

PROGRAMAÇÃO NA PLATAFORMA IBMQ E ANÁLISE COMPARATIVA DA PERFORMANCE DE COMPUTADORES QUÂNTICOS

Anton Albino¹, Raphael Fortes² e Rodrigo Bloot.³

¹Universidade Federal da Integração Latino-Americana (UNILA), Foz do Iguaçu-PR, Brasil (e-mail do autor principal)

² Universidade Federal da Integração Latino-Americana (UNILA), Foz do Iguaçu-PR, Brasil

³Universidade Federal da Integração Latino-Americana (UNILA), Foz do Iguaçu-PR, Brasil

Resumo: O presente estudo analisa a performance de protótipos de computadores quânticos desenvolvidos pela IBM, que podem ser programados por meio da *framework Qiskit*. A interface dos protótipos foi utilizada para analisar a performance dos algoritmos quânticos de Bernstein-Vazirani, Simon e Grover, com base nos diagramas de taxas de erros e conectividade entre *qubits*. No final, discutimos a influência de possíveis processos de decoerência nos resultados das simulações.

Palavras-chave: computação quântica; algoritmos; emaranhamento; decoerência.

INTRODUÇÃO

Dispositivos computacionais desempenham papel fundamental em problemas de natureza científica e tecnológica. Devido às limitações físicas impostas aos processos de miniaturização de componentes eletrônicos, Richard Feynman apresentou em 1959 a primeira proposta da necessidade de explorar fenômenos quânticos para realizar trabalhos computacionais através da manipulação de átomos individuais, cujo pioneirismo precedeu a criação de áreas de pesquisa como a Nanotecnologia e a Informação Quântica. Desde então, houve um avanço considerável no desenvolvimento de experimentos capazes de testar as previsões teóricas da Teoria Quântica. Essas iniciativas, aliadas a investimentos expressivos em pesquisa e desenvolvimento na última década por parte de universidades, centros de pesquisa, bancos, governos e empresas, culminaram no lançamento de protótipos para os primeiros computadores quânticos.

Em sistemas computacionais clássicos, um bit é representado pelos dígitos binários discretos 0 e 1 e equivale à menor unidade de informação que pode ser utilizada em um determinado procedimento. Por outro lado, nos computadores quânticos, um *qubit* pertence a uma faixa contínua de valores contida nas diversas configurações que um sistema físico de dois

níveis pode assumir. Exemplos de *qubits* em sistemas reais são encontrados nos mais variados cenários, como na restrição das energias do átomo de hidrogênio ao conjunto que engloba seu estado fundamental e o primeiro estado excitado, nos diferentes arranjos para a transmissão de fótons polarizados em eixos perpendiculares e em propriedades intrínsecas da matéria (como as orientações *up* e *down* do *spin* de um elétron em uma determinada direção, quando este é submetido à presença de um campo magnético). A descrição matemática de um *qubit* é caracterizada por um vetor de duas dimensões expresso como uma combinação linear de estados de base ortonormais

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1)$$

os quais, por sua vez, estão associados a um espaço de Hilbert.

Os avanços tecnológicos introduzidos com a adoção de sistemas quânticos em tarefas informacionais refletem-se principalmente no aumento da velocidade de processamento de dados e na redução do número total de bits necessários para a implementação de algoritmos específicos, graças à existência de uma propriedade imprescindível para a execução da maioria destas tarefas: o emaranhamento. Todavia, quando um sistema puro interage com um meio

externo e as informações relativas ao estado $|\psi\rangle$ são expostas a este meio, ocorre um fenômeno que afeta o estado global no qual este sistema está inserido e destrói a pureza da função de onda original: a decoerência. Se um estado emaranhado puro possui ao menos um *qubit* exposto à ação de um sistema externo, o processo de decoerência não só reduz a amplitude das superposições existentes entre todos os seus constituintes como também afeta o grau de emaranhamento entre os mesmos.

Recentemente, a IBM anunciou um plano de desenvolvimento de uma classe de computadores quânticos que seriam projetados resolver problemas contemporâneos de alto nível de complexidade. Os primeiros protótipos já foram apresentados, e a previsão é de que o poder de processamento desses dispositivos deve aumentar consideravelmente dentro dos próximos dez anos. Alguns desses modelos – como o *IBMQ* - foram disponibilizados com livre acesso para o público, os quais constituem o objeto de estudo primordial do presente trabalho. A interface dos protótipos foi utilizada para analisar a performance dos algoritmos quânticos de Bernstein-Vazirani, Simon e Grover. Um circuito quântico, em geral, é representado por um diagrama que contém um estado inicial (entrada) de n -*qubits*, seguido de operações unitárias (que atuam sobre um ou mais *qubits*) e um estado final (saída); o qual é comumente medido na base computacional padrão e expresso como uma sequência de bits registradas em um canal clássico.

O algoritmo de Bernstein-Vazirani se propõe a determinar uma *string* s a partir de um estado de entrada e de uma função oráculo

$$f(x) = s \cdot a(\text{mod}2), \quad (2)$$

a qual é registrada em um *qubit* auxiliar para uma única iteração. Por sua vez, o algoritmo de Simon encontra o período r de uma função $z(x)$ armazenada em um segundo registrador, onde, se $z(x_1) = z(x_2)$, é garantido que $x_1 \oplus x_2 = r$. Este procedimento é executado em um intervalo de tempo exponencialmente mais rápido quando comparado a outros modelos clássicos. Já o algoritmo de Grover (também conhecido como Iterações de Grover) é um método de busca que pode superar os protocolos convencionais em uma escala de tempo que decresce em proporção quadrática. Para esse modelo, a entrada é descrita por uma superposição de estados $|s\rangle$ (onde cada autoestado pode ser interpretado como um item de uma lista), e associamos uma notação particular $|w\rangle$ ao estado procurado para diferenciá-lo dos demais. Cada iteração é composta por duas operações algébricas consecutivas, a saber: uma inversão de fase expressa por

$$U_f|w\rangle = -|w\rangle, \quad (3)$$

que inverte a fase do autoestado procurado, seguida de uma inversão sobre a média dada por

$$U_s = 2|s\rangle\langle s| - 1, \quad (4)$$

as quais devem ser repetidas em um ciclo de aproximadamente \sqrt{N} vezes. A cada iteração, a probabilidade de encontrar o estado procurado (através de medidas nos registradores da saída do circuito) aumenta.

Os algoritmos descritos foram implementados não só para analisar os resultados das simulações apresentadas nesse trabalho, mas também para comparar a performance dos diferentes simuladores e dispositivos reais disponibilizados pela plataforma *IBMQ*.

MATERIAL E MÉTODOS

Os códigos e o acesso à plataforma foram implementados por meio da framework *Qiskit* (desenvolvida pela própria IBM) em linguagem *Python*. Informações sobre todos os dispositivos analisados foram retiradas do *dashboard* da página *IBM Quantum Experience*, que ainda fornece planilhas de calibração com informações detalhadas para download sobre as taxas de erro de operações envolvendo um ou dois *qubits*. A Fig. 1 exibe as interconexões referentes aos dispositivos quânticos que serão analisados na próxima seção, onde as menores taxas de erro e conectividade são indicadas pelas cores mais claras.

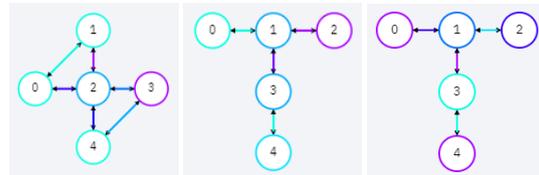


Figura 1: Diagrama taxa de erros e conectividade entre *qubits* (Dispositivos: *ibmqx2*, *ibmq_burlington* e *ibmq_vigo*).

A análise prévia dos diagramas (também conhecidos como mapas de acoplamento) contribui para dirimir a propagação de erros nos códigos. Se o projeto do circuito contém uma operação que deve realizada sobre um par de *qubits* quando estes não possuem uma conexão direta entre si, os sistemas da IBM irão substituí-la por um circuito equivalente. A Fig. 2 apresenta um exemplo para as portas *SWAP* e *CNOT*, onde é possível inferir que a mudança de conexões e a permuta entre os *qubits* podem gerar um aumento significativo de operações no circuito, contribuindo para a propagação de erros que influencia o resultado final.

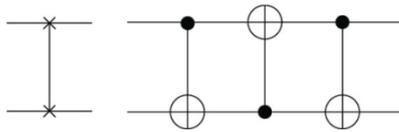


Figura 2: Exemplo de equivalência entre circuitos quânticos. A operação associada à porta SWAP pode ser implementada com o uso de três portas CNOT.

RESULTADOS E DISCUSSÃO

Para implementar o Algoritmo de Bernstein-Vazirani, foram utilizados três *qubits* na estrutura principal do código e um quarto auxiliar, o qual desempenhou a função de armazenar o resultado da operação descrita na Eq. (2). A Fig. 3 exibe a estrutura do circuito quântico utilizado para executar o algoritmo.

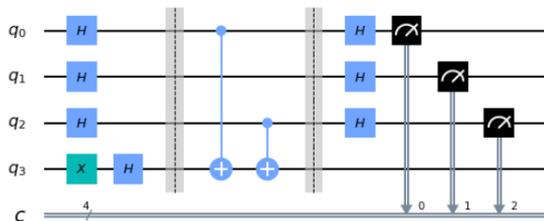


Figura 3: Circuito quântico projetado para executar o algoritmo de Bernstein-Vazirani.

Observe que a função $f(x)$ (implementada entre as linhas tracejadas na cor cinza) é composta por duas portas CNOTS, as quais requerem o uso de dois *qubits*: um para o alvo, e outro para o controle (ambos indicados na cor azul). As demais portas lógicas quânticas são constituídas de apenas um *qubit*, e as operações Hadamard e NOT são indicadas respectivamente pelas leras H (azul-escuro) e X (verde). Os registradores (na cor preta) representam as medidas efetuadas individualmente sobre os *qubits* situados nas linhas do circuito indicadas por q_0 , q_1 e q_2 .

Para analisar a performance dos respectivos dispositivos, o algoritmo foi executado 1024 vezes em ambos os sistemas. A Fig. 4 mostra que o computador quântico *ibmqx2* apresentou o valor correto em 69,0% das vezes, superando consideravelmente a taxa de sucesso de 34,2% do *ibmq_burlington*. Os resultados concordam com as informações apresentadas nas Figs. 1 e 2. Enquanto a Fig. 1 revela que só existe conexão física entre os *qubits* q_2 e q_3 no dispositivo *ibmqx2* (além de uma taxa de erro reduzida inerente às operações em *qubits* individuais), o circuito exibido na Fig. 2 indica que foram utilizadas duas operações entre dois *qubits* na função oráculo (indicadas por $cx_{(0,3)}$ e $cx_{(2,3)}$). Assim, quando não há conexão direta entre os *qubits* e uma operação envolvendo estes estados é implementada, outros fatores - como um possível aumento da taxa local de decoerência - podem influenciar a permuta entre esses *qubits*, elevando o número de resultados indesejados.

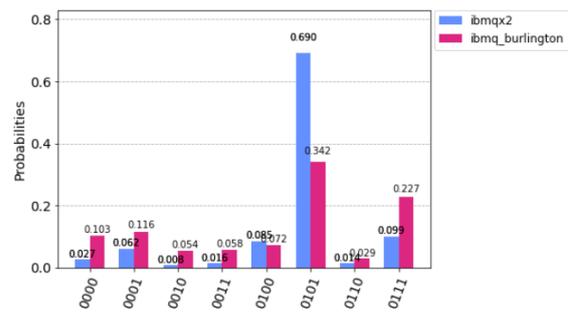


Figura 4: Histograma dos diferentes resultados em dois dispositivos da IBM.

O algoritmo de Simon também foi implementado com quatro *qubits*. Para esse modelo, a função oráculo foi construída com um período implícito dado por $r=|11\rangle$ (equivalente à representação binária do número decimal 3), e a Fig. 5 exibe o circuito projetado para executar o protocolo. Repare que a implementação da função periódica $z(x)$ (a qual é armazenada no segundo registrador) está inserida entre as linhas tracejadas, e que o estado do primeiro registrador permanece inalterado.

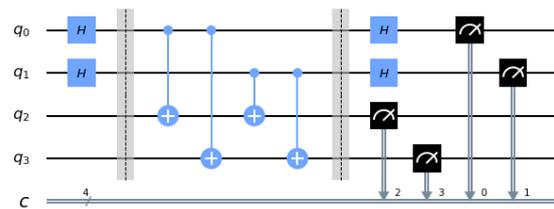


Figura 5: Circuito quântico projetado para executar o algoritmo de Simon.

Nesse caso, uma única medida no segundo registrador (*qubits* q_2 e q_3) é efetuada a fim de colapsar a função de onda do primeiro registrador para uma superposição de seus respectivos autoestados. Na sequência, efetuam-se novas medidas sobre q_0 e q_1 para obter-se o estado final. A Fig. 6 mostra os resultados gerados através dos dispositivos *ibmqx2* e *ibmq_vigo*.

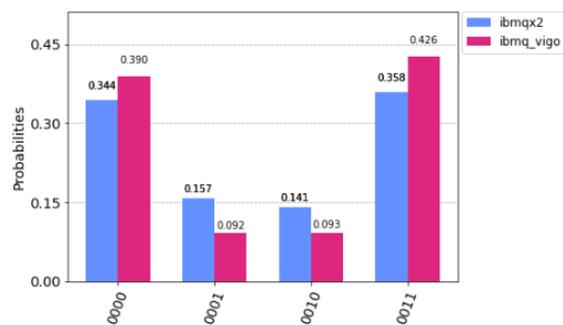


Figura 6: Comparação da performance de dois dispositivos para o algoritmo de Simon.

O modelo teórico prevê que os estados ‘00’ e ‘11’ seriam obtidos em 50% das vezes. Nesse caso, o

período $r = 11$ pode ser determinado com a solução do sistema dado por

$$\begin{aligned} r \cdot 00 &= 0; \\ r \cdot 11 &= 0. \end{aligned}$$

O resultado real confirma que os dispositivos se aproximam das previsões teóricas. Os estados '10' e '01', medidos em menor frequência, estão relacionados com a presença de ruído, possivelmente devido à erros de operação nas portas lógicas *CNOT*. Para esse algoritmo, o *ibmq_vigo* apresenta uma performance superior à do *ibmqx2*: enquanto a taxa máxima de erros em operações em *qubits* individuais do primeiro é igual a 5.174×10^{-4} , a do segundo é dada por 2.643×10^{-3} . Assim, verificamos novamente que se não há uma conexão física prévia entre os *qubits*, o aumento do número de operações eleva consideravelmente a taxa de erros durante o processamento dos cálculos.

O algoritmo de Grover, por sua vez, foi implementado com três *qubits*, e a operação U_f (Eq. 3) foi adaptada para buscar o estado $|101\rangle = |5\rangle$. A Fig. 7 reproduz a estrutura desse circuito. Note que a sequência $U_f U_s$ está contida entre as linhas tracejadas,

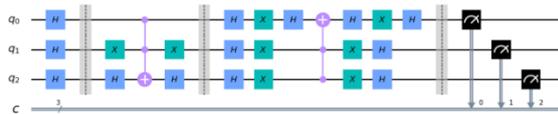


Figura 7: Circuito quântico projetado para executar o algoritmo de Grover.

Para três *qubits*, o número de repetições necessárias é de aproximadamente $\sqrt{N} = \sqrt{2^3} = 2,82$ e, portanto, duas iterações são suficientes para concluir o procedimento. O resultado comparativo obtido entre o dispositivo *ibmqx2* e o *qasm_simulador* após a primeira iteração é mostrado na Fig. 8.

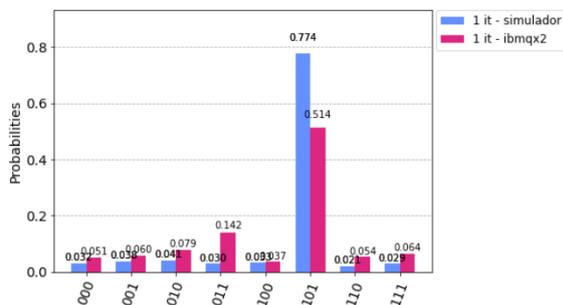


Figura 8: Resultados do algoritmo de Grover para uma iteração (Comparativo entre o simulador e o dispositivo *ibmqx2*).

Para realizar a segunda iteração, é necessário aplicar a operação $U_f U_s$ no estado de saída do bloco anterior. Como cada etapa é constituída de vinte operações lógicas, uma sequência de duas iterações gera o dobro do número total de cálculos efetuados. Após a

segunda iteração, o estado de saída foi medido 1024 vezes, resultando na distribuição de probabilidades mostrada na Fig. 9.

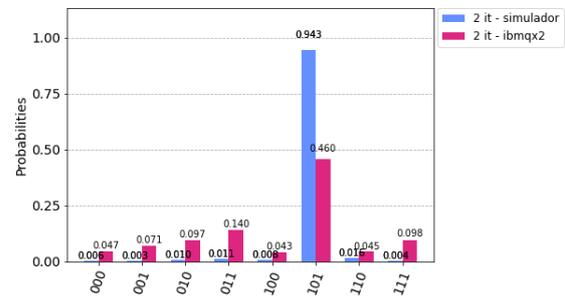


Figura 9: Resultados do algoritmo de Grover para duas iterações (Comparativo entre o simulador e o dispositivo *ibmqx2*).

O mesmo procedimento comparativo entre o simulador e o *ibmqx2* foi aplicado para o dispositivo *ibmq_vigo*. Os resultados para uma e duas iterações são mostrados nas Fig. 10 e 11, respectivamente.

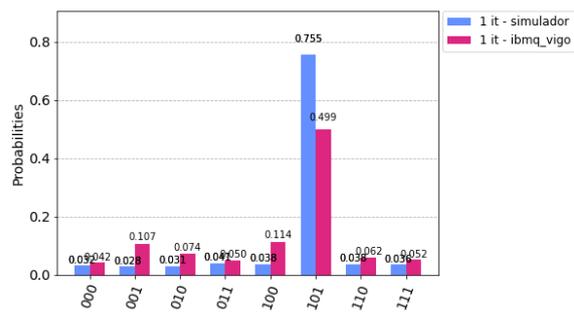


Figura 10: Resultados do algoritmo de Grover para uma iteração (Comparativo entre o simulador e o dispositivo *ibmq_vigo*).

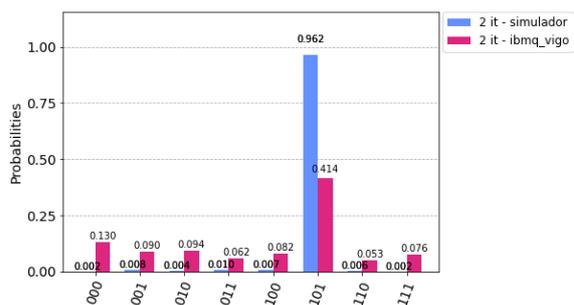


Figura 11: Resultados do algoritmo de Grover para duas iterações (Comparativo entre o simulador e o dispositivo *ibmq_vigo*).

Em seguida, comparamos os dois dispositivos reais utilizados para executar o algoritmo (*ibmqx2* e *ibmq_vigo*). As Figs. 12 e 13 apresentam os resultados obtidos em ambos os dispositivos.

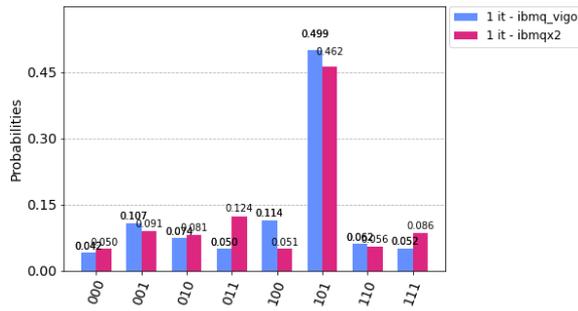


Figura 12: Resultados do algoritmo de Grover para uma iteração (Comparativo entre *ibmqx2* e *ibmq_vigo*).

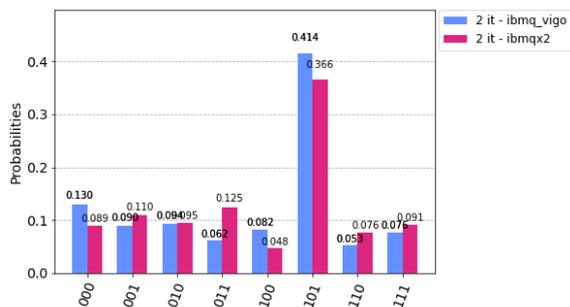


Figura 13: Resultados do algoritmo de Grover para duas iterações (Comparativo entre *ibmqx2* e *ibmq_vigo*).

Como previsto pela teoria, a probabilidade de o item buscado ('5') ser medido no simulador aumenta de ~75% para ~95% após a segunda iteração. O algoritmo também foi testado nos dispositivos reais *ibmqx2* e *ibmq_vigo* e, apesar de o estado procurado ainda apresentar uma probabilidade maior de ser medido após a segunda iteração, a probabilidade decresce em relação à primeira, devido a um crescimento significativo no número de operações realizadas. Nos dois últimos gráficos, nota-se uma queda no número de vezes que o estado buscado foi medido na segunda iteração em relação à primeira: de 49,9% para 41,4% no *ibmq_vigo*, e de 46,2% para 36,6% no *ibmqx2*. Nesse caso, podemos inferir que a perda na precisão está diretamente relacionada ao número de portas lógicas utilizadas para executar as duas iterações.

CONCLUSÃO

Os resultados obtidos para o algoritmo de Bernstein-Vazirani demonstram que, para esse algoritmo, o dispositivo *ibmqx2* apresenta desempenho superior ao *ibmq_burlington* devido a uma melhor conectividade física entre os *qubits*, descartando a necessidade da utilização de um grande número de operações equivalentes à troca desses estados.

Para o algoritmo de Simon, verificamos que os dados obtidos aproximam-se da distribuição prevista pelo modelo teórico, que determina uma frequência de 50%

de colapso das medidas no sistema para os estados $|00\rangle$ e $|11\rangle$. Todavia, a presença indesejada dos estados $|01\rangle$ e $|10\rangle$ não pode ser descartada, indicando que possíveis flutuações intrínsecas aos sistemas podem ser decorrentes de processos de decoerência. Além disso, verificamos que o dispositivo *ibmq_vigo* apresenta resultados mais precisos que o *ibmqx2*, devido às taxas de erros do primeiro (ocasionadas por operações que atuam em *qubits* individuais) serem menores do que as do segundo.

Por fim, mostramos que o algoritmo de Grover apresenta resultados similares para os dispositivos *ibmqx2* e *ibmq_vigo*. Apesar de haver semelhança entre ambos, podemos observar novamente uma superioridade do último, o que pode ser explicado pelo elevado número de portas lógicas que atuam em *qubits* individuais utilizadas para implementar esse algoritmo. Em dispositivos reais, a diminuição da probabilidade de se medir o estado almejado na segunda iteração está diretamente relacionada ao aumento na quantidade de operações lógicas necessárias para a execução dos protocolos.

Desse modo, conclui-se que uma análise prévia e cuidadosa das informações disponíveis na plataforma *IBM Q Experience* contribui para mitigar erros na execução dos códigos. Os resultados demonstram que a performance dos protocolos pôde ser otimizada com a escolha do dispositivo ideal para cada modelo, e que as informações dos diagramas de taxas de erros e conectividade entre *qubits* são fundamentais para a construção dos algoritmos. Com base nos experimentos, é razoável esperar que o aumento do número de *qubits* na implementação dos circuitos pode incorrer em um efeito cascata no processo de decoerência, resultando em um acúmulo significativo de erros nos cálculos efetuados pelo sistema físico. Consequentemente, o aprofundamento em estudos referentes à atenuação de efeitos de decoerência para circuitos em maior escala torna-se fundamental para o propósito de tornar o uso destes computadores prático e aplicável no futuro.

AGRADECIMENTOS:

Agradecemos a IBM Quantum Experience por permitir o uso das plataformas online dos dispositivos *ibmqx2*, *ibmq_burlington* e *nibmq_vigo*, e pela assistência disponibilizada através de tutoriais.

REFERÊNCIAS

Aspuru-Guzik, A. Algorithms for Quantum Computers. Scientific American, September 14, 2018. Disponível em <<https://www.scientificamerican.com/article/algorithms-for-quantum-computers/>>.

Brooks, M. Beyond Quantum Supremacy: The Hunt for Useful Quantum Computers. Scientific American, October 3, 2019. Disponível em <<https://www.scientificamerican.com/article/beyond-quantum-supremacy-the-hunt-for-useful-quantum-computers/>>.

Diagramas de conectividade entre *qubits*. Disponível em: <<https://quantum-computing.ibm.com/>>.

Feynman, R.P. There's plenty of room at the bottom: An invitation to enter a new field of physics. In Handbook of Nanoscience, Engineering, and Technology, 3rd ed.; CRC Press: Boca Raton, FL, USA, 2012.

Fortes, R., Teletransporte Quântico na Presença de Ruído. Tese. Universidade Federal de São Carlos, São Carlos, SP, 2016.

Mermin, N. D. Quantum Computer Science An Introduction. Cambridge University Press. New York, NY, USA, 2007.

Nielsen, M. A. and Chuang, I. L. Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, New York, NY, USA, 2011.

Pakin, S. and Coles, P. The Problem with Quantum Computers, June 10, 2019. Disponível em <<https://blogs.scientificamerican.com/observations/the-problem-with-quantum-computers/>>.

Roukes, M. Espaço suficiente lá embaixo. Scientific American Brasil. Edição Especial: Nanotecnologia, Número 22, Pág. 6-13, Editora Duetto, 2008.

Somma, R. Are We Ready for Quantum Computers? March 13, 2020. Disponível em <<https://blogs.scientificamerican.com/observations/are-we-ready-for-quantum-computers/>>.

Vazirani, U.V. EDX' course: Quantum Mechanics and Quantum Computation. Disponível em <<https://www.edx.org/course/quantum-mechanics-and-quantum-computation>>.