

exemplo_circuito_01

November 15, 2025

Autor: Osvaldo L. Santos-Pereira (olsp@if.ufrj.br / osvald23@gmail.com)

- **Webpage:** <https://ozsp12.github.io/>
- **GitHub:** <https://github.com/ozsp12>
- **ResearchGate:** <https://www.researchgate.net/profile/Osvaldo-Santos-Pereira>
- **Google Scholar:** <https://scholar.google.com/citations?user=HIZpOX8AAAAJ&hl=en>
- **ORCID:** <https://orcid.org/0000-0003-2231-517X>
- **LinkedIn:** <https://www.linkedin.com/in/ozsp12>
- **Twitter (X):** <https://x.com/ozsp12>
- **TikTok:** <https://www.tiktok.com/@ozsp12>
- **YouTube:** <https://www.youtube.com/@ozlsp12>

1 Introdução

- O qubit q_0 é usado como **controle lógico** de operações subsequentes.
- O qubit q_1 atua como **alvo** das rotações e da operação CNOT.
- As duas portas Hadamard em q_0 criam e depois reconfiguram a base de superposição.
- A porta $R_Y(\pi)$ é responsável por introduzir uma rotação contínua (analogica) na esfera de Bloch, produzindo interferência de fase.

Em conjunto, essas operações geram um circuito capaz de **preparar, manipular e medir estados de Bell**. O circuito atua primeiro sobre o qubit q_0 , aplicando a porta X , seguida da porta H , e depois um $CNOT$ no qual q_0 é o qubit de controle. Após essa etapa de controle, ainda em q_0 , aplica-se uma segunda porta H . No qubit q_1 , que funciona como alvo do $CNOT$, aplica-se ao final a rotação $R_Y(\pi)$. Em resumo, trata-se de um circuito em que q_0 controla q_1 , com operações adicionais que preparam e transformam o estado antes e depois do acoplamento entre eles.

No qubit q_0 são aplicadas as seguintes portas:

1. X
2. H
3. $\bullet-H$ (controle de um CNOT seguido de um H)

No qubit q_1 são aplicadas as seguintes portas:

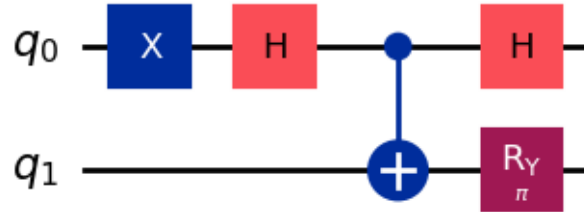
1. \oplus (alvo do CNOT)
2. $R_Y(\pi)$

Ou seja, a porta *CNOT* é aplicada de modo que o qubit q_0 desempenha o papel de controle, determinando se a operação é ou não acionada, enquanto o qubit q_1 atua como alvo, sofrendo a inversão condicionada ao estado de q_0 . Fisicamente, dizer que um qubit desempenha o papel de controle enquanto o outro é o alvo significa que a evolução de um deles determina de maneira coerente o que acontece com o outro. No caso da porta *CNOT*, o qubit de controle não sofre nenhuma alteração direta: ele apenas “decide”, pelo seu estado, se o alvo será modificado. Se o controle estiver em $|0\rangle$, nada acontece; se estiver em $|1\rangle$, o alvo sofre um flip por meio da operação X . O ponto crucial é que isso ocorre sem medição: se o controle estiver em superposição, a operação acontece de forma condicional e simultânea em cada ramo da superposição, produzindo correlações quânticas e gerando emaranhamento. Assim, o controle é o sistema que aciona a interação, e o alvo é o sistema que recebe o efeito da operação condicionada, tudo preservando a coerência e a linearidade da mecânica quântica.

```
[1]: # Imports principais
from qiskit import QuantumCircuit, QuantumRegister
from qiskit.quantum_info import Statevector
from qiskit.visualization import plot_bloch_multivector
import numpy as np
# comando que faz com que os gráficos gerados usando Matplotlib sejam exibidos
↳ diretamente dentro do próprio notebook
%matplotlib inline

# Cria um circuito com 2 qubits
qc = QuantumCircuit(2)
# Aplica a porta X no qubit 0
qc.x(0)
# Aplica uma porta Hadamard (H) no qubit 0
qc.h(0)
# Aplica uma porta CNOT com controle em q0 e alvo em q1
qc.cx(0, 1)
# Aplica uma rotação R_y(π) no qubit 1
qc.ry(np.pi, 1)
# Aplica novamente uma porta Hadamard no qubit 0
qc.h(0)
# Exibe o circuito
qc.draw("mpl")
```

[1]:



2 Explicação passo a passo

2.1 Criando a registradora

No trecho a seguir, um circuito quântico é inicializado com dois qubits e sua representação gráfica é exibida no notebook por meio do comando `qc.draw("mpl")`.

```
[2]: # Cria um circuito com 2 qubits
qc = QuantumCircuit(2)
# Exibe o circuito
qc.draw("mpl")
```

[2]:

q_0 —

q_1 —

No trecho a seguir, um circuito quântico é inicializado com dois qubits, sua representação gráfica é exibida no notebook e o vetor de estado associado às operações presentes no circuito é extraído. Esse vetor de estado é então apresentado em notação LaTeX por meio de `psi.draw('latex')`. O vetor de estado inicial é o

$$|\psi\rangle = |00\rangle$$

```
[3]: # Extrai o vetor de estado resultante do circuito atual
psi = Statevector.from_instruction(qc)
#
psi.draw('latex')
```

[3]:

$|00\rangle$

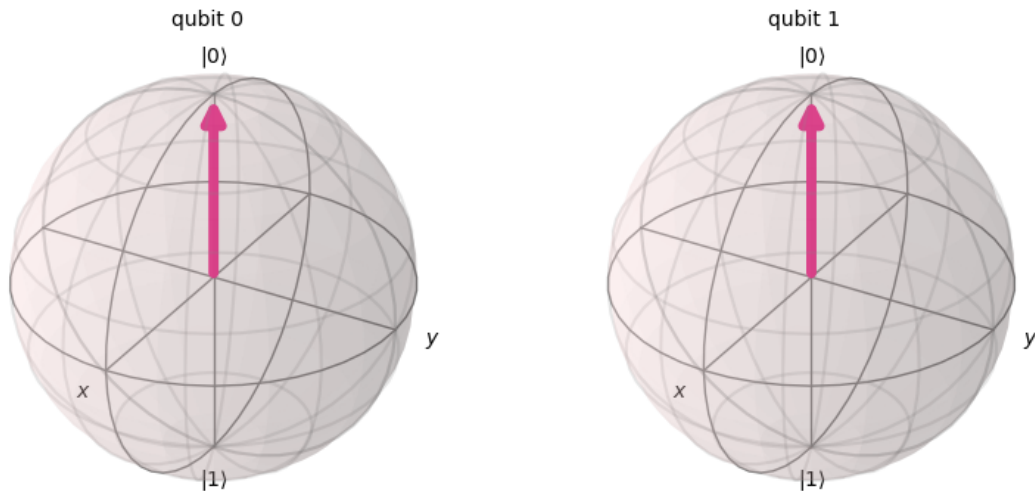
No trecho a seguir, o vetor de estado previamente obtido é representado em sua forma textual em LaTeX. Essa representação é retornada por meio do comando `psi.draw('latex_source')`.

```
[4]: #  
psi.draw('latex_source')
```

```
[4]: ' |00\\rangle'
```

No trecho a seguir, a esfera de Bloch correspondente ao vetor de estado extraído do circuito é gerada e exibida no notebook. A visualização é produzida pelo comando `plot_bloch_multivector(psi)` e apresentada por meio de `display()`.

```
[5]: # plta esfera de Bloch do estado do circuito  
display(plot_bloch_multivector(psi))
```



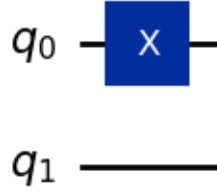
2.2 Aplicando a porta X no qubit 0

No trecho a seguir, um circuito quântico é inicializado com dois qubits, a porta X é aplicada ao qubit 0 e a representação gráfica resultante é exibida no notebook por meio de `qc.draw("mpl")`. Na notação adotada pelo Qiskit, o estado computacional é escrito no formato $|q_1q_0\rangle$, de modo que o qubit 0 ocupa a posição menos significativa e aparece na última posição do ket.

```
[6]: # Cria um circuito com 2 qubits  
qc = QuantumCircuit(2)  
# Aplica a porta X no qubit 0  
qc.x(0)  
# Exibe o circuito
```

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

[6] :



2.3 Explicação teórica da aplicação de X no qubit 0

No circuito considerado, trabalha-se no espaço de Hilbert $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2$, com base computacional ordenada como

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\},$$

A base computacional de dois qubits é representada matricialmente por vetores coluna de dimensão 4. Na convenção usual (ordenando os estados como $|00\rangle, |01\rangle, |10\rangle, |11\rangle$), tem-se

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

e, na convenção do Qiskit, o ket é escrito como $|q_1q_0\rangle$, em que q_0 ocupa a posição menos significativa (última posição). A porta de Pauli- X é representada pela matriz

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

definida pelos autovetores da base computacional por

$$X|0\rangle = |1\rangle,$$

$$X|1\rangle = |0\rangle.$$

No circuito, a operação X é aplicada apenas ao qubit 0, logo o operador global U (controlador padrão de portas) atuando em \mathcal{H} é dado por

$$U = I \otimes X,$$

onde I é a matriz identidade em \mathbb{C}^2 associada ao qubit 1, e X é a porta aplicada no qubit 0. A forma matricial do operador $U = I \otimes X$ é obtida a partir das matrizes

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

O produto de Kronecker resulta em

$$U = I \otimes X = \begin{pmatrix} 1 \cdot X & 0 \cdot X \\ 0 \cdot X & 1 \cdot X \end{pmatrix} = \begin{pmatrix} X & 0 \\ 0 & X \end{pmatrix}.$$

Substituindo explicitamente os blocos das submatrizes X , obtém-se

$$U = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

O estado inicial do registrador é dado por

$$|\psi_0\rangle = |00\rangle = |0\rangle_1 \otimes |0\rangle_0.$$

A evolução unitária induzida por U é escrita como

$$\begin{aligned} |\psi_1\rangle &= U|\psi_0\rangle \\ &= (I \otimes X)(|0\rangle_1 \otimes |0\rangle_0) \\ &= |0\rangle_1 \otimes X|0\rangle_0 \\ &= |0\rangle_1 \otimes |1\rangle_0 \\ &= |01\rangle. \end{aligned}$$

Em representação de matrizes

$$|\psi_1\rangle = U|\psi_0\rangle = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = |01\rangle.$$

O comando `Statevector.from_instruction(qc)` constrói o vetor de estado $|\psi_1\rangle$ associado ao circuito, e a instrução `psi.draw('latex')` exibe exatamente o resultado $|01\rangle$ na notação de kets adotada pelo Qiskit.

```
[7]: # Extrai o vetor de estado resultante do circuito atual
psi = Statevector.from_instruction(qc)
```

```
#
psi.draw('latex')
```

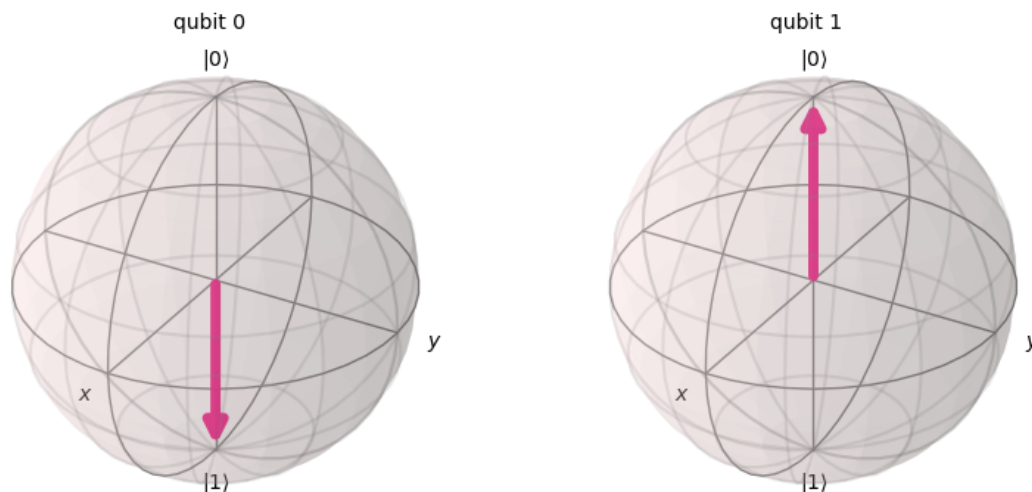
[7]:

$|01\rangle$

2.4 A esfera de Bloch

A figura apresentada mostra a representação na esfera de Bloch dos dois qubits que compõem o estado $|\psi_1\rangle = |01\rangle$. Na convenção adotada, o qubit 0 corresponde ao último dígito do ket e o qubit 1 ao primeiro. A esfera à esquerda representa o qubit 0, cujo estado é $|1\rangle$. Esse estado está associado ao polo sul da esfera de Bloch, o que é indicado pela seta apontada para baixo. A esfera à direita corresponde ao qubit 1, cujo estado é $|0\rangle$, localizado no polo norte da esfera de Bloch e representado pela seta apontando para cima. Cada esfera exibe o estado puro correspondente como um vetor unitário alinhado com o eixo z , refletindo que não há superposição entre $|0\rangle$ e $|1\rangle$ em nenhum dos qubits.

```
[8]: # plta esfrea de Bloch do estado do circuito
display(plot_bloch_multivector(psi))
```



2.5 Acrescentando Hadamard ao circuito

O circuito exibido aplica duas operações sucessivas ao qubit 0. A primeira operação é a porta X , que transforma $|0\rangle$ em $|1\rangle$ e vice-versa. Em seguida, aplica-se a porta de Hadamard H , que leva cada autoestado computacional para uma superposição:

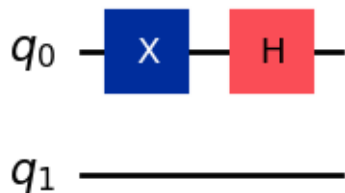
$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Como essas operações são aplicadas somente ao qubit 0, a linha correspondente ao qubit 1 permanece inalterada. O diagrama produzido reflete exatamente essa sequência: um bloco X seguido de um

bloco H na linha de q_0 , enquanto a linha de q_1 permanece vazia, indicando ausência de operações.

```
[9]: # Cria um circuito com 2 qubits
qc = QuantumCircuit(2)
# Aplica a porta X no qubit 0
qc.x(0)
# Aplica uma porta Hadamard (H) no qubit 0
qc.h(0)
# Exibe o circuito
qc.draw("mpl")
```

[9]:



2.6 Explicação teórica da atuação de Hadamard

O vetor de estado obtido corresponde ao resultado da aplicação sequencial das portas X e H no qubit 0, com o qubit 1 permanecendo no estado $|0\rangle$. O estado inicial do registrador é

$$|\psi_0\rangle = |00\rangle.$$

A porta X atua apenas no qubit menos significativo, produzindo

$$(I \otimes X)|00\rangle = |01\rangle.$$

Em seguida, a porta de Hadamard atua no mesmo qubit. Usando

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

obtém-se

$$\begin{aligned} |\psi_2\rangle &= (I \otimes H)|01\rangle \\ &= |0\rangle_1 \otimes H|1\rangle_0 \\ &= |0\rangle_1 \otimes \frac{|0\rangle_0 + |1\rangle_0}{\sqrt{2}} \\ &= \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle). \end{aligned}$$

Esse é exatamente o estado exibido pelo comando `psi.draw('latex')`, representado como

$$|\psi_2\rangle = \frac{\sqrt{2}}{2}|00\rangle - \frac{\sqrt{2}}{2}|01\rangle.$$

```
[10]: # Extrai o vetor de estado resultante do circuito atual
psi = Statevector.from_instruction(qc)
#
psi.draw('latex')
```

[10]:

$$\frac{\sqrt{2}}{2}|00\rangle - \frac{\sqrt{2}}{2}|01\rangle$$

2.7 Estado em superposição na esfera de Bloch

A imagem mostra a representação de Bloch de cada qubit no estado

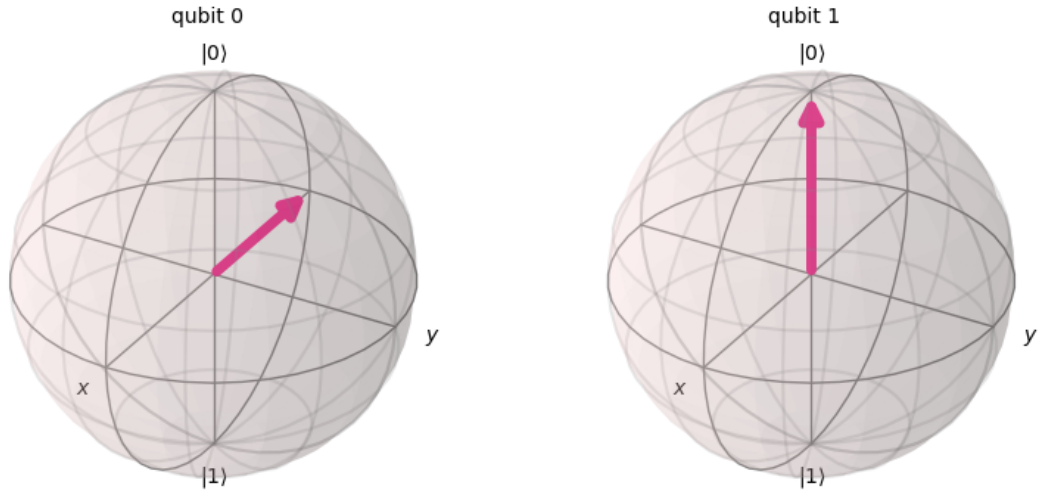
$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle).$$

Nesse estado, o qubit 1 permanece em $|0\rangle$, pois o primeiro dígito do ket não é afetado pelas operações realizadas. Na esfera à direita, isso aparece como um vetor alinhado com o polo norte, indicando ausência de superposição desse qubit. Já o qubit 0 encontra-se no estado

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

que corresponde a um estado puro com componentes reais e fase relativa negativa entre $|0\rangle$ e $|1\rangle$. A esfera à esquerda mostra esse estado como um vetor inclinado no semiplano xz , refletindo que o qubit está em superposição não trivial entre os autoestados computacionais. Assim, cada esfera exibe a posição geométrica do estado individual de cada qubit, obtida após decompor o estado global em seus estados reduzidos.

```
[11]: # plta esfera de Bloch do estado do circuito
display(plot_bloch_multivector(psi))
```



2.8 Acrescentando a porta CNOT ao circuito

O circuito apresentado aplica três operações sucessivas. Primeiro, a porta X é aplicada ao qubit 0, invertendo seu estado. Em seguida, a porta de Hadamard H atua no mesmo qubit, produzindo uma superposição linear dos estados computacionais. Após essas operações locais, é aplicada a porta $CNOT$, na qual o qubit 0 funciona como controle e o qubit 1 funciona como alvo. O que implica que o segundo qubit é invertido se, e somente se, o primeiro qubit estiver no estado $|1\rangle$. Em termos descritivos, a porta $CNOT$ opera em um registrador quântico de dois qubits e realiza um flip no qubit alvo exatamente quando o qubit de controle ocupa o estado $|1\rangle$.

Antes		Depois	
Controle	Alvo	Controle	Alvo
$ 0\rangle$	$ 0\rangle$	$ 0\rangle$	$ 0\rangle$
$ 0\rangle$	$ 1\rangle$	$ 0\rangle$	$ 1\rangle$
$ 1\rangle$	$ 0\rangle$	$ 1\rangle$	$ 1\rangle$
$ 1\rangle$	$ 1\rangle$	$ 1\rangle$	$ 0\rangle$

Se $\{|0\rangle, |1\rangle\}$ são os únicos valores de entrada permitidos para ambos os qubits, então a saída do **qubit alvo** da porta $CNOT$ corresponde ao resultado de uma porta XOR clássica. Mantendo o **qubit de controle** fixado em $|1\rangle$, a saída do qubit alvo da porta $CNOT$ produz o resultado de uma porta NOT clássica. De forma mais geral, as entradas podem estar em uma superposição linear dos estados $|0\rangle, |1\rangle$. A porta $CNOT$ transforma o estado quântico

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$

em

$$a|00\rangle + b|01\rangle + c|11\rangle + d|10\rangle.$$

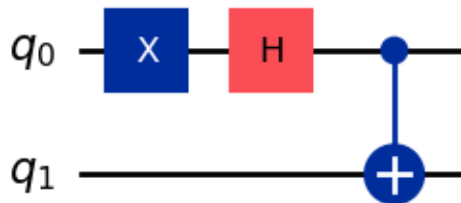
A ação da porta CNOT pode ser representada pela matriz (na forma de uma matriz de permutação):

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

O diagrama exibido no notebook reflete essa estrutura: um bloco X seguido de um bloco H na linha de q_0 , e a operação controlada conectando q_0 ao qubit 1 por meio do símbolo de controle na linha superior e do símbolo de soma na linha inferior.

```
[12]: # Cria um circuito com 2 qubits
qc = QuantumCircuit(2)
# Aplica a porta X no qubit 0
qc.x(0)
# Aplica uma porta Hadamard (H) no qubit 0
qc.h(0)
# Aplica uma porta CNOT com controle em q0 e alvo em q1
qc.cx(0, 1)
# Exibe o circuito
qc.draw("mpl")
```

[12]:



2.9 Estado entrelaçado

Esse resultado corresponde ao estado produzido pela sequência de operações aplicadas no circuito: a porta X , seguida da porta de Hadamard H no qubit 0, e por fim a porta CNOT com controle em q_0 e alvo em q_1 . O estado inicial do registrador é

$$|\psi_0\rangle = |00\rangle.$$

Após a aplicação de X no qubit menos significativo, obtém-se

$$(I \otimes X)|00\rangle = |01\rangle.$$

Em seguida, a porta Hadamard atua nesse mesmo qubit:

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

o que produz

$$(I \otimes H)|01\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle).$$

Por fim, a porta CNOT atua no circuito, e os termos individuais são transformados como

$$|00\rangle \mapsto |00\rangle, \quad |01\rangle \mapsto |11\rangle,$$

pois apenas no segundo termo o qubit de controle está em $|1\rangle$. Assim, o estado final é

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle,$$

que é exatamente o resultado exibido pelo comando `psi.draw('latex')`.

```
[13]: # Extrai o vetor de estado resultante do circuito atual
psi = Statevector.from_instruction(qc)
#
psi.draw('latex')
```

[13]:

$$\frac{\sqrt{2}}{2}|00\rangle - \frac{\sqrt{2}}{2}|11\rangle$$

2.10 Estado entrelaçado não pode ser representado na esfera de Bloch

Esse estado final,

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle,$$

corresponde a um estado emaranhado de dois qubits. Cada termo da superposição envolve ambos os qubits simultaneamente, e por isso o estado não pode ser escrito como um produto de estados individuais, isto é,

$$|\psi\rangle \neq |\phi_1\rangle \otimes |\phi_0\rangle.$$

A representação na esfera de Bloch exige exatamente essa fatorabilidade: a esfera de Bloch descreve apenas **estados puros de um único qubit**, que pertencem ao espaço de Hilbert de dimensão 2.

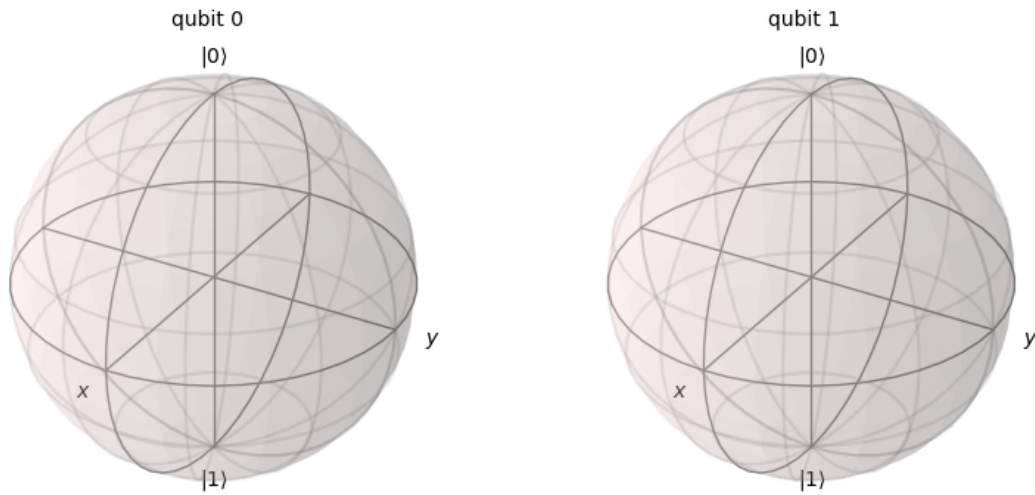
No caso presente, o estado pertence ao espaço \mathbb{C}^4 e não pode ser reduzido a estados puros de cada qubit separadamente. De fato, ao tentar obter os estados individuais, deve-se calcular as matrizes densidade reduzidas por meio da operação traço parcial. Para um estado emaranhado como este, cada qubit isolado encontra-se em um estado **misto**, descrito por uma matriz densidade que não é um projetor de posto 1. Por exemplo, para este estado específico tem-se que

$$\rho_0 = \text{Tr}_{q_1}(|\psi\rangle\langle\psi|) = \frac{1}{2}I,$$

$$\rho_1 = \text{Tr}_{q_0}(|\psi\rangle\langle\psi|) = \frac{1}{2}I,$$

sendo I a matriz identidade 2×2 . Esses estados são puramente mistos e correspondem ao **centro** da esfera de Bloch, não a um vetor localizado na superfície. Portanto, não é possível representar o estado global $|\psi\rangle$ em uma esfera de Bloch, pois a esfera descreve apenas estados puros de um único qubit. A esfera só pode ser usada para representar os estados reduzidos de cada qubit, que neste caso são mistos e aparecem como o ponto central da esfera, sem direção definida.

```
[14]: # plta esfrea de Bloch do estado do circuito
display(plot_bloch_multivector(psi))
```



2.11 Acrescentando rotação em y ao circuito

O circuito mostrado aplica uma sequência de quatro operações. Primeiro, o qubit 0 recebe a porta X , que inverte seu estado. Em seguida, aplica-se a porta de Hadamard H no mesmo qubit, colocando-o em superposição. Depois dessas operações locais, a porta CNOT é aplicada, com q_0 atuando como qubit de controle e q_1 como qubit alvo; nessa operação, o alvo é invertido somente quando o controle está no estado $|1\rangle$. Por fim, o qubit 1 recebe a rotação $R_y(\pi)$, que corresponde a uma rotação de ângulo π em torno do eixo y . A operação $R_y(\theta)$ é definida a partir das matrizes de Pauli. Em particular, a matriz de Pauli associada ao eixo y é

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

uma das três matrizes de Pauli $\sigma_x, \sigma_y, \sigma_z$, que formam uma base para o espaço dos operadores hermitianos 2×2 e representam rotações fundamentais em \mathbb{C}^2 . Essas matrizes são geradoras das rotações em $SU(2)$, e o operador de rotação em torno do eixo y é definido como

$$R_y(\theta) = e^{-i\theta\sigma_y/2}.$$

A exponenciação de operadores leva à forma fechada

$$R_y(\theta) = \begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix},$$

que, para $\theta = \pi$, resulta em

$$R_y(\pi) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Essa operação atua nos estados computacionais como

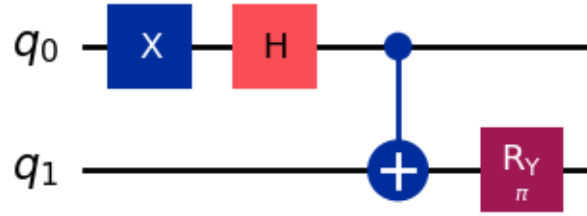
$$R_y(\pi)|0\rangle = |1\rangle,$$

$$R_y(\pi)|1\rangle = -|0\rangle,$$

mostrando que a rotação de π em torno de y produz essencialmente uma inversão dos estados, com uma fase relativa no segundo caso. O diagrama do circuito representa exatamente essa sequência: as portas X e H em q_0 , a porta CNOT conectando os dois qubits e, ao final, a rotação $R_y(\pi)$ aplicada ao qubit 1.

```
[15]: # Cria um circuito com 2 qubits
qc = QuantumCircuit(2)
# Aplica a porta X no qubit 0
qc.x(0)
# Aplica uma porta Hadamard (H) no qubit 0
qc.h(0)
# Aplica uma porta CNOT com controle em q0 e alvo em q1
qc.cx(0, 1)
# Aplica uma rotação R_y(pi) no qubit 1
qc.ry(np.pi, 1)
# Exibe o circuito
qc.draw("mpl")
```

[15]:



2.12 Explicação teórica sobre a atuação da rotação em y no circuito

Esse vetor de estado corresponde ao resultado da aplicação sequencial das quatro operações do circuito: as portas X e H no qubit 0, o CNOT com controle em q_0 e alvo em q_1 , e finalmente a rotação $R_y(\pi)$ no qubit 1. O estado inicial é

$$|\psi_0\rangle = |00\rangle.$$

Após a porta X em q_0 , obtém-se

$$|\psi_1\rangle = |01\rangle.$$

A porta Hadamard no qubit 0 produz

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|01\rangle.$$

A seguir, a porta CNOT resulta em

$$|\psi_3\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle.$$

Por fim, aplica-se a rotação $R_y(\pi)$ no qubit 1 resulta em

$$\begin{aligned} |\psi_4\rangle &= R_y(\pi)|\psi_3\rangle \\ &= R_y(\pi) \left[\frac{1}{\sqrt{2}}|0\rangle_1 \otimes |0\rangle_0 - \frac{1}{\sqrt{2}}|1\rangle_1 \otimes |1\rangle_0 \right] \\ &= \frac{1}{\sqrt{2}}|1\rangle_1 \otimes |0\rangle_0 + \frac{1}{\sqrt{2}}|0\rangle_1 \otimes |1\rangle_0 \\ &= \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle. \end{aligned}$$

o que é matematicamente equivalente à forma simplificada acima. Esse é exatamente o estado exibido pelo comando `psi.draw('latex')`.

```
[16]: # Extrai o vetor de estado resultante do circuito atual
psi = Statevector.from_instruction(qc)
#
psi.draw('latex')
```

[16]:

$$\frac{\sqrt{2}}{2}|01\rangle + \frac{\sqrt{2}}{2}|10\rangle$$

2.13 Acrescentando mais uma porta Hadamard

O circuito aplica cinco operações sucessivas sobre dois qubits. Primeiro, o qubit q_0 recebe a porta X , que inverte seu estado. Em seguida, aplica-se uma porta de Hadamard H no mesmo qubit, colocando-o em superposição. Depois, o circuito aplica uma porta CNOT, na qual q_0 atua como controle e q_1 atua como alvo; o alvo é invertido somente quando o controle está no estado $|1\rangle$. A operação seguinte é uma rotação $R_y(\pi)$ aplicada ao qubit q_1 , implementando uma rotação de ângulo π em torno do eixo y , definida pelo operador

$$R_y(\pi) = \exp\left(-\frac{i\pi}{2}\sigma_y\right) = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},$$

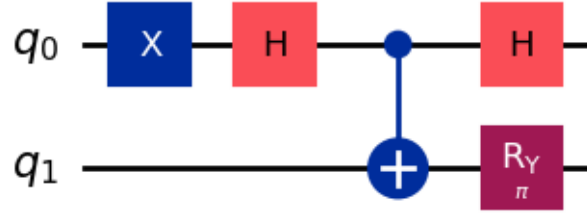
sendo σ_y a matriz de Pauli correspondente ao eixo y ,

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}.$$

Por fim, uma segunda porta Hadamard é aplicada novamente ao qubit q_0 . O diagrama gerado pelo Qiskit reflete exatamente essa sequência, com a linha superior mostrando as operações locais e o ponto de controle do CNOT, e a linha inferior mostrando o alvo do CNOT e a rotação $R_y(\pi)$.

```
[17]: # Cria um circuito com 2 qubits
qc = QuantumCircuit(2)
# Aplica a porta X no qubit 0
qc.x(0)
# Aplica uma porta Hadamard (H) no qubit 0
qc.h(0)
# Aplica uma porta CNOT com controle em q0 e alvo em q1
qc.cx(0, 1)
# Aplica uma rotação R_y(pi) no qubit 1
qc.ry(np.pi, 1)
# Aplica novamente uma porta Hadamard no qubit 0
qc.h(0)
# Exibe o circuito
qc.draw("mpl")
```

[17]:



O vetor de estado apresentado pelo Qiskit corresponde ao resultado da aplicação sucessiva das cinco operações do circuito. O registrador é inicializado no estado $|00\rangle$, e a porta X aplicada ao qubit q_0 produz o estado $|01\rangle$. Em seguida, a porta de Hadamard é aplicada ao mesmo qubit, fazendo com que o registrador passe a ser descrito por

$$\frac{1}{\sqrt{2}}(|00\rangle - |01\rangle).$$

A porta CNOT é então aplicada, com q_0 atuando como qubit de controle e q_1 como alvo, de modo que apenas o segundo termo da superposição é modificado. Como o controle encontra-se em $|1\rangle$ nesse termo, o alvo é invertido, e o estado passa a ser

$$\frac{1}{\sqrt{2}}(|00\rangle - |11\rangle).$$

A operação seguinte consiste na rotação $R_y(\pi)$ aplicada ao qubit q_1 . Essa rotação é definida pela ação

$$R_y(\pi)|0\rangle = |1\rangle, \quad R_y(\pi)|1\rangle = -|0\rangle,$$

de modo que os termos da superposição são transformados segundo

$$|00\rangle \mapsto |10\rangle, \quad |11\rangle \mapsto -|01\rangle.$$

Como consequência, o registrador passa a ser descrito por

$$\frac{1}{\sqrt{2}}(|10\rangle + |01\rangle).$$

Finalmente, uma segunda porta de Hadamard é aplicada ao qubit q_0 . Utilizando-se as transformações

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

a superposição anterior é convertida no estado

$$\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle + |11\rangle),$$

o qual coincide exatamente com o vetor exibido pelo comando `psi.draw('latex')`.

```
[18]: # Extrai o vetor de estado resultante do circuito atual
psi = Statevector.from_instruction(qc)
#
psi.draw('latex')
```

[18]:

$$\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

3 Apêndice: traço parcial e estados mistos

Tomar o **traço parcial** de um estado composto significa descartar matematicamente uma das partes do sistema quântico para obter a descrição do que resta. Quando dois qubits formam um estado conjunto ρ , eles vivem no espaço de Hilbert $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_0$. No entanto, se estamos interessados apenas em um dos qubits — por exemplo, o qubit 0 — precisamos eliminar a informação do qubit 1. Esse processo é descrito pela operação chamada **traço parcial**, denotada por

$$\rho_0 = \text{Tr}_{q_1}(\rho).$$

O traço parcial combina duas ideias: (1) não estamos observando o qubit descartado; (2) toda informação que depende exclusivamente desse qubit é eliminada. Em termos formais, ele atua somando todas as componentes de ρ que pertencem ao qubit descartado, preservando apenas a estrutura matemática relevante para o qubit restante. No caso de um estado puro de dois qubits,

$$\rho = |\psi\rangle\langle\psi|,$$

o traço parcial produz uma matriz densidade 2×2 que descreve o estado de um único qubit como se o outro não existisse mais. Para estados separáveis (não emaranhados), o resultado é novamente um estado puro; já para estados emaranhados, como

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle),$$

o traço parcial produz

$$\rho_0 = \rho_1 = \frac{1}{2}I,$$

que é um estado misto: não aponta para nenhum ponto específico da esfera de Bloch, pois representa incerteza total sobre o qubit isolado. Isso ocorre porque o estado global possui correlações quânticas que não podem ser atribuídas a nenhum dos qubits individualmente. Em resumo, **tirar o traço** significa eliminar matematicamente a parte do sistema que não se deseja descrever, resultando em um estado reduzido que captura apenas a estatística observável do subsistema restante.