Arquitetura híbrida para Loterias em Blockchain com compressão de estado via Merkle Tree

Romulo de Moraes[1], Arthur G. Bubolz[1], Denner G. Ayres[1], Vinícius G. Pinto[1], Bruno L. Dalmazo[1]

[1]Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG)
Caixa Postal 474 – 96201-900 – Rio Grande – RS – Brazil

{demoraes.romulo, arthurgomesbubolz, dennerayres}@furg.br

{vinicius.pinto, dalmazo}@furg.br

Abstract. *This paper presents an architecture for decentralized lotteries on the Ethereum network, based on smart contracts, verifiable randomness, and storage optimization using a Merkle Tree. The proposed design aims to reduce the average transaction cost (gas fees) and improve on-chain scalability by comparing three storage approaches: array, mapping, and Merkle Tree. The results show that gas consumption reveals a significant advantage for the Merkle Tree, reducing the total cost by up to three orders of magnitude, confirming its efficiency and suitability for high-demand decentralized applications.*

Resumo. *Este artigo apresenta uma arquitetura para loterias descentralizadas na rede Ethereum, baseada em contratos inteligentes, aleatoriedade verificável e otimização de armazenamento por meio de Merkle Tree. A proposta visa reduzir o custo médio das transações (gas fees) e aprimorar a escalabilidade on-chain, comparando três abordagens distintas de armazenamento: array, mapping e Merkle Tree. Os resultados mostram que o consumo de gas evidencia uma vantagem expressiva da Merkle Tree, reduzindo em até três ordens de magnitude o custo total, o que confirma sua eficiência e potencial para aplicações descentralizadas de alta demanda.*

1. Introdução

Loterias são sistemas de distribuição de prêmios baseados em sorteios, nos quais os participantes adquirem bilhetes e aguardam a seleção aleatória dos vencedores. Nas loterias tradicionais, uma entidade centralizada controla processos-chave como emissão de bilhetes, realização do sorteio e pagamento dos prêmios. Frequentemente, mantém sob sigilo detalhes do procedimento e do algoritmo de seleção dos números [Sher 1980]. Essa falta de transparência restringe o direito de qualquer interessado em verificar, de forma independente, a correção e a integridade do sorteio, gerando preocupações sobre possível manipulação de resultados ou fraude por parte do operador.

Diante desses desafios de confiança, as tecnologias de registro distribuído (*blockchain*) [Nakamoto 2008] e contratos inteligentes (*smart contracts*) [Filippi et al. 2021] emergem como alternativas robustas para aumentar a transparência em sistemas de loteria. Em uma blockchain pública como Ethereum [Buterin 2013], todas as transações, incluindo a compra de bilhetes e chamadas aos contratos, são registradas de forma imutável

e acessível a qualquer interessado. Assim, o código do contrato inteligente que implementa o sorteio torna-se auditável publicamente, e sua execução automática elimina a necessidade de um operador central de confiança.

No entanto, a execução de contratos inteligentes enfrenta limitações inerentes de escalabilidade, especialmente relacionadas ao custo de armazenamento e modificação do estado na *Ethereum Virtual Machine (EVM)*. Cada operação executada em um contrato inteligente consome uma unidade de medida denominada *gas*, que representa a quantidade de esforço computacional necessária para processar a transação [Buterin 2013]. Operações que gravam dados permanentes, como adicionar elementos a um *array* ou atualizar um *mapping*, consomem substancialmente mais *gas* do que operações de leitura, pois demandam escrita em memória persistente da blockchain. Conforme documentado em [Antonopoulos and Wood 2018], essas instruções de escrita são ordens de magnitude mais caras devido ao custo energético e ao esforço de consenso distribuído necessários para validar o novo estado entre todos os nós da rede. Como consequência, aplicações que exigem múltiplas inserções (como o registro de milhares de participantes em uma loteria) apresentam crescimento linear de custo, tornando-se economicamente inviáveis na camada principal da Ethereum.

Para contornar essa limitação, adota-se a estratégia de mesclar processamento *off-chain* e *on-chain*, delegando tarefas computacionalmente intensivas a componentes externos e mantendo apenas as informações essenciais na blockchain. Essa abordagem híbrida reduz o consumo de *gas* ao custo de reintroduzir um pequeno grau de confiança no operador responsável pelo processamento externo, um compromisso aceitável entre descentralização absoluta e viabilidade prática [Bubolz et al. 2025]. Essa linha de raciocínio orienta o desenvolvimento da arquitetura apresentada neste artigo, que propõe o uso de estruturas criptográficas (*Merkle Trees*) e oráculos de aleatoriedade verificável para otimizar o custo e a escalabilidade de sorteios descentralizados.

A estrutura deste trabalho é a seguinte: na Seção 2, apresentamos os principais estudos relacionados. A proposta é detalhada na Seção 3. Os testes e os resultados obtidos são discutidos na Seção 4, e por fim, encerramos com as considerações finais e trabalhos futuros na Seção 5.

2. Trabalhos Relacionados

Nos últimos anos, diversas pesquisas têm explorado o uso de blockchain e contratos inteligentes para automatizar e auditar sorteios de forma descentralizada. A maioria das abordagens busca garantir transparência, imparcialidade e aleatoriedade verificável.

[Hire et al. 2023] propuseram uma loteria descentralizada em Ethereum, automatizando o registro de participantes, seleção de ganhadores e transferência de prêmios via oráculo de aleatoriedade verificável (VRF) [Breidenbach et al. 2021]. Embora garanta verificabilidade e elimine intermediários, o estudo não apresenta métricas quantitativas de desempenho, como consumo de *gas* ou tempo de execução, limitando a análise de escalabilidade.

[Pan et al. 2022] introduziram o *FPLotto*, que combina VRF, interpolação de Lagrange e *zk-SNARKs* (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) para preservar a privacidade dos participantes. Apesar de fornecer provas crip-

tográficas auditáveis, a complexidade computacional e o alto custo de verificação inviabilizam sua adoção em ambientes públicos de larga escala.

[Zhang et al. 2024] propuseram a SM2VRF, uma função aleatória verificável baseada em criptografia SM2, demonstrando redução de 68% no tempo de verificação com provas em lote. Entretanto, sua adoção é limitada pela falta de compatibilidade com o ecossistema Ethereum.

Em síntese, embora as soluções existentes avancem na geração segura de aleatoriedade, elas negligenciam a otimização do estado *on-chain*. Poucos trabalhos discutem a compressão de dados ou mecanismos escaláveis de armazenamento em sorteios. Essa lacuna motiva a presente pesquisa, que propõe o uso de Merkle Trees como meio de reduzir o consumo de *gas* mantendo a verificabilidade e integridade dos dados.

3. Proposta

A arquitetura proposta busca mitigar os gargalos de escalabilidade presentes em contratos inteligentes, especialmente aqueles relacionados às operações que modificam o estado da blockchain [G. Bubolz et al. 2025]. Conforme discutido anteriormente, gravações em estruturas como *arrays* e *mappings* acarretam alto custo de *gas* devido à natureza persistente do armazenamento da EVM [Antonopoulos and Wood 2018]. Para contornar essa limitação, a solução adota uma abordagem híbrida, combinando processamento *off-chain* e *on-chain*, com o objetivo de reduzir o consumo de recursos computacionais e otimizar o custo por transação. O sistema é estruturado em quatro camadas interdependentes (Figura 1).

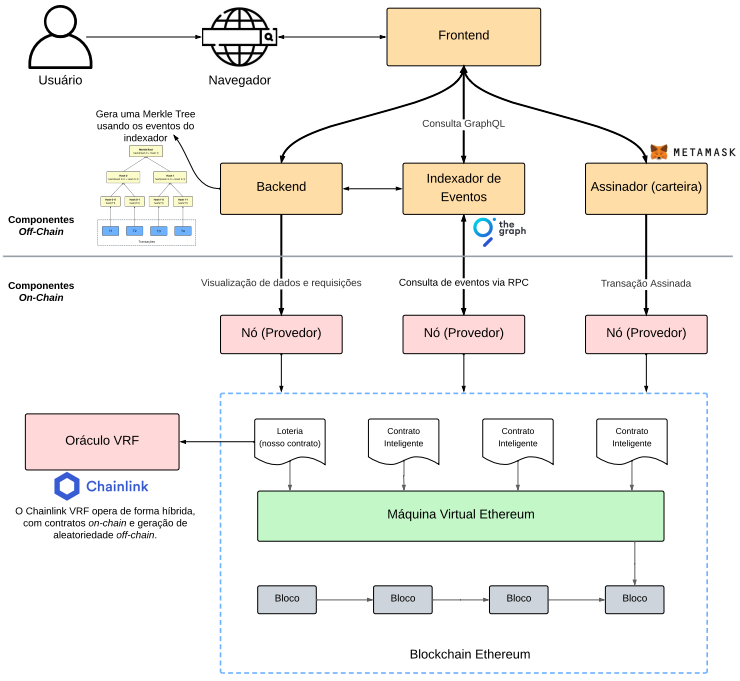

Figura 1. Arquitetura híbrida, integrando componentes *on-chain* e *off-chain*.

A principal inovação da arquitetura reside na compressão de estado *on-chain*. Em vez de armazenar individualmente os endereços dos participantes, o contrato mantém

apenas um resumo criptográfico do conjunto total por meio da raiz da Merkle Tree. Essa estratégia transforma o custo de armazenamento, que seria linear em relação ao número de bilhetes ($O(N)$), em custo constante ($O(1)$) para cada atualização da raiz.

As provas de inclusão (*Merkle Proofs*) podem ser verificadas *on-chain*, permitindo que qualquer usuário comprove sua participação de forma independente. Essa característica preserva a auditabilidade e a integridade do sistema, mesmo com parte do processamento sendo realizado fora da blockchain. Assim, a arquitetura mantém transparência e verificabilidade pública, com uma economia significativa de *gas* em comparação a implementações puramente *on-chain*.

Embora o processamento *off-chain* introduza um ponto de confiança no operador responsável por gerar a Merkle Tree (dono do contrato), que em um cenário adverso pode atuar como agente malicioso, essa dependência é amplamente mitigada pela natureza determinística das provas criptográficas e pela possibilidade de verificação independente por parte dos usuários. Com isso, o sistema preserva um equilíbrio adequado entre descentralização, eficiência e auditabilidade, aspectos fundamentais para aplicações em larga escala na Ethereum [Buterin 2013, Santo et al. 2023].

Qualquer alteração, omissão ou manipulação nos dados de entrada, como a exclusão de um bilhete válido, gera necessariamente uma raiz distinta, cuja inconsistência pode ser identificada por meio de uma verificação simples. Como todos os eventos relevantes são públicos, cada usuário pode reconstruir sua própria prova de inclusão e compará-la com a raiz registrada no contrato. Caso haja discrepância, a tentativa de fraude se torna imediatamente evidente. Assim, o mecanismo de verificação autônoma garante que comportamentos maliciosos por parte do operador *off-chain* sejam detectáveis, impedindo que a segurança do sistema dependa exclusivamente da sua honestidade.

4. Avaliação e Discussão

A avaliação experimental foi conduzida localmente em uma rede simulada utilizando o framework Hardhat [Hardhat - Ethereum Foundation sd]. Foram comparadas três estratégias para armazenamento de bilhetes, *array*, *mapping* e Merkle Tree, em cenários variando de 5 a 100.000 bilhetes. O objetivo foi mensurar tanto o consumo de *gas* quanto o tempo de execução das principais funções do contrato. Para possibilitar a simulação de cenários com grande volume de participantes, o limite de *gas* da rede local foi propositalmente elevado. Observou-se que as abordagens tradicionais rapidamente atingem o limite de *gas* da EVM devido ao custo proporcional ao número de inserções, ao passo que a solução baseada em eventos empregada pela Merkle Tree permanece funcional, pois não se aproxima desse limite mesmo em cenários de alto volume.

Os resultados (Figura 2) indicam que a abordagem com Merkle Tree apresenta maior tempo de execução devido ao custo computacional de *hashing* ($O(N \log N)$), enquanto as estruturas *array* e *mapping* escalam linearmente ($O(N)$). No entanto, apesar dessa diferença no tempo de processamento, o consumo de *gas* (Figura 3) evidencia uma vantagem expressiva da Merkle Tree, reduzindo em até três ordens de magnitude o custo total para 100 mil participantes. Em termos práticos, isso implica que loterias descentralizadas de grande porte tornam-se inviáveis com abordagens tradicionais, uma vez que o custo de gravação na EVM cresce proporcionalmente ao número de bilhetes inseridos, enquanto a Merkle Tree mantém esse custo praticamente constante.

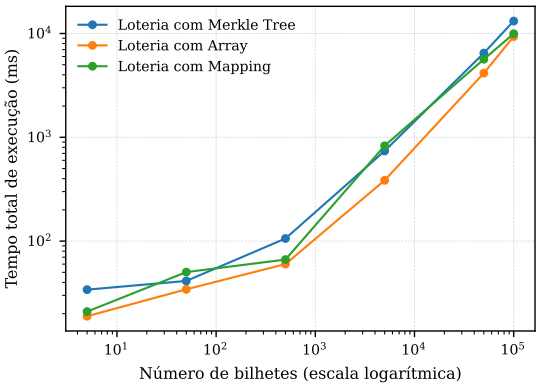

Figura 2. Tempo de execução

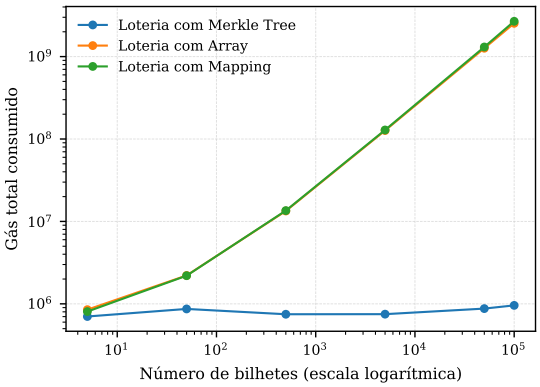

Figura 3. Consumo de *gas*

É fundamental notar que a geração da Merkle Tree ocorreu apenas após o conjunto completo de bilhetes estar definido, simulando um sorteio periódico. Além disso, o tempo de execução medido reflete tanto a preparação da transação quanto a inclusão do bloco na rede local, que é minerado sob demanda no Hardhat [Hardhat - Ethereum Foundation sd]. Por fim, vale ressaltar que apenas o tempo de execução é afetado pelas características da máquina utilizada, ou seja, o consumo de gás, por outro lado, é uma propriedade determinística da EVM e permanece invariável entre diferentes ambientes.

5. Considerações Finais

Este trabalho apresentou uma arquitetura híbrida para loterias descentralizadas em blockchain, combinando contratos inteligentes, aleatoriedade verificável e compressão criptográfica de estado. A proposta visa conciliar transparência, auditabilidade e viabilidade econômica, enfrentando diretamente o desafio do custo elevado de operações que modificam o estado *on-chain* na EVM [Antonopoulos and Wood 2018].

Através do uso do oráculo Chainlink VRF [Breidenbach et al. 2021], o sistema garante geração de aleatoriedade justa e comprovável, eliminando a necessidade de confiar em um operador central. A integração de estruturas de Merkle Tree permite representar grandes volumes de participantes de forma condensada, reduzindo o consumo de *gas* em até três ordens de magnitude em comparação a modelos tradicionais baseados em *arrays* e *mappings*. É importante destacar que o modelo híbrido implica um pequeno grau de confiança no operador responsável pelo cálculo das raízes de Merkle. Entretanto, esse risco é mitigado pela capacidade de qualquer usuário reconstruir a árvore e verificar a autenticidade dos dados, garantindo descentralização prática mesmo em ambientes parcialmente delegados. Do ponto de vista científico, este trabalho contribui ao evidenciar que otimizações estruturais, mais do que apenas soluções de segunda camada, podem redefinir o custo-benefício de aplicações baseadas em contratos inteligentes.

Como trabalhos futuros, pretende-se formalizar a arquitetura proposta como um padrão mais geral para aplicações descentralizadas com grande número de participantes e compressão de estado via Merkle Tree, tendo a loteria como um caso particular. Também será investigado o uso de *Sparse Merkle Trees* em memória (em ambientes L2), e mecanismos de mitigação da confiança no operador *off-chain*, como auditorias amostrais e esquemas básicos de *fraud proofs*, reforçando a auditabilidade do sistema.

Referências

Antonopoulos, A. M. and Wood, G. (2018). *Mastering Ethereum: Building Smart Contracts and DApps*. O'Reilly Media.

Breidenbach, L., Cachin, C., Chan, B., Coventry, A., Ellis, S., Juels, A., Koushanfar, F., Miller, A., Magauran, B., and et al., D. M. (2021). Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Chainlink Labs*, 1:1–136.

Bubolz, A. G., Lucca, G., Oliveira, L. d. S., Teixeira, T., Berri, R. A., Borges, E. N., and Dalmazo, B. L. (2025). Analysis of bitcoin trends through the integration of on-chain financial indicators and machine learning. In Gervasi, O., Murgante, B., Garau, C., Karaca, Y., Taniar, D., C. Rocha, A. M. A., and Apduhan, B. O., editors, *Computational Science and Its Applications – ICCSA 2025*, pages 35–50, Cham. Springer Nature Switzerland.

Buterin, V. (2013). Ethereum white paper. *GitHub Repository*. Available at: `https://ethereum.org/en/whitepaper/`.

Filippi, P. D., Wray, C., and Sileno, G. (2021). Smart contracts. *Internet Policy Review*, 10(2):1–9.

G. Bubolz, A., C. Freitas, M., Lucca, G., A. Berri, R., N. Borges, E., and L. Dalmazo, B. (2025). Towards bitcoin trend prediction: A machine learning approach using blockchain-derived data. In Martínez, L., Camacho, D., Yin, H., Dutta, B., Yera, R., Rodríguez Domínguez, R. M., and Tallón-Ballesteros, A., editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2025*, pages 471–482, Cham. Springer Nature Switzerland.

Hardhat - Ethereum Foundation (s.d.). Hardhat documentation. `https://hardhat.org/`. Acesso em: 6 jul. 2025.

Hire, A., Lanjewar, H., Haridas, P., Jadhav, M., and Rane, M. (2023). Decentralized lottery using blockchain. In *Proceedings of the 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN)*, pages 1035–1041. IEEE.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Online: `https://bitcoin.org/bitcoin.pdf`*.

Pan, Y., Zhao, Y., Liu, X., Wang, G., and Su, M. (2022). Fplotto: A fair blockchain-based lottery scheme for privacy protection. In *Proceedings of the 2022 IEEE International Conference on Blockchain (Blockchain)*, pages 21–28. IEEE.

Santo, Y., Immich, R., Dalmazo, B. L., and Riker, A. (2023). Fault detection on the edge and adaptive communication for state of alert in industrial internet of things. *Sensors*, 23(7).

Sher, G. (1980). What makes a lottery fair? *Noûs*, pages 203–216.

Zhang, Y., Yang, J., Lei, H., Bao, Z., Lu, N., Shi, W., and Chen, B. (2024). Verifiable random function schemes based on sm2 digital signature algorithm and its applications for committee elections. *IEEE Open Journal of the Computer Society*.