Universidad Nacional Autónoma de México
Facultad de Ciencias
Computación Cuántica I | 7138

Práctica 2 : | Circuitos en qiskit

Sosa Romo Juan Mario | 320051926

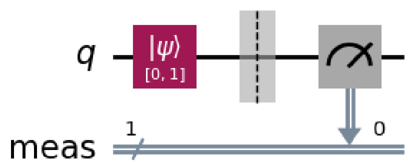
9/09/24

1. **Usando qiskit, diseñe tres circuitos cuya salida, antes de la medición resulte en cada uno de los siguientes estados:**

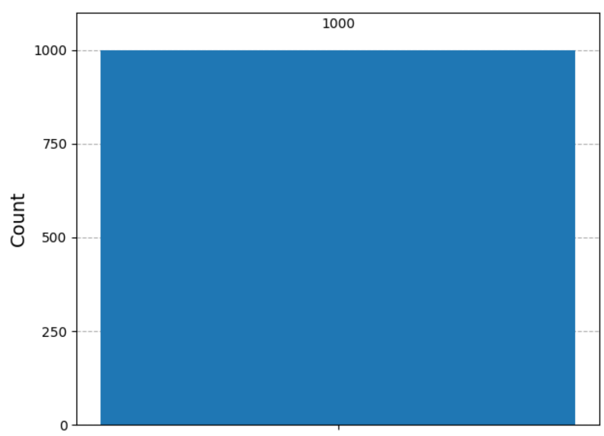
   (a) $|1\rangle$

   Usando estado inicial

```python
1 estado_inicial = [0,1]
2 circuito1 = QuantumCircuit(1)
3 circuito1.initialize(estado_inicial,0)
4 circuito1.measure_all()
5 circuito1.draw('mpl')
```



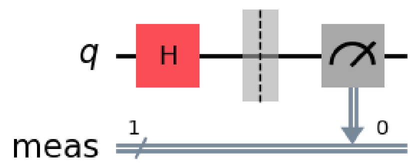
Este es bastante sencillo lo único que hay que hacer es utilizar el estado inicial $|1\rangle$ que se hace en la linea 1 dándole $\alpha = 0, \beta = 1$. Si medimos la salida nos dara:



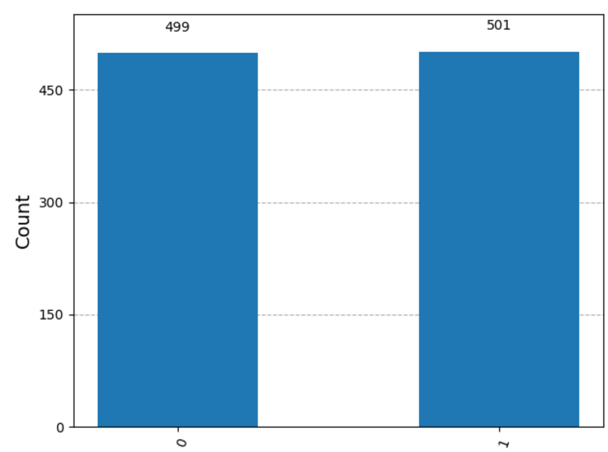
(b) $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

Por definición

```
circuitoMas = QuantumCircuit(1)
circuitoMas.h(0)
circuitoMas.measure_all()
circuitoMas.draw('mpl')
```

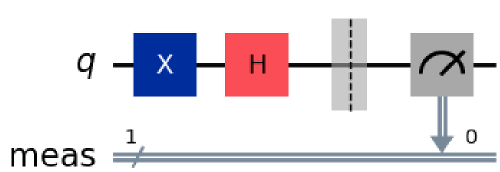


Este es un poquito mas complicado pero sale de la propiedad de que $\hat{H}|0\rangle = |+\rangle$ y como nuestros `qubits` se inicializan en 0 por defecto, solo es aplicar la compuerta H. Si la medimos nos queda así:

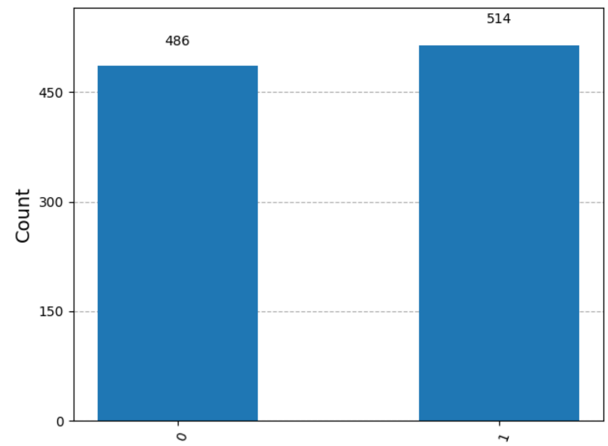


(c) $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

```
circuitoMenos = QuantumCircuit(1)
circuitoMenos.x(0)
circuitoMenos.h(0)
circuitoMenos.measure_all()
circuitoMenos.draw('mpl')
```

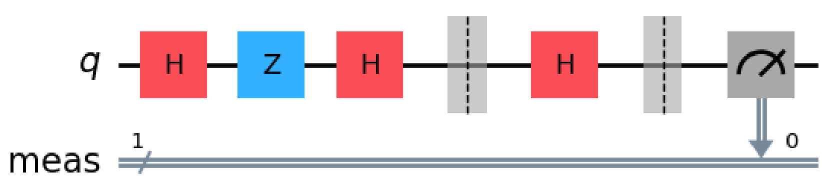


Este es básicamente lo mismo que el anterior pero antes de aplicar H usamos X para convertir nuestro **qubit** en un 1. Después usamos la propiedad: $\hat{H}\,|1\rangle = |-\rangle$ y terminamos. Al medirlo nos sale lo mismo.

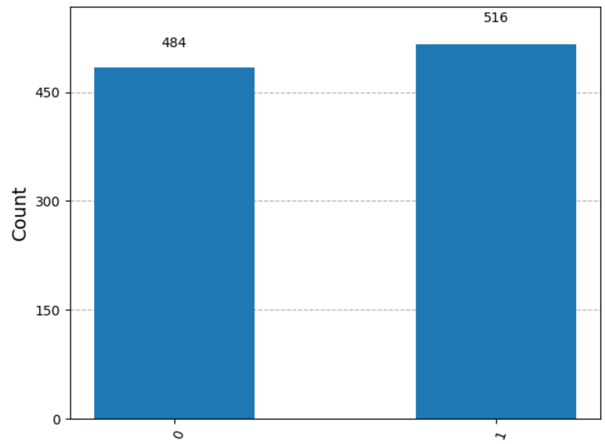


2. **Ejecute el circuito generado mediante las siguientes instrucciones:**

```
circuito2 = QuantumCircuit(1)

circuito2.h(0)
circuito2.z(0)
circuito2.h(0)

circuito2.barrier(0)

circuito2.h(0)

circuito2.measure_all()

circuito2.draw('mpl')
```

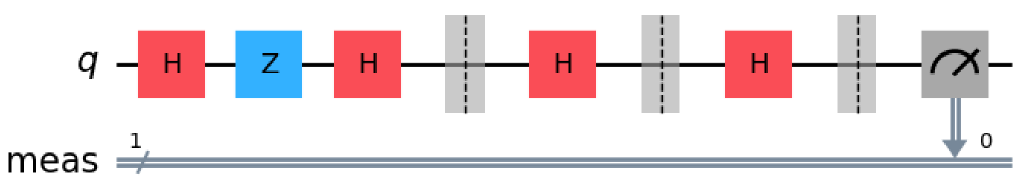


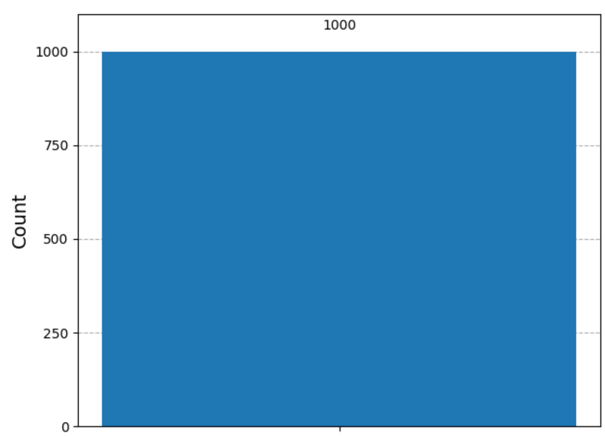
Que nos genera el siguiente histograma:



A partir del histograma, ¿puede determinar el estado final del sistema? Si no es posible, ¿cómo podría solucionarlo?

Únicamente usando el histograma no nos es posible determinar con exactitud el estado final del sistema pues se encuentra en superposición, podemos decir que por la cantidad tan cercana a .5 se trata del estado $|+\rangle$ o $|-\rangle$, sin embargo, para saberlo con exactitud podemos operar matemáticamente con las compuertas o usar código. Veámoslo, con compuertas primero notar que la compuerta $\hat{H}$ aplicada 2 veces seguida es la identidad por lo que podemos de hecho ignorar estas ultimas 2. Lo siguiente es que sabemos que nuestro qubit se inicializa en $|0\rangle$ por defecto, por lo que al aplicar la primera compuerta tenemos que $\hat{H}\,|0\rangle = |+\rangle$ y Z hace cambio de fase lo que nos lleva a $|-\rangle$. También se puede utilizar una compuerta $\hat{H}$ sobre el resultado antes de medir para obtener un 1 y saber que en efecto se trataba de dicha compuerta. Las imágenes del histograma y el circuito están aquí:

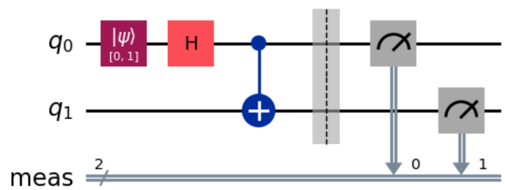




3. **Diseñe 3 circuitos diferentes de 2 qubits para generar los estados de Bell listados a continuación. Explique por qué sus circuitos funcionan y añada una imagen de cada uno de ellos.**

(a) $|\phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$

```
estado_inicial = [0,1]
circuito_bell2 = QuantumCircuit(2)
circuito_bell2.initialize(estado_inicial,0)
circuito_bell2.h(0)
circuito_bell2.cx(0,1)                #cx: controlled x
circuito_bell2.measure_all()
circuito_bell2.draw('mpl')
```



Para crear todos los estados de Bell en realidad se utiliza casi el mismo circuito con diferentes estados iniciales, es decir con diferentes qubits iniciales. En este caso pasa:

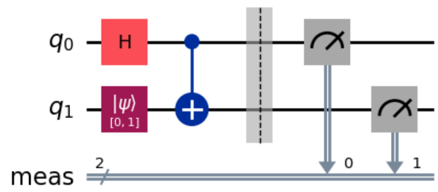
$$\hat{C}_x \hat{H} |10\rangle = \hat{C}_x \left(\frac{1}{\sqrt{2}}(|00\rangle - |10\rangle) \right) = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

Podemos comprobar sacando el statevector, recordando que la base es $00, 01, 10, 00$.

```
Statevector([ 0.70710678+0.j,  0.        +0.j,  0.        +0.j,
             -0.70710678+0.j],
            dims=(2, 2))
```

(b) $|\psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

```
estado_inicial = [0,1]
circuito_bell4 = QuantumCircuit(2)
circuito_bell4.initialize(estado_inicial,0)
circuito_bell4.initialize(estado_inicial,1)
circuito_bell4.h(0)
circuito_bell4.cx(0,1)                #cx: controlled x
circuito_bell4.measure_all()
circuito_bell4.draw('mpl')
```



En este caso, usamos la combinación inicial $|01\rangle$:

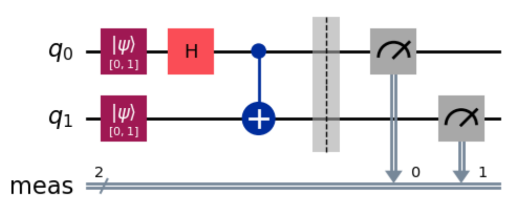
$$\hat{C}_x\hat{H}\,|01\rangle = \hat{C}_x\left(\frac{1}{\sqrt{2}}(|01\rangle + |11\rangle)\right) = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Podemos comprobar sacando el statevector, recordando que la base es $00, 01, 10, 00$.

```
Statevector([0.        +0.j, 0.70710678+0.j, 0.70710678+0.j,
             0.        +0.j],
            dims=(2, 2))
```

(c) $|\psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

```
estado_inicial = [0,1]
circuito_bell4 = QuantumCircuit(2)
circuito_bell4.initialize(estado_inicial,0)
circuito_bell4.initialize(estado_inicial,1)
circuito_bell4.h(0)
circuito_bell4.cx(0,1)                #cx: controlled x
circuito_bell4.measure_all()
circuito_bell4.draw('mpl')
```



En este caso, usamos la combinación inicial $|11\rangle$:

$$\hat{C}_x\hat{H}\,|11\rangle = \hat{C}_x\left(\frac{1}{\sqrt{2}}(|01\rangle - |11\rangle)\right) = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

Podemos comprobar sacando el statevector, recordando que la base es $00, 01, 10, 00$.

```
Statevector([ 0.        +0.j, -0.70710678+0.j,  0.70710678+0.j,
              0.        +0.j],
            dims=(2, 2))
```

Algo raro en esta parte es que al sacar el statevector se le aplica una fase global pero no es algo importante para esto y el circuito es correcto.

¿Cuál es la importancia de los estados de Bell en el cómputo cuántico?

Son los primeros 4 estados en donde podemos ver `qubits` con comportamiento entrelazado y superposición en donde, al medir uno el estado de ambos colapsa. De manera aplicada lo podemos pensar como la manera en la que los qubits se comunican y dependen entre si unos de otros.