

- ✦ Bloque 3: Algoritmos Cuánticos

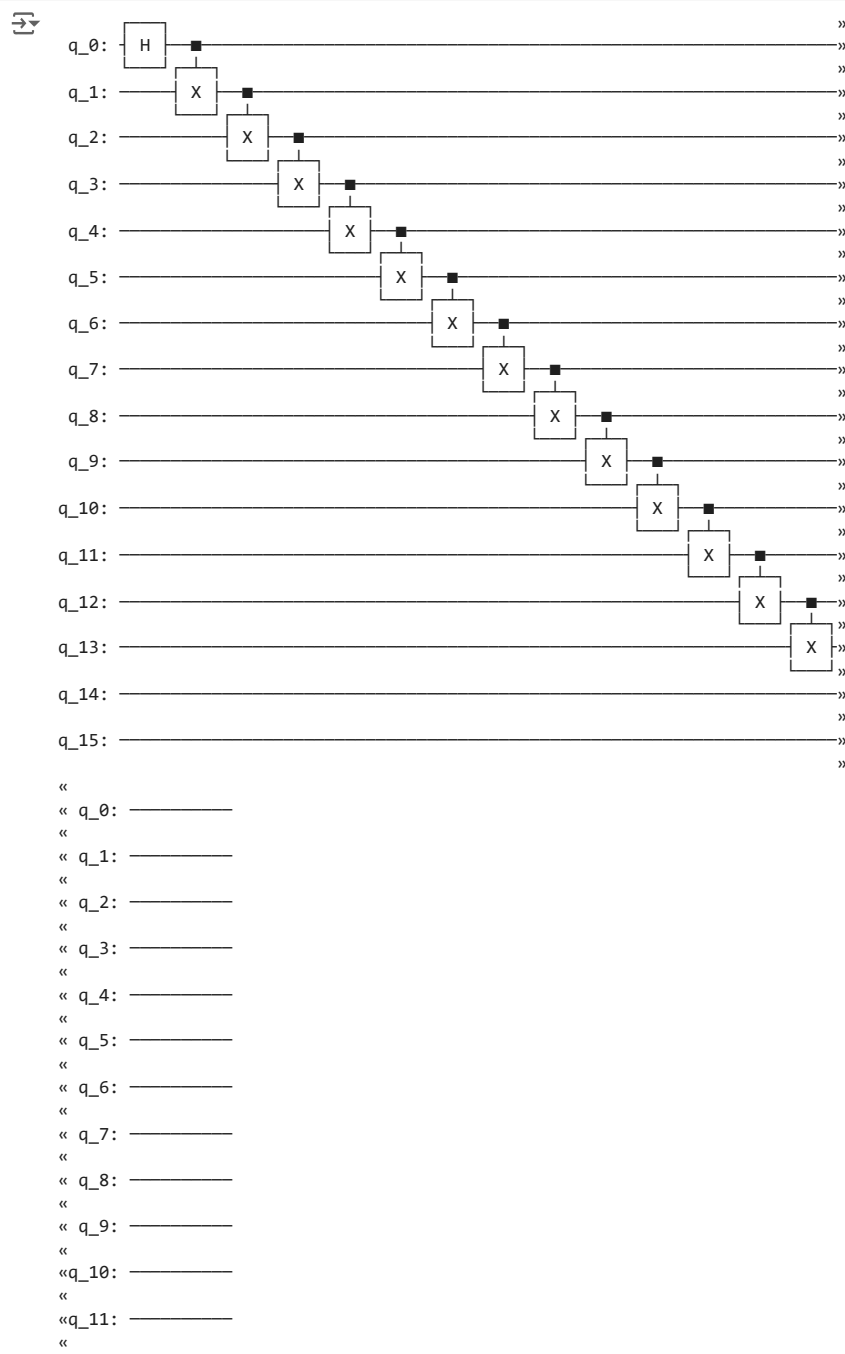
Alumno: Álvaro Manuel Aparicio Morales

I Certificado de Extensión Universitaria en Computación Cuántica (2024-2025)

✓ Ejercicio 52

```
from qiskit import QuantumCircuit
import numpy as np
```

```
# Circuito de 16 qubits
qc = QuantumCircuit(16)
qc.h(0)
for q in range(1,qc.num_qubits):
    qc.cx(q-1,q)
print(qc)
```



Para poder optimizar el circuito anterior, podemos trabajar con el circuito mediante la aplicación de puertas en paralelo. Para lograrlo se puede seguir la estructura de un árbol binario donde cada nivel es una etapa donde se pueden aplicar puertas cnots paralelas.

A continuación generamos el circuito optimizado:

```
# Crear un nuevo circuito
qc_optimized = QuantumCircuit(16)

# 1. Aplicar la puerta Hadamard al primer qubit
qc_optimized.h(0)

# 2. Usar CNOT en paralelo dividiendo los qubits
step = 1
while step < qc.num_qubits:
    for i in range(0, qc.num_qubits - step, 2 * step):
        qc_optimized.cx(i, i + step)
    step *= 2

# Dibujar el circuito optimizado
print("Circuito optimizado con profundidad 5:")
print(qc_optimized)
```

→ Circuito optimizado con profundidad 5:

