Mackenzie

SSCAD 24

XXV Simpósio em Sistemas Computacionais de Alto Desempenho

Minicurso 2
Introdução à Computação Quântica com IBM/Qiskit

Calebe P. Bianchini
Giancarlo P. Gamberi
Ryan M. A. Santos

Sorte?

Manipulação de probabilidade

Resultados favoráveis

Minicurso

“Entender” princípios da mecânica quântica

Entender funcionamento da computação quântica

Aprender sobre qiskit

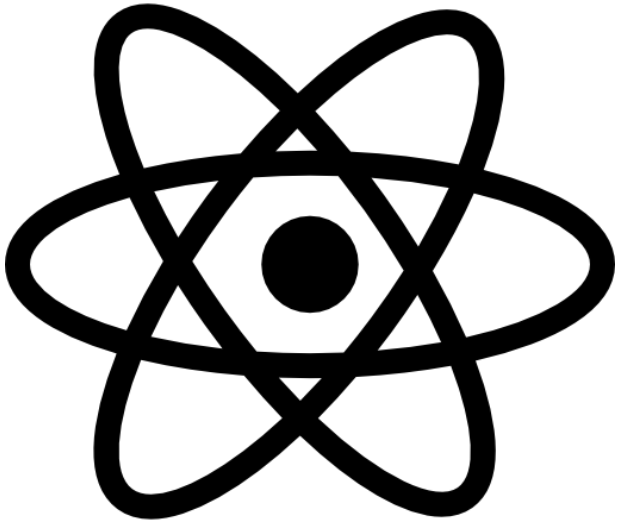
Operar/experimentar circuitos quânticos (básicos/intermediários)

Introdução

Computação quântica:

- Primeiras propostas por Feynman, 1982
- Turing completo por Deutsch, 1985
- Fatoração eficiente por Shor, 1994
- Busca $O(\sqrt{N})$ por Grover, 1996
- Primeiro demonstração experimental de um computador quântico em 1998
- Supremacia quântica em 2019

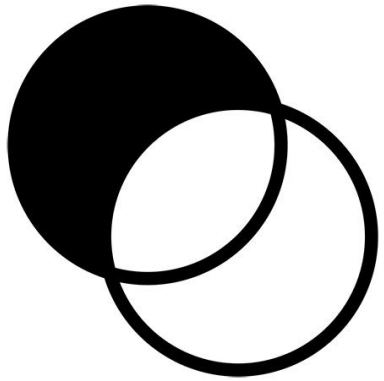
Mecânica quântica



Superposição

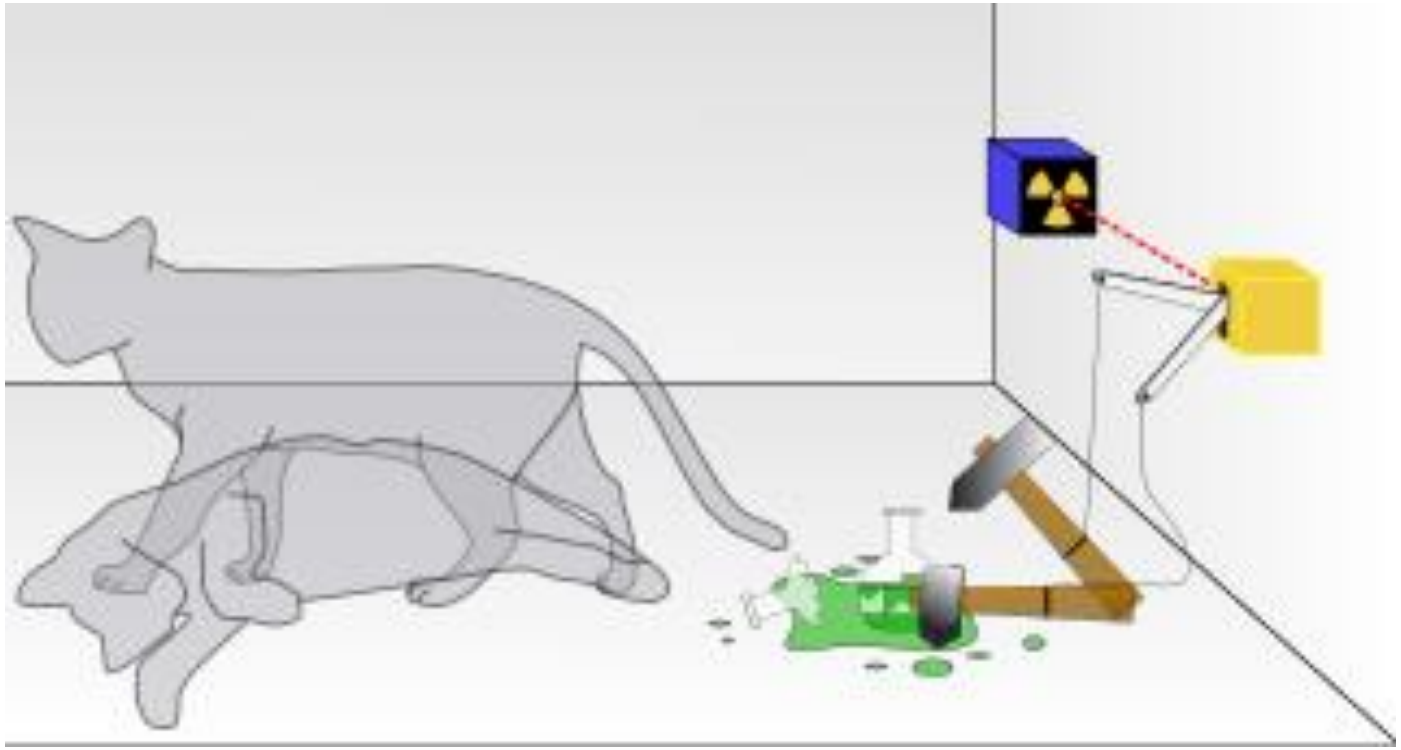
Emaranhamento

Decoerência



Superposição

Princípio fundamental a computação quântica



Gato de Schrödinger

- Gato
- Veneno
- Substancia radioativa

Observação

- Superposição desfeita
- Um dos possíveis resultados é resolvido no momento da observação

Emaranhamento

Comunicação em sistemas quânticos

Partículas correlacionadas

- Estados entrelaçados
- Independe da distancia
- Observação desfaz superposição em ambos

Ruido

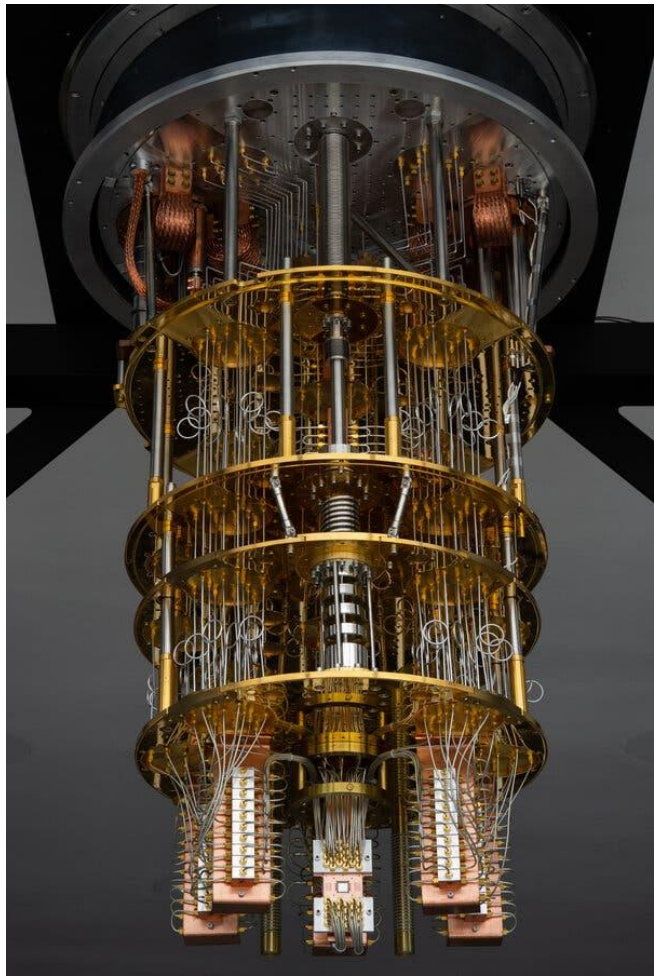
Incerteza

Flutuações

Inerente

Decoerência

Aleatório



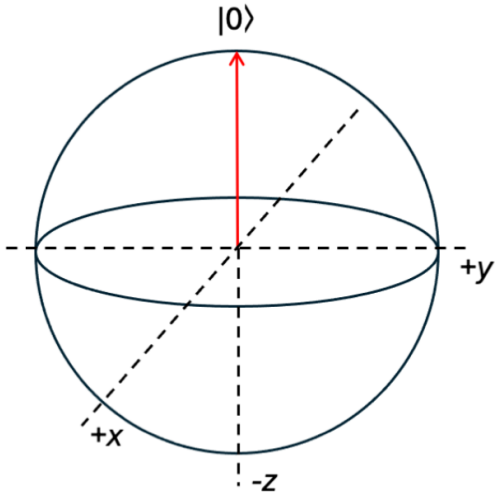
Computação quântica

Como tais conceitos são utilizados para computação?

0 | 1

Qubit Vs Bit

Vs



Qubit

- Superposição
- Portas quânticas
- Paradigma inédito

Bit

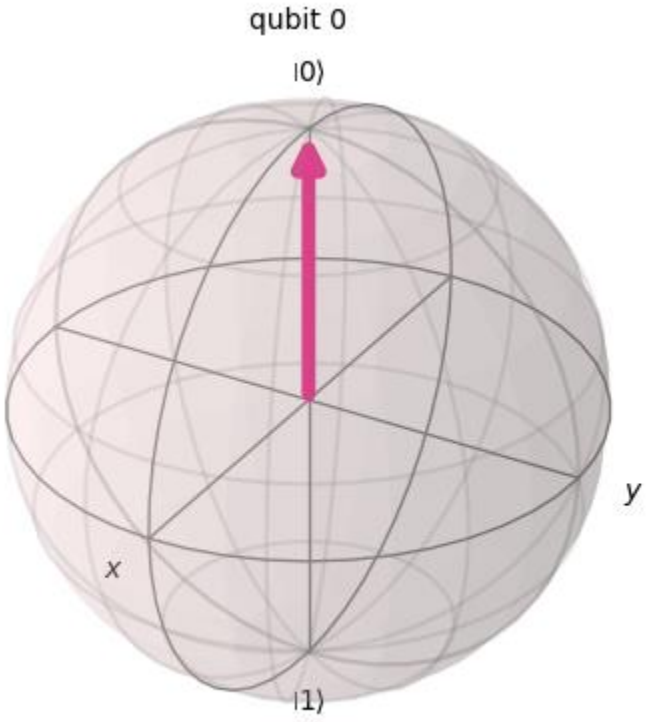
- Dois estados
- Portas clássicas
- Paradigma moderno

Representação

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha|0\rangle + \beta|1\rangle = |\psi\rangle \qquad |\alpha|^2 + |\beta|^2 = 1$$

Esfera de bloch



Efetiva visualmente

Ineficiente para operações complexas

Múltiplos qubits

Produto tensorial:

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad |10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad |11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$|000\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Portas logicas

- Identidade
  - Pauli-X
  - Pauli-Y
  - Pauli-Z
- Hadamard

Identidade

Mantem o estado

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$I|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \alpha|0\rangle + \beta|1\rangle = |\psi\rangle$$

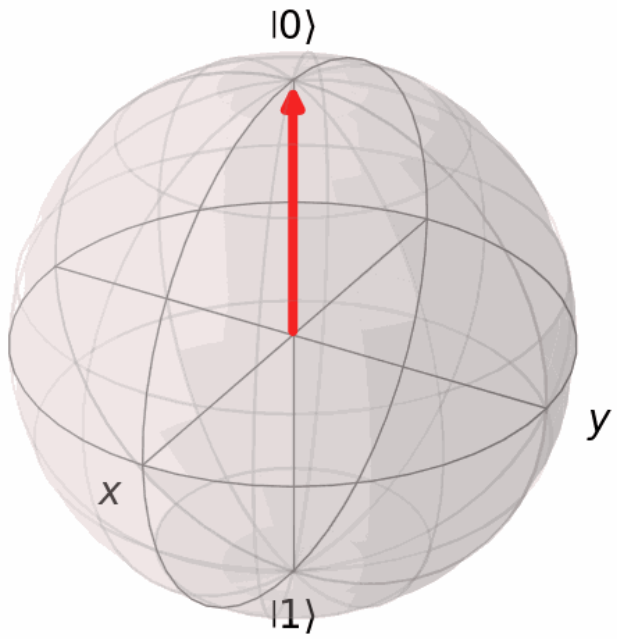
Pauli-X

Inversão de bit

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$$

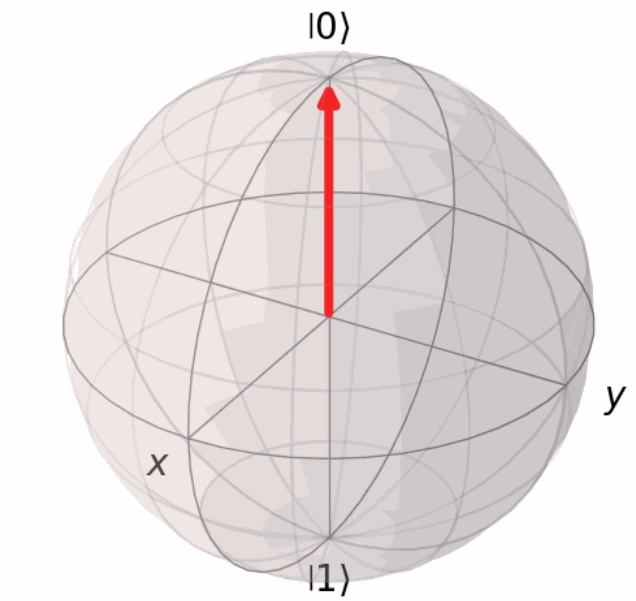


Pauli-Y

Inversão de bit através do eixo Y

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Y|0\rangle = i|1\rangle$$
$$Y|1\rangle = -i|0\rangle$$



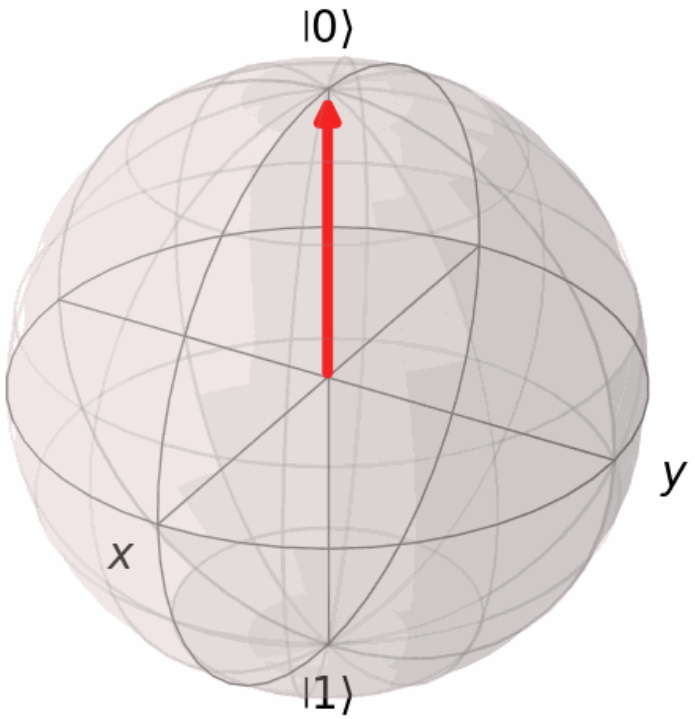
Iniciando uma superposição

Hadamard

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

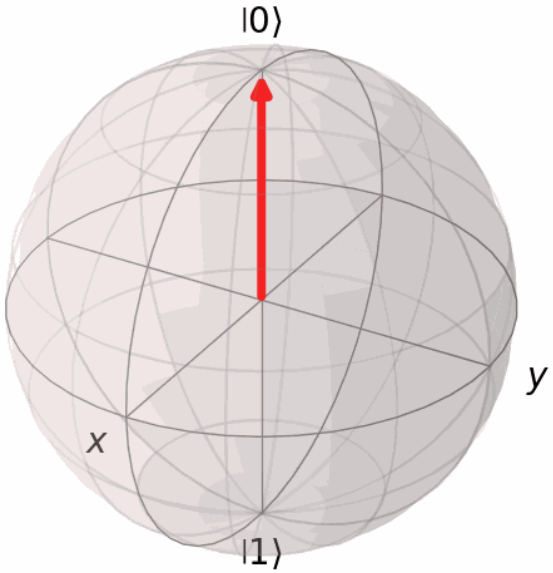


Pauli-Z

Inversão de bit através do eixo Y

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$Z|0\rangle = |0\rangle$$
$$Z|1\rangle = -|1\rangle$$



- IBM
- Biblioteca python
- Experimentação de sistemas quânticos

Desenvolvendo um Circuito quântico

Bibliotecas utilizadas/recomendadas:

- Qiskit
- Matplotlib
- Numpy

Importações

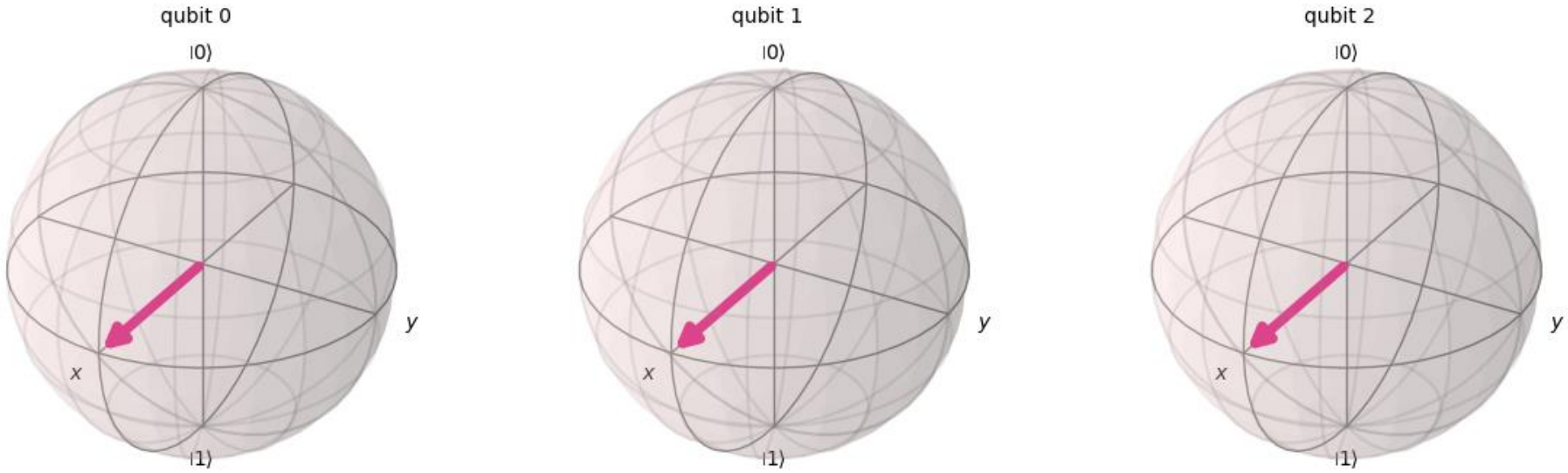
```
1. from qiskit import QuantumCircuit, transpile
2. from qiskit.providers.basic_provider import BasicProvider
3. from qiskit.quantum_info import Statevector
4. from qiskit.visualization import
   plot_histogram,plot_bloch_multivector
5. from math import pi
```

Inicialização

```
1. qc = QuantumCircuit(3)
2. qc.h(0)
3. qc.h(1)
4. qc.h(2)
```

Visualização esferas de bloch

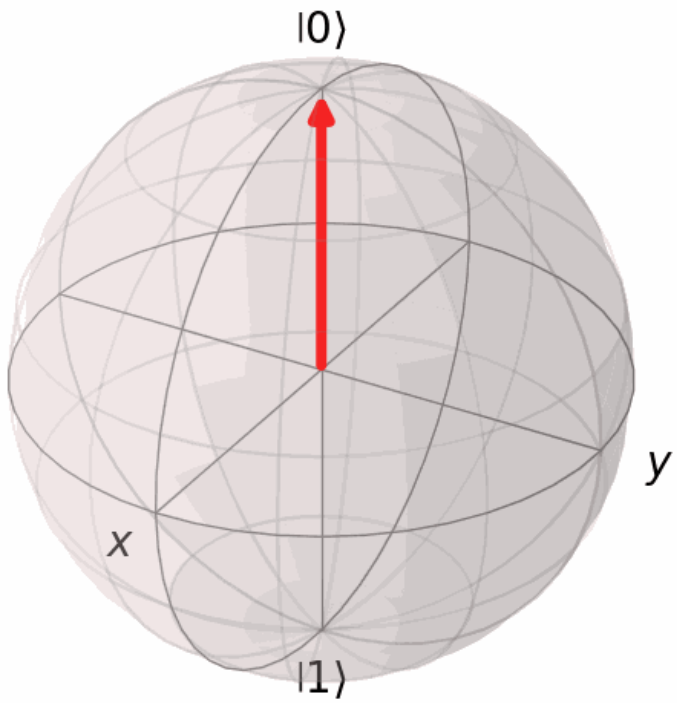
```
plot_bloch_multivector(Statevector(qc))
```



Algumas operacões teste...

No primeiro qubit:
Rotação Z de 135◦ = 3π/4

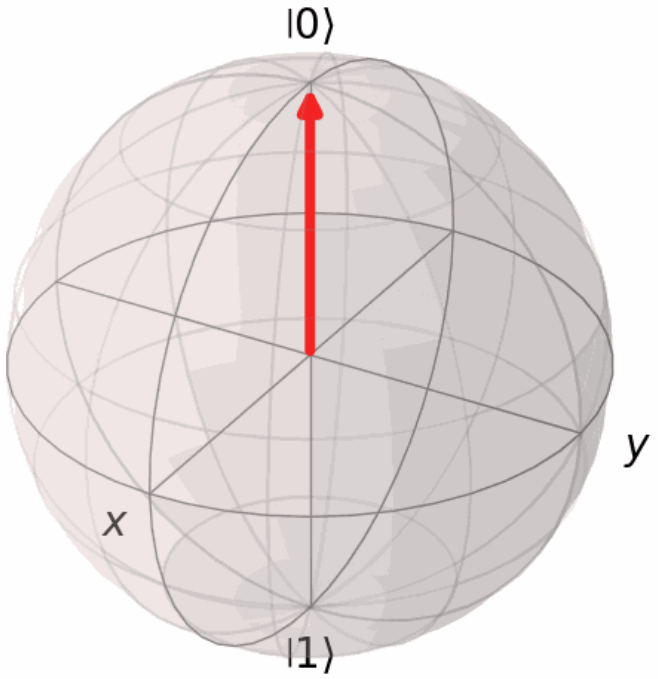
```
1. qc.rz(3*pi/4,0)
```



Algumas operacões teste...

No segundo qubit:
Rotação Y de 45∘ = π/4

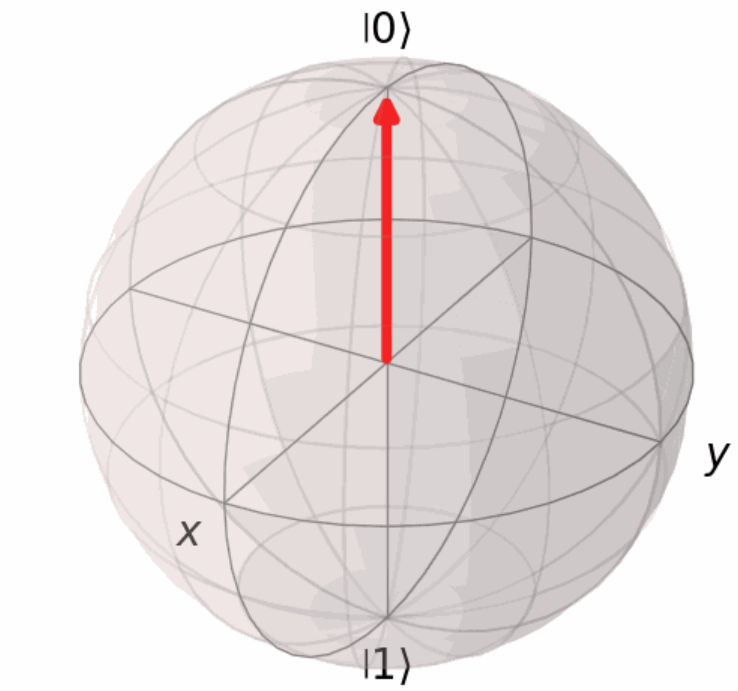
```
1. qc.ry(pi/4,1)
```



Algumas operacões teste...

No terceiro qubit:
Rotação Y de -45◦ | 315 ◦ =
-π/4 seguida por uma
rotação X de 45◦ = π/4

```
1. qc.ry(-pi/4,2)
2. qc.rx(pi/4,2)
```



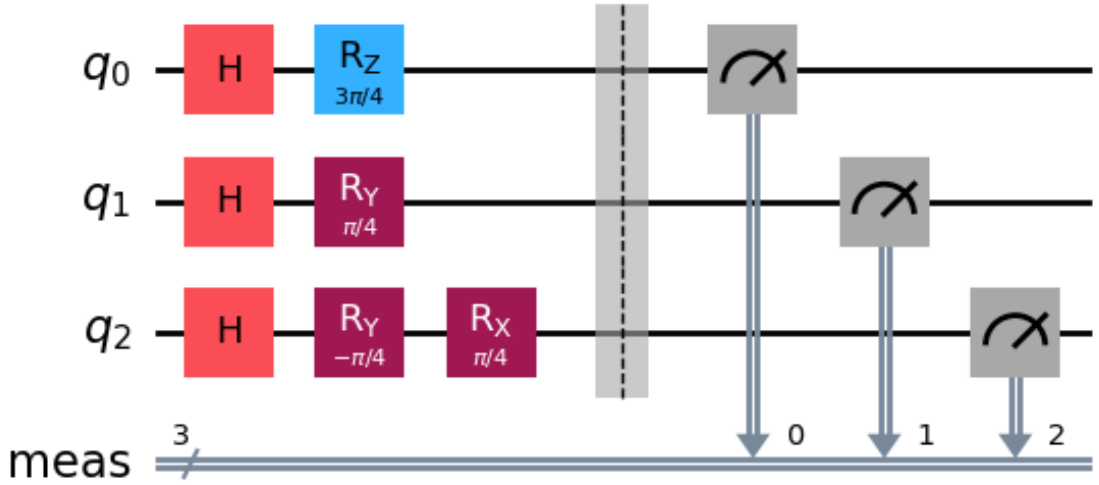
Exercício

Criar um circuito de 3 qubits

Aplicar rotações aos qubits

Observar comportamento

Medição



Aplicar medição

```
1. qc.measure_all()
```

```
1. qc.draw("mpl")
```

Provedores

Fornecedores para computação quântica

IBM

Google

Microsoft

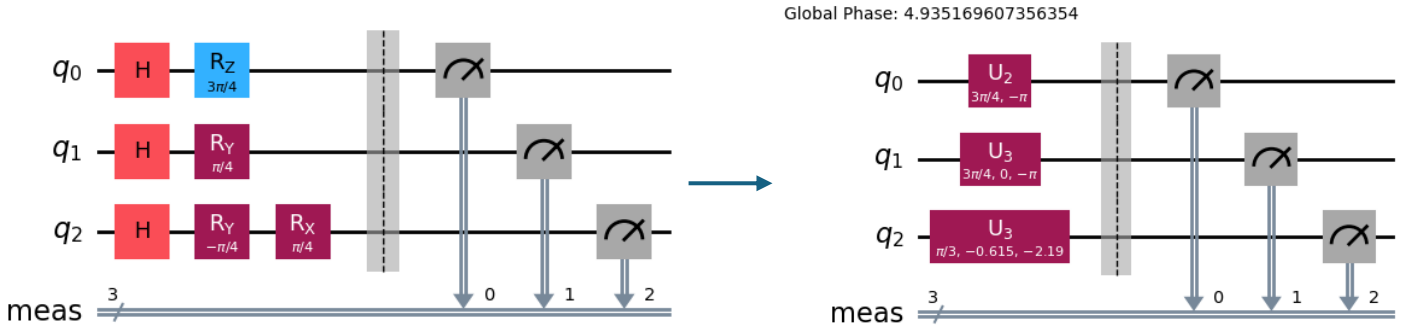
Simuladores

```
1. provider = BasicProvider()
2. backend=provider.get_backend("basic_simulator")
```

Transpilação

"compilador" quântico

```
1. new_circuit = transpile(qc, backend)
```



Execução

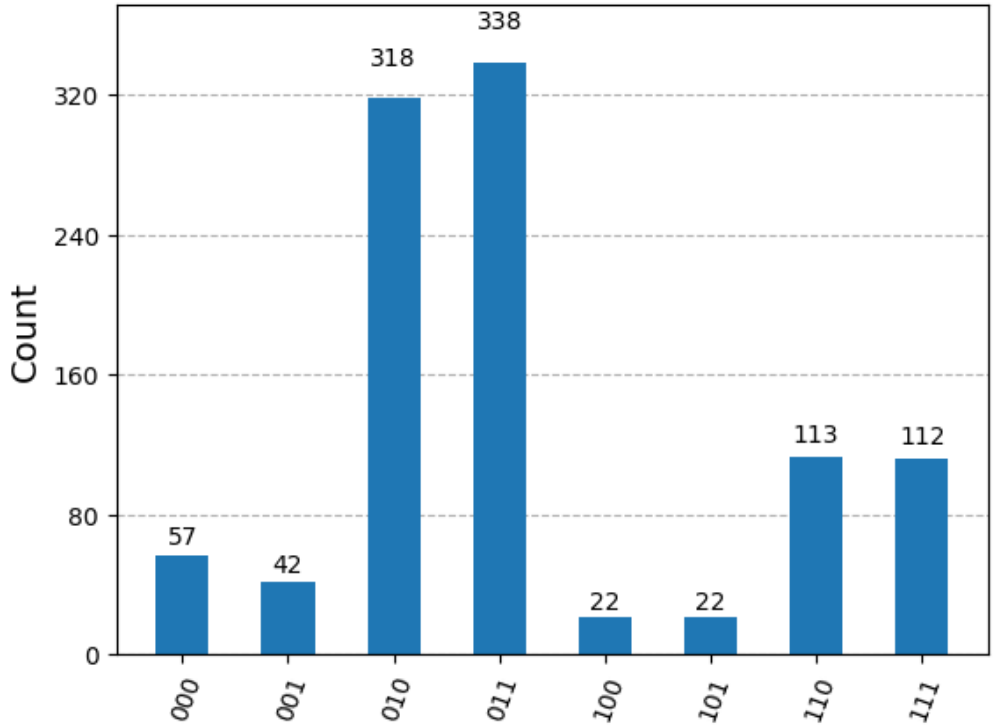
Definição final da execução

Esperar a execução

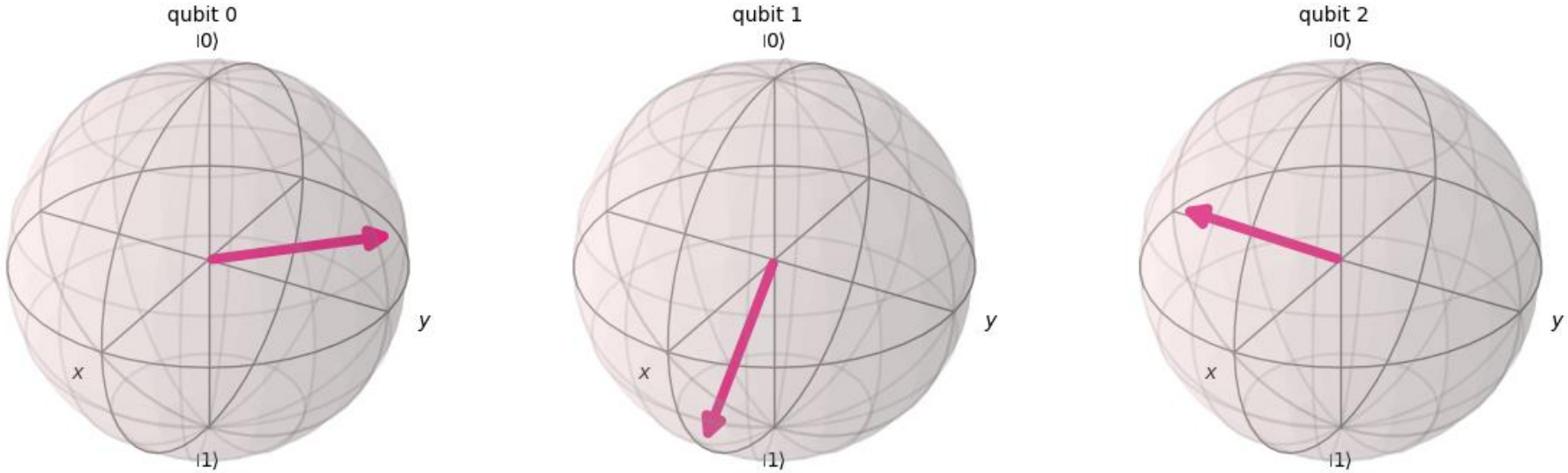
```
1. job = backend.run(new_circuit, shots=1024)
2. result = job.result()
```

Resultados

```
1. counts = result.get_counts(qc)
2. plot_histogram(counts)
```



Por que?



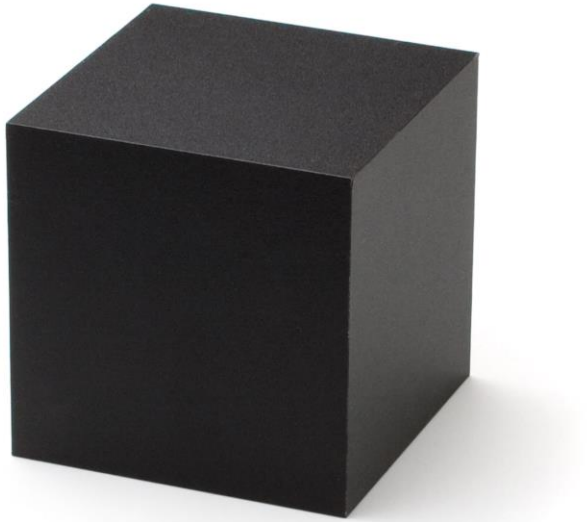
Aplicação de um circuito

Alguns exemplos de aplicações:

Busca de Grover

Autômato celular

Oraculo



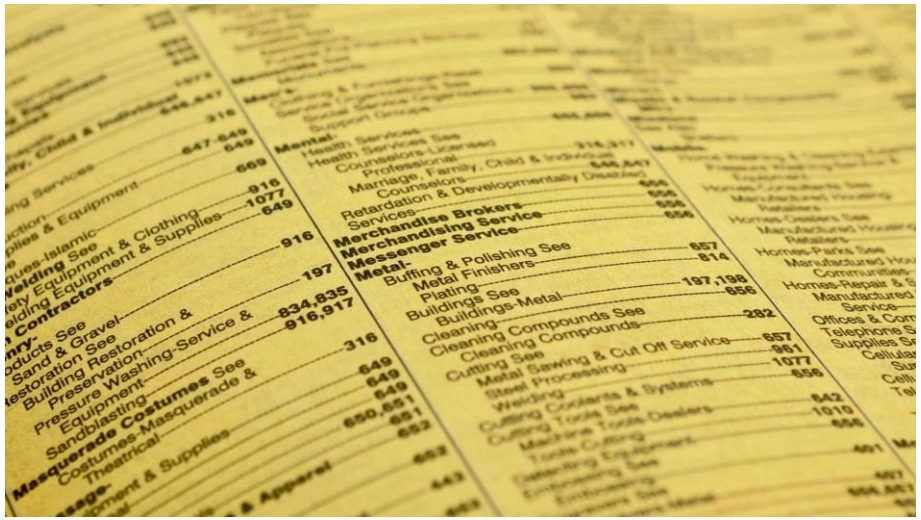
Caixa preta

Entrada e saída de informação

Não *precisamos* saber o que faz

Comum a algoritmos quânticos

Mas não necessariamente exclusivo



Busca de Grover

1996

Busca em lista desordenada

$O(\sqrt{N})$

O algoritmo (original)...

1. Inicializar o circuito em uma superposição
2. Repetir as seguintes etapas $\sqrt{N}$ vezes:
   1. Rotacionar a fase buscada em π radianos
   2. Aplicar a matriz de difusão D no circuito, definida por D = HRH, onde H éa porta Hadamard e R é uma matriz diagonal cujo primeiro elemento é 1, e o restante é −1
3. Observar o circuito88

Exercícios:
Criar um oraculo

```
1. N=3
2. faseBuscada='011’
3.  rotacaoFase = np.ones(2**N,dtype=int)
4.  rotacaoFase[int(faseBuscada,2)] = -1
5.  oraculo = DiagonalGate(rotacaoFase)
6.  oraculo.name='oraculo'
```

Inicializar superposição

Rotacionar alguma fase em Z

Sugestão:

Diagonal

Matriz de rotação

Montar matriz de rotação

Matriz diagonal

Primeiro elemento da diagonal 1

Restante -1

```
1. diagonalMatrizRotacao = -np.ones(2**N,dtype=int)
2. diagonalMatrizRotacao[0] = 1
3. matrizRotacao = DiagonalGate(diagonalMatrizRotacao)
4. matrizRotacao.name = 'R'
```

Matriz de difusão

Montar circuito de matriz de difusão:

Hadamard

Rotação

Hadamard

```
1. matrizDifusao = QuantumCircuit(N, name='D')
2. matrizDifusao.h(range(N))
3. matrizDifusao.append(matrizRotacao,range(N))
4. matrizDifusao.h(range(N))
```

Montando Grover

Oraculo

Difusão

$\sqrt{N}$ vezes

```
1. for i in range(floor(sqrt(N))):
2.     Grover.append(oraculo,range(N))
3.     Grover.append(matrizDifusao,range(N))
```

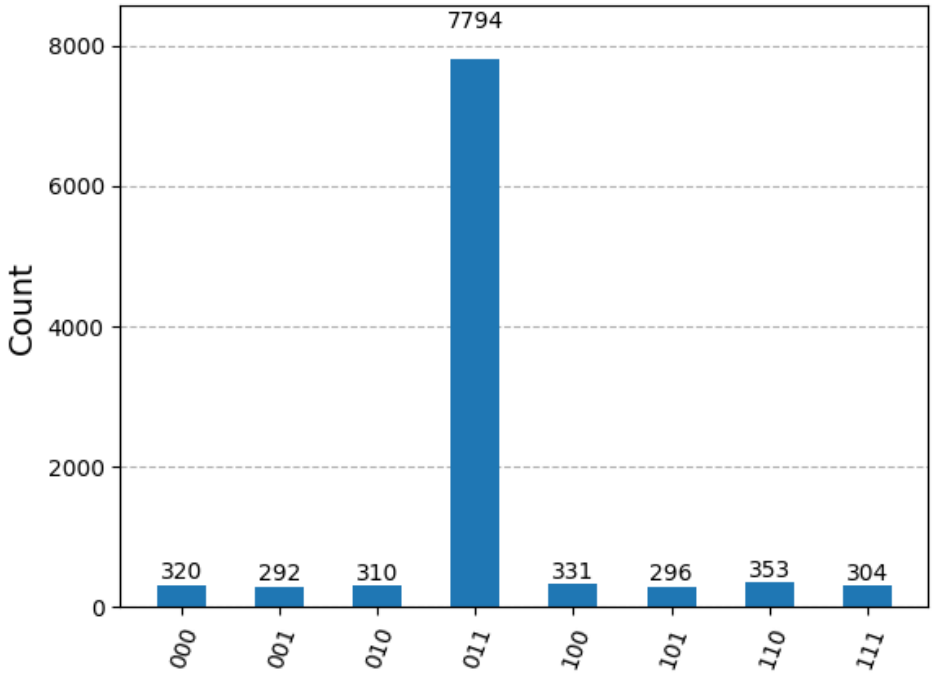
Execução

```
1. provider = BasicProvider()
2. backend=provider.get_backend("basic_simulator")
3. new_circuit = transpile(Grover, backend)
4. job = backend.run(new_circuit, shots=10000)
5. result = job.result()
6. counts = result.get_counts(Grover)
7. plot_histogram(counts)
```

Executar o circuito

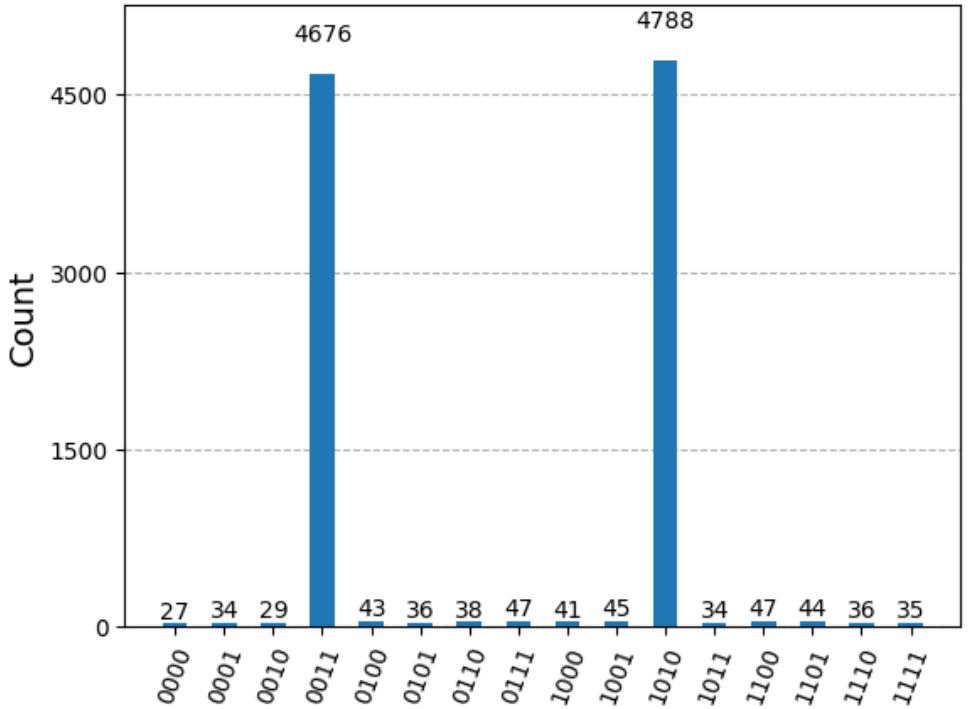
Visualizar os resultados

Resultados



Estado “marcado” acentuado

Múltiplas buscas



Expectativas

Avanço na quantidade de qubits

Redução da decoerência

Comunicação quântica

Investimento

Simulação física

IA

Repositório com os notebooks

SSCAD24-QuantumComputing

Isso é tudo pessoal!

Giancarlo P. Gamberi
giangamberi@hotmail.com.br

Calebe P. Bianchini
calebe.bianchini@mackenzie.br

Ryan M. A. Santos
ryan.marco.ismart@gmail.com