.; SETHUMADHAVAN, M. On blockchain applications: hyperledger fabric and ethereum. International Journal of Pure and Applied Mathematics, v. 118, n. 18, p. 2965–2970, 2018.
- SAYEED, S.; MARCO-GISBERT, H.; CAIRA, T. Smart contract: Attacks and protections. IEEE Access, IEEE, v. 8, p. 24416–24427, 2020.
- SUBRAMANIAM, C.; SEN, R.; NELSON, M. L. Determinants of open source software project success: A longitudinal study. Decision Support Systems, Elsevier, v. 46, n. 2, p. 576–585, 2009.
- TIKHOMIROV, S.; VOSKRESENSKAYA, E.; IVANITSKIY, I.; TAKHAVIEV, R.; MARCHENKO, E.; ALEXANDROV, Y. Smartcheck: Static analysis of Ethereum smart contracts. In: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. 2018. p. 9–16.
- VIRIYASITAVAT, W.; HOONSOPON, D. Blockchain characteristics and consensus in modern business processes. Journal of Industrial Information Integration, Elsevier, v. 13, p. 32–39, 2019.
- YUAN, Y.; WANG, F.-Y. Blockchain and cryptocurrencies: Model, techniques, and applications. IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE, v. 48, n. 9, p.1421–1428, 2018.