Apostila Primeiros Passos na Computação Quântica

Amanda G. Valério

Setembro, 2020

1 Introdução

Nesta pequena apostila vamos introduzir alguns conceitos básicos para começar os estudos em computação quântica. Trataremos principalmente da base matemática e explicação de conceitos fundamentais da física. Depois, entraremos no básico da computação quântica, com qubits, portas quânticas e um breve resumo da situação da área atualmente. Ao final, traremos algumas sugestões de plataformas para simular circuitos e algoritmos quânticos, além de material complementar para um aprofundamento.

Não existem pré-requisitos para este minicurso. A fundamentação mais básica está resumida nesta apostila. Obviamente, para aprofundar seus estudos pode ser necessário também aprofundar nos conceitos que trouxemos aqui, principalmente se tratando de Álgebra Linear. Mas esta é uma preocupação que pode ser deixada para o futuro.

Também é importante ressaltar que este minicurso é focado em entender a computação quântica através do modelo matemático. Este é uma abstração matemática sobre o funcionamento dos computadores quânticos, que não leva em conta informações como materiais usados, tipo de qubit (fóton, elétron,...), etc. Desta forma, conseguimos entender os circuitos e funcionamento dos algoritmos sem precisarmos total domínio prévio sobre áreas avançadas da física.

2 Embasamento Matemático

2.1 Álgebra Linear

Primeiramente, vamos precisar de conceitos básicos de álgebra linear como vetores e matrizes. Basicamente, toda a computação quântica é descrita por matrizes, então é fundamental um bom domínio desta ferramenta para seus estudos. Como não vamos aprofundar muito a princípio, apenas entender estes conceitos e aprender a operação de multiplicação é suficiente.

Primeiro falaremos sobre vetores. Se você tiver um ponto, o segmento de reta que vai da origem do plano (ex.: plano cartesiano) até este ponto é um vetor. Para o caso de você ainda não ter estudado sobre isso ou não se lembrar mais, A figura 2, que trataremos mais a frente, se trata de um vetor.

Agora, vamos tratar sobre matrizes. Provavelmente você já deve ter visto matrizes na programação. As matrizes da álgebra linear são muito parecidas: também são tabelas de números que representam alguma situação, mas com a diferença que as operações básicas já estão definidas. Dentre estas, veremos apenas a de multiplicação e o produto interno, que usaremos mais tarde em nossos exemplos. Mas é fortemente recomendado que você estude a fundo este ferramental matemático se quiser estudar computação quântica mais a fundo. A multiplicação de matrizes é feita de acordo com o diagrama na figura 1.

O Produto Interno é uma multiplicação de matrizes em que cada elemento da primeira matriz multiplica todos os outros elementos da segunda matriz. Esta operação é muito frequente quando se está trabalhando com as portas quânticas. Falaremos melhor sobre elas um pouco mais adiante. A equação (1) mostra um exemplo de uma operação de Produto Interno:

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 2\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \\ 3\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 4\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix} \tag{1}$$

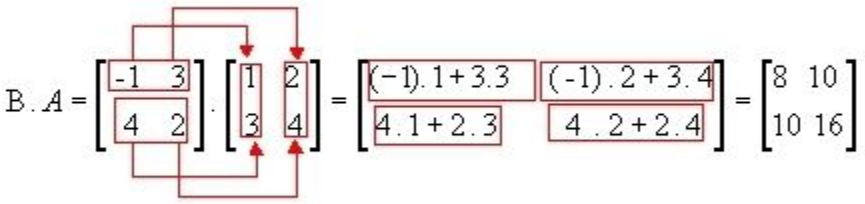

Figure 1: Fonte: https://www.somatematica.com.br/emedio/matrizes/matrizes4.php

2.2 Números Complexos

Introduziremos os números complexos nesta apostila para auxiliar a quem quiser ter um entendimento melhor e mais aprofundado sobre a computação quântica. Para entender o minicurso não é necessário, mas é uma área fundamental se voe for continuar seus estudos na área.

Os números complexos é um conjunto numérico da matemática, que engloba os números reais e os números imaginários, e todos os outros conjuntos abaixo dele (naturais, inteiros e racionais). Os números complexos são representados por $a + bi$, onde a é a parte real, bi é a parte imaginária, que sempre está acompanhada pelo i (ou j) para identificação.

Se você também pode pensar o número imaginário como um vetor de duas dimensões, em que a primeira coordenada se refere à parte real e a segunda se refere à parte imaginária do vetor. Na figura 2 é possível visualizar melhor isto. Vendo deste modo, um número complexo também pode ser escrito através da norma[1] do vetor e o ângulo θ.

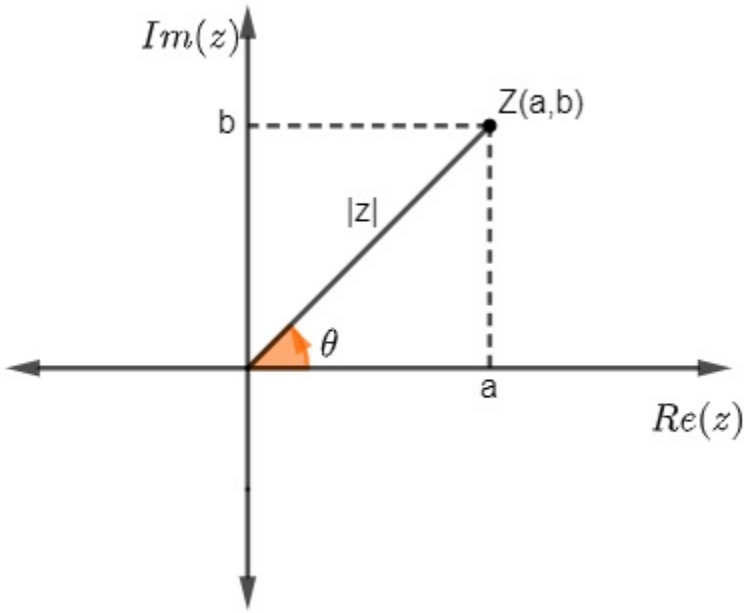

Figure 2: z = (a, b) = a + bi. Fonte: https://brasilescola.uol.com.br/matematica/numeros-complexos.htm

Trabalharemos visualizando o número complexo como um vetor, por ser mais recorrente na computação quântica. A seguir, iremos aprender como realizar as operações básicas neste novo formato de número:

[1] módulo, comprimento

2.2.1 Adição e Subtração

Basta somar ou subtrair parte real com parte real e parte imaginária com parte imaginária. Se um dos números não tiver uma das partes, você pode considerar ela como 0 e realizar a conta da mesma forma.

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \tag{2}$$

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \tag{3}$$

2.2.2 Multiplicação

A multiplicação entre números complexos é da seguinte forma:

$$(x_1, y_1) \cdot (x_2, y_2) = (x_1x_2 - y_1y_2, x_1y_2 + x_2y_1) \tag{4}$$

Multiplicar um número real com um número complexo é mais fácil. O inverso acontece da mesma forma, garantido pela propriedade de associatividade. Sendo λ um número real, basta multiplicar apenas as partes reais e manter a parte imaginária, como no exemplo:

$$\lambda(x_1, y_1) = (\lambda x_1, y_2) \tag{5}$$

2.2.3 Conjugado

Este é um conceito um pouco diferente, que não temos nos números reais. O conjugado é como o inverso de um número complexo e basta inverter o sinal da parte imaginária para forma-lo. Isto pode parecer não ser o suficiente para torná-lo o inverso, mas podemos ver melhor isso na forma gráfica:

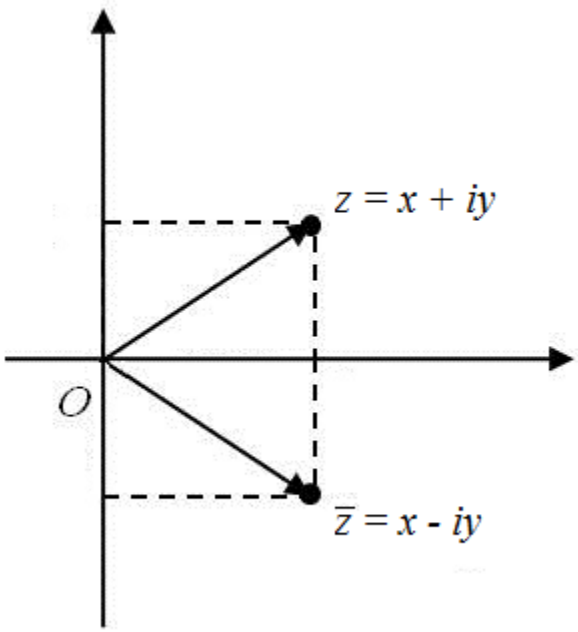

Figure 3: Fonte: `https://wikiciencias.casadasciencias.org/wiki/index.php/Conjugado_de_um_nmero_complexo`

2.2.4 Divisão

Agora que conhecemos o que é um conjugado, podemos passar para a divisão. Por exemplo, se tempos um número $z = a + bi$ e um número $w = c + di$, a divisão é feita da seguinte forma:

$$z\nabla \cdot w = \frac{z}{w} \cdot \frac{\overline{w}}{\overline{w}} \tag{6}$$

3 Princípios Básicos da Computação Quântica

Os computadores quânticos tem a capacidade de processar mais rapidamente muitos tipos de problemas que são extremamente complexos para os computadores atuais. Isto acontece graças às suas propriedades de superposição e emaranhamento, que faremos uma breve explicação mais adiante. Apesar de seu poder computacional, eles não são capazes de resolver problemas que um computador clássico não conseguiria resolver, como por exemplo o problema da parada[2].

3.1 O que é um qubit?

Fisicamente, o qubit é uma partícula usada para computar. O bit que nós utilizamos em computação clássica é um pulso de sinal elétrico, onde o valor de .... pode significar 0 ou 1. Então, para utilizar uma partícula como um bit, ela precisa ter dois estados bem definidos para simular o 0 e o 1 de um bit, como a polarização de um fóton por exemplo. Mas diferente dos bits, os qubits podem estar no estado 0 e 1 ao mesmo tempo. Isso acontece por causa da natureza probabilística da Mecânica Quântica e damos o nome de superposição à este efeito. Uma forma de representar graficamente um qubit é usando a Bloch Sphere:

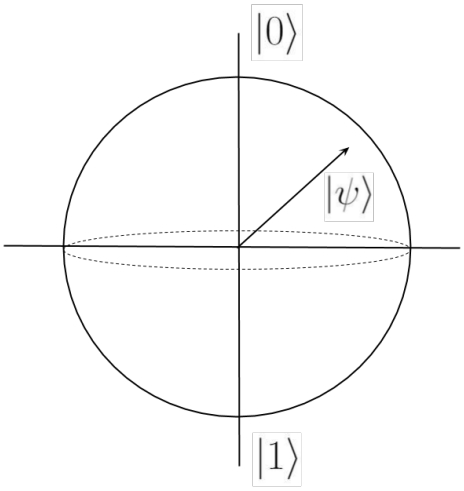

Figure 4: Fonte: autoria própria

Na figura 4, o $|1\rangle$ e o $|0\rangle$ representam os estados que podem ser medidos para o qubit $|psi\rangle$. Então, você pode pensar um qubit como $|psi\rangle = \alpha\,|0\rangle + \beta\,|0\rangle$, onde α e β são números complexos e marcam as probabilidades de o qubit estar em $|0\rangle$ e $|1\rangle$, respectivamente. Apesar disso, ao capturarmos a informação de um qubit (medir seu estado), ele vai colapsar e retornar apenas $|0\rangle$ ou $|1\rangle$.

Um outro fenômeno muito interessante é o emaranhamento. Nele, as partículas podem se correlacionar e trocar informações instantaneamente entre si, mesmo não estando próximas. Podemos representar matematicamente o entrelaçamento como uma combinação linear dos estados. Desta forma, quando você aplica uma porta em um qubit, os outros qubits emaranhados também vão sofrer a ação daquela porta.. Então ao invés de termos dois qubits variando entre $|0\rangle$ e $|1\rangle$, temos dois qubits combinados representados por $|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$, onde α, β, γ e δ são complexos e também marcam as probabilidades do qubit estar nos estados $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$, respectivamente.

[2]Uma máquina MT simula uma máquina M que executa um algoritmo. A máquina M retorna sim ou não. A máquina MT recebe a resposta e deve retornar sim caso a máquina M tenha retornado uma resposta e não caso ela tenha entrado em loop (processamento infinito). O problema é que se a máquina M não parar e retornar um resultado, a máquina MT nunca saberá a diferença entre um loop e a execução normal do algoritmo e então também nunca parará a simulação e não retornará qualquer resultado.

3.2 Portas Quânticas

Para programar computadores clássicos, utilizamos as portas lógicas. Elas são componentes eletrônicos que funcionam como operadores da lógica Booleana e são formadas combinando AND (E), OR (OU) e NOT (negação) sobre os valores 0 e 1 (verdadeiro ou falso). Na figura 5 podemos visualizar um circuito integrado com seu esquema de portas lógicas OU.

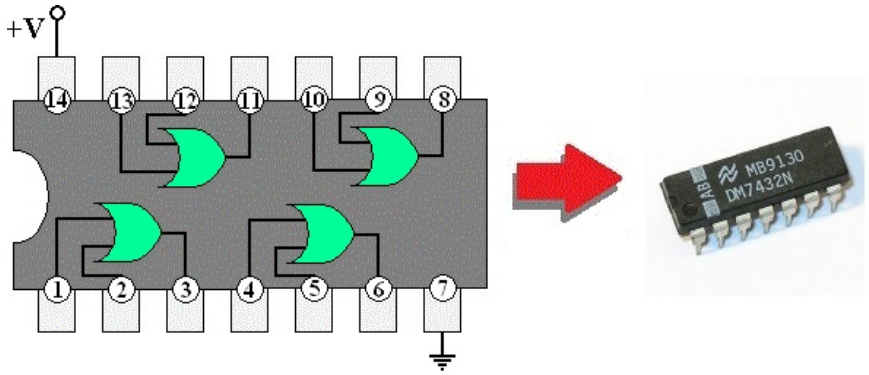

Figure 5: Fonte: `https://wiki.ifsc.edu.br/mediawiki/index.php/AULA_3_-_Eletrnica_Digital_1_-_Graduao`

Porém, na computação quântica estas portas já não funcionam. Aqui, representaremos as portas quânticas utilizando matrizes de números complexos. Elas alteram as probabilidades de resultado dos qubits utilizando rotações. A seguir, traremos algumas das principais portas quânticas utilizadas:

$$\text{Hadamard} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\text{Pauli-X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\text{Pauli-Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\text{Pauli-Z} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

$$\frac{\pi}{8} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{pmatrix}$$

$$\text{Swap} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Controlled-Not} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\text{Controll-ed-Phase} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{pmatrix}$$

3.3 Algoritmos Quânticos

Os algoritmos quânticos foram desenvolvidos muito antes de termos computadores quânticos poderosos o suficiente para rodá-los. Na verdade, hoje ainda não podemos dizer que temos estes computadores, pois

mesmo que já tenha sido feita a implementação do Algoritmo de Shor, ele ainda não produz um resultado útil.

O Algoritmo de Shor utiliza as propriedades quânticas para fatorar um número em dois primos. É este o algoritmo que alertou sobre o poder da computação quântica para quebrar a criptografia que usamos atualmente. Mais precisamente, os padrões que se baseiam em números primos, como o RSA.

Porém, pela formulação do algoritmo podemos deduzir que o computador precisaria de no mínimo qubits suficientes para representar o número em binário. Esta é uma tarefa complicada, tanto pela dificuldade quanto por custos monetários, de se agrupar muitos qubits em um processador. Isto sem falar na topologia e em interferência. Por exemplo, para fatorar o número 15 seriam necessários 5 qubits para a representação em binário, enquanto uma chave em RSA pode utilizar pelo menos 512 bits.

Um outro algoritmo também ainda longe de ser implementado largamente, porém muito famoso é o Algoritmo de Grover. Com ele, utilizando principalmente o emaranhamento das partículas, é possível buscar um elemento em uma base de dados desordenada utilizando, no pior caso, $O(\sqrt{N})$ iterações[3].Isto é um grande feito, já que um dos problemas atualmente é resgatar informações em grandes bancos de dados. Geralmente, a forma mais eficiente para trabalhar com dados é ordenando-os, pois facilita a busca. Mas o custo para ordenar cresce à medida que novos dados vão sendo inseridos e estes dados também precisam ser ordenados na inserção. Com o Algoritmo de Grover já não teríamos mais este problema.

3.4 A Computação Quântica atualmente

No ano passado, a Google publicou um artigo juntamente com a Nasa, alegando ter alcançado a supremacia quântica. O time de pesquisadores construiu um processador quântico de 53 qubits, chamado Sycamore, que executou uma tarefa em cerca de 3 minutos que um computador clássico levaria 10.000 anos para concluir. Depois que a notícia se espalhou, a IBM se pronunciou que o seu supercomputador Summit conseguiria fazer uma simulação da mesma tarefa em dois dias e meio. A discussão entre as duas empresas aos poucos foi deixada de lado, porém nada disso tira o mérito do experimento.

Em uma palestra do dia 25/09/2020, no evento da Semana de Sistemas de Informação da USP[4], o palestrante da noite Genaro Costa afirmou que já tem empresas trabalhando em placas PCI comerciais com tecnologia quântica para os computadores atuais. O lançamento, segundo ele, seria dentro de uns 5 anos em média.

Atualmente, algumas empresas estão utilizando a nuvem para disponibilizar computadores quânticos para que outras pessoas e organizações também possam testar seus algoritmos e circuitos. Algumas tem planos gratuitos, mas todas elas tem seus planos pagos. Este é o caso da IBM, que oferece tanto processamento gratis quanto pago. Já a Amazon (Braket) oferece apenas planos de processamento pagos.

Mas além de pesquisas em hardware para as máquinas e criação circuitos e algoritmos simples, também estão sendo realizadas pesquisas sobre internet quântica, aprendizado de máquina e inteligência artificial neste novo paradigma. Também está aberta para o desenvolvimento de novos simuladores, bibliotecas e arquiteturas híbridas (que interliguem processamento clássico e quântico), e desenvolvimento de algoritmos para simular fenômenos quimicos e físicos. Inclusive, em agosto, foi publicado um artigo sobre uma simulação molecular realizada no processador Sycamore.

Um outro tema que desperta o interesse de empresas e de desenvolvedores é com relação à criptografia. Graças ao algoritmo de Shor, um computador quântico pode quebrar diversos padrões de criptografia que usamos atualmente, inclusive o RSA. Para resolver este problema, os pesquisadores estão desenvolvendo a criptografia pós-quântica, onde o problema que gera o algoritmo de criptografia (no caso do RSA é a fatoração de um número primo enorme) é custoso para ambos os tipos de computadores quebrarem. Já existem algoritmos desenvolvidos, mas a área também está aberta para pesquisas.

Também é importante ressaltar que a grande maioria de bibliotecas para simulação de computação quântica é programada em Python. Então, se você for seguir seus estudos na área, um bom conselho seria: comece a estudar Python, pois ele irá te abrir um leque maior de possíveis ferramentas para desenvolvimento.

[3] notação assintótica de crescimento de custo

[4] disponível em: `https://www.twitch.tv/each_ssi/video/752189748`

4 Simuladores disponíveis atualmente

4.1 Microsoft Q# e QDK

O QDK (Quantum Development Kit) desenvolvido pela Microsoft é um conjunto de ferramentas para a computação quântica, com bibliotecas e simuladores para a programação em Q#, sua linguagem de programação. Mais informações podem ser encontradas em `https://docs.microsoft.com/pt-br/quantum/overview/what-is-qsharp-and-qdk`.

4.2 IBM QisKit

Esta é uma ferramenta Open Source disponibilizada pela IBM para programar circuitos quânticos, localmente ou online (utilizando a IBM Quantum Experience).Também é possível utilizar os computadores quânticos da IBM para executar online os circuitos e algoritmos, porém nem todos eles são gratuitos. É necessário instalar o Python 3.5 ou superior para instalar o QisKit. Maiores informações em `https://qiskit.org/overview#quick-start`.

4.3 QCL

O QCL é um simulador feito por Bernhard Omer e disponível para sistemas Linux. Ele pode ser baixado em `http://tph.tuwien.ac.at/~oemer/qcl.html`, onde também estão todos os artigos que o descrevem e fazem o papel de documentação. Para a sua instalação, é necessário instalar também os pacotes: bison, libplot, ncurses, readline5, flex.

4.4 Cirq

É uma biblioteca Open Source em Python para programação de pequenos circuitos. Conta com ferramentas de compatibilidade execução nos computadores quânticos da Google. A documentação e os links para download podem ser encontrados em `https://cirq.readthedocs.io/en/stable/install.html`.

4.5 Quirk

Disponível em: `https://algassert.com/quirk`. Usando o Quirk, você pode programar clicando e arrastando as portas para organizar o circuito, além de circuitos prontos, como o algoritmo de Grover e a Transformada de Fourier Quântica. No menu principal da aplicação podemos encontrar o link para o tutorial de como utilizar.

5 Bibliografia

SIPSER, Michael. Introduction to the Theory of Computation. Cengage learning, 2012.

Nielsen, M. A. and Chuang, I. (2010).Quantum Computation and Quantum Information.Cambridge University Press.

Curso Physical Basics of Quantum Computing, Universidade Estadual de São Petersburgo, disponível em: https://www.coursera.org/learn/physical-basis-quantum-computing

YUAN, Xiao. A quantum-computing advantage for chemistry. Science, v. 369, n. 6507, p. 1054-1055, 2020.

SHOR, Peter W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review, v. 41, n. 2, p. 303-332, 1999.

ARUTE, Frank et al. Quantum supremacy using a programmable superconducting processor. Nature, v. 574, n. 7779, p. 505-510, 2019.

GROVER, Lov K. A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. 1996. p. 212-219.

6 Materiais Complementares

CAMARGO, Ivan de; BOULOS, Paulo. Geometria analítica: um tratamento vetorial. 3ª edição. Renovada e ampliada–São Paulo: Prentice Hall, 2005.

WALLACE, Julia. Quantum computer simulators-a review version 2.1. 1999.

Lista de simuladores: https://www.quantiki.org/wiki/list-qc-simulators

Podcast do Dragões de Garagem sobre Computação Quântica: disponível em: `http://dragoesdegaragem.com/podcast/dragoes-de-garagem-157-computacao-quantica/`

NAKAHARA, Mikio; OHMI, Tetsuo. Quantum computing: from linear algebra to physical realizations. CRC press, 2008.

Cursos no Coursera da Universidade Estadual de São Petesburgo