# PQCinBlock: uma ferramenta para análise do impacto de algoritmos de Criptografia Pós-Quântica em blockchains via benchmark e simulação

**Alison G. Schemitt [1], Marco A. C. Rodegheri [1], Henrique C. P. Horch [1]**
**Roben C. Lunardi [1,2], Avelino F. Zorzo [1], Henrique F. da Silva [3],**
**Diego Kreutz [3], Rodrigo B. Mansilha [3]**

[1]PPGCC – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
[2]Instituto Federal do Rio Grande do Sul (IFRS)
[3]AI Horizon Labs | PPGES – Universidade Federal do Pampa (UNIPAMPA)

{alison.schemitt, marco.rodegheri, henrique.horch}@edu.pucrs.br
roben.lunardi@zonanorte.ifrs.edu.br, avelino.zorzo@pucrs.br,
{henriquefan.aluno, diegokreutz, mansilha}@unipampa.edu.br

***Abstract.*** *We present PQCinBlock, a tool to evaluate the performance and storage impact of Post-Quantum Digital Signature Algorithms on blockchain systems through benchmarks and simulations. Our work has two main objectives: (1) the direct comparison with classical algorithms (e.g., ECDSA) and (2) different computational environment tests. PQCinBlock integrates with blockchain simulators (e.g., BlockSim), offering a practical way to analyze how different signature algorithms affect the behavior and performance of blockchain networks. This approach enables the evaluation of everything from processing times and the storage space required for cryptographic artifacts to the impacts of using these algorithms in blockchains.*

***Resumo.*** *Apresentamos o PQCinBlock, uma ferramenta para avaliação de impactos de desempenho e de armazenamento para algoritmos de assinatura digital pós-quântica em sistemas blockchain através de benchmarks e simulações. Visamos a dois objetivos principais: (1) a comparação direta com esquemas clássicos (e.g., ECDSA) e (2) testes em ambientes computacionais diversos. O PQCinBlock possui integração com simuladores de blockchain (e.g., BlockSim), oferecendo uma forma prática de analisar como diferentes algoritmos de assinatura afetam o comportamento e o desempenho de redes blockchain. Essa abordagem permite avaliar desde tempos de processamento e o espaço necessário para armazenar os artefatos criptográficos até impactos do uso desses algoritmos em blockchains.*

## 1. Introdução

A computação quântica, que explora fenômenos da mecânica quântica para resolver determinados problemas computacionalmente complexos [Upama et al. 2022], tem recebido crescente atenção da comunidade científica nos últimos anos. A disponibilidade de um Computador Quântico Criptograficamente Relevante (CRQC) representará uma ameaça significativa aos sistemas criptográficos atuais, pois o algoritmo de Shor [Shor 1994] permite resolver, em tempo polinomial, problemas matemáticos fundamentais como a fatoração de inteiros e o logaritmo discreto. Consequentemente, algoritmos criptográficos amplamente utilizados, como RSA e ECC, que dependem da dificuldade computacional desses problemas, poderão tornar-se vulneráveis.

O processo de padronização de algoritmos de criptografia pós-quântica (PQC) conduzido pelo *National Institute of Standards and Technology* (NIST) resultou, até o momento, na seleção de dois esquemas de Encapsulamento de Chaves (KEM), ML-KEM e HQC, e três esquemas de Assinatura Digital (DSA), *ML-DSA*, *SLH-DSA* e Falcon [NIST 2024, Alagic et al. 2025]. Os algoritmos ML-KEM, ML-DSA e SLH-DSA correspondem, respectivamente, aos esquemas originalmente submetidos como CRYSTALS-Kyber, CRYSTALS-Dilithium e SPHINCS+, que foram selecionados pelo NIST e posteriormente padronizados. Falcon e HQC também foram selecionados pelo NIST, mas ainda não haviam recebido, à época considerada, uma versão final de seus respectivos padrões. Além disso, o NIST conduz um processo adicional de padronização voltado à seleção de novos esquemas de assinatura digital baseados em problemas matemáticos distintos dos utilizados nos algoritmos já selecionados.

As *blockchains* dependem fortemente de mecanismos criptográficos para garantir a integridade, a autenticidade e a segurança das transações. O *Elliptic Curve Digital Signature Algorithm* (ECDSA), empregado em redes amplamente utilizadas, como *Bitcoin* e *Ethereum*, baseia sua segurança na dificuldade do problema do logaritmo discreto em curvas elípticas e, portanto, é vulnerável a ataques quânticos baseados no algoritmo de Shor [Shor 1994, Yang et al. 2024]. Essa vulnerabilidade evidencia a necessidade de avaliar e, progressivamente, adaptar as infraestruturas *blockchain* para o uso de mecanismos criptográficos resistentes a ataques quânticos. Entretanto, a adoção de algoritmos de criptografia pós-quântica (PQC) pode introduzir diferentes custos computacionais e impactos de desempenho, tornando necessária uma análise sistemática de seu comportamento em diferentes sistemas e ambientes computacionais.

Como um passo importante para auxiliar desenvolvedores e pesquisadores a compreender o impacto de novos algoritmos de assinatura digital em sistemas *blockchain*, e apoiar nossa pesquisa [Schemitt et al. 2025, Schemitt et al. 2026], desenvolvemos o PQCinBlock[1], uma ferramenta modular de *benchmark*, automatizada e integrada com simuladores de *blockchain*. A ferramenta foi concebida com três objetivos principais: (*i*) análise comparativa entre algoritmos clássicos e pós-quânticos, (*ii*) extensibilidade para incorporação de novos esquemas criptográficos e diferentes simuladores, e (*iii*) adaptabilidade a diferentes ambientes computacionais, desde simulações até sistemas em produção.

O PQCinBlock combina duas camadas de avaliação. A primeira realiza medições dos tempos de processamento para operações criptográficas (*e.g.*, geração e verificação de assinaturas digitais) e a medição dos tamanhos dos artefatos criptográficos gerados pelos algoritmos. Os resultados alimentam a segunda camada, na qual integramos as métricas obtidas com simuladores de *blockchain* (*e.g.*, *BlockSim* [Alharby and van Moorsel 2020]). Essa integração nos permite aproximar e extrapolar o impacto no desempenho sistêmico e de armazenamento em blockchains causado pelo uso de PQC.

## 2. PQCinBlock: Visão Geral e Implementação

A Figura 1 mostra o fluxo de dados da ferramenta, dividido em três partes principais: Entradas, Etapas de Processamento e Saídas. A seguir, são descritas as etapas e os fluxos da ferramenta, bem como são apresentados os cenários de armazenamento considerados e a implementação da ferramenta.

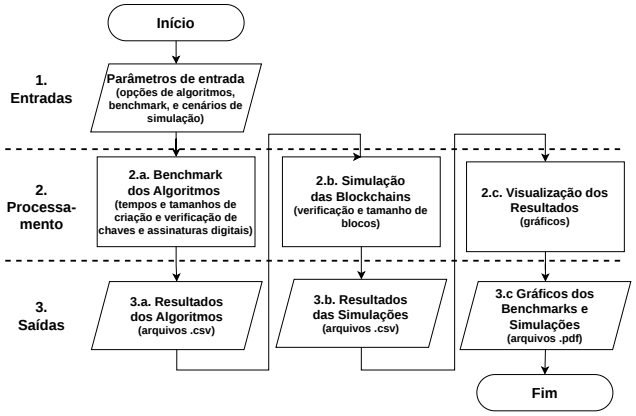
### 2.1. Etapas e Fluxo de Dados

Como **Entradas**, selecionamos quais algoritmos serão avaliados, as opções para o *benchmark* e opções para a simulação, incluindo os cenários de simulação dos artefatos criptográficos. A descrição dos Parâmetros de Entrada está detalhada na Tabela 1.

[1] `https://github.com/conseg/PQCinBlock`

O **Processamento** contém três módulos. O módulo de **Benchmark** executa os algoritmos e níveis selecionados, com o número de execução de estabilização (aquecimento) e de coleta de dados definidos nos parâmetros de entrada. Os dados coletados são o tempo necessário para realização das operações criptográficas e o tamanho dos artefatos criptográficos de cada algoritmo. O módulo de **Simulação** usa os resultados do benchmark para modelar como os algoritmos se comportariam na verificação de blocos em uma rede *blockchain* e o tamanho médio dos blocos gerados utilizando as assinaturas digitais de cada algoritmo. É possível definir o número de execuções da simulação, quais modelos de blockchain (Bitcoin ou Ethereum) a serem simulados e qual cenário de simulação de armazenamento desejado. O módulo de **Visualização** prepara os resultados das etapas anteriores para visualização final.

**Figura 1. Etapas e Fluxo de Dados.**




Como **Saídas**, os resultados ficam disponíveis em formatos estruturados (CSV) e como gráficos, facilitando a análise dos dados brutos e dos resultados das simulações.

**Tabela 1. Descrição dos argumentos da ferramenta PQCinBlock**

| Argumento | Descrição |
|---|---|
| `--help`, `-h`, | Exibe a mensagem de ajuda com a descrição de todos os argumentos disponíveis e instruções de uso da ferramenta. |
| `--list-algorithm` | Exibe todos os algoritmos de assinatura digital disponíveis na ferramenta. |
| `--algorithm` | Lista de algoritmos de assinatura digital a serem avaliados. Suporta múltiplos valores, incluindo algoritmos clássicos (*e.g.*, *ECDSA*) e pós-quânticos (*e.g.*, *Dilithium*, *Falcon*). |
| `--benchmark`, `-r` | Número de execuções de cada algoritmo para coleta de métricas via *benchmark*. |
| `--warm-up`, `-wp` | Número de execuções de aquecimento (*warm-up*) antes do *benchmark* principal, para estabilização de desempenho. |
| `--levels`, `-l` | Define os níveis de segurança do *NIST* (1 a 5) dos algoritmos a serem testados. Pode receber múltiplos valores. |
| `--blockchain-model` | Define o modelo de blockchain disponível no *BlockSim* (i.e., 1: Bitcoin, 2: Ethereum) a ser utilizado. Pode receber múltiplos valores. |
| `--simulation` | Número de execuções da simulação de blockchain usando o *BlockSim*. |
| `--simulation-scenario` | Define o cenário de simulação do impacto dos artefatos criptográficos(i.e., 1: apenas tamanho de assinatura digital, 2: tamanho da assinatura digital + tamanho da chave pública, 3: 1 chave pública e várias assinaturas digitais correspondentes). Padrão: 2 |

## 2.2. Cenários de Armazenamento

Foram definidos 3 cenários de armazenamento para análise e simulação do impacto causado pelos diferentes algoritmos de assinatura digital. O cenário 1 armazena apenas a

assinatura digital na blockchain; esse cenário reflete casos em que a chave pública é recuperada de outra forma que não seja o armazenamento em um bloco. O cenário 2 armazena tanto a chave pública quanto a assinatura digital correspondente na blockchain; esse cenário é análogo ao funcionamento comum da maioria das blockchains. Por fim, o cenário 3 armazena uma única chave pública por bloco e diversas assinaturas digitais correspondentes; esse cenário representa blockchains como a appendable-block blockchain, na qual cada bloco contém diversas transações assinadas com uma mesma chave.

### 2.3. Implementação

O PQCinBlock consiste em três módulos principais e um programa de controle. O módulo de *benchmark* utiliza as bibliotecas Cryptography[2] (versão 45.0.2) para o algoritmo ECDSA e liboqs-python (versão 0.12.0) da Open Quantum Safe[3] para os algoritmos PQC. As bibliotecas são escritas em Python e são *wrappers* para implementações em *C/C++*. As operações criptográficas de geração de chaves, criação da assinatura digital e a verificação da assinatura, com a medição do tempo necessário para cada operação e o tamanho dos artefatos gerados, são feitas de forma sucessiva, com o descarte dos artefatos antigos a cada execução do *benchmark*.

O módulo de *simulação* utiliza uma versão modificada do *BlockSim* [Alharby and van Moorsel 2020], estendida para avaliar o impacto dos algoritmos criptográficos no desempenho da verificação de assinaturas digitais e nos requisitos de armazenamento dos artefatos criptográficos em *blockchains*. A análise considera três cenários de armazenamento de chaves públicas e assinaturas digitais, definidos de acordo com o uso de algoritmos criptográficos clássicos ou pós-quânticos.

O módulo de *visualização* gera gráficos a partir dos dados produzidos pelos outros módulos. O *programa de controle* coordena a execução desses módulos de acordo com os parâmetros de entrada. Atualmente, a ferramenta suporta *Python 3.11.2+* e os algoritmos, com suas variações, listados na Tabela 2.

A ferramenta foi projetada para permitir a adição de novos algoritmos sem modificar os módulos existentes. Para isso, basta criar um novo arquivo *Python* no diretório `algorithms`, contendo obrigatoriamente a variável `ALGORITHMS`, que define os algoritmos e seus respectivos níveis de segurança, e a função `time_evaluation(variant: str, runs: int)`, responsável por realizar as medições de desempenho das operações criptográficas e dos artefatos criptográficos.

## 3. Avaliação

O PQCinBlock foi avaliado em diferentes configurações de *hardware* e sistemas operacionais. A Tabela 3 apresenta os ambientes computacionais testados. Em nossa avaliação, selecionamos o conjunto de algoritmos de assinatura digital pós-quânticos disponíveis na liboqs e o algoritmo clássico ECDSA em todas as variações e níveis de segurança disponíveis. Foram utilizadas todas as variantes dos algoritmos da Tabela 2. Vale notar que o algoritmo `ML-DSA` equivale ao `CRYSTALS-Dilithium` na nova nomenclatura estabelecida pelo *NIST* em 2024 [NIST 2024], enquanto o `Falcon-padded` é uma variação do `Falcon` com assinatura de tamanho fixo.

Utilizando a Tabela 1, configuramos os testes com: 100 execuções de benchmark (`--benchmark 100`), precedidas por 10 de aquecimento (`--warm-up 10`); níveis de segurança `--levels 1 2 3 5`; e 100 simulações de blockchain

[2] `https://cryptography.io/en/45.0.2/`
[3] `https://openquantumsafe.org/`

**Tabela 2. Algoritmos *PQC* disponíveis (como não existe algoritmo de Nível 4 a coluna correspondente foi suprimida).**

| # | Algoritmo (16) | Variantes (46) Nível 1 (14) | Nível 2 (2) | Nível 3 (14) | Nível 5 (16) |
|---|---|---|---|---|---|
| 1 | ECDSA | P-256 | — | P-384 | P-521 |
| 2 | ML-DSA* | — | ML-DSA-44 | ML-DSA-65 | ML-DSA-87 |
| 3 | Dilithium* | — | Dilithium2 | Dilithium3 | Dilithium5 |
| 4 | Falcon | Falcon-512 | — | — | Falcon-1024 |
| 5 | Falcon-padded | Falcon-padded-512 | — | — | Falcon-padded-1024 |
| 6 | MAYO | MAYO-2 | — | MAYO-3 | MAYO-5 |
| 7 | SPHINCS+-SHA-s | SHA2-128s-simple | — | SHA2-192s-simple | SHA2-256s-simple |
| 8 | SPHINCS+-SHA-f | SHA2-128f-simple | — | SHA2-192f-simple | SHA2-256f-simple |
| 9 | SPHINCS+-SHAKE-s | SHAKE-128s-simple | — | SHAKE-192s-simple | SHAKE-256s-simple |
| 10 | SPHINCS+-SHAKE-f | SHAKE-128f-simple | — | SHAKE-192f-simple | SHAKE-256f-simple |
| 11 | CROSS-RSDP-small | CROSS-RSDP-128-small | — | CROSS-RSDP-192-small | CROSS-RSDP-256-small |
| 12 | CROSS-RSDPG-small | CROSS-RSDPG-128-small | — | CROSS-RSDPG-192-small | CROSS-RSDPG-256-small |
| 13 | CROSS-RSDP-balanced | CROSS-RSDP-128-balanced | — | CROSS-RSDP-192-balanced | CROSS-RSDP-256-balanced |
| 14 | CROSS-RSDPG-balanced | CROSS-RSDPG-128-balanced | — | CROSS-RSDPG-192-balanced | CROSS-RSDPG-256-balanced |
| 15 | CROSS-RSDP-fast | CROSS-RSDP-128-fast | — | CROSS-RSDP-192-fast | CROSS-RSDP-256-fast |
| 16 | CROSS-RSDPG-fast | CROSS-RSDPG-128-fast | — | CROSS-RSDPG-192-fast | CROSS-RSDPG-256-fast |

* Inclui versões pré- e pós-padronização.

**Tabela 3. Máquinas testadas**

| Máquina | CPU | Memória | Sistema Operacional |
|---|---|---|---|
| Desktop com WSL | Intel Pentium G5400 (3.7GHz) | WSL: 8 GB<br>Host: 16 GB | WSL: Ubuntu 24.04.4 LTS Linux Kernel 6.18.33.2-microsoft<br>Host: Windows 11 Pro 10.0.22631 |
| Raspberry PI4 | Cortex-A72 (1.8 GHz) | 4 GB | Raspberry Pi OS 12 (bookworm) Linux Kernel 6.12.62-v8+ |

(`--simulation 100`) nos modelos Bitcoin e Ethereum (`--blockchain-model 1 2`) sob os 3 cenários da Seção 2.2. Estes parâmetros servem para demonstração; avaliações mais robustas são detalhadas em [Schemitt et al. 2025, Schemitt et al. 2026].

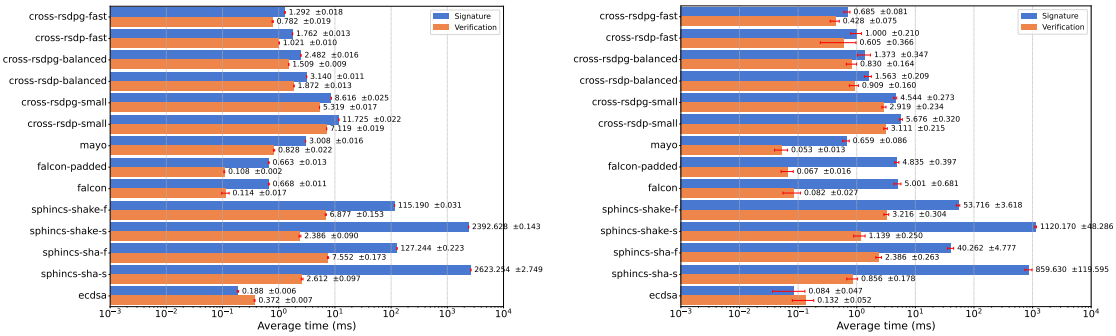
### 3.1. Benchmark dos Algoritmos

Primeiro apresentamos os resultados de desempenho e de armazenamento do benchmark dos algoritmos de assinatura digital, seguidos pela análise do impacto via simulação de *blockchain*. Esta abordagem nos permite examinar os algoritmos tanto individualmente quanto seu comportamento no contexto de um sistema blockchain.

A Figura 2 exibe os resultados obtidos para níveis 1 e 5 de segurança. Nós os selecionamos pois representam os extremos de complexidade computacional. Vale destacar que nem todos os algoritmos se aplicam a todos os níveis (ver Tabela 2). Cada figura mostra o tempo para cada ambiente computacional criar (azul) e verificar (laranja) as assinaturas digitais usando cada algoritmo. Como esperado, os tempos de execução aumentam proporcionalmente à complexidade das operações em níveis mais altos de segurança. Observamos que os tempos médios das operações mostram padrões similares entre ambas as máquinas. Como esperado, o Raspberry Pi 4 apresenta tempos maiores, pois possui menor capacidade computacional. Em geral, o ECDSA é superado na operação de verificação pelos algoritmos Falcon e ML-DSA/Dilithium, com a operação de criação variando em cada algoritmo. Entretanto, as famílias dos algoritmos SPHINCS+ e CROSS operam constantemente mais lentas que ECDSA.

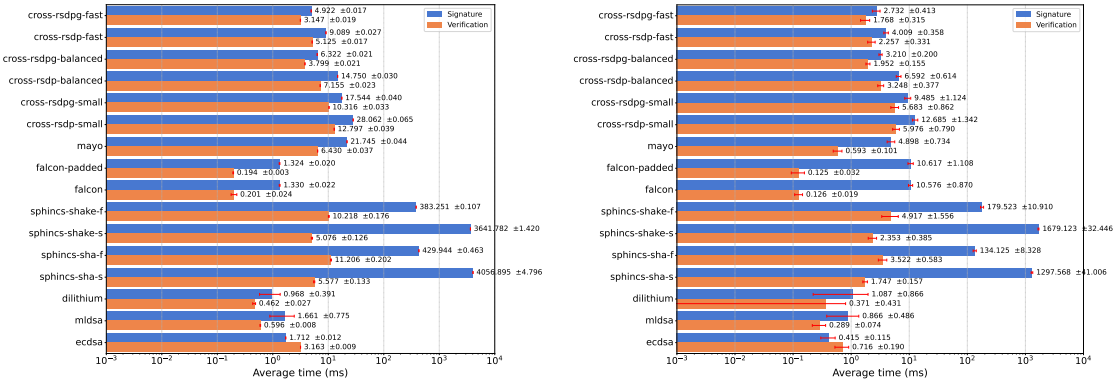
A Figura 3 apresenta o benchmark dos artefatos criptográficos, sem distinção de máquina, pois o resultado é o mesmo para ambas. Apesar da chave pública de alguns algoritmos possuir tamanho similar às chaves do ECDSA, a assinatura digital é consideravelmente maior, sendo um impeditivo para aplicações com armazenamento limitado.

**Figura 2. Resultados de desempenho - Níveis 1 e 5 de segurança**




(a) Raspberry Pi 4 - Nv. 1

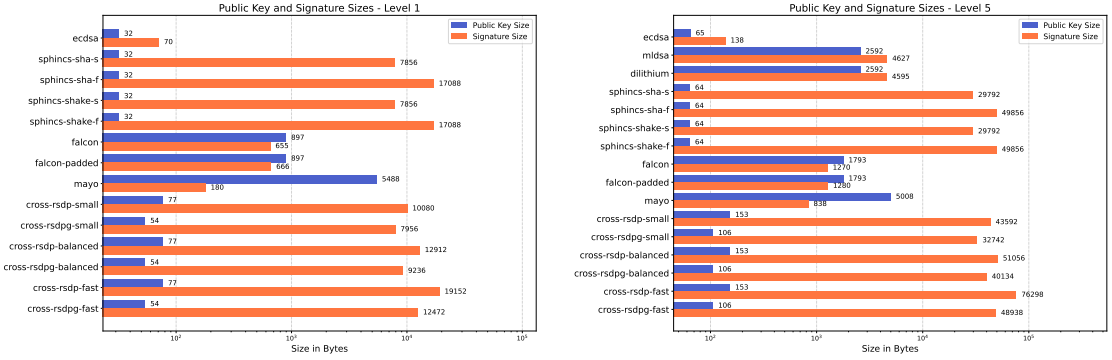
(b) Windows com WSL - Nv. 1




(c) Raspberry Pi 4 - Nv. 5

(d) Windows com WSL - Nv. 5

**Figura 3. Resultados de armazenamento - Níveis 1 e 5 de segurança**




(a) Nível 1

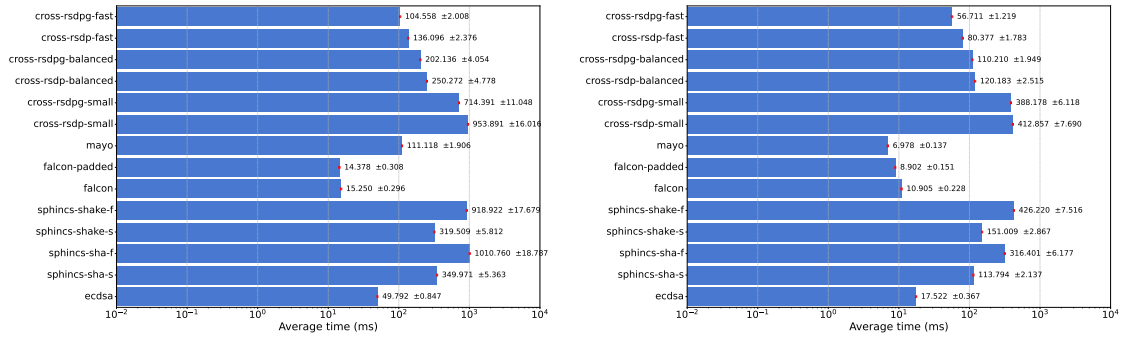
(b) Nível 5

## 3.2. Simulação de Blockchain

Os resultados do *benchmark* alimentam o simulador *BlockSim* conforme os parâmetros da Seção 3 (ver Tabela 1). A Figura 4 mostra a verificação de blocos no nível 1 do Bitcoin, selecionado pois, por ter mais transações por bloco, gera mais assinaturas e torna o impacto pós-quântico mais evidente. Observa-se padrão similar de tempo em ambas as máquinas, proporcional à complexidade das operações e mantendo a relação entre algoritmos, pois a verificação consiste em checar múltiplas assinaturas.

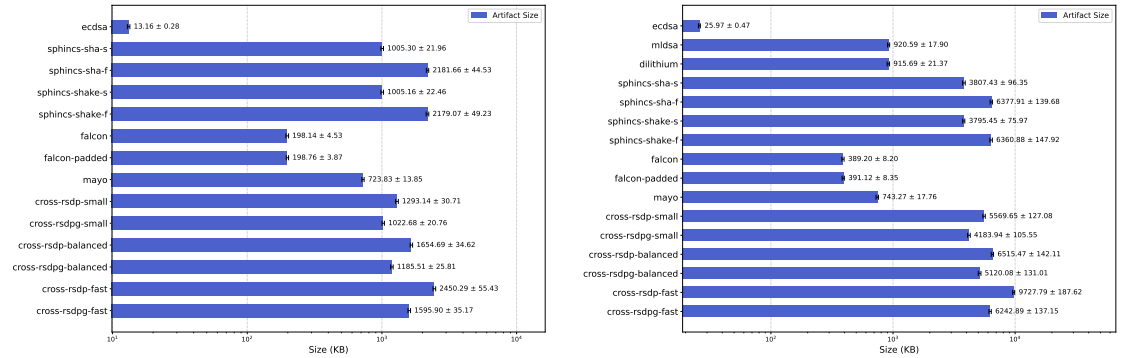
A Figura 5 apresenta a simulação de tamanho dos artefatos criptográficos nos

**Figura 4. Simulação de Desempenho - Nível 1 de segurança (Bitcoin)**




(a) Raspberry Pi 4 - Nv. 1 (b) Windows com WSL - Nv. 1

**Figura 5. Simulação de armazenamento - Níveis 1 e 5 de segurança (Bitcoin)**




(a) Nível 1 - Cenário 2 (b) Nível 5 - Cenário 2

níveis 1 e 5 para o Bitcoin no cenário 2, que armazena a assinatura e a chave pública no mesmo bloco; os cenários 1 e 3 foram omitidos para priorizar o cenário de uma blockchain clássica na demonstração. Novamente apresentamos sem distinção de máquina, pois a diferença entre elas é mínima, decorrente do funcionamento do próprio simulador. Os resultados mostram que há um aumento de $\approx 15\times$ a $\approx 374\times$ dos artefatos criptográficos ao substituir o ECDSA por qualquer outro algoritmo.

## 4. Discussão e Trabalhos Relacionados

A consistência dos resultados entre diferentes configurações de *hardware* sugere que o desempenho relativo dos algoritmos permanece estável. A Tabela 4 mostra que a maioria dos trabalhos relacionados avalia algoritmos PQC de forma isolada, enquanto [Juaristi et al. 2025] aborda *blockchain*, mas com foco em serviços específicos de PQC. O *PQCinBlock* avança nesse cenário ao integrar avaliação criptográfica, simulação de *blockchain* e arquitetura extensível, permitindo analisar tanto o desempenho dos algoritmos quanto seus impactos sistêmicos. A ferramenta já foi utilizada em trabalhos anteriores sobre o tema [Schemitt et al. 2025, Schemitt et al. 2026].

## 5. Considerações Finais

Desenvolvemos o PQCinBlock para comparar sistematicamente o impacto de desempenho e de armazenamento de algoritmos de assinatura digital clássicos e pós-quânticos, com foco especial em aplicações *blockchain*. A ferramenta integra medi-

**Tabela 4. Trabalhos Relacionados: Algoritmos e Simulação.**

| Trabalho | Algoritmos Suportados/Avaliados | Simul. BC |
|---|---|---|
| [Commey et al. 2025] | Kyber, RSA e ECDSA | - |
| [Holmes 2023] | ML-DSA, SLH-DSA, Falcon, LMS/HSS e XMSS/XMSSmt | - |
| [Ramzan and Cimato 2025] | ML-DSA, SLH-DSA e ECDSA | - |
| [Yokubov and Gan 2021] | ML-DSA (Dilithium), Falcon e Rainbow | - |
| [Juaristi et al. 2025] | Dilithium, Falcon, SPHINCS+ e ECDSA | ✓ |
| [Schemitt et al. 2025] | ML-DSA (Dilithium), SLH-DSA (SPHINCS+), ECDSA, MAYO, Falcon, Falcon-padded e Cross (RSDP/RSDPG) | ✓ |
| [Schemitt et al. 2026] | ML-DSA (Dilithium), ECDSA, MAYO, Falcon e Falcon-padded | ✓ |
| **PQCinBlock** | **ML-DSA (Dilithium), SLH-DSA (SPHINCS+), ECDSA, MAYO, Falcon-512, Falcon-1024 e Cross (RSDP/RSDPG)** | ✓ |

ções diretas de desempenho dos algoritmos com simulações usando o simulador *BlockSim* [Alharby and van Moorsel 2020] modificado, permitindo avaliar tanto o tempo necessário para as operações criptográficas e o armazenamento necessário para os respectivos artefatos criptográficos quanto o seu impacto na verificação e tamanho de blocos.

Nossos resultados mostram que, embora os algoritmos pós-quânticos apresentem diferenças consideráveis de desempenho e armazenamento em relação ao *ECDSA*, essas variações são consistentes com os níveis de segurança oferecidos. A análise sugere que a substituição de algoritmos vulneráveis ao algoritmo de Shor é viável, desde que se considere a arquitetura e nível de segurança necessários.

Para trabalhos futuros, planejamos duas direções: incluir novos algoritmos de assinatura digital, tanto pós-quânticos quanto clássicos; e estender o simulador para suportar outras arquiteturas de blockchain, como baseadas em Appendableblock [Lunardi et al. 2022] e DAGs [Sealey et al. 2022].

**Demonstração**. O código fonte, documentação e instruções de instalação[4] estão disponíveis no repositório do PQCinBlock[5]. Nossa demonstração incluirá: explicação dos parâmetros de entrada e algoritmos suportados; execução da ferramenta com e sem simulação da *blockchain*; e apresentação dos resultados gerados em cada cenário. Será utilizado equipamento dos próprios autores.

**Agradecimentos:** Este estudo foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Brasil; pela Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) (25/2551-0002572-9, 24/2551-0001368-7, 24/2551-0000726-1); pela FAPESP (2020/05183-0) e 2023/00816-2); e pelo CNPq/MCTI/FNDCT Nº 22/2024, projeto #444727/2024-8. Roben Lunardi é apoiado pelo IFRS e é bolsista de pós-doutorado PIPD/CAPES. O trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

**Declaração de Uso de IA Generativa:** Durante a preparação deste trabalho, os autores utilizaram a ferramenta NotebookLM para fins de revisão ortográfica e gramatical. Os autores revisaram e editaram o conteúdo conforme necessário e assumem total responsabilidade pelo conteúdo final da publicação.

[4] `https://youtu.be/lVWv9Dv7tQk`
[5] `https://github.com/conseg/PQCinBlock`

## Referências

Alagic, G., Bros, M., Ciadoux, P., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Silberg, H., Smith-Tone, D., and Waller, N. (2025). Status report on the fourth round of the NIST post-quantum cryptography standardization process. Technical Report NIST IR 8545, National Institute of Standards and Technology (U.S.), Gaithersburg, MD.

Alharby, M. and van Moorsel, A. (2020). Blocksim: An extensible simulation tool for blockchain systems. Frontiers in Blockchain, Volume 3 - 2020.

Commey, D., Appiah, B., Klogo, G. S., Bagyl-Bac, W., and Gadze, J. D. (2025). Performance analysis and deployment considerations of post-quantum cryptography for consumer electronics. Preprint arXiv:2505.02239 [cs.CR].

Holmes, S. A. (2023). Impact of post-quantum signatures on blockchain and DLT systems. CEUR Workshop.

Juaristi, P., Agudo, I., Rios, R., and Ricci, L. (2025). Benchmarking post-quantum cryptography in ethereum-based blockchains. In Computer Security. ESORICS 2024 International Workshops, pages 340–353, Cham. Springer Nature Switzerland.

Lunardi, R. C., Michelin, R. A., Nunes, H. C., Neu, C. V., Zorzo, A. F., and Kanhere, S. S. (2022). Consensus algorithms on appendable-block blockchains: Impact and security analysis. Mob. Netw. Appl., 27(4):1408–1420.

NIST (2024). Nist releases first 3 finalized post-quantum encryption standards. Disponível em: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards.

Ramzan, M. T. and Cimato, S. (2025). Blockchain in the Quantum Era: Surveying Security Challenges and Post-Quantum Cryptography. In 2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC), pages 2218–2223.

Schemitt, A. G., da Silva, H. F., Lunardi, R. C., Kreutz, D., Mansilha, R. B., and Zorzo, A. F. (2025). Assessing the Impact of Post-Quantum Digital Signature Algorithms on Blockchains. In 2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pages 2373–2380. ISSN: 2324-9013.

Schemitt, A. G., Lunardi, R. C., Zorzo, A. F., Silva, H. F. d., Kreutz, D., and Mansilha, R. B. (2026). Análise de Viabilidade da Implantação de Algoritmos de Criptografia Pós-Quântica em Blockchains para IoT. Anais do Colóquio em Blockchain e Web Descentralizada (CBlockchain); 2026: Anais do IV Colóquio em Blockchain e Web DescentralizadaDO - 10.5753/cblockchain.2026.22588.

Sealey, N., Aijaz, A., and Holden, B. (2022). Iota tangle 2.0: Toward a scalable, decentralized, smart, and autonomous iot ecosystem. In 2022 International Conference on Smart Applications, Communications and Networking (SmartNets), pages 01–08.

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, SFCS '94, page 124–134, USA. IEEE Computer Society.

Upama, P. B., Faruk, M. J. H., Nazim, M., Masum, M., Shahriar, H., Uddin, G., Barzanjeh, S., Ahamed, S. I., and Rahman, A. (2022). Evolution of quantum computing: A systematic survey on the use of quantum computing tools. In 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), pages 520–529.

Yang, Z., Alfauri, H., Farkiani, B., Jain, R., Pietro, R. D., and Erbad, A. (2024). A survey and comparison of post-quantum and quantum blockchains. Commun. Surveys Tuts., 26(2):967–1002.

Yokubov, B. and Gan, L. (2021). Comprehensive Comparison of Post-Quantum Digital Signature Schemes in Blockchain. In 2021 International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB), pages 158–161.