

Aplicando o algoritmo de Grover e qRAM para resolver o puzzle game lights out

Palestrante: José Victor Soares Scursulim

Mestre em Física | IBM Qiskit Advocate | Certified
Qiskit Developer



THE
DEVELOPER'S
CONFERENCE



O Qiskit é uma SDK open-source para trabalhar com computadores quânticos ao nível de pulsos, circuitos e aplicações modulares. O Qiskit é o elemento principal para usar os computadores quânticos da IBM via nuvem.

Se você programa em Python, você está a um **pip install qiskit** de distância dos seus primeiros passos na computação quântica.

Links importantes:

Página principal: <https://qiskit.org/>

GitHub: <https://github.com/Qiskit/qiskit>

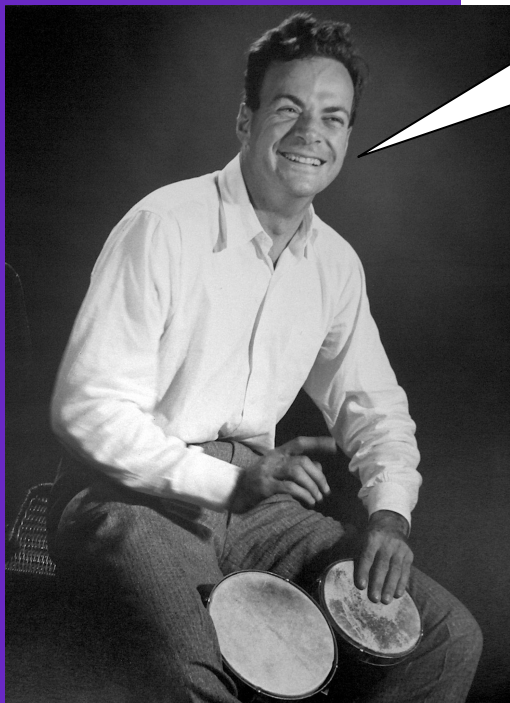
Livro texto: <https://qiskit.org/textbook/preface.html>



QR Code que leva a um vídeo meu explicando como dar os primeiros passos no Qiskit.

1. Introdução

“Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.”



Richard Feynman

<https://catonmat.net/ftp/simulating-physics-with-computers-richard-feynman.pdf>



Physics of Computation Conference Endicott House MIT May 6-8, 1981

1 Freeman Dyson
2 Gregory Chaitin
3 James Crutchfield
4 Norman Packard
5 Panos Ligomenides
6 Jerome Rothstein
7 Carl Hewitt
8 Norman Hardy
9 Edward Fredkin
10 Tom Toffoli
11 Rolf Landauer
12 John Wheeler

13 Frederick Kantor
14 David Leinweber
15 Konrad Zuse
16 Bernard Zeigler
17 Carl Adam Petri
18 Anatol Holt
19 Roland Vollmar
20 Hans Bremerman
21 Donald Greenspan
22 Markus Buettiker
23 Otto Floberth
24 Robert Lewis

25 Robert Suaya
26 Stan Kugell
27 Bill Gosper
28 Lutz Priese
29 Madhu Gupta
30 Paul Benioff
31 Hans Moravec
32 Ian Richards
33 Marian Pour-El
34 Danny Hillis
35 Arthur Burks
36 John Cocke

37 George Michaels
38 Richard Feynman
39 Laurie Lingham
40 Thiagarajan
41 ?
42 Gerard Vichniac
43 Leonid Levin
44 Lev Levitin
45 Peter Gacs
46 Dan Greenberger

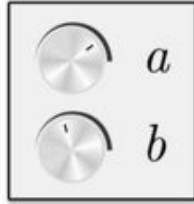
Classical

Bit



Qubit

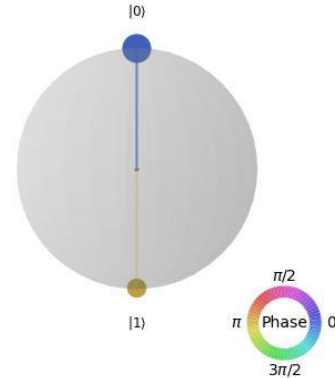
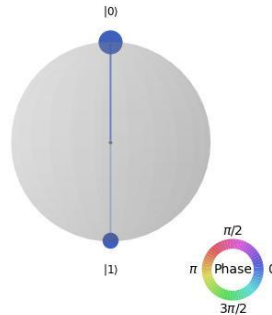
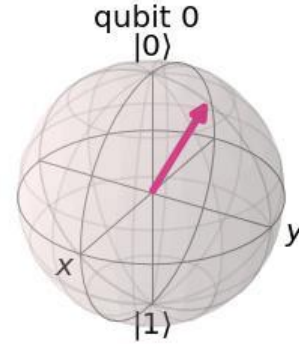
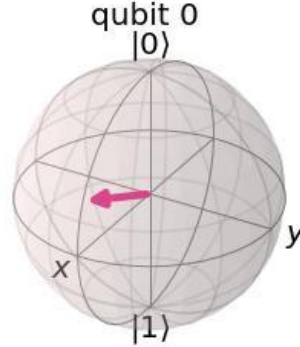
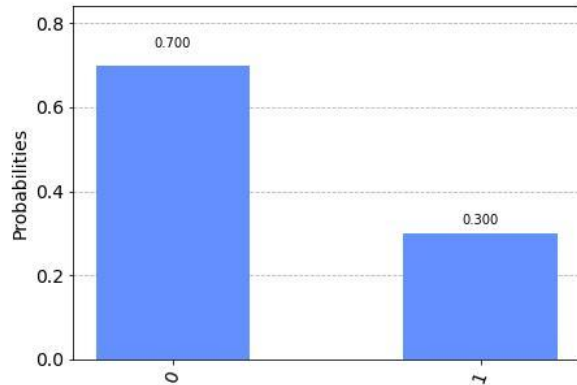
$$a|0\rangle + b|1\rangle$$



bit - clássico: 0 ou 1

qubit - quântico: 0 e 1

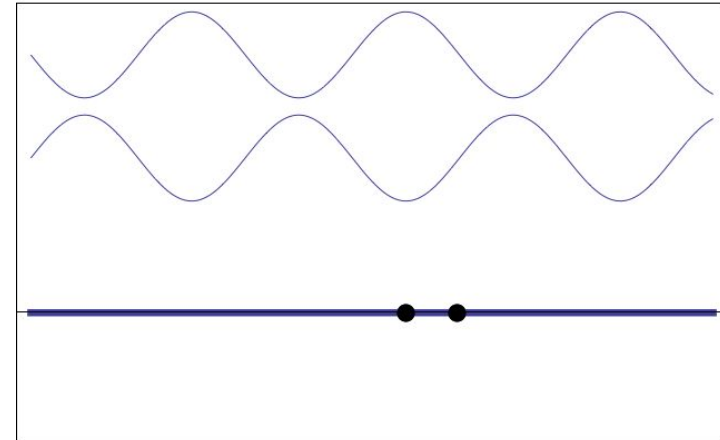
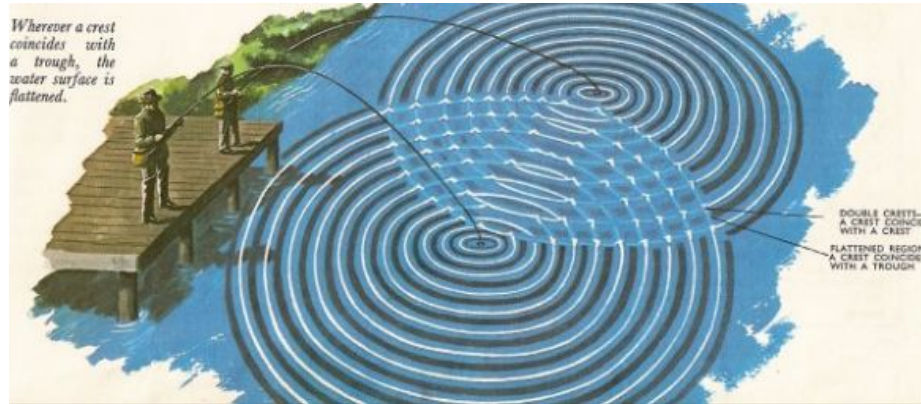
$$|\psi\rangle = \alpha|0\rangle + e^{i\theta}\beta|1\rangle$$



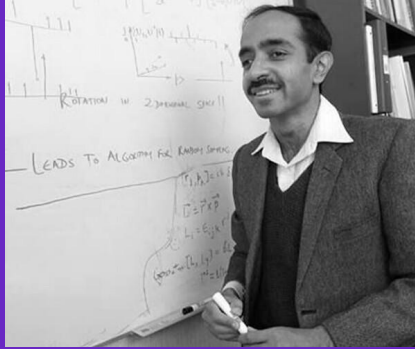
A computação quântica é intrinsecamente probabilística, e explora as seguintes propriedades da física quântica para obter vantagem computacional em relação ao paradigma clássico:

- Superposição de estados
- Emaranhamento
- Fases relativas
- Interferência de amplitudes de probabilidade

$$\frac{1}{\sqrt{2}}|\text{cat sitting}\rangle + \frac{1}{\sqrt{2}}|\text{cat lying}\rangle$$



2. Algoritmo de Grover



Lov Grover

<https://arxiv.org/abs/quant-ph/9605043>

O algoritmo de Grover foi proposto em 1996 e com ele demonstrou-se que a computação quântica resolveria de forma mais eficiente o problema de buscas em base de dados não estruturados.

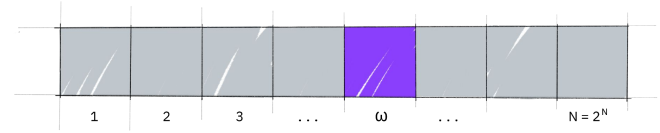


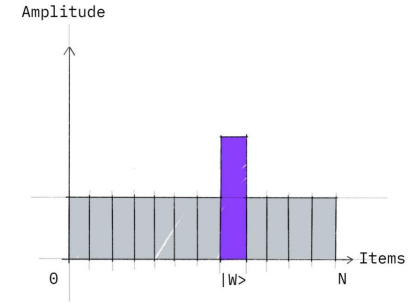
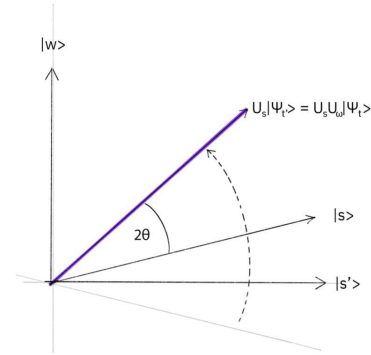
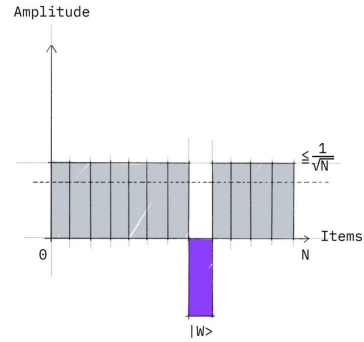
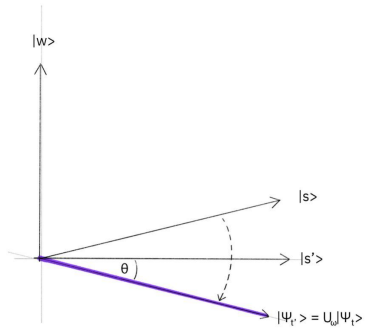
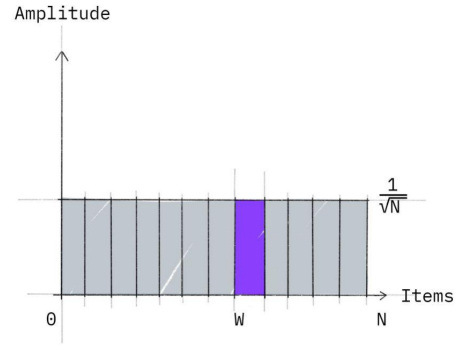
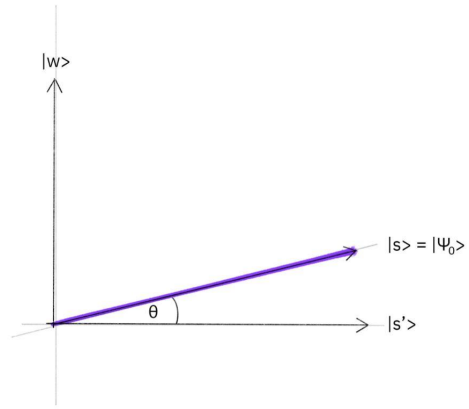
Complexidade computacional
Clássico: $O(N)$ vs Quântico: $O(\sqrt{N})$



Aplicações:

- Buscas
- Problemas de otimização do tipo satisfatibilidade
- Sistemas de equações lineares em módulo 2
- Outras





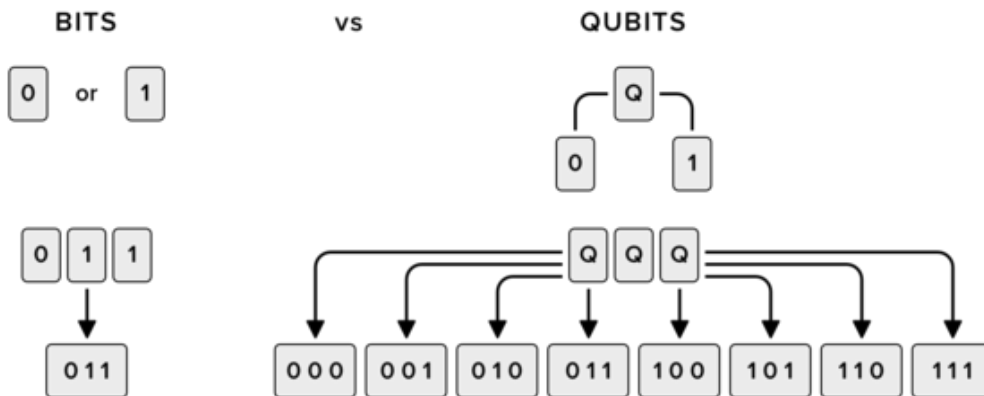
2. qRAM



Seth Lloyd

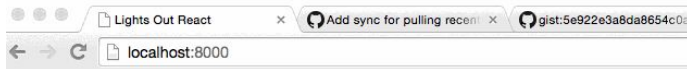
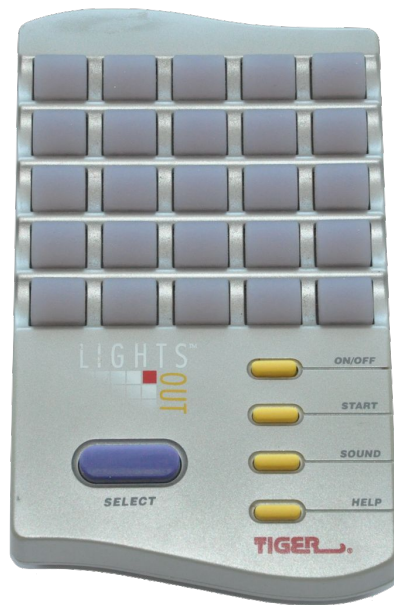
<https://arxiv.org/abs/0708.1879>

Em 2008, Vittorio Giovannetti, Seth Lloyd e Lorenzo Maccone publicaram um paper onde propuseram uma memória de acesso aleatório quântica, a qRAM. Este tipo de memória faz uso de qubits para endereçar superposições quânticas em células de memória, para n qubits podemos endereçar 2^n superposições. Espera-se que a qRAM possa acelerar tarefas como reconhecimento de padrões, e que também seja utilizada na transformada de Fourier quântica, em algoritmos de buscas e em algoritmo de quantum machine learning.

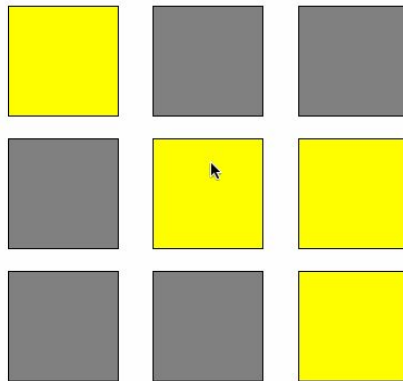


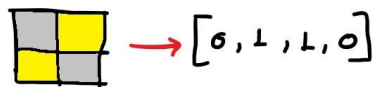
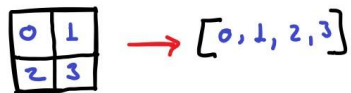
3. Lights Out

O Lights Out é um jogo eletrônico criado pela Tiger em 1995. O jogo consiste em um layout quadrado de lâmpadas que inicialmente são ligadas aleatoriamente. O objetivo é apagar todas as lâmpadas ligando ou desligando lâmpadas do layout, lembrando que ao acionar uma lâmpada ela liga ou desliga vizinhas na vertical ou na horizontal.



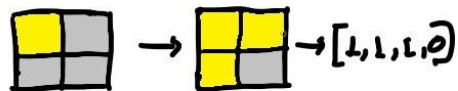
Lights Out!





 = 1
 = 0

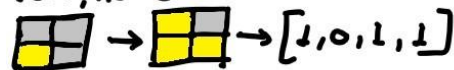
LIGANDO A LÂMPADA
NA POSIÇÃO ZERO (0)



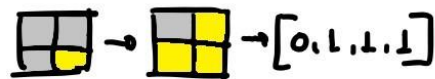
Posição UM (1)



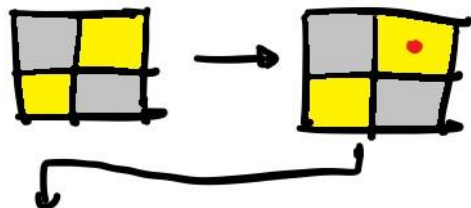
Posição DOIS (2)



Posição TRÊS (3)



Exemplo 2x2



Solução : DESLIGAR
Posição 1
Posição 2

Solução MAPEADA

[0, 1, 1, 0]

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix}$$

$$(1) x_1 + x_2 + x_3 = L_1$$

$$(2) x_1 + x_2 + x_4 = L_2$$

$$(3) x_1 + x_3 + x_4 = L_3$$

$$(4) x_2 + x_3 + x_4 = L_4$$

4. Demonstração de código

Hora de colocar em prática o
que foi discutido até aqui!



https://github.com/jvscursulim/tdc_talk



Obrigado pela atenção!

