

Básico

Imports e ajustes iniciais

```
from qiskit import *  
%matplotlib inline
```

Circuitos

Inicialização

```
circuitoQuantico = QuantumCircuit(n_qubits,n_bits)
```

Adição de portas

```
circuitoQuantico.nome_porta(parâmetro, índice_qubit)
```

Parâmetro pode existir(por exemplo porta rx que leva como parâmetro quanto será a rotação ou não(por exemplo porta X) .

Medindo

```
circuitoQuantico.measure(índice_qubit, índice_bit)
```

Rodando o circuito(Simulador)

```
simulador = Aer.get_backend('qasm_simulator')  
job = execute(circuitoQuantico, simulador, shots = 10000)  
Resultados = job.result()
```

O parâmetro shots em execute pode ser nulo

Rodando o circuito(IBMQ)

```
from qiskit import IBMQ
```

```
IBMQ.load_account()  
provedor = IBMQ.get_provider('ibm-q')  
computadorq = provedor.get_backend('ibmq_16_melbourne')  
executar = execute(circuitoQuantico, backend= computadorq)
```

```
resultado = executar.result()
```

Display

Diagrama do circuito

```
circuitoQuantico.draw(output='mpl')
```

Esfera de Bloch(pelo vetor de bloch)

```
from qiskit.visualization import plot_bloch_vector
```

```
plot_bloch_vector(vetor_bloch, title="titulo_esfera")
```

Esfera de Bloch(rodando um circuito)

```
from qiskit.tools.visualization import plot_bloch_multivector
```

```
simulador = Aer.get_backend('statevector_simulator')  
resultado = execute(circuitoQuantico, backend = simulador).result()
```

```
vetor_de_estado = result.get_statevector()  
plot_bloch_multivector(vetor_de_estado)
```

Histograma do resultado do circuito

```
simulador = Aer.get_backend('qasm_simulator')  
job = execute(circuitoQuantico, simulador, shots = 10000)  
Resultados = job.result()
```

O parâmetro shots em execute pode ser nulo

