

Qiskit | Advocates

Pasos

1. Hacer contribuciones sobre computación cuántica o actividades relacionadas a la comunidad Qiskit
2. Obtener 20 puntos
3. Aprobar el test Qiskit Developer Certification (es supervisado, costo \$200, posibilidad de solicitar un voucher)
4. Llenar aplicación en verano (posibilidad de aplicar una vez al año)

A partir de 16 años.

Tipos de contribución

1 a 4 puntos

- Contribuir al Qiskit textbook
- Participar en discusiones en canal de Slack de Qiskit
- Respuestas a preguntas relacionadas con Qiskit en StackExchange

4 a 8 puntos

- Contribuciones pequeñas a Qiskit
- Hacer contenido sobre Qiskit (videos, blogs, talleres, podcasts)
- Pequeños proyectos con Qiskit
- Participar en eventos de Qiskit

8 a 12 puntos

- Proyectos importantes con Qiskit
- Comenzar un club de computación cuántica
- Organizar un hackathon cuántico
- Crear un canal popular de computación cuántica (YouTube, blog)
- Qiskit Localization Medalla de bronze o plata

12 a 15 puntos

- Introducir funcionalidades nuevas a Qiskit
- Publicación científica usando Qiskit
- Ganar un evento de IBM Quantum
- Qiskit Localization Medalla de oro o platino

¿Qué debes saber para el examen?

- Instalar Qiskit
- Definir un circuito cuántico, un registro cuántico y un registro clásico
- Visualizar circuitos y qubits
- Aplicar compuertas (simples, de dos qubits, mediciones y barreras)
- Calcular la profundidad del circuito
- Uso de simuladores y de backends
- Ejecutar un circuito y obtener las cuentas de la medición
- Entender operadores
- OpenQasm
- Visualizar resultados de circuitos

Ejemplos

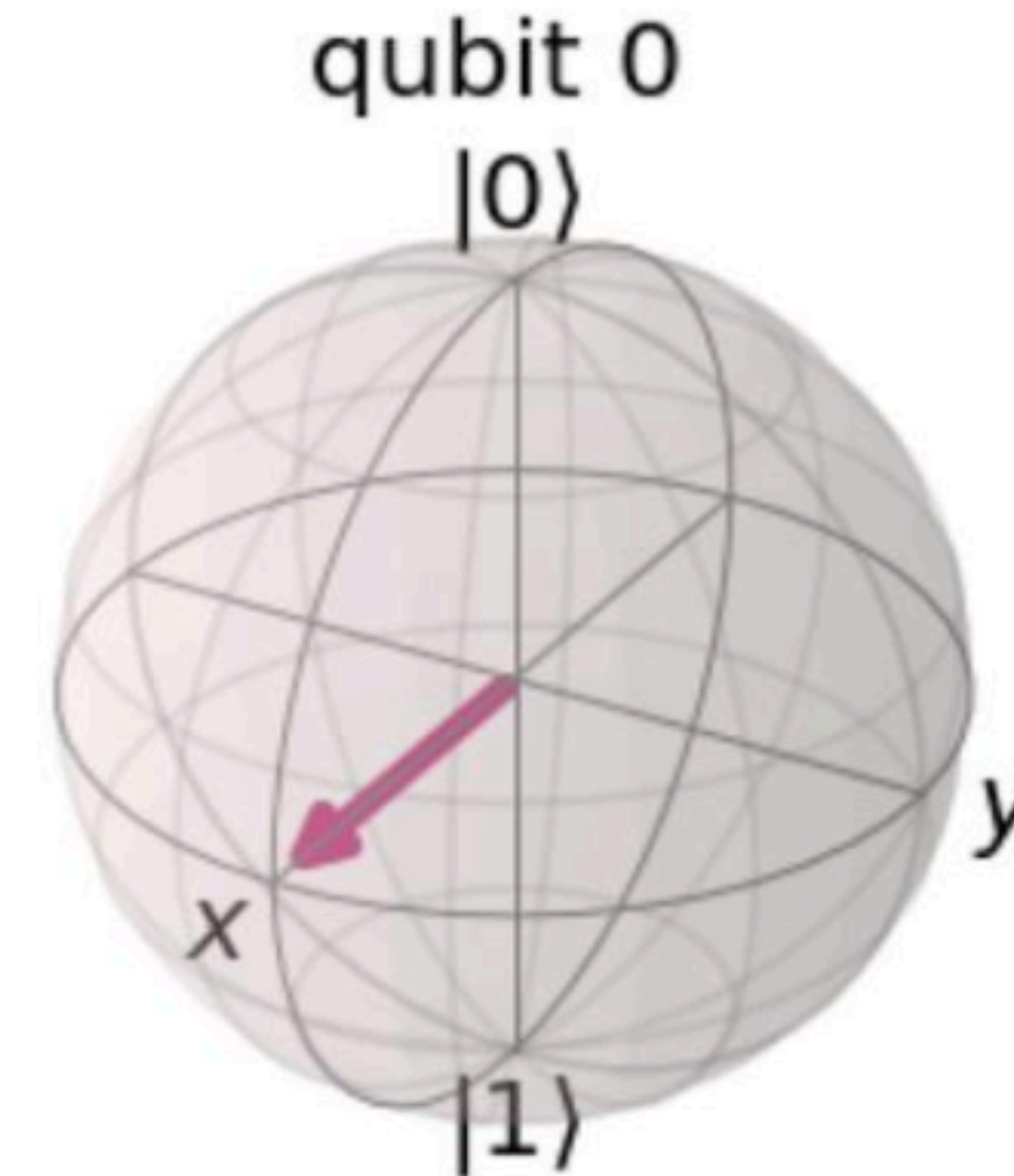
5. Which code fragment will produce a maximally entangled, or Bell, state?

- A. `bell = QuantumCircuit(2)`
`bell.h(0)`
`bell.x(1)`
`bell.cx(0, 1)`
- B. `bell = QuantumCircuit(2)`
`bell.cx(0, 1)`
`bell.h(0)`
`bell.x(1)`
- C. `bell = QuantumCircuit(2)`
`bell.h(0)`
`bell.x(1)`
`bell.cz(0, 1)`
- D. `bell = QuantumCircuit(2)`
`bell.h(0)`
`bell.h(0)`

6. Given this code, which two inserted code fragments result in the state vector represented by this Bloch sphere?

```
qc = QuantumCircuit(1,1)
# Insert code fragment here

simulator = Aer.get_backend('statevector_simulator')
job = execute(qc, simulator)
result = job.result()
outputstate = result.get_statevector(qc)
plot_bloch_multivector(outputstate)
```



- A. `qc.h(0)`
- B. `qc.rx(math.pi / 2, 0)`
- C. `qc.ry(math.pi / 2, 0)`
- D. `qc.rx(math.pi / 2, 0)`
`qc.rz(-math.pi / 2, 0)`
- E. `qc.ry(math.pi, 0)`

Enlaces

- Examen: <https://www.ibm.com/training/certification/C0010300>
- Aplicación: <https://github.com/qiskit-advocate/application-guide>
- Lista de advocates: <https://qiskit.org/advocates>