

# "Quantum"tempo leva? Análise de desempenho de bibliotecas de simulação de circuitos quânticos

João Bessa, Ricardo Miranda-Filho, Mauro Nooblath e Rosiane de Freitas

**Resumo**— A ONU declarou 2025 como o Ano Internacional da Ciência e Tecnologia Quântica (YIQ), como forma de reconhecer e celebrar os 100 anos desde o desenvolvimento inicial da mecânica quântica. Baseada nos fenômenos quânticos de superposição, emaranhamento e decoerência, a computação quântica está diretamente relacionada a avanços em processos de criptografia e de inteligência computacional, dentre outros. Entretanto, como os computadores quânticos reais ainda são de uso e acesso muito restritos, os simuladores quânticos surgem para que estudos e experimentações ajudem a entender o funcionamento e desempenho de processos computacionais quânticos. Este trabalho apresenta uma plataforma de testes para análise comparativa das diferentes bibliotecas de simulação de computação quântica disponíveis em linguagens de programação Python e Julia, sendo elas : Qiskit, Pennylane, Cirq, Qulacs projectq, Yao e Pyquest. Os resultados demonstram que as bibliotecas Yao, Qulacs e Pyquest obtiveram melhor desempenho em portas quânticas, enquanto a PennyLane obteve o melhor desempenho na resolução de circuitos variacionais.

**Palavras-Chave**— circuitos quânticos, computação quântica, inteligência artificial, programação quântica, simuladores quânticos.

**Abstract**— The UN declared 2025 the International Year of Quantum Science and Technology (IQS) to recognize and celebrate 100 years since the initial development of quantum mechanics. Based on the quantum phenomena of superposition, entanglement, and decoherence, quantum computing is directly related to advances in cryptography and computational intelligence, among other processes. However, since real quantum computers are still very limited in use and access, quantum simulators have emerged to enable studies and experiments to help understand the functioning and performance of quantum computational processes. This work presents a testbed for comparative analysis of the different quantum computing simulation libraries available in Python and Julia programming languages: Qiskit, Pennylane, Cirq, Qulacs projectq, Yao, and Pyquest. The results demonstrate that the Yao, Qulacs and Pyquest libraries achieved the best performance in quantum gates, while PennyLane achieved the best performance in solving variational circuits.

**Keywords**— computational intelligence, quantum circuits, quantum computing, quantum programming, quantum simulators.

## I. INTRODUÇÃO

O avanço da computação quântica representa uma das fronteiras mais emocionantes da ciência e tecnologia contemporâneas. Aproveitando fenômenos fundamentais como

João Bessa<sup>1</sup>, Ricardo Miranda-Filho<sup>1</sup>, Mauro Nooblath<sup>1</sup>, Rosiane de Freitas<sup>1</sup>, <sup>1</sup>Universidade Federal do Amazonas, Manaus-AM, e-mails: {joao.bessa, ricardo.filho, mauronooblath061296, rosiane}@icomp.ufam.edu.br. Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES-PROEX) - Código de Financiamento 001, pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pela Fundação de Amparo à Pesquisa do Estado do Amazonas - FAPEAM - através do POSGRAD 2024-2025.

a superposição e o emaranhamento quântico, esta área tem o potencial de revolucionar setores como inteligência computacional, criptografia, otimização e simulação de sistemas complexos.

Apesar de seu enorme potencial, computadores quânticos plenamente operacionais e acessíveis ainda não são realidade para a maioria dos pesquisadores e desenvolvedores. Neste contexto, as bibliotecas de simulação quântica surgem como ferramentas essenciais, permitindo a experimentação, o desenvolvimento e o teste de algoritmos quânticos em ambientes clássicos.

Na base desses algoritmos estão as portas quânticas, operações matemáticas (unitárias) que manipulam o estado dos qubits. Diferentemente dos algoritmos clássicos, que operam sobre bits usando portas lógicas tradicionais (AND, OR, NOT), os algoritmos quânticos se apoiam em portas como Hadamard, CNOT, Pauli-X/Y/Z, e portas de rotação, capazes de criar superposição e emaranhamento. Essas operações possibilitam explorar simultaneamente múltiplos caminhos de solução e acelerar determinadas tarefas.

Este aspecto é particularmente relevante quando olhamos para os algoritmos aproximativos e heurísticos tão usados em inteligência artificial (IA). Algoritmos quânticos como o Quantum Approximate Optimization Algorithm (QAOA) e o Variational Quantum Eigensolver (VQE) combinam circuitos parametrizados com técnicas de otimização clássica, criando um fluxo híbrido que lembra algoritmos heurísticos: não garantem soluções exatas, mas produzem soluções de alta qualidade em problemas complexos, muitas vezes NP-difíceis. Além disso, as abordagens de quantum machine learning exploram circuitos treináveis, similares a redes neurais, onde os parâmetros das portas quânticas são ajustados para reconhecer padrões em dados.

Neste trabalho, apresentamos uma análise comparativa de algumas das bibliotecas mais relevantes para simulação quântica disponíveis atualmente, incluindo Qiskit, Pennylane, Cirq, Qulacs, ProjectQ, Yao e PyQuEST. Exploramos suas principais características, vantagens e limitações, com foco especial em aplicações voltadas para inteligência computacional, aprendizado de máquina quântico e heurísticas aproximativas. O objetivo deste trabalho é fornecer um panorama claro e atualizado que auxilie pesquisadores, estudantes e profissionais a escolherem as ferramentas mais adequadas para seus projetos, compreendendo as nuances de desempenho, compatibilidade e aplicabilidade de cada biblioteca.

## II. PORTAS E CIRCUITOS QUÂNTICOS

A computação quântica afasta-se dos paradigmas da computação clássica ao utilizar os princípios da mecânica quântica.

Na computação quântica, a informação é codificada em qubits, que funcionam num espaço de Hilbert bidimensional. Ao contrário dos bits clássicos que representam estados discretos “0” ou “1”, os qubits podem existir em sobreposições destes estados, regidos por amplitudes complexas  $\alpha_0$  e  $\alpha_1$ . Esta capacidade permite a computação paralela em múltiplos estados em simultâneo, transformando potencialmente campos que dependem do processamento e otimização de dados complexos. A aprendizagem quântica de máquina (do inglês: Quantum Machine Learning – QML) tira partido destas propriedades quânticas para aumentar a eficiência computacional e resolver tarefas computacionais complexas que os métodos clássicos têm dificuldade em gerir eficazmente.

Matematicamente, um qubit pode ser expresso pela equação 1 e os quadrados dos valores absolutos das amplitudes  $\alpha_0$  e  $\alpha_1$  representam as probabilidades de medir o qubit nos estados  $|0\rangle$  ou  $|1\rangle$ , respetivamente. Ao manipular um estado quântico utilizando Operadores Matriciais Unitários (ou Portas Lógicas Quânticas), a transformação para outro estado quântico preserva a conservação da probabilidade. Isto garante que a soma dos quadrados das amplitudes permanece igual a um(1).

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad \text{onde} \quad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (1)$$

A visualização gráfica de como o estado de um qubit é descrito pode ser ilustrada através da Esfera de Bloch. Podemos observar a transformação do estado de um qubit através das rotações nesta esfera. A equação de estado na Esfera de Bloch é facilmente obtida convertendo a equação 1 para coordenadas esféricas, definindo  $\alpha_0 = \cos(\frac{\theta}{2})$  e  $\alpha_1 = e^{i\phi} \sin(\frac{\theta}{2})$ . Isto resulta na equação 2.

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right)|1\rangle, \quad \text{onde} \quad \begin{matrix} \theta \in [0, \pi], \\ \phi \in [0, 2\pi]. \end{matrix} \quad (2)$$

Na computação quântica, existem diferentes tipos de portas lógicas quânticas. Algumas portas, como as portas X, Y e Z, operam num único qubit e permitem a manipulação das amplitudes e da fase do estado de base do qubit. A porta Hadamard (H) é crucial, pois coloca um qubit num estado de sobreposição, igualando as suas amplitudes em diferentes estados. Outro grupo essencial é o das portas de rotação, tais como  $R_x(\frac{\theta}{2})$ ,  $R_y(\frac{\theta}{2})$ ,  $R_z(\frac{\theta}{2})$ , que giram os qubits em torno de eixos específicos na esfera de Bloch por um ângulo  $\theta$ .

As portas multi-qubit, como as portas de controle que podem ser  $C_x$ ,  $C_z$  e  $C_y$ , operam em dois qubits, influenciando o estado de um qubit com base no estado de outro, que serve de controle. Estas portas são essenciais para gerar emaranhamento entre qubits, uma propriedade fundamental na computação quântica. Existem várias técnicas para criar emaranhamento: O emaranhamento linear liga cada qubit  $i$  ao qubit  $i + 1$  para  $i = 0, 1, \dots, n - 2$ . O emaranhamento total assegura que cada qubit está emaranhado com todos os outros qubits do sistema. O emaranhamento circular é semelhante ao linear, mas com uma ligação adicional entre o primeiro e o último qubits. O emaranhamento circular alternado (SCA) é uma variante do emaranhamento circular em que o padrão de ligação muda

um qubit em cada bloco subsequente, alternando os papéis de qubits de controle e qubits alvo. Por último, a porta SWAP troca os estados de dois qubits.

O desenvolvimento de algoritmos em computação quântica exige uma compreensão profunda da forma como as portas quânticas influenciam os estados quânticos. Esta compreensão é crucial para a construção de estados que codificam soluções para problemas e para a obtenção desses estados aquando da medição do qubit. Para tal, é necessário manipular as amplitudes e fases dos estados dos qubits através de portas quânticas, garantindo que as medições produzem o estado de solução codificado desejado. Devido à natureza probabilística da mecânica quântica, é imperativo realizar um número suficiente de medições para obter distribuições probabilísticas fiáveis. Esta abordagem garante a identificação precisa da solução correta com elevada confiança.

### A. Circuitos Quânticos

Combinando as transformações de sinal e algébricas que cada porta faz com o fenómeno de emaranhamento, propagando as operações para mais de um Qubit, pode-se criar circuitos quânticos capazes de resolver problemas difíceis para a computação atual, como o algoritmo de Shor para fatoração rápida e circuitos parametrizados para problemas de otimização e Quantum Machine Learning(2).

O algoritmo de **Shor** é usado para fatorar números inteiros de forma eficiente, resolvendo um problema que é computacionalmente difícil para computadores clássicos. Ele se baseia em encontrar o período de uma função relacionada ao número a ser fatorado. A parte quântica do algoritmo usa a Transformada de Fourier Quântica para identificar esse período com alta probabilidade. Com essa informação, é possível encontrar os fatores primos do número de entrada. Esse algoritmo ameaça a segurança de sistemas criptográficos clássicos, como RSA, pois pode quebrá-los em tempo polinomial.

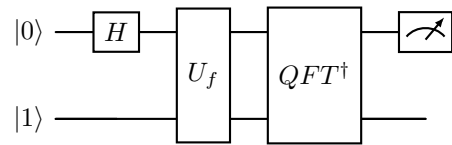


Fig. 1. Circuito simplificado do algoritmo de Shor, exibindo a preparação em superposição, aplicação de operações modulares controladas e a transformada de Fourier quântica inversa para estimar o período.

Diferente do algoritmo de Shor, o algoritmo de **Grover** resolve problemas de busca não estruturada, como encontrar uma entrada específica em uma lista não ordenada. Em um cenário clássico, seriam necessárias em média  $N/2$  buscas em uma lista de  $N$  elementos; Grover reduz esse tempo para aproximadamente  $\sqrt{N}$ , oferecendo uma aceleração quadrática. Ele funciona aplicando uma operação de inversão de fase seguida por uma difusão (amplificação de amplitudes), repetidas vezes, até que a probabilidade de encontrar o item desejado seja maximizada.

**Simon** propôs um algoritmo que resolve um problema específico de encontrar um *padrão oculto* em uma função que satisfaz a propriedade  $f(x) = f(x \oplus s)$  para um vetor

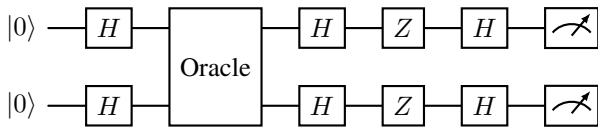


Fig. 2. Circuito simplificado do algoritmo de Grover, mostrando preparação em superposição, aplicação do oracle, inversão sobre a média (difusor) e medição.

secreto  $s$ . Esse problema foi criado para demonstrar uma separação exponencial entre algoritmos quânticos e clássicos. O algoritmo de Simon utiliza superposição e medição para coletar informações lineares sobre o vetor  $s$ , e depois resolve um sistema de equações lineares módulo 2 para determiná-lo. Ele é considerado um precursor do algoritmo de Shor, pois usa ideias semelhantes de interferência quântica para revelar propriedades ocultas de funções.

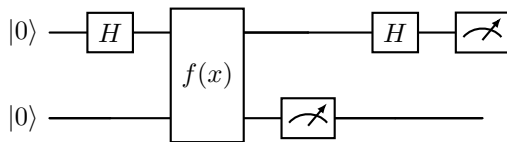


Fig. 3. Circuito simplificado do algoritmo de Simon, ilustrando a aplicação do oracle, medição parcial e transformada de Hadamard para extração do vetor  $s$ .

### B. Circuitos Variacionais

Os Circuitos Quânticos Variacionais (VQC – Variational Quantum Circuits) são uma abordagem híbrida de computação quântica e computação clássica para resolver problemas de otimização, simulação de sistemas físicos, química quântica e aprendizado de máquina quântico(3).

Um circuito quântico parametrizado é construído. As portas quânticas dependem de parâmetros ajustáveis. O circuito executa e mede o sistema quântico para obter uma função de custo. Um otimizador clássico (ex.: gradiente descendente, Adagrad) ajusta os parâmetros do circuito para minimizar (ou maximizar) a função de custo. Este processo é repetido iterativamente até que a solução ótima (ou satisfatória) seja encontrada. Como exemplos de circuitos podem ser citados: VQE (Variational Quantum Eigensolver), usado para encontrar estados de menor energia de moléculas na química quântica; QAOA (Quantum Approximate Optimization Algorithm), usado para resolver problemas de otimização combinatória; Além de usos em estratégias de Aprendizagem Quântica de Máquina.

### III. DESEMPENHO DE SIMULADORES DE CIRCUITOS QUÂNTICOS

Avaliar o desempenho de bibliotecas de simulação de circuitos quânticos é essencial para garantir sua eficiência, escalabilidade e adequação a diferentes aplicações. À medida que os algoritmos quânticos se tornam mais complexos e os requisitos de simulação crescem, torna-se fundamental entender como cada biblioteca lida com o tempo de execução, uso de memória e fidelidade na simulação. Comparações entre

diferentes frameworks permitem identificar quais são mais apropriados para determinadas tarefas, como simulação de algoritmos variacionais, otimização quântica ou aprendizado de máquina quântico.

Para este artigo, a ideia é avaliar, sobre o prisma da inteligência computacional, o desempenho das bibliotecas de simulação de circuitos quânticos através da análise do tempo de execução de diferentes rotinas de simulação em ambiente de recursos computacionais domésticos. As simulações estão configuradas para rodar em ambiente de única thread e placa de vídeo disponível apenas nos cenários em que CUDA é explicitado. A única rotina Multithread é a de Circuitos variacionais.

### A. Trabalhos Relacionados

Na literatura há diversos trabalhos postulando diretrizes de como avaliar sistemas quânticos reais. O artigo de (4) apresenta um sistema para benchmark de portas aleatórias multiQubit e descreve o processo de aferição dos resultados. Enquanto o artigo de (5) apresenta uma revisão abrangente e técnica das principais técnicas de caracterização e benchmarking de portas e circuitos quânticos. O foco central destes trabalhos é a avaliação precisa dos erros que afetam a fidelidade dos circuitos quânticos, especialmente em aplicações sensíveis como simulações químicas e computação quântica tolerante a falhas. Embora no ambiente de simulação não exista ruído normal e a recuperação das informações é ótima, entender como funciona os métodos de avaliação de sistemas reais, ajuda a compreender que processos são difíceis de serem emulados e se de fato correspondem a um possível resultado real.

Para ambientes de simulação o artigo de (6) apresenta uma revisão técnica que avalia o desempenho de diversos simuladores de estado vetorial usados para simular computadores quânticos em plataformas de computação de alto desempenho (HPC). O foco está em comparar a eficiência computacional de diferentes pacotes de software ao executar três tarefas quânticas: simulação da dinâmica do modelo de Heisenberg, amostragem de circuitos quânticos aleatórios (RQC), e a Transformada de Fourier Quântica (QFT). É interessante perceber que embora este artigo seja sobre simulação, uma das tarefas(ROC) é foco de estudos de Benchmark em sistemas reais, presente nos trabalhos de (5) (4).

### IV. ESTRUTURA DE BENCHMARK

Este trabalho apresenta 3 baterias de teste, portas básicas:  $X, H, CNOT, T$ , circuitos parametrizados ordinários( regressor simples) e Variational Quantum Eigensolver para Quisquit e PennyLane.

Recursos Computacionais:

- **CPU:** Ryzen 5 1600AF (Multithread não habilitado).
- **RAM:** 32 Gb
- **GPU:** RTX 3060 (Apenas para testes com CUDA explicitado)
- Bibliotecas e interpretadores: Python 3, Julia, Conda, CUDA. Bibliotecas de simulação na sua versão estável disponível em Q1/2025.

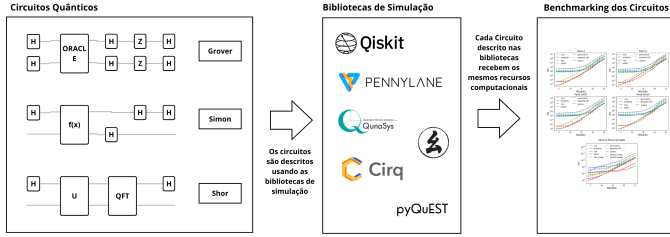


Fig. 4. Fluxo de execução do benchmark de simulação de circuitos quânticos.

Com foco em compreender o desempenho dos simulares em sistemas domésticos, sem recursos computacionais de alto desempenho, os testes foram executados em uma única thread e a aceleração gráfica só foi utilizada em circuitos parametrizados para comparar a diferença do uso da aceleração. Cada Instância de teste representa a média de 10 execuções. O esquema da **Figura 4** simplifica o fluxo de execução do benchmark.

### V. RESULTADOS

Para a primeira parte do benchmark foi realizado testes com 4 portas Quânticas Hadamard, X, CNOT e CCNOT (Toffoli), o objetivo deste teste é verificar como se comportam as bibliotecas quando é simulado circuitos de até 25 Qubits em portas unitárias em paralelo(sem emaranhamento). O segundo cenário verifica o desempenho de um circuito variacional, para verificar o potencial da aceleração por GPU quando disponível e a possibilidade de usos em otimização. A figura a seguir mostra o resultado destes cenários:

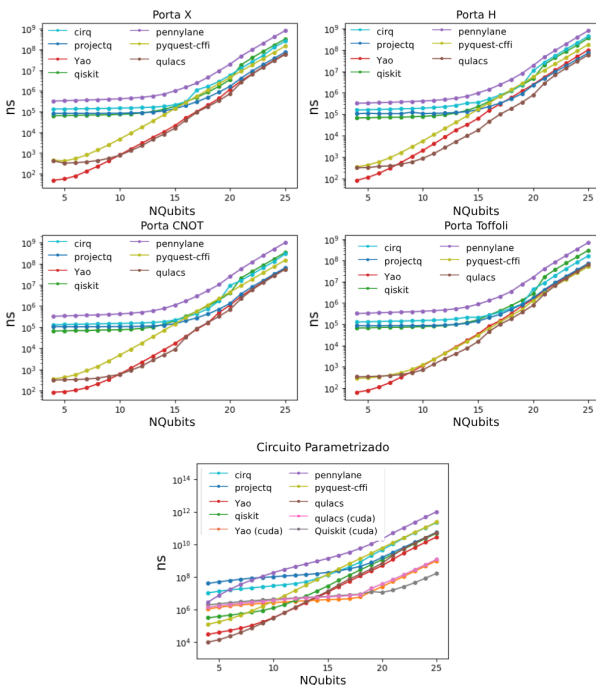


Fig. 5. Resultado do benchmark das portas Hadamard, X, CNOT e Toffoli em relação ao tempo de execução.

O primeiro comportamento a ser notado é que a biblioteca Yao teve um piso de tempo de execução menor em todas as portas básicas. É interessante mencionar que o Yao é executado

em Julia, uma linguagem de programação científica semelhante ao Python. Entretanto Julia possui um compilador Just in time, similar ao PyPy, isso traz ganhos de performance em tempo de execução, como a quantidade de Qubits inicial é fácil de simular, o Yao se sobressai por conta da sua linguagem de codificação. Esse comportamento pode ser observado também em Qulacs e Pyquest que possuem apenas uma interface de interação em python e possuem toda a parte de simulação em C e C++.

A biblioteca com maior tempo de execução foi o pennylane e qiskit, embora sejam as mais lentas nestes testes de porta única, são as bibliotecas mais usadas para simulação em python. PennyLane por sua vez possui um ferramental muito vasto para otimização e Quantum Machine Learning, podendo ser utilizada aceleração gráfica nesses cenários. Qiskit além de ser a linguagem base dos computadores quânticos da IBM, elogiada por sua precisão (6) e integração com o ecossistema python, possibilitando métodos híbridos, além do suporte a aceleração por GPU. Ademais, a aceleração de GPU potencializou todas as bibliotecas que as foi ativada. Demonstrada no gráfico de circuitos parametrizados presente na figura anterior, na simulação circuitos mais complexos é essencial o uso de aceleração gráfica dado o aumento acentuado do tempo de execução.

Portas singulares apresentadas nos resultados anteriores compõem circuitos mais complexos capazes de resolver problemas difíceis para a computação clássica. Como o algoritmo de Shor que aproveita características especiais da computação quântica como as mudanças de base rápidas e a transformada quântica de fourier. Para esta parte foram Selecionados 3 algoritmos da literatura: Shor, Groover e Simon para o benchmark. O resultado destes cenários estão presentes na figura abaixo: As 3 bibliotecas de simulação de melhor desempenho em portas singulares também obtiveram melhor desempenho relativo as demais bibliotecas, vale ressaltar que este cenário é de experimentos de menor escala, iterando sobre problemas menores, o suporte a aceleração gráfica e integração a bibliotecas e até mesmo computadores quânticos reais como Qiskit e CirQ podem ser mais relevantes em cenários de maior complexidade.

Embora mais complexos, estratégias baseadas em circuitos variacionais como o VQE e Máquinas de vetores de suporte Quântica ainda podem ser executadas em computadores domésticos em cenários mais simples. Diante desta realidade a ultima parte do benchmark busca verificar o tempo de execução da minimização energética de 2 moléculas: Íon de 3 Hidrogênios  $H_3^+$  e água  $H_2O$ , a partir de um circuito VQE. Foram selecionadas apenas as bibliotecas de YAO, Qiskit e PennyLane para este teste pois estas possuem já implementadas em si ( ou em bibliotecas auxiliares) diversos cálculos além do VQE necessários para minimização energética, como a estratégia para estimar a energia de um único átomo.

A PennyLane que diante dos cenários anteriores sempre obteve os piores tempos, neste cenário obteve o melhor desempenho, mesmo comparado a Yao e sua compilação otimizada. A PennyLane ainda possui grande facilidade de escrita de código nesse cenário, justificando a maior adesão da comunidade à biblioteca. A Yao por sua vez se mostrou rápida

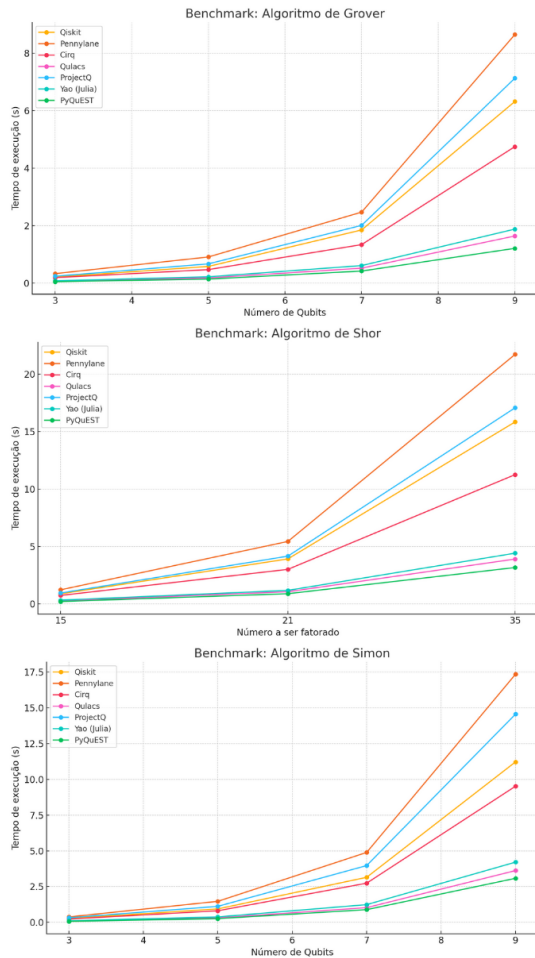
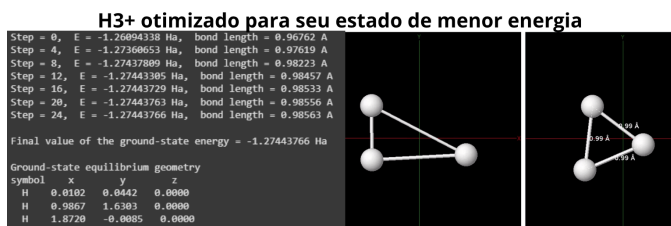


Fig. 6. Resultado do benchmark dos Circuitos do Algoritmo de Shor, Simon e Groover em relação ao tempo de execução.



**Tempo de execução VQE**

	Qiskit	PENNYLANE	⚡
<b>H3+</b>	<b>19m21s</b>	<b>16m14s</b>	<b>18m3s</b>
<b>H2O</b>	<b>27m12s</b>	<b>19m11s</b>	<b>23m31s</b>

Fig. 7. Resultado da execução do VQE em cada Linguagem e o resultado da otimização do Ion  $H_3^+$ .

em todos os cenários, demonstrando potencial para simulação de circuitos mais complexos.

**VI. CONSIDERAÇÕES FINAIS**

O trabalho apresentado apresentou o desempenho de 7 bibliotecas para simulação de circuitos Quânticos sendo elas: Qiskit, Pennylane, Cirq, Qulacs, projectq, Yao e pyquest.

Durantes os testes das portas e dos circuitos de Simon, Grover e Shor, as bibliotecas qulacs, pyquest e Yao obtiveram melhor tempo de execução. Em circuitos parametrizados, as simulações que tiveram aceleração gráfica habilitada obtiveram os melhores tempos. No cenário mais complexo, as bibliotecas otimizadas para cenários otimização do PennyLane o deixou com o melhor desempenho.

Diante do exposto, podemos ver como diferenças de implementação e finalidade de uso podem impactar em rotinas de simulação de circuitos quânticos em diferentes cenários. Destacando a Yao, com bom desempenho em todos os cenários e a Pennylane, embora tenha desempenho pior em cenários mais básicos, quando utilizado em cenários complexos possui ótimo desempenho e facilidade de implementação. Em trabalhos futuros é interessante investigar mais aspectos além do tempo de execução, como precisão dos circuitos em simulação de ruído, alocação eficiente de recursos, bem como cenários mais complexos.

**AGRADECIMENTOS**

Este trabalho foi parcialmente financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES-PROEX) - Código de Financiamento 001, pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pela Fundação de Amparo à Pesquisa do Estado de Amazonas - FAPEAM - através do POSGRAD 2024-2025.

**REFERÊNCIAS**

- [1] Griffith, D. & Others Mecânica Quântica. (PEARSON,2011)
- [2] Horvat, S. & Others Universal quantum computation via quantum controlled classical operations. *Journal Of Physics A: Mathematical And Theoretical*. **55**, 075301 (2022,1), <https://dx.doi.org/10.1088/1751-8121/ac4393>
- [3] Huang, H. Near-term quantum computing techniques: Variational quantum algorithms, error mitigation, circuit compilation, benchmarking and classical simulation.. *Sci. China Phys. Mech. Astron.* (2023)
- [4] Gaebler, J., Meier, A., Tan, T., Bowler, R., Lin, Y., Hanneke, D., Jost, J., Home, J., Knill, E., Leibfried, D. & Wineland, D. Randomized Benchmarking of Multiqubit Gates. *Phys. Rev. Lett.* **108**, 260503 (2012,6), <https://link.aps.org/doi/10.1103/PhysRevLett.108.260503>
- [5] Tripathi, V., Kowsari, D., Saurav, K., Zhang, H., Levenson-Falk, E. & Lidar, D. Benchmarking quantum gates and circuits. *Chemical Reviews*. (2025)
- [6] Jamadagni, A., Läuchli, A. & Hempel, C. Benchmarking quantum computer simulation software packages: state vector simulators. *ArXiv Preprint ArXiv:2401.09076*. (2024)