

(\*) IBM's quantum device. Source: Wikipedia (CC BY 4.0)

# Como Programar Computadores Quânticos Reais

Franklin de Lima Marquezino  
Universidade Federal do Rio de Janeiro  
franklin@cos.ufrj.br

Maio/2023

# Objetivos desta apresentação

- **O que é a CQ?**

Trata-se de usar a mecânica quântica para resolver alguns problemas mais eficientemente

- **O que temos hoje?**

Posso MESMO programar um computador quântico real?

- **O que a CQ não é?**

Não é mágica! Ou seja, nem todo problema fica eficiente em computadores quânticos

- **O que teremos no futuro?**

Quais setores beneficiados no curto, médio e longo prazo

# Os limites da computação clássica

## Lei de Moore

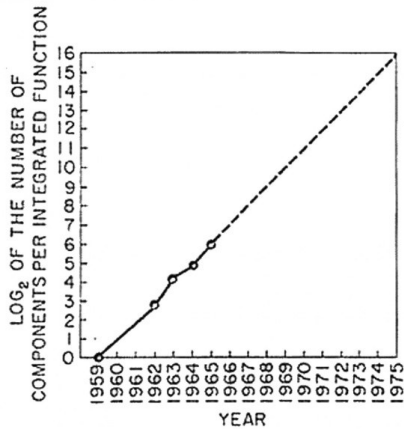
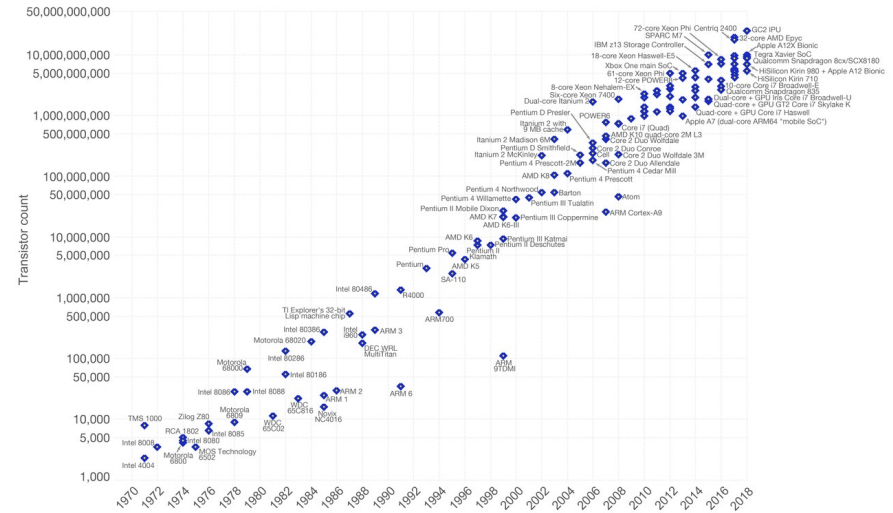


Fig. 2 Number of components per integrated function for minimum cost per component extrapolated vs time.

(\*) Gordon Moore's original graph, Source: Electronics, 1965



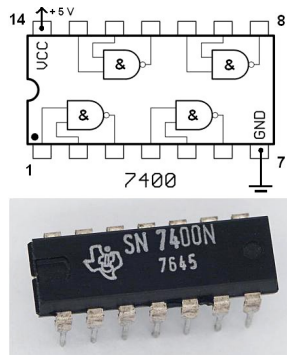
Moore's Law – The number of transistors on integrated circuit chips (1971-2018)  
 Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore's law.



Data source: Wikipedia ([https://en.wikipedia.org/wiki/Transistor\\_count](https://en.wikipedia.org/wiki/Transistor_count))  
 The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

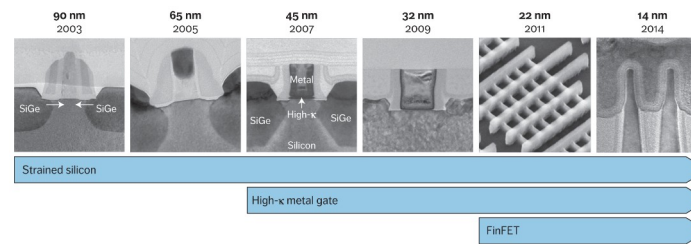
Licensed under CC-BY-SA by the author Max Roser.

## Miniaturização



1960s

(\*) Texas IC containing 4 NAND gates. Source: Wikipedia



(\*) Source: IEEE Micro, www.computer

2015



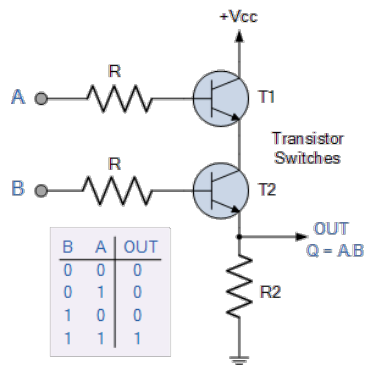
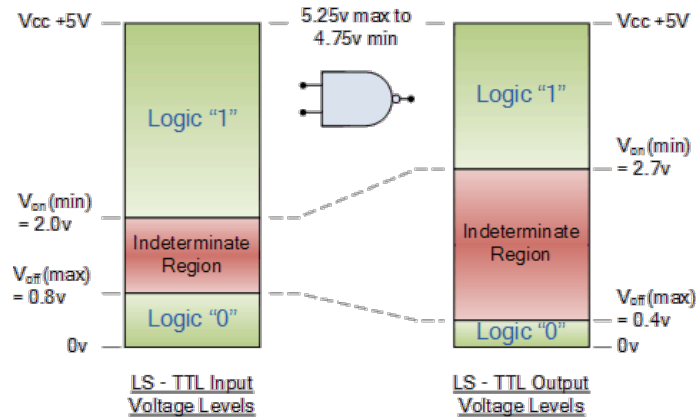
# Mecânica quântica



- **As regras do jogo, ou seja, os postulados da mecânica quântica**
  - ▷ Representação
  - ▷ Evolução
  - ▷ Medições
  - ▷ Composição
- **Conselho para estudantes:**  
**Keep Calm**  
**and**  
**Learn Linear Algebra**

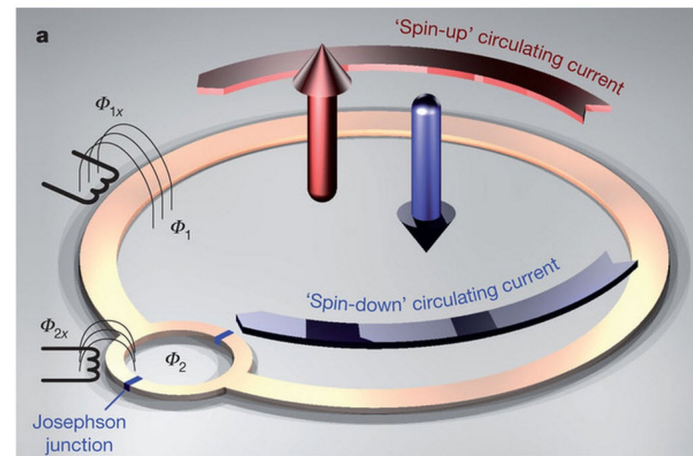
# Apresentando os bits quânticos

- Bits (clássicos) podem ser 0 ou 1



(\*) Source: www.electronics-tutorials.ws

- Bits quânticos (qubits) pode ser um vetor (1,0) ou um vetor (0,1) ou qualquer combinação deles



(\*) Superconducting qubit. Source: "Quantum annealing with manufactured spins", Nature, 2011

# Aplicações de computadores quânticos

## ▪ Simulação

- Química quântica
- Design de materiais
- Design de fármacos

## ▪ Otimização

- Finanças
- Óleo & gás

## ▪ Machine learning

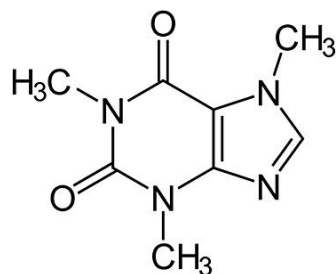
## ▪ Buscas

## ▪ Fatoração

## ▪ etc.

mais cedo

mais tarde



Simulação clássica,  $10^{48}$  bits  
(Mais que o número de átomos de nosso planeta!)

Simulação quântica, 160 qubits

**Quantum Computers  
Destroy Internet Security**

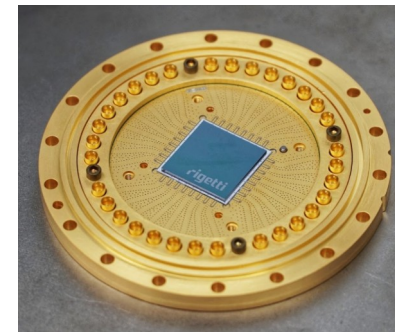


(\*) minutephysics, Youtube

Requer muitos qubits e tolerância a falhas

# Computadores quânticos atuais

- Para rodar um programa quântico, você vai precisar de um computador quântico!
- Já existem vários computadores QREI disponíveis (poucos qubits, ruidosos)
- Alguns vocês já podem acessar
  - ▷ IBM Q
  - ▷ Dwave Leap (não no Brasil)
  - ▷ AWS Braket (pago)
  - ▷ Azure Quantum (pago)
- Vários outros sendo construídos em diversos lugares



# Empresas

Software e consultoria



Zapata Computing



Q<sup>x</sup>Branch



quantumbenchmark

QILIMANJARO

|EeroQ>

Hardware quântico



XANADU



Microsoft



... e muitas outras



# Empresas

## THE EUROPEAN QUANTUM COMPUTING STARTUP LANDSCAPE

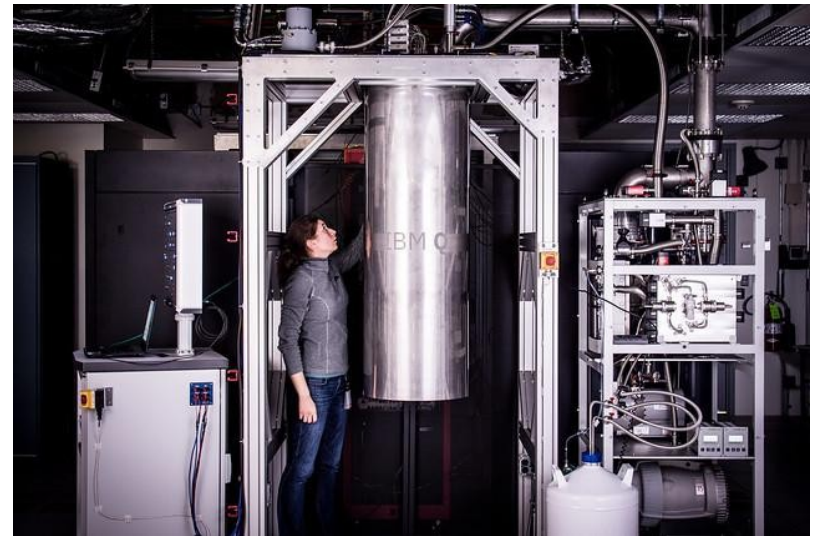


# Programando um computador quântico

- IBM: circuit composer, Qasm, Qiskit (Python)
- Rigetti: pyQuil (Python)
- Xanadu: PennyLane (Python)
- Dwave: Ocean (Python)
- Microsoft: Q#
- etc

# IBM Quantum

- Boa oportunidade para experimentar computadores QREI
- Sistemas premium
  - ▷ Até 433 qubits
- Sistemas públicos
  - ▷ Vários com 5 ou 7 qubits
  - ▷ IBM QASM Simulator (até 32 qubits simulados)
- Acessem em <https://quantum-computing.ibm.com>



(\*) IBM Research

# Como programar no IBM Quantum

- Comece online, sem precisar instalar nada!
- Vá até <https://quantum-computing.ibm.com>
- Comece pelo Circuit Composer

The screenshot displays the IBM Quantum Experience web interface. At the top, there's a navigation bar with 'File', 'Edit', 'Inspect', 'View', 'Share', and 'Help'. Below that, a toolbar contains various quantum gates like H, CNOT, T, S, Z, S†, T†, U1, and single-qubit rotation gates (RX, RY, RZ, U3, Y). The main workspace shows a quantum circuit with three qubits (q0, q1, c2). Q0 has an H gate, followed by a CNOT with q1 as the control and q0 as the target. Q1 has a CNOT with q0 as the control and q1 as the target. Both q0 and q1 are then measured. Below the circuit, a 'Measurement Probabilities' bar chart shows two bars for outcomes '00' and '11', both at approximately 50%. To the right, a 'Q-sphere' visualization shows the state of the qubits on a Bloch sphere. On the far right, a 'Code editor' window shows the QASM code for the circuit.

```
QASM
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[2];
5 creg c[2];
6
7 h q[0];
8 cx q[0],q[1];
9 measure q[0] -> c[0];
10 measure q[1] -> c[1];
```

(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

# Como programar o IBM Q

- O código é gerado automaticamente em QASM e em Python (Qiskit)

The screenshot displays the IBM Quantum Experience web interface. At the top, there's a navigation bar with 'File', 'Edit', 'Inspect', 'View', 'Share', and 'Help' menus. Below this, the current circuit is titled 'Circuits / DemonstracaoSlide Saved'. A toolbar contains various quantum gates like H, CNOT, T, S, Z, S†, T†, U1, and others. The main workspace shows a quantum circuit with three qubits: q0, q1, and c2. q0 has an H gate, followed by a CNOT gate with q1 as the control. q1 has a CNOT gate with q0 as the control. Both q0 and q1 are then measured. The measurement results are shown in a 'Measurement Probabilities' bar chart, which displays two bars for the states '00' and '11', both with a probability of approximately 50%. To the right of the chart is a 'Q-sphere' visualization showing the state of the qubits on a Bloch sphere. A code editor on the right side shows the corresponding Qiskit Python code:

```
Qiskit v
Read only
1 from qiskit import QuantumRegister,
  ClassicalRegister, QuantumCircuit
2 from numpy import pi
3
4 qreg_q = QuantumRegister(2, 'q')
5 creg_c = ClassicalRegister(2, 'c')
6 circuit = QuantumCircuit(qreg_q,
  creg_c)
7
8 circuit.h(qreg_q[0])
9 circuit.cx(qreg_q[0], qreg_q[1])
10 circuit.measure(qreg_q[0], creg_c[0])
11 circuit.measure(qreg_q[1], creg_c[1])
```

(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

# Como programar o IBM Q

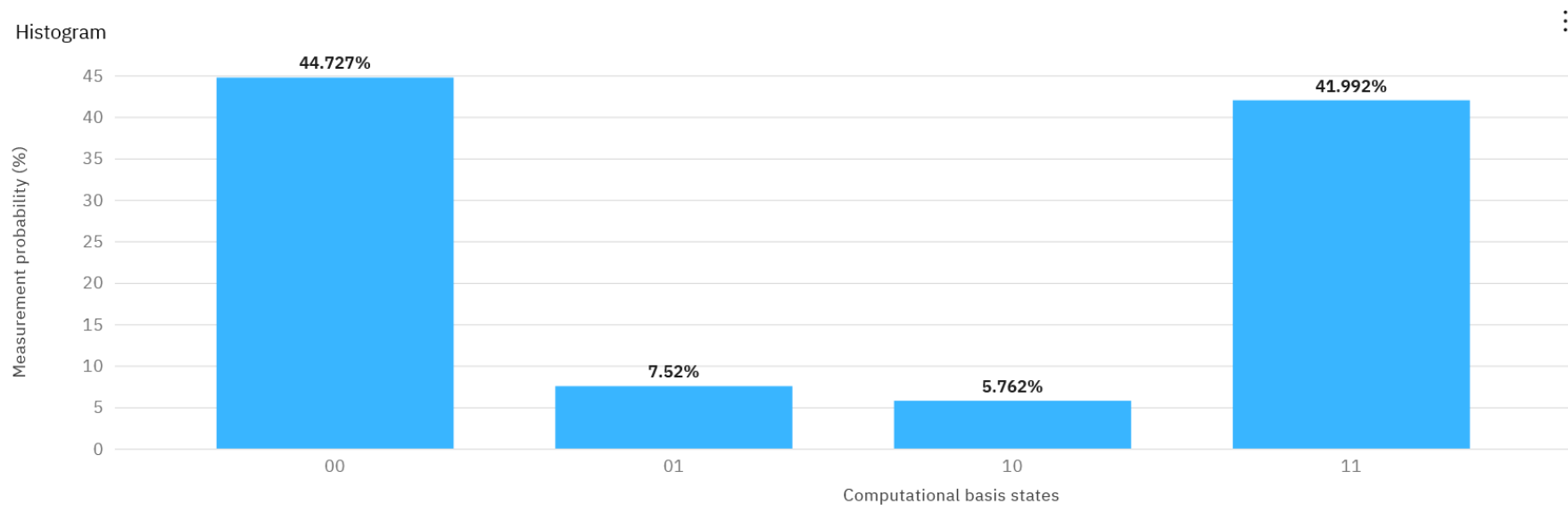
- Jobs submetidos para o computador quântico real podem demorar! Também é possível simular

The screenshot displays the IBM Quantum Experience web interface. At the top, there's a navigation bar with 'File', 'Edit', 'Inspect', 'View', 'Share', and 'Help' menus. A 'Run Settings' dropdown is set to 'Run on ibmq\_bogota'. The main workspace shows a quantum circuit with three qubits (q0, q1, c2). Qubit q0 has an H gate, followed by a CNOT gate with q1 as the control and q0 as the target. Qubit q1 has a CNOT gate with q0 as the control and q1 as the target. Both qubits q0 and q1 are measured at the end of the circuit. Below the circuit, the 'Measurement Probabilities' section shows a bar chart with two bars for computational basis states '00' and '11', both at approximately 50% probability. To the right, the 'Q-sphere' visualization shows a Bloch sphere with a vertical line connecting the poles, representing the state of the qubits. A 'Phase' indicator is visible at the bottom left of the sphere. On the right side of the interface, a 'Jobs' panel shows 'Pending jobs (1)' with a job ID '5f7cc9b8146c6100138599...' in a 'QUEUED' state. Below it, 'Completed jobs (0)' are listed. The job details include 'Backend: ibmq\_bogota', 'Provider: ibm-q-research/Franklin-Marquez/main', and 'Expected to run: in 2 hours'.

(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

# Como ler os resultados

## Result



(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

# Mapa de acoplamentos

online

ibmq\_santiago (5 qubits, QV32)



Queue: 34 jobs

online

ibmq\_bogota (5 qubits, QV32)



Queue: 14 jobs

Reservable

online

ibmq\_rome (5 qubits, QV32)



Queue: 32 jobs

1 upcoming reservations

ibmq\_bogota v1.0.4

online

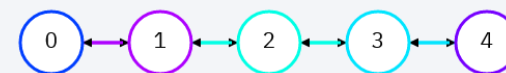


Queue: 14 jobs

Reservable

Providers with access:

ibmq-research/Franklin-Marquez /main



Single-qubit U2 error rate



CNOT error rate



Download Calibrations

Qubits  
5

Online since  
Jun 03, 2020

Maximum shots  
8192

Quantum Volume  
32

Basis gates  
u1, u2, u3, cx, id

Maximum circuits  
900

(C) IBM Quantum Experience. Disclaimer: I am not an IBM Employee and I do not represent IBM.



# Programando em alto nível

The screenshot displays the IBM Quantum Experience interface. At the top, there is a menu bar with 'File', 'Edit', 'Inspect', 'View', 'Share', and 'Help'. On the right, there are 'Run Settings' and a 'Run on ibmq\_bogota' button. Below the menu, the breadcrumb 'Circuits / DemonstrationSlideTranspiler Saved' is visible, along with 'Code', 'Docs', and 'Jobs' buttons.

The central part of the interface shows a quantum circuit with five qubits (q0 to q4) and a classical register (c5). The circuit starts with an H gate on q0, followed by CNOT gates from q0 to q1, q1 to q2, q2 to q3, and q3 to q4. Each qubit then has a Z gate. The circuit concludes with measurements on qubits 0, 1, and 2, which are stored in classical bits c[0], c[1], and c[2] respectively.

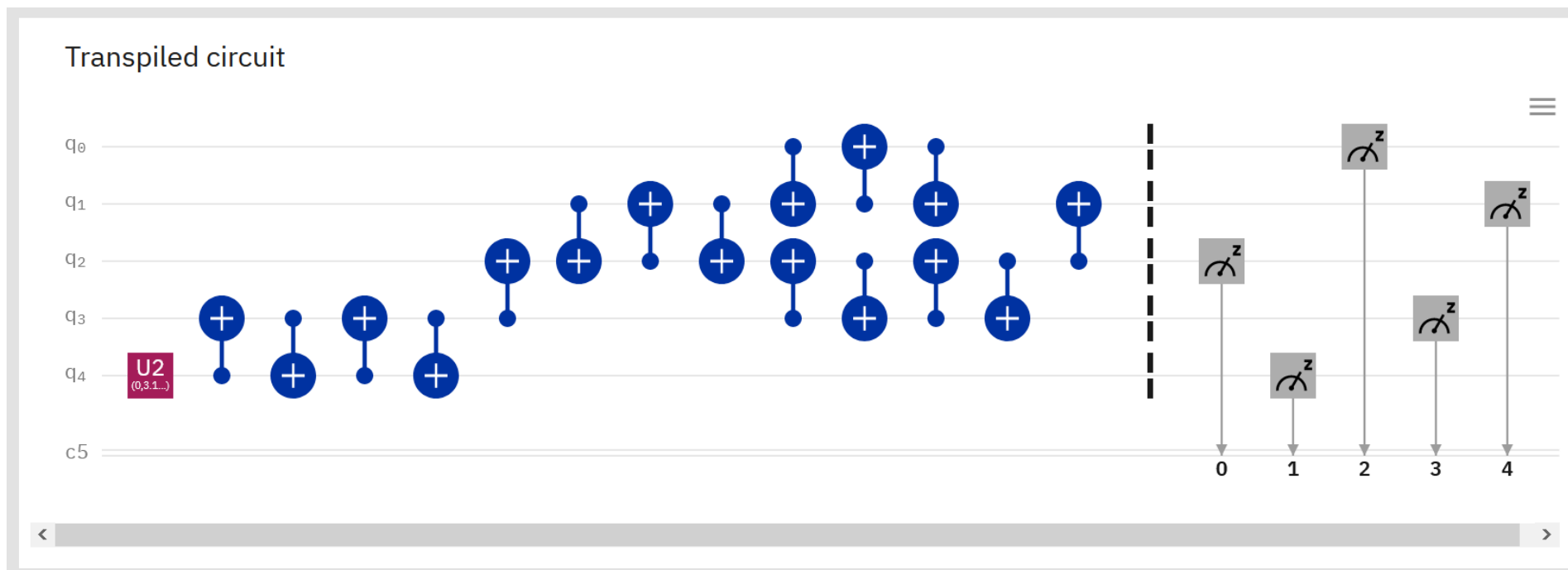
Below the circuit, there are sections for 'Measurement Probabilities' and 'Q-sphere'. The 'Q-sphere' section has checkboxes for 'State' (checked) and 'Phase angle' (unchecked).

On the right side, a 'Code editor' window is open, showing the QASM code for the circuit:

```
QASM
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[5];
5 creg c[5];
6
7 h q[0];
8 cx q[0],q[1];
9 cx q[0],q[2];
10 cx q[0],q[3];
11 cx q[0],q[4];
12 measure q[0] -> c[0];
13 measure q[1] -> c[1];
14 measure q[2] -> c[2];
```

(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

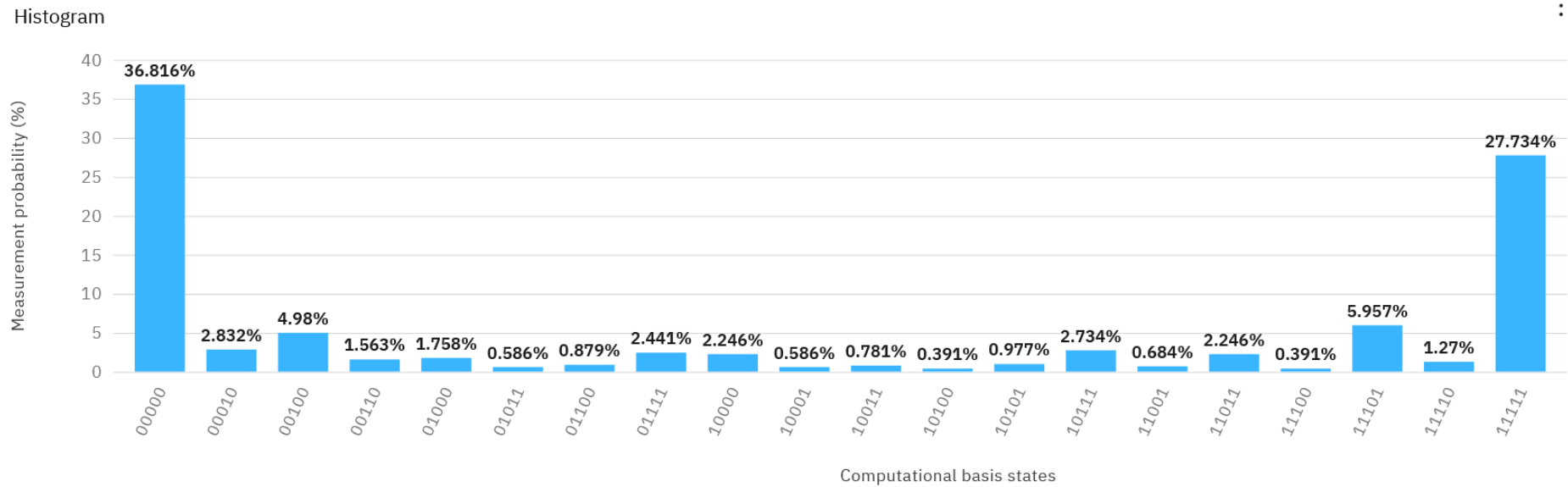
# Transpilação



(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

# Resultado obtido pelo circuito transpilado

## Result



(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

# Executando no Quantum Lab (Notebook)

The screenshot displays the IBM Quantum Experience Jupyter Notebook interface. The top navigation bar includes the text "IBM Quantum Experience", a "Feedback" button, and search and user icons. Below this is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The right side of the menu bar shows "Trusted", "Python 3", and the "jupyter" logo. A toolbar contains icons for file operations, a "Run" button, and other controls. The main area shows two code cells. The first cell, labeled "In [7]:", contains the following Python code:

```
qreg_q = QuantumRegister(2, 'q')
creg_c = ClassicalRegister(2, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)

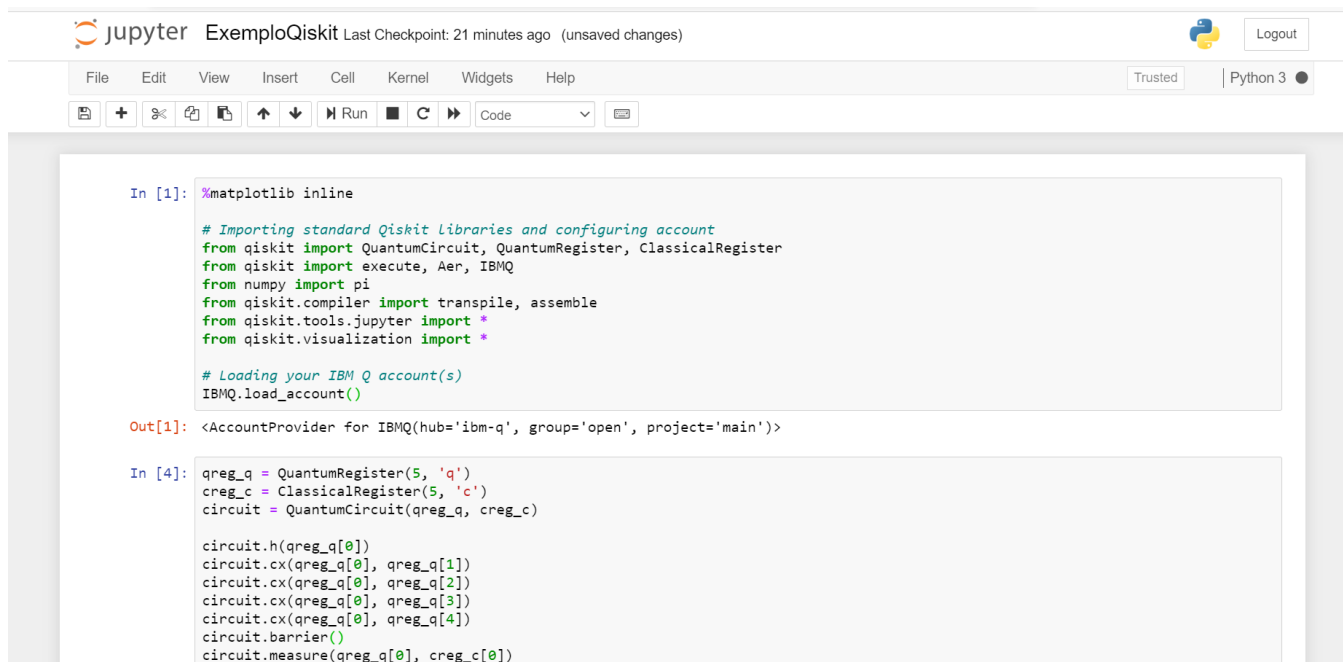
circuit.h(qreg_q[0])
circuit.cx(qreg_q[0], qreg_q[1])
```

The output of this cell is a string representation of the instruction set: `<qiskit.circuit.instructionset.InstructionSet at 0x7f24414ec550>`. The second cell, labeled "In [8]:", contains the code `circuit.draw('mpl')`. Below this cell, a quantum circuit diagram is shown for qubit  $q_0$ . It features a blue square gate labeled "H" followed by a blue dot on the wire, representing a measurement operation.

(C) IBM Quantum Experience.  
Disclaimer: I am not an IBM Employee and I do not represent IBM.

# Executando localmente

- Necessário Python 3
- Recomendável Jupyter Notebook
- Baixe o qiskit em <https://qiskit.org> ou via `pip install qiskit`
- Recomendável `pip install qiskit[visualization]`



The screenshot shows a Jupyter Notebook titled "ExemploQiskit" with a "Python 3" kernel. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The notebook contains two code cells:

```
In [1]: %matplotlib inline

# Importing standard Qiskit Libraries and configuring account
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister
from qiskit import execute, Aer, IBMQ
from numpy import pi
from qiskit.compiler import transpile, assemble
from qiskit.tools.jupyter import *
from qiskit.visualization import *

# Loading your IBM Q account(s)
IBMQ.load_account()

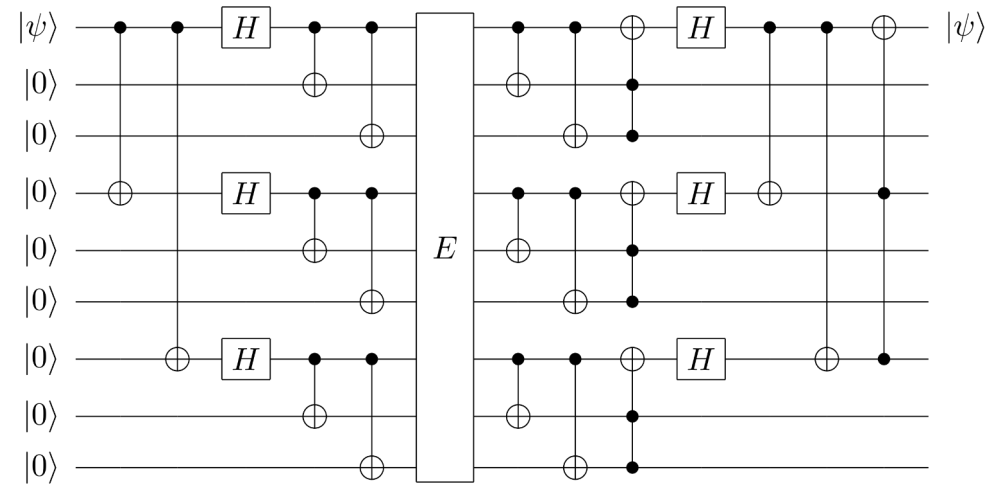
Out[1]: <AccountProvider for IBMQ(hub='ibm-q', group='open', project='main')>

In [4]: qreg_q = QuantumRegister(5, 'q')
        creg_c = ClassicalRegister(5, 'c')
        circuit = QuantumCircuit(qreg_q, creg_c)

        circuit.h(qreg_q[0])
        circuit.cx(qreg_q[0], qreg_q[1])
        circuit.cx(qreg_q[0], qreg_q[2])
        circuit.cx(qreg_q[0], qreg_q[3])
        circuit.cx(qreg_q[0], qreg_q[4])
        circuit.barrier()
        circuit.measure(qreg_q[0], creg_c[0])
```

# Correção de erros quânticos

- É possível corrigir erros
- Nós *devemos* corrigir erros de computadores quânticos
- Entretanto, é caro (requer muitos qubits)
- Taxa de erros precisa estar abaixo de um limiar



(\*) Shor code: 1 logical qubit is mapped to 9 physical qubits.  
Source: Wikipedia.

# Algoritmos importantes

- **Algoritmos**

- ▷ Shor (1994)
- ▷ Grover (1996)
- ▷ Element distinctness (2004)
- ▷ HHL (2009)

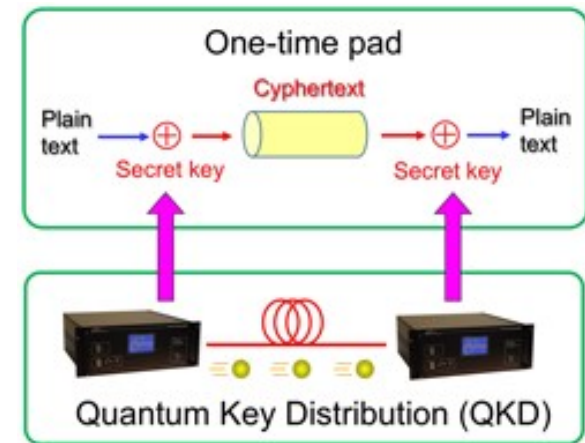
- **Técnicas**

- ▷ HSP
- ▷ Amplitude amplification
- ▷ Quantum walks
- ▷ QAOA
- ▷ Quantum annealing

- **Para uma lista completa veja <http://quantumalgorithmzoo.org/>**

# Consequências para criptografia

- CQ quebra RSA (mas requer muitos qubits)
- Motiva a criptografia pós-quântica
- Novas oportunidades para troca de chaves e comunicação segura



(\*) Source: [www.nict.go.jp](http://www.nict.go.jp)



(\*) Source: [physicsworld.com](http://physicsworld.com)



# Como se preparar nessa área

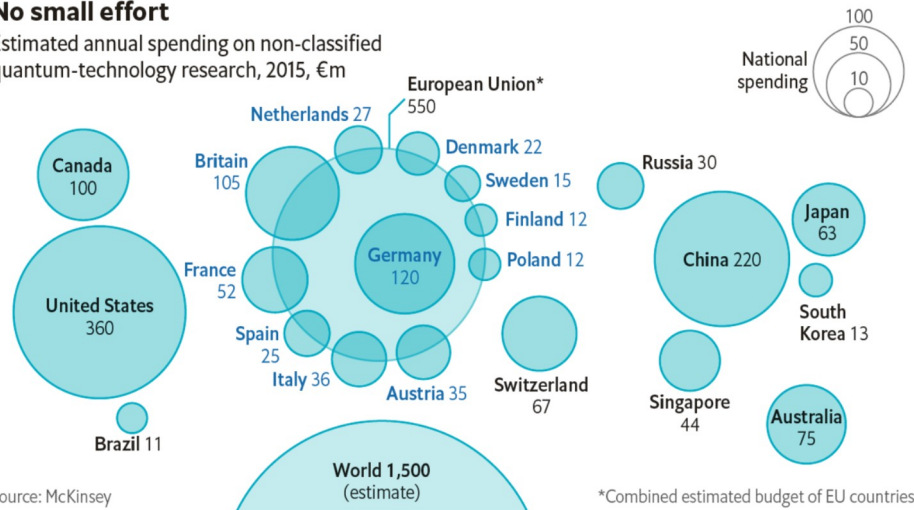
- **Graduação: matemática, computação, engenharia, física etc (várias possibilidades)**
- **Muita atenção aos cursos de Álgebra Linear**
- **Aprenda a programar os computadores quânticos da IBM (github programaquantica)**
- **Iniciação científica, se possível, é interessante!**
- **Mestrado/Doutorado**

# Conclusão

- QREI / Impacto limitado em negócios: próximos (quantos?) anos
- Correção de erros / Impacto mais amplo: médio/longo prazo
- Tolerância a falhas / Escala completa: longuíssimo prazo... paciência!
- Fiquem atentos! Muitas oportunidades na academia e na indústria

## No small effort

Estimated annual spending on non-classified quantum-technology research, 2015, €m



Source: McKinsey

The New York Times

## *The Next Tech Talent Shortage: Quantum Computing Researchers*

By Cade Metz

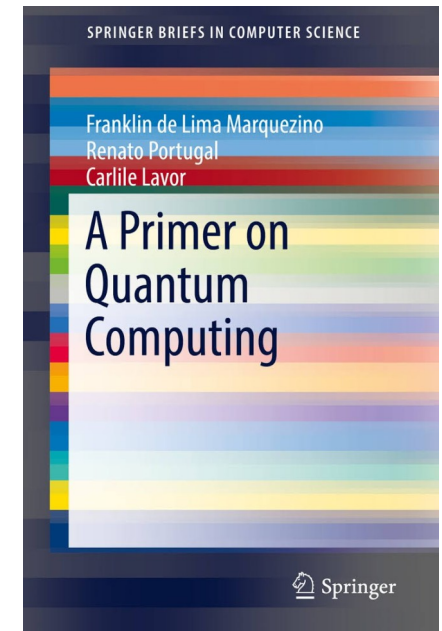
Oct. 21, 2018

Christopher Savoie, founder and chief executive of a start-up called Zapata, offered jobs this year to three scientists who specialize in an increasingly important technology called quantum computing. They accepted.

Several months later, the Cambridge, Mass., company was still waiting for the State

# Para saber mais

- **Apostila**  
[github.com/programaquantica](https://github.com/programaquantica)
- **Livro**  
[www.springer.com/gp/book/9783030190651](http://www.springer.com/gp/book/9783030190651)
- **Seminário PESC**  
[www.youtube.com/watch?v=MXovwCMx3uA%22](https://www.youtube.com/watch?v=MXovwCMx3uA%22)



# Obrigado!

## Como Programar Computadores Quânticos Reais

Franklin de Lima Marquezino  
Universidade Federal do Rio de Janeiro  
franklin@cos.ufrj.br