Segurança com computação quântica e algoritmo de Grover

Jefferson Deyvis

Sumário

- Algoritmo de Shor
- Distribuição de chaves quânticas
- Algoritmo de Grover

Fatoração de inteiros

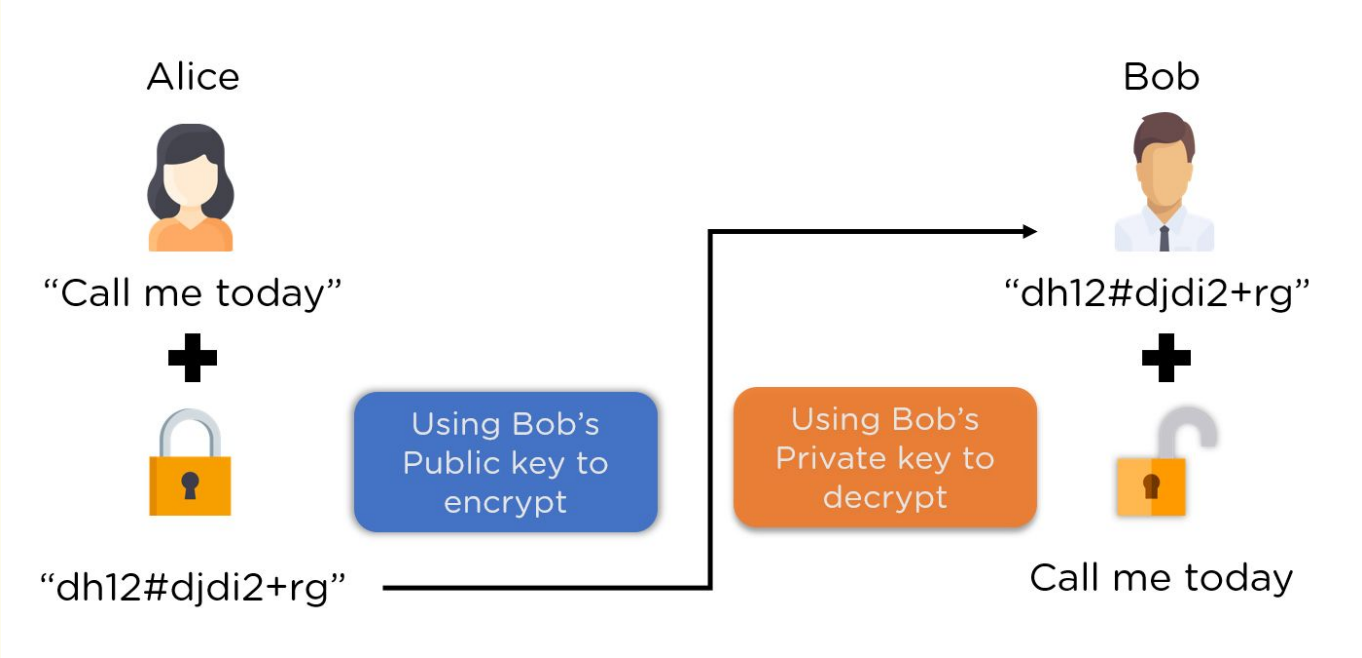
Dado um número inteiro positivo N, encontrar todos os números primos P1, P2, ..., Pn tais que:

$$N = P1 * P2 * ... * Pn$$

O desafio está na dificuldade de fatorar números inteiros grandes, principalmente quando eles são o produto de dois números primos muito grandes.

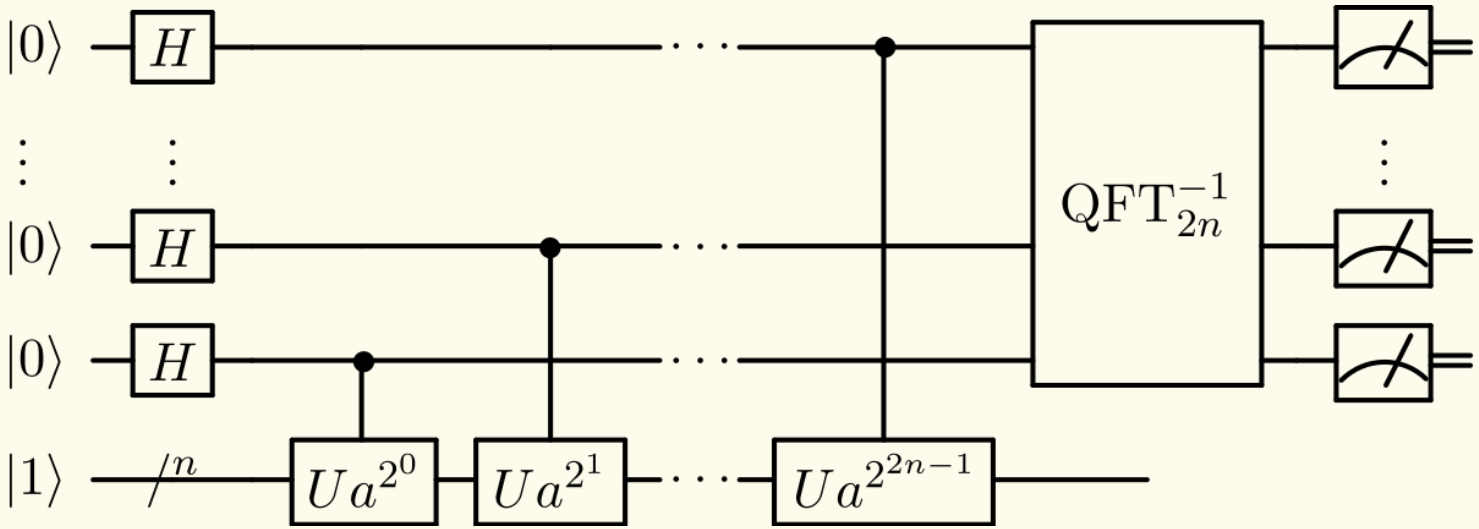
Fatoração de inteiros na criptografia

A criptografia de chave pública baseada na dificuldade desse problema é amplamente usada para proteger informações sensíveis na internet, mas a segurança desses sistemas depende da incapacidade de fatorar números inteiros grandes em tempo razoável.

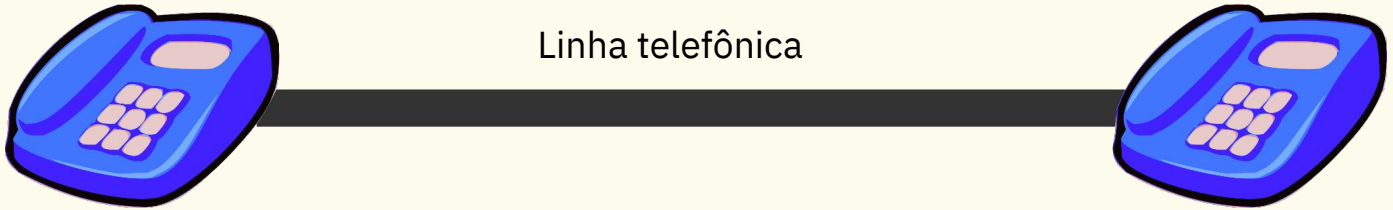


Algoritmo de Shor

calcular a "ordem" de a módulo N, onde N é um Inteiro grande e a < N é escolhido aleatoriamente.

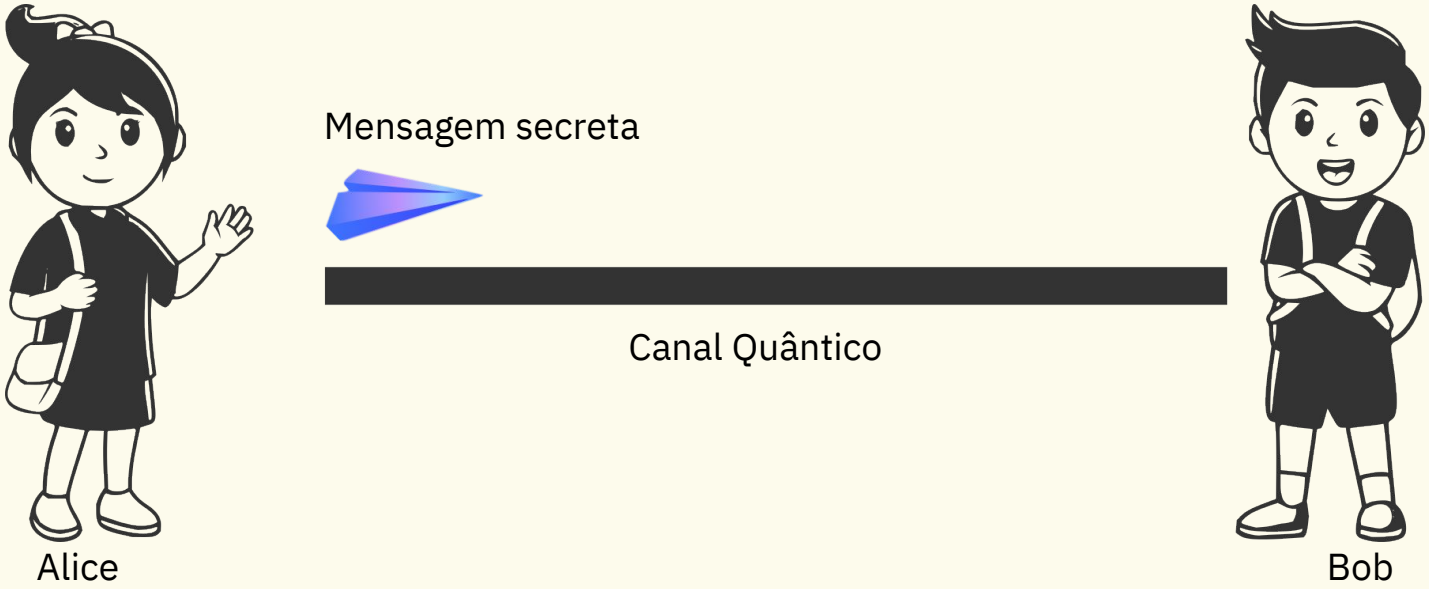


Canal de comunicação clássico



Invasão através do canal de comunicação

Canal de comunicação quântico



Protocolo BB84

Estados da base computacional de um único qubit

Estados ortogonais que formam a base Z

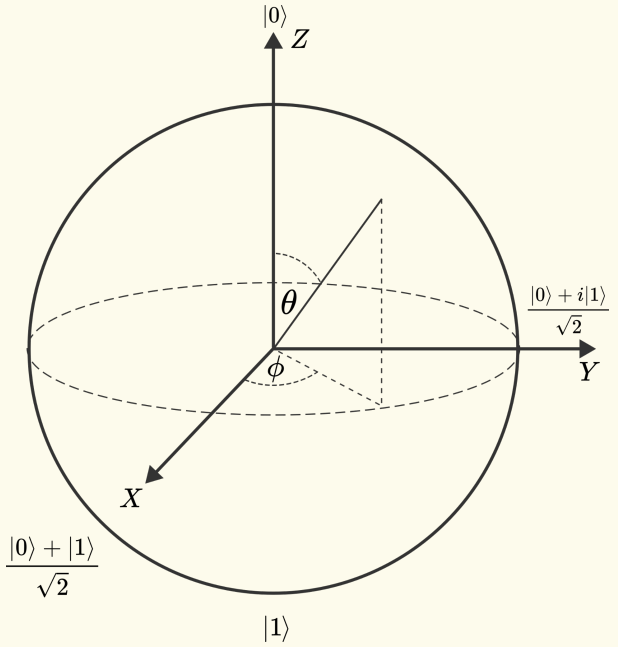
$$|0\rangle \qquad |1\rangle$$

Estados ortogonais que formam a base X

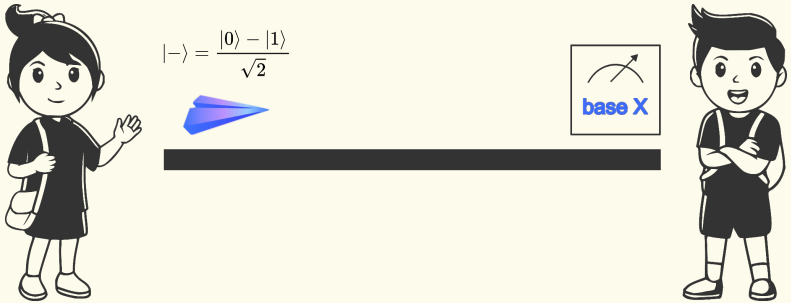
$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

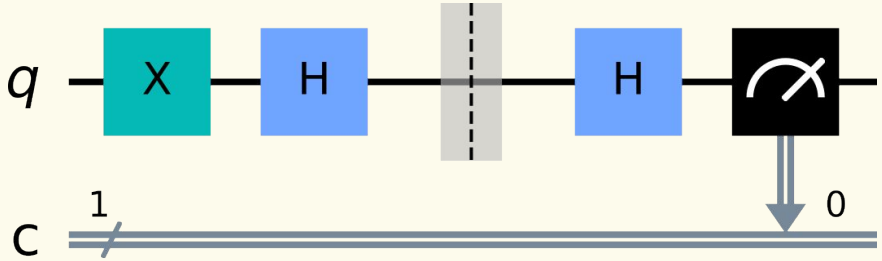
Estados ortogonais que formam a base Y

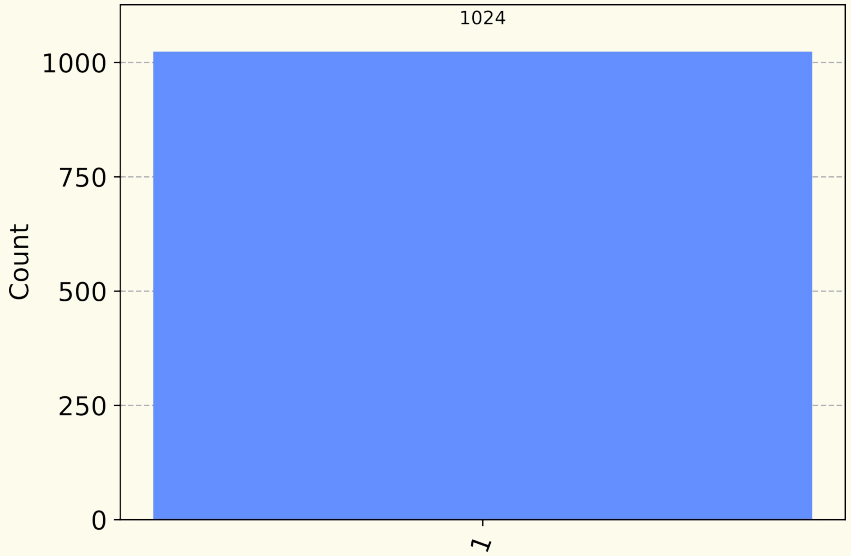
$$|R\rangle = \frac{|0\rangle + i|1\rangle}{\sqrt{2}} \qquad |L\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}}$$



Distribuição de chaves quânticas

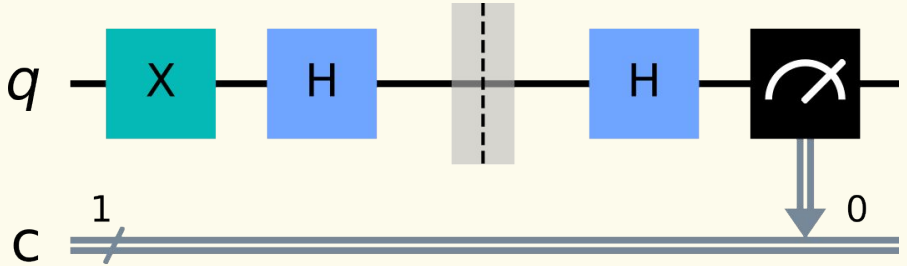




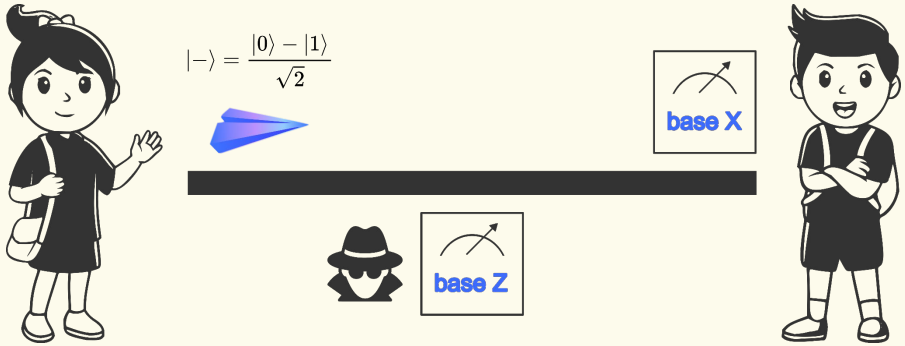


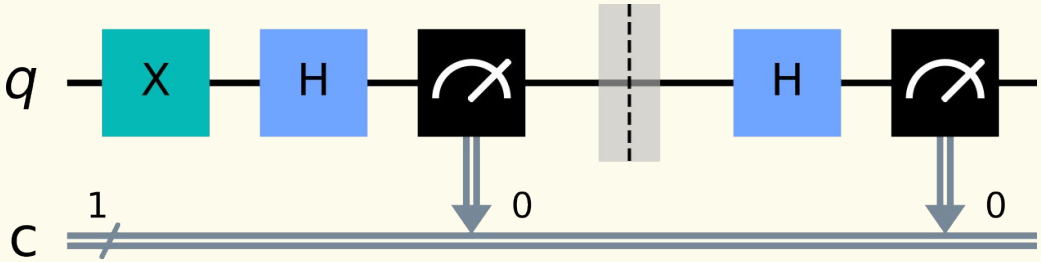
Implementação

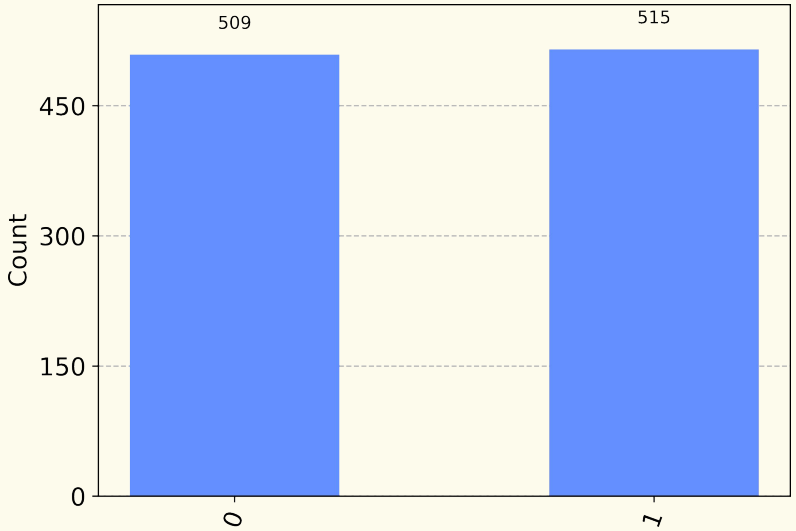
```
qc = QuantumCircuit(1,1)
# Alice prepares qubit in state |->
qc.x(0)
qc.h(0)
qc.barrier()
# Alice now sends the qubit to Bob
# who measures it on the X-basis
qc.h(0)
qc.measure(0,0)

# Draw and simulate circuit
display(qc.draw())
aer_sim = Aer.get_backend('aer_simulator')
job = aer_sim.run(qc)
plot_histogram(job.result().get_counts())
```



Distribuição de chaves quânticas



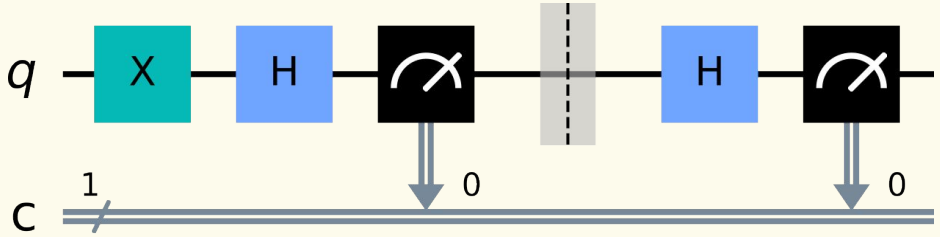




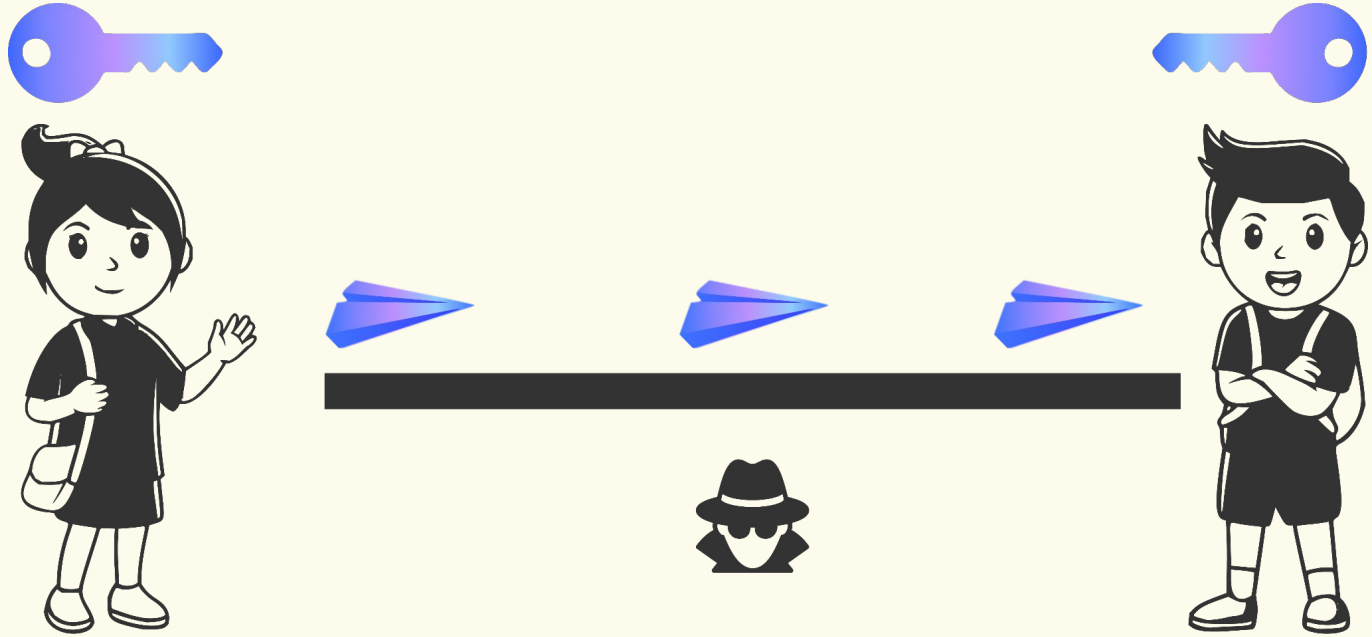
Qiskit | Fall Fest 2023

Implementação

```
qc = QuantumCircuit(1,1)
# Alice prepares qubit in state |->
qc.x(0)
qc.h(0)
# Alice now sends the qubit to Bob
# but Eve intercepts and tries to read it
qc.measure(0, 0)
qc.barrier()
# Eve then passes this on to Bob
# who measures it in the X-basis
qc.h(0)
qc.measure(0,0)

# Draw and simulate circuit
display(qc.draw())
aer_sim = Aer.get_backend('aer_simulator')
job = aer_sim.run(qc)
plot_histogram(job.result().get_counts())
```



Exemplo do Qiskit: sem interceptação

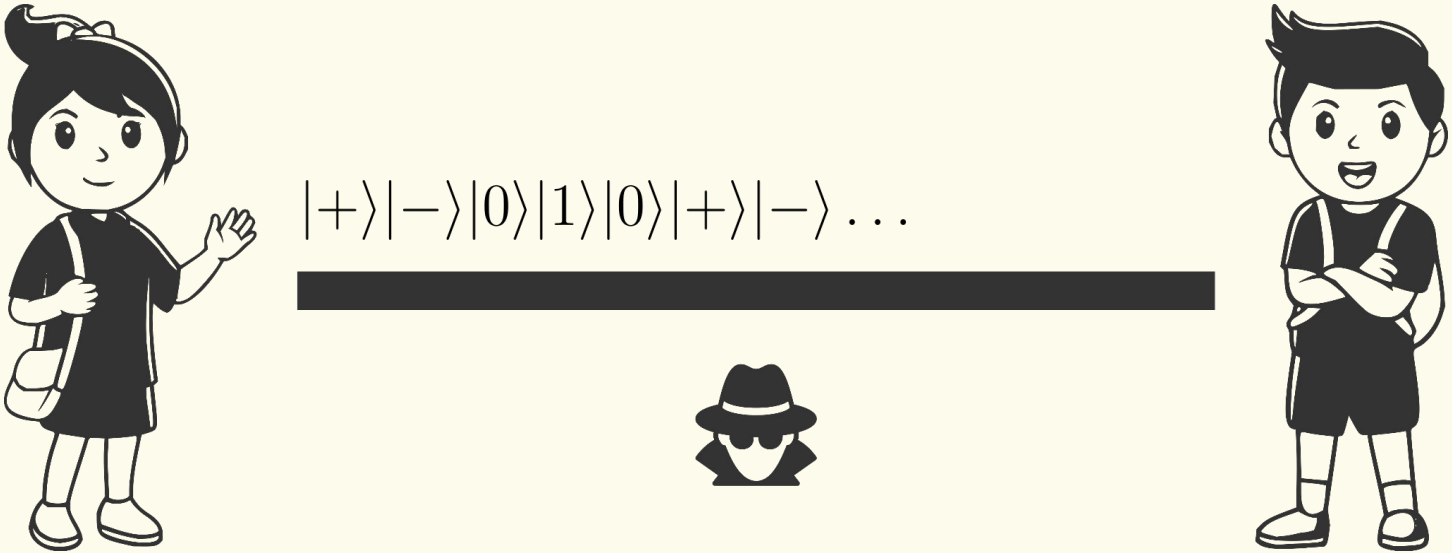


Exemplo do Qiskit: sem interceptação

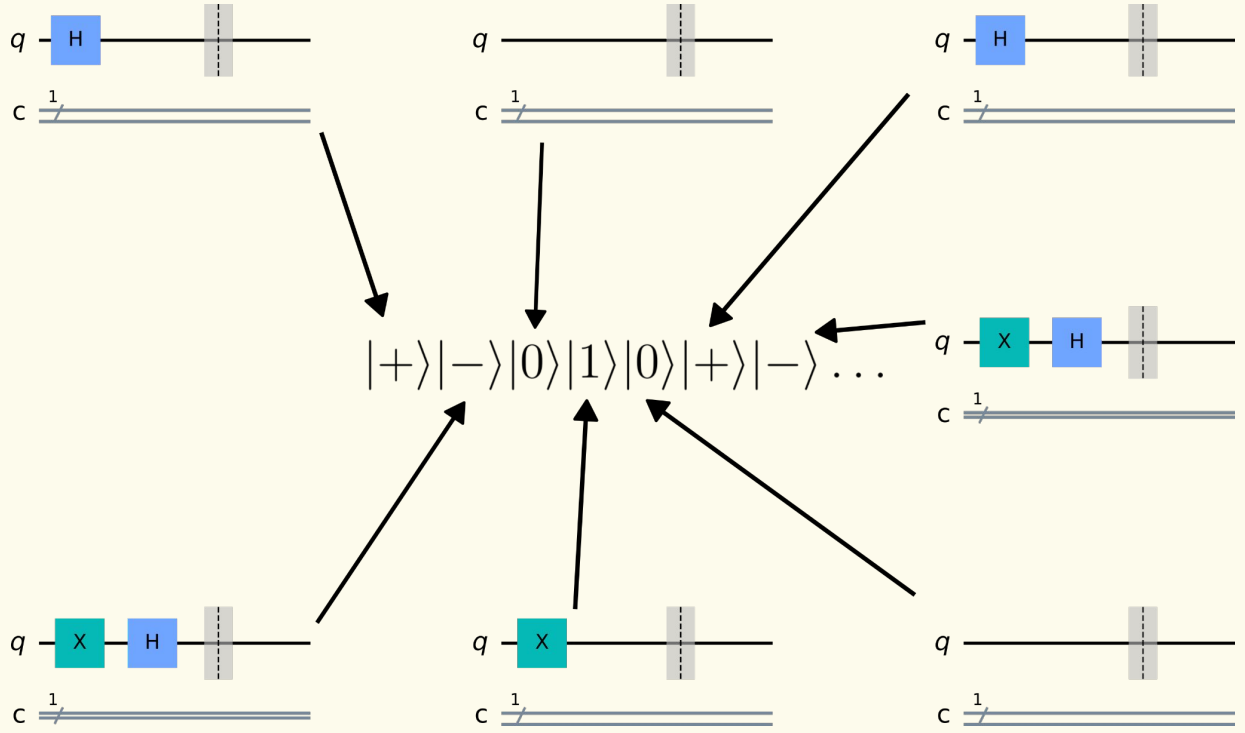
bits de alice = [0,1,0,1,0,0,1,...]
base de alice = [1,1,0,0,0,1,1,...]

Exemplo do Qiskit: sem interceptação

bits de alice = [0,1,0,1,0,0,1,...]
base de alice = [1,1,0,0,0,1,1,...]



Exemplo do Qiskit: sem interceptação



Exemplo do Qiskit: sem interceptação

Exemplo do Qiskit: sem interceptação

base de bob = [1,1,0,0,0,1,1,...]
resultados de bob = [1,0,0,1,1,0,1,...]

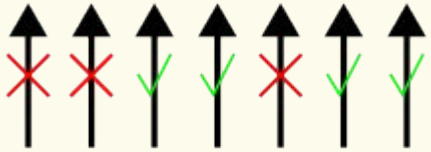
Exemplo do Qiskit: sem interceptação

bits de alice = [0,1,0,1,0,0,1,...]

resultados de bob = [1,0,0,1,1,0,1,...]

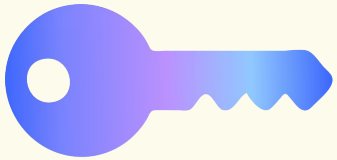
Exemplo do Qiskit: sem interceptação

[0,1,0,1,0,0,1,...]



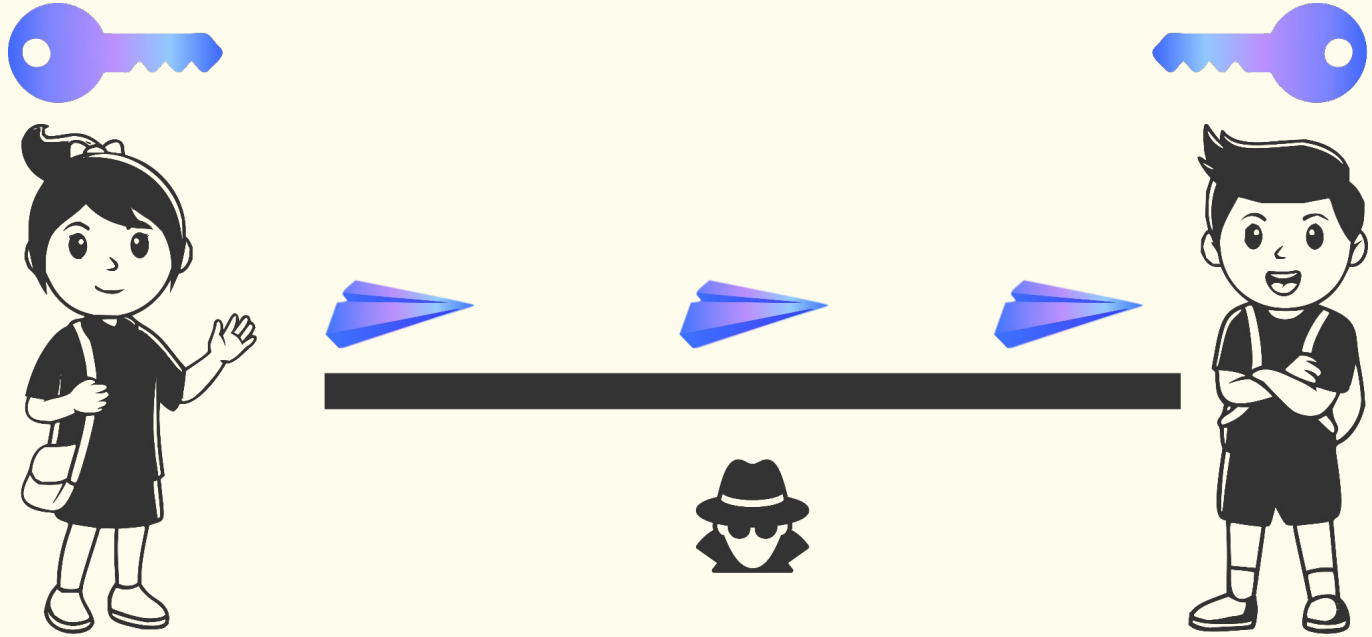
[1,0,0,1,1,0,1,...]

[0,1,0,1,...]

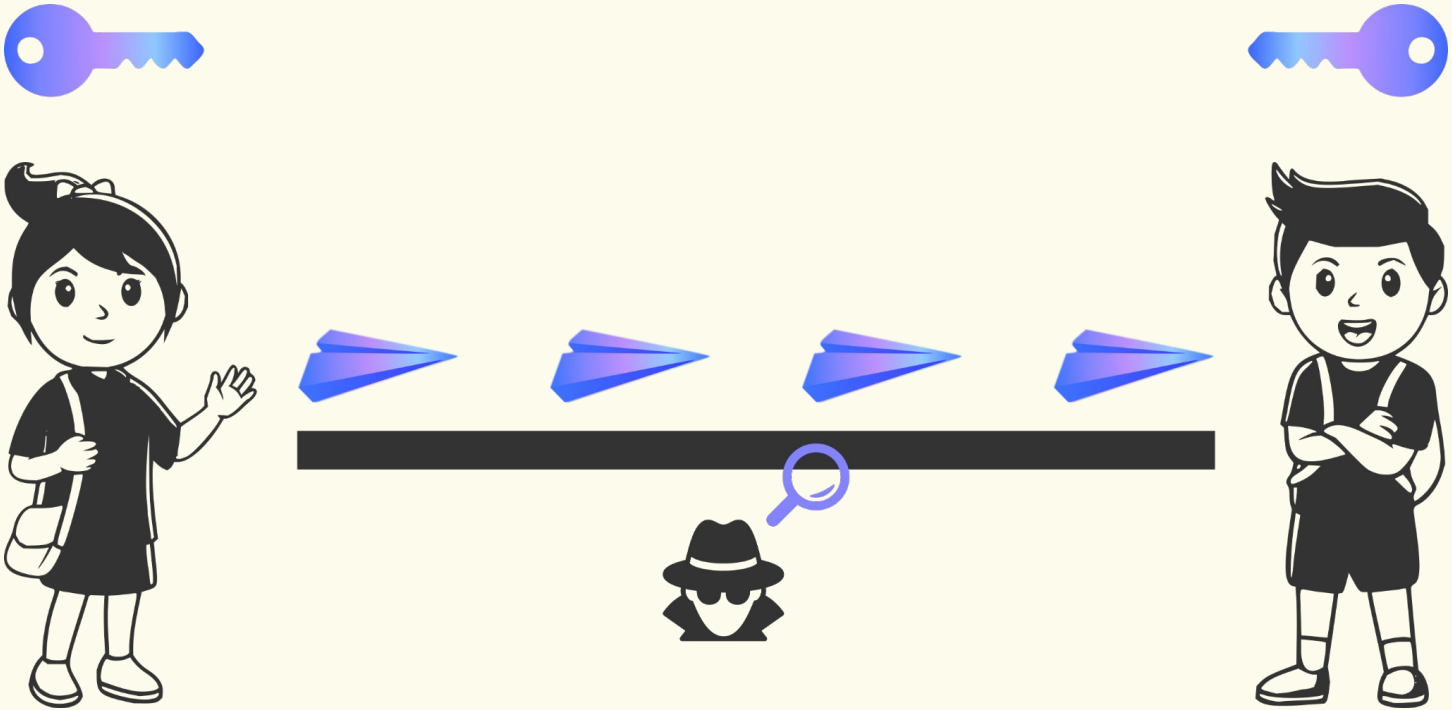


testar em uma amostra

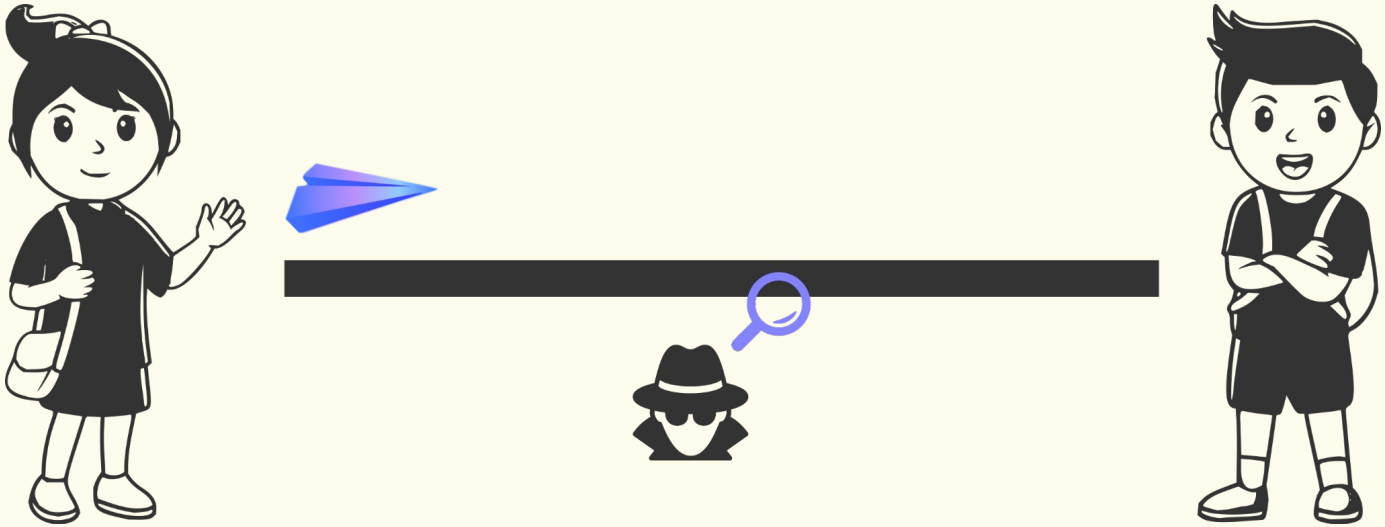
Exemplo do Qiskit: sem interceptação



Exemplo do Qiskit: com interceptação



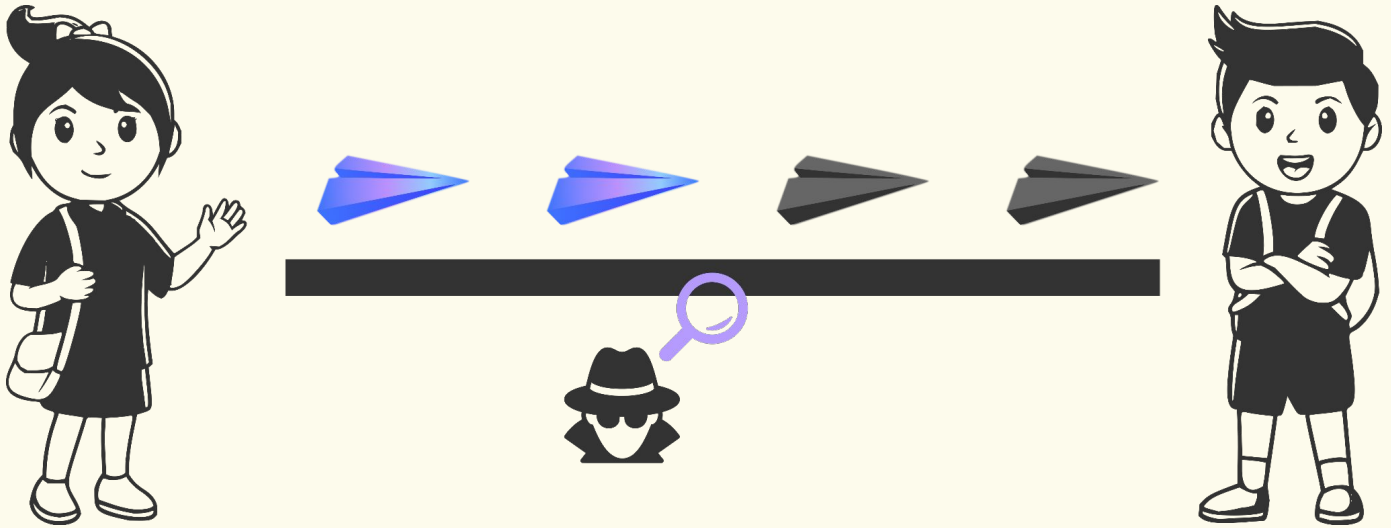
Exemplo do Qiskit: com interceptação



bits de alice = [0,1,0,1,0,0,1,...]
base de alice = [1,1,0,0,0,1,1,...]

$$|+\rangle|-\rangle|0\rangle|1\rangle|0\rangle|+\rangle|-\rangle\ldots$$

Exemplo do Qiskit: com interceptação

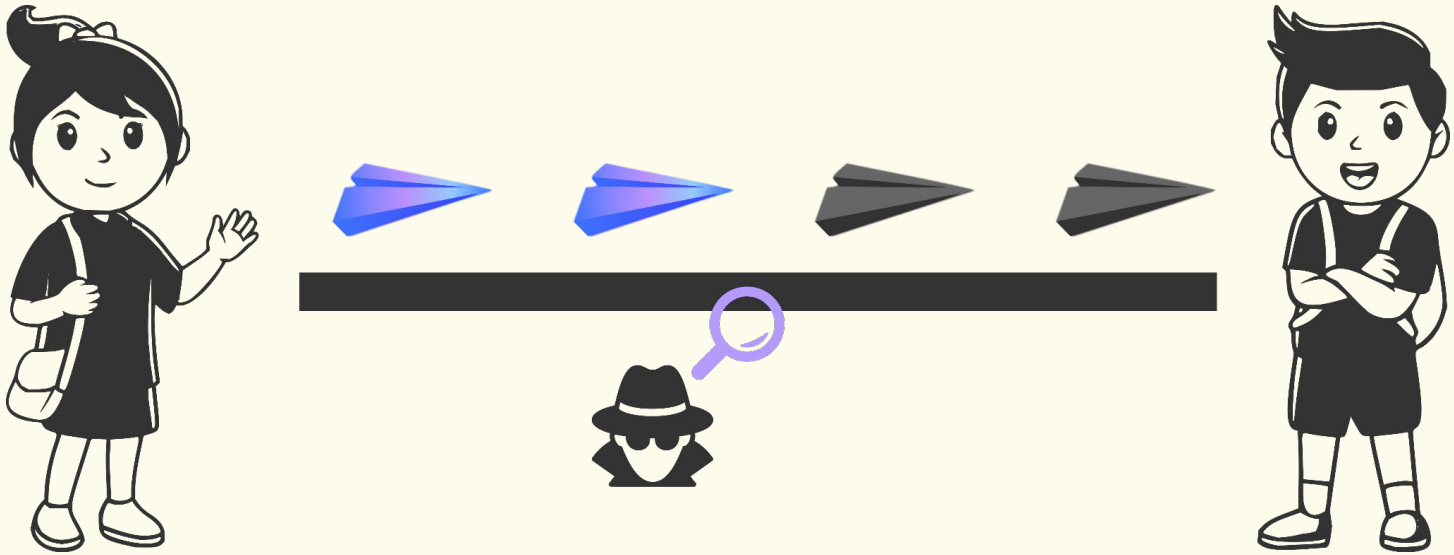


$$|+\rangle|-\rangle|0\rangle|1\rangle|0\rangle|+\rangle|-\rangle\ldots$$

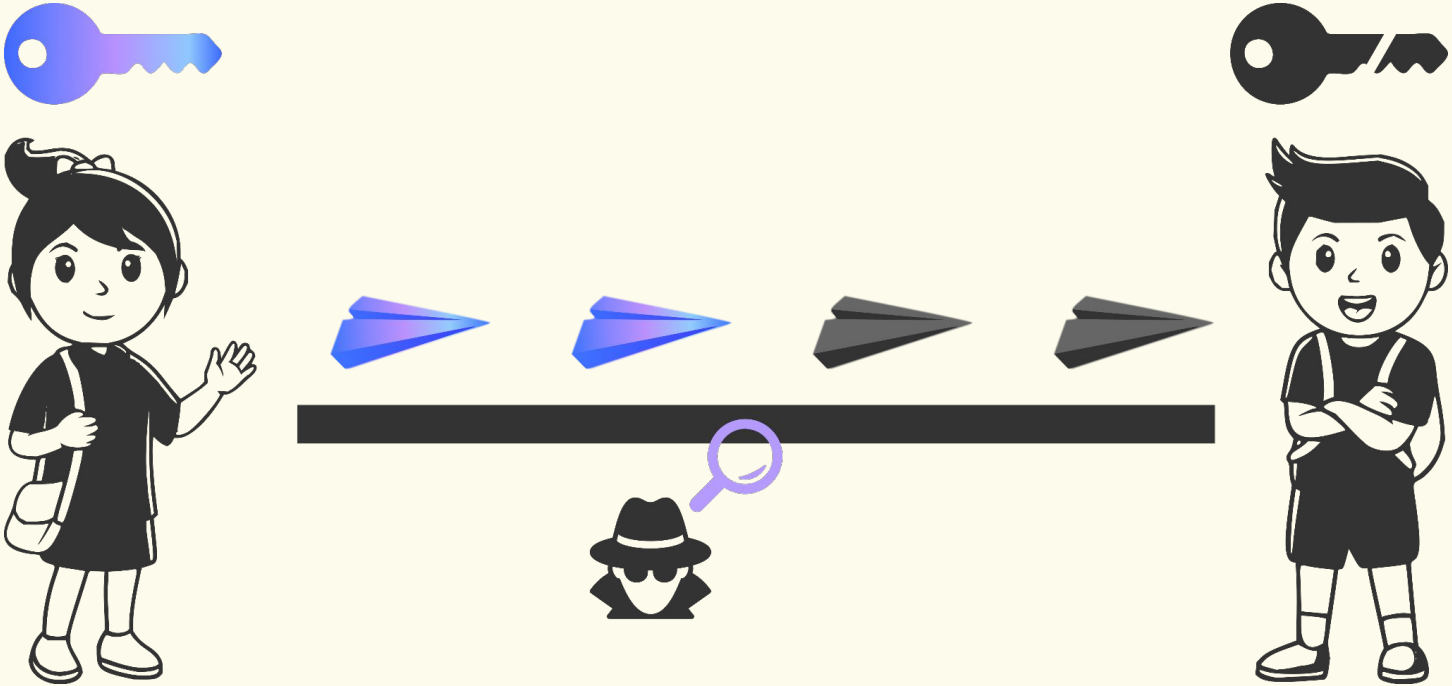
base de eva = [0,0,0,1,0,1,0,...]
resultados de eva = [1,1,0,0,1,0,0,...]

Exemplo do Qiskit: sem interceptação

base de bob = [1,1,0,0,0,1,1,...]
resultados de bob = [1,0,0,1,1,0,1,...]



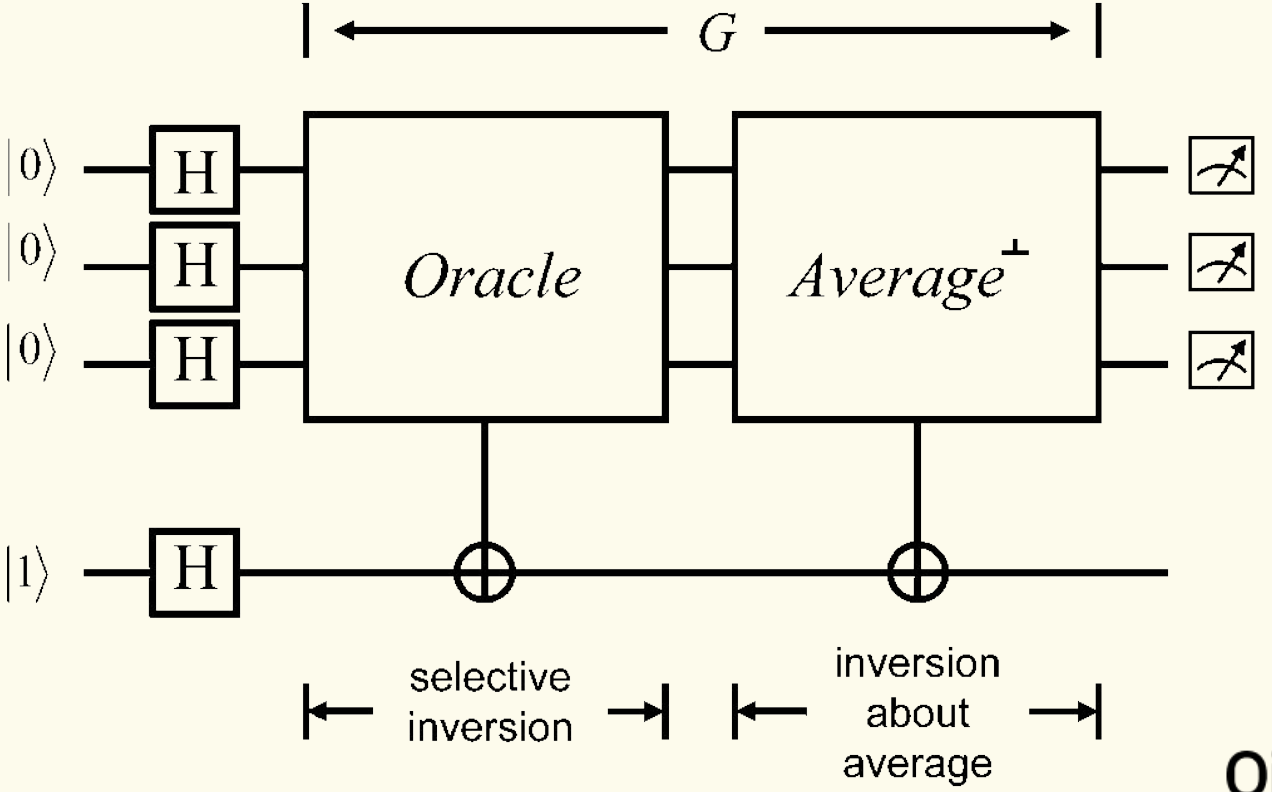
Exemplo do Qiskit: sem interceptação



Algoritmo de Grover

O algoritmo de Grover é um algoritmo quântico desenvolvido por Lov Grover em 1996, projetado para buscar uma solução eficientemente em uma lista não ordenada de dados.

Algoritmo de Grover



Algoritmo de Grover

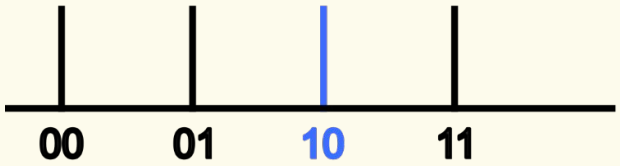
$$f : \{0,1\}^n \rightarrow \{0,1\}$$

$$f(x) = \begin{cases} 1 & \text{se } x = x_0 \\ 0 & \text{caso contrário} \end{cases}$$

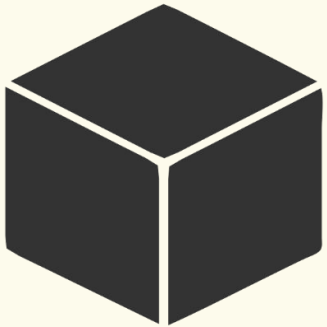
para n = 2, x = {00, 01, 10, 11}

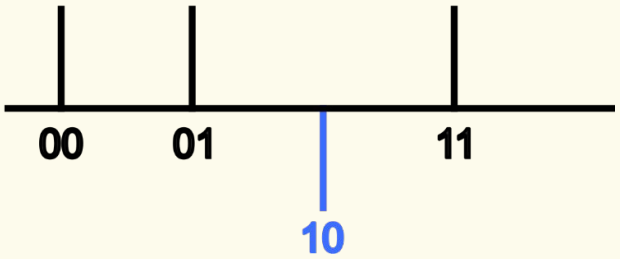
Algoritmo de Grover

$$\frac{1}{\sqrt{4}}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$



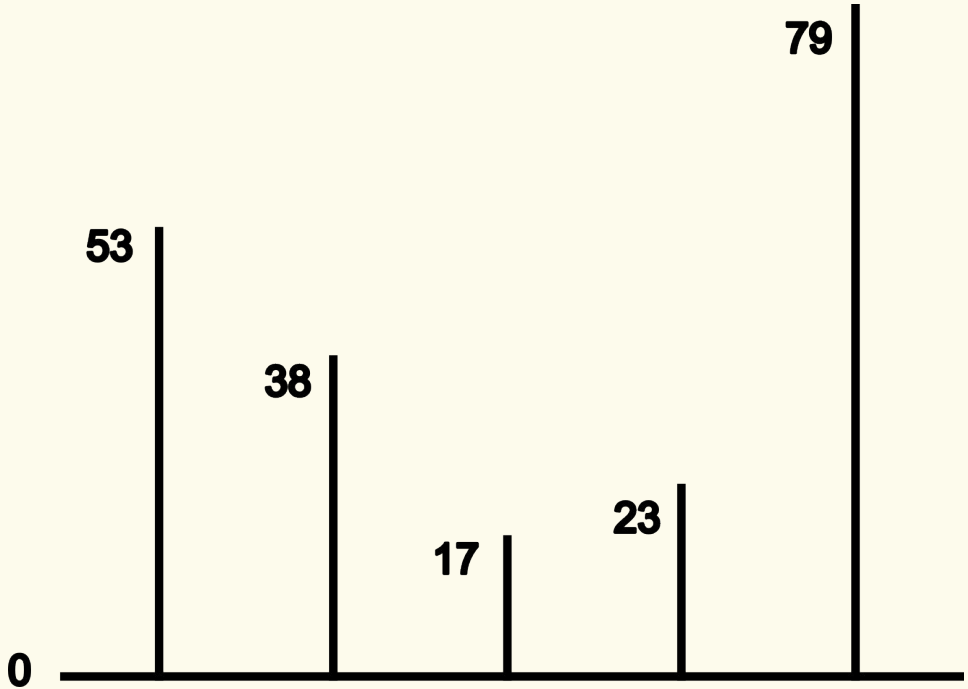
oráculo





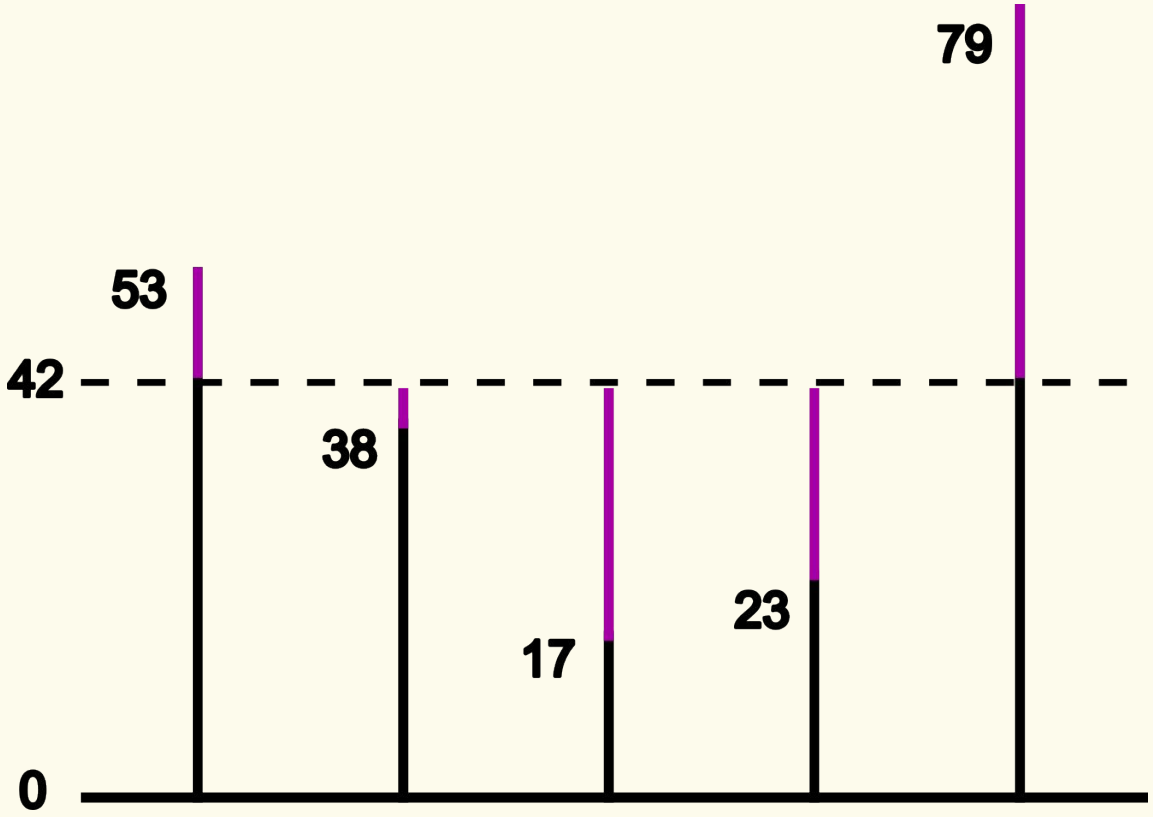
$$I_{|x_0\rangle} = (-1)^{f(x)}\,|x\rangle$$

Algoritmo de Grover



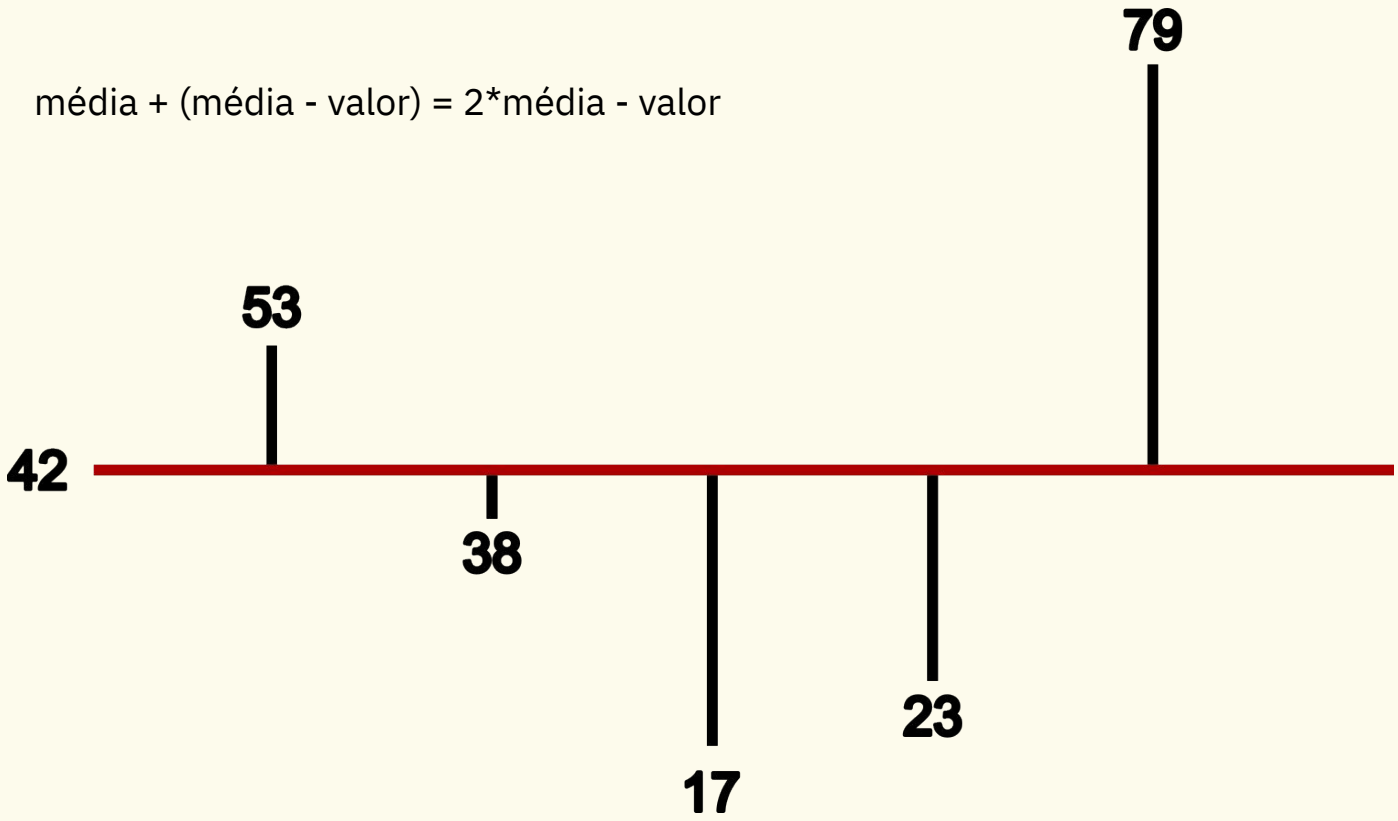
[53, 38, 17, 23, 79] = “amplitudes”

Algoritmo de Grover



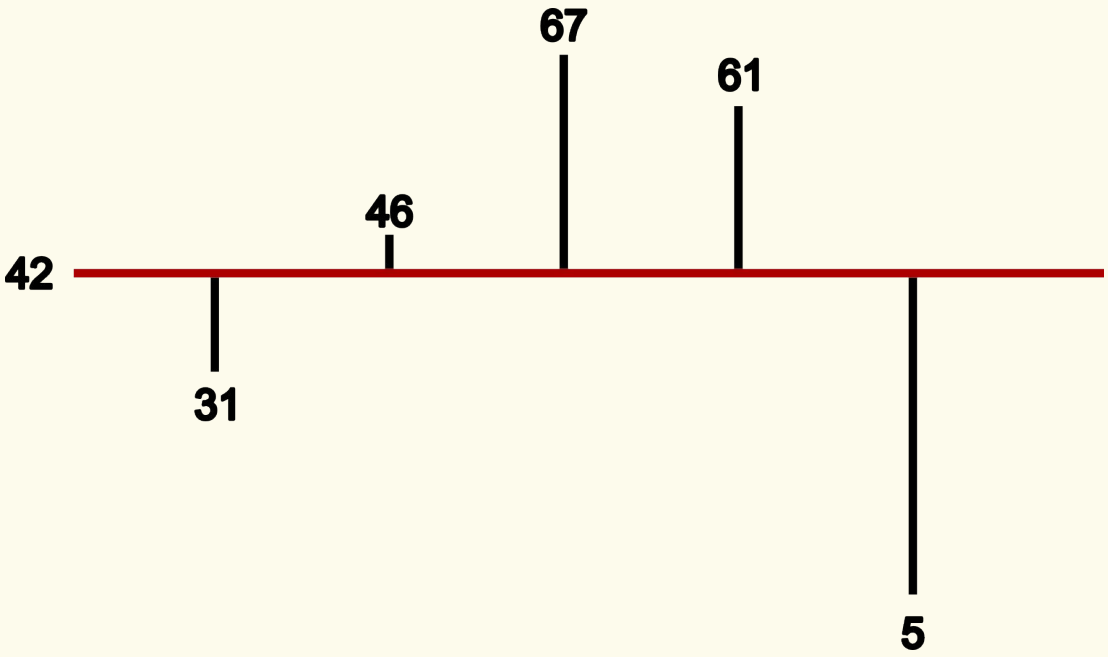
Algoritmo de Grover

média + (média - valor) = 2*média - valor



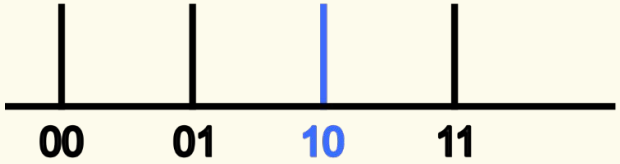
Algoritmo de Grover

42 + (42 - 53) = 31

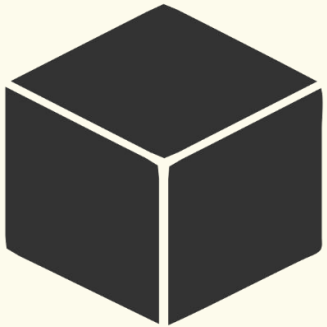


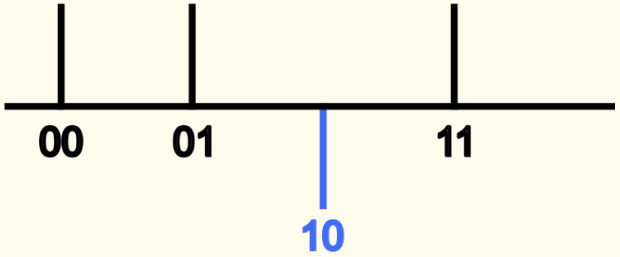
Algoritmo de Grover

$$\frac{1}{\sqrt{4}}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$

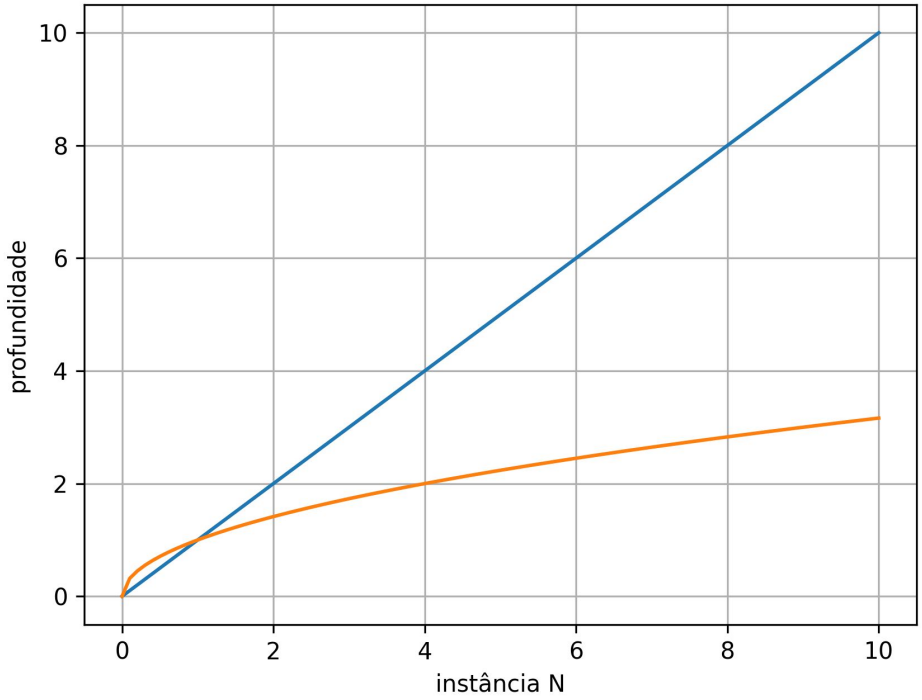


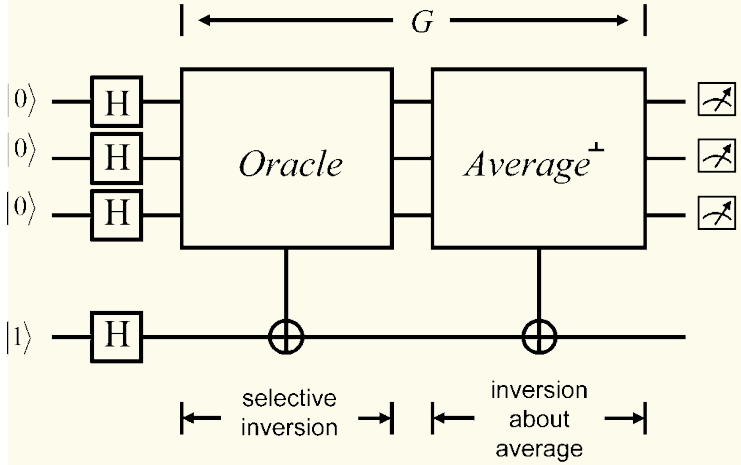
oráculo





Algoritmo de Grover





Obrigado