Team 3

Quantum programs testing tool

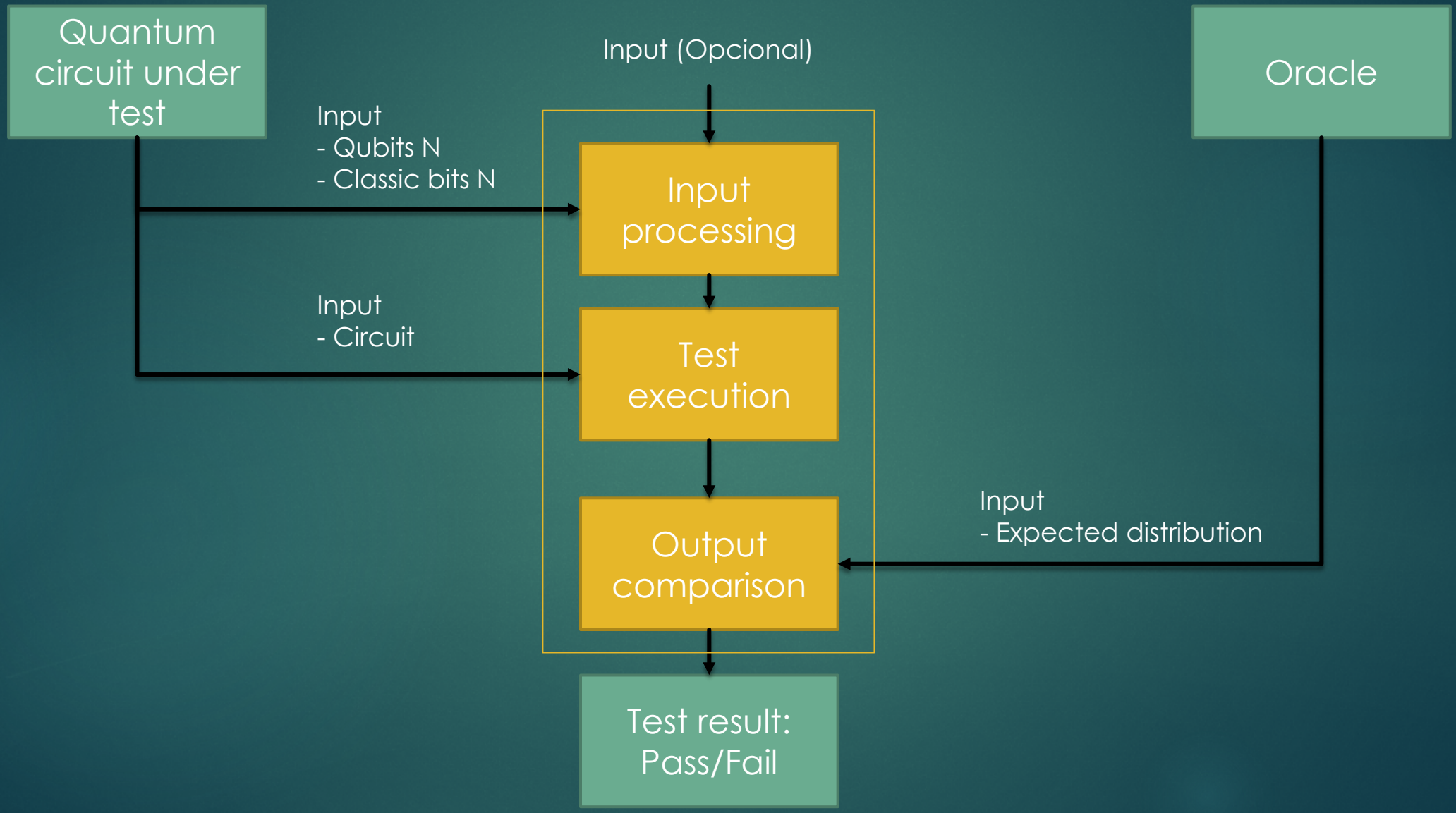
HACKATHON QUANTUM MADRID

Motivación

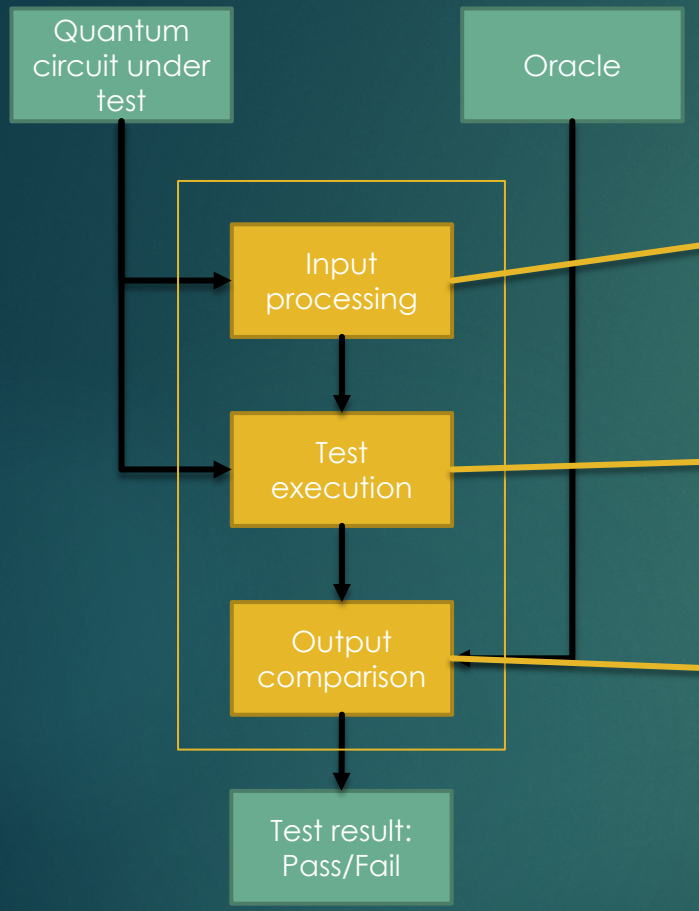
"To fully exploit QC's potential, it is important to ensure the correctness of quantum programs. Doing so via software testing is, however, very challenging because of QC's inherent properties: superposition and entanglement." X. Wang et al.

- Existe teoría pero no herramientas
- Los sistemas clásicos no estan adaptados

Estructura


Quantum circuit under test
Input (Opcional)
Oracle
Input
- Qubits N
- Classic bits N
Input processing
Input
- Circuit
Test execution
Output comparison
Input
- Expected distribution
Test result: Pass/Fail

Testing Quantum Programs

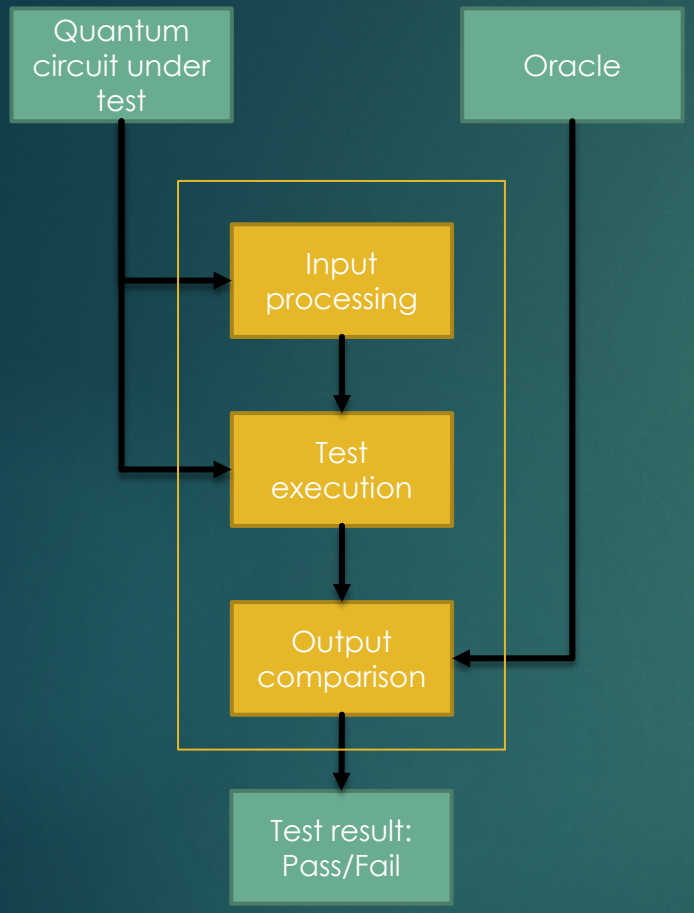


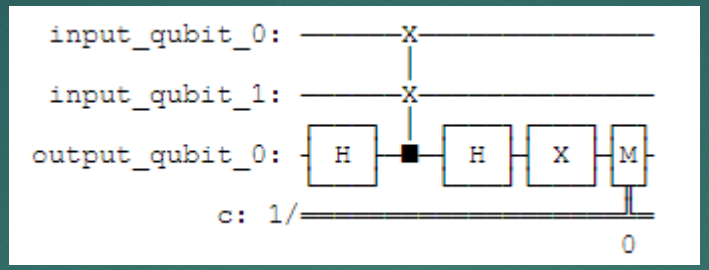

```
#Función iqc_individual
#INPUTS: value: array of ints, num_total_qubits: int, num_cl_bits: int
#OUTPUTS: qc: QuantumCircuit
def iqc_individual(value=[], num_total_qubits=0, num_cl_bits=0):
```

```
#Función test_circuit
#INPUTS: iqc: QuantumCircuit, qc: QuantumCircuit, shots: int
#OUTPUTS: histogram: array of ints
def test_circuit(iqc, qc,  shots=1024):
```

```
#Función chi_square
#INPUTS: outputs: dict<string, int>, oracle: dict<string, int>
#OUTPUTS: boolean
def chi_square(outputs,oracle):
```

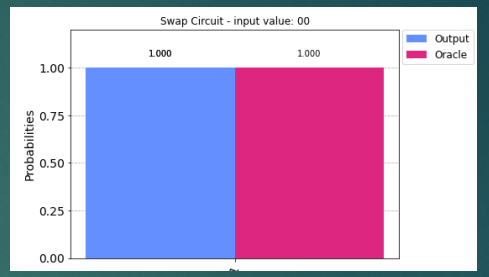
Testing Quantum Programs

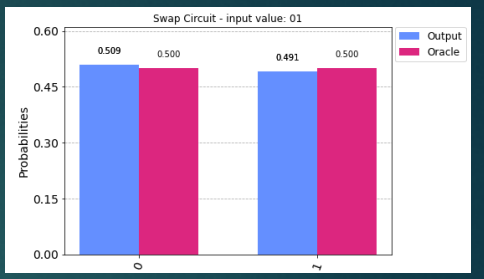


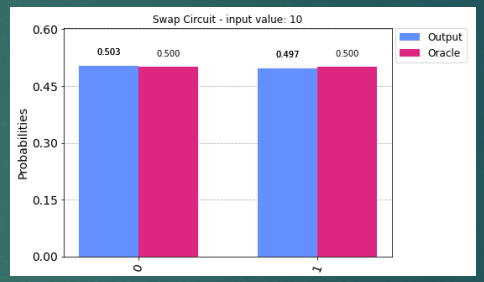


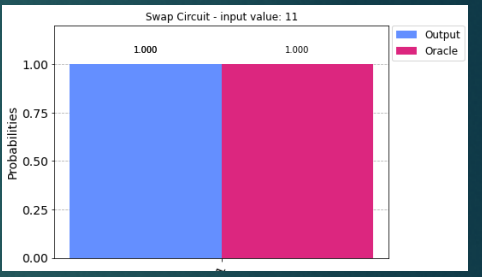

```
Circuit passed evaluation for input [0, 0].
Circuit passed evaluation for input [0, 1].
Circuit passed evaluation for input [1, 0].
Circuit passed evaluation for input [1, 1].
```

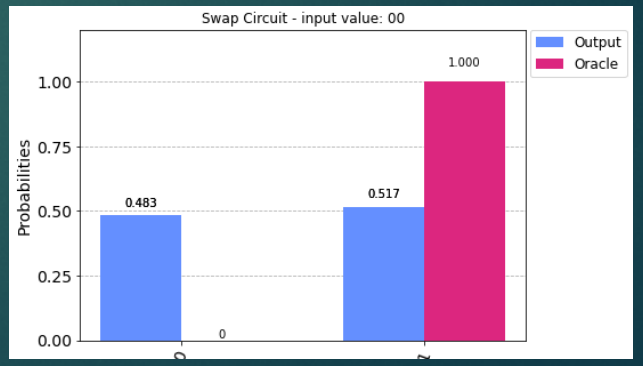
```
Circuit failed evaluation for input [0, 0]! There is some error.
Circuit returned {'1': 549, '0': 475}
Circuit passed evaluation for input [0, 1].
Circuit passed evaluation for input [1, 0].
Circuit failed evaluation for input [1, 1]! There is some error.
Circuit returned {'1': 530, '0': 494}
```



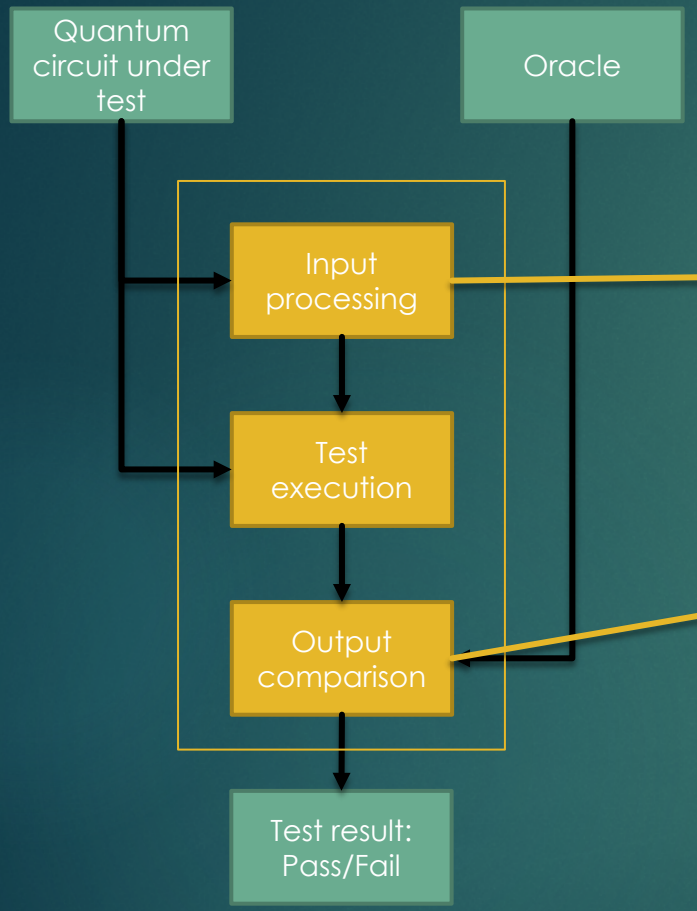








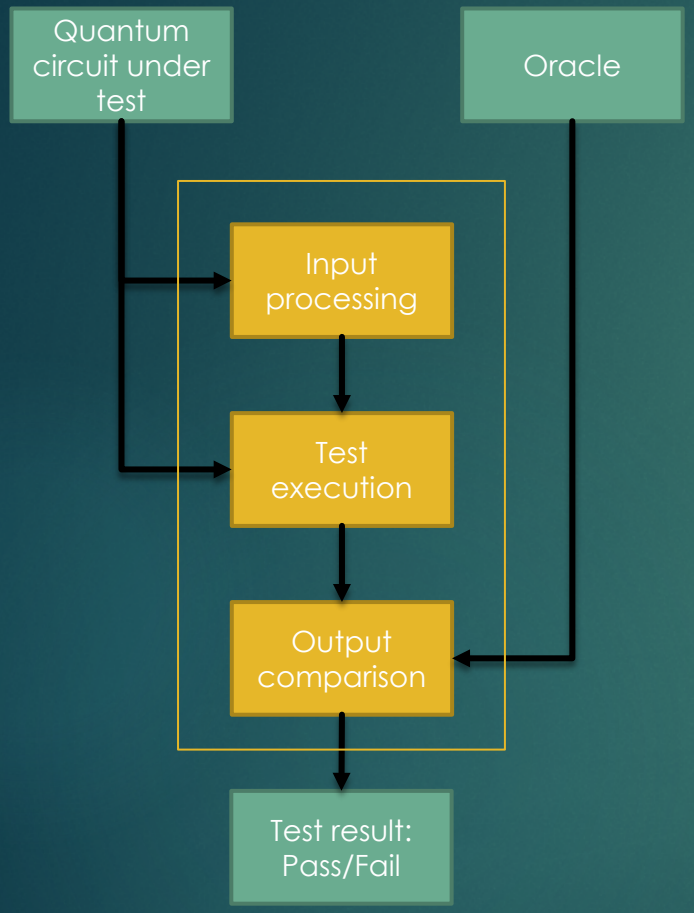
Probando Múltiples Inputs

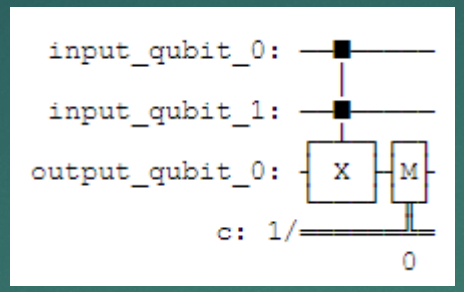



```
#Función iqc_superposition
#INPUTS: num_inputs: int, num_total_qubits: int, num_cl_bits: int
#OUTPUTS: qc: QuantumCircuit
def iqc_superposition(num_inputs=0, num_total_qubits=0, num_cl_bits=0)
```

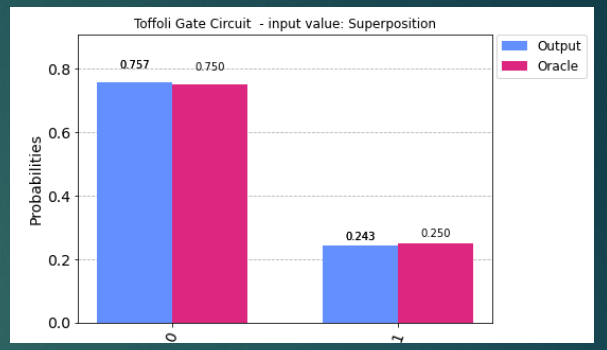
```
#Función oracle_union
#INPUTS: oracles: dict<string, dict<string, int>>
#OUTPUTS: oracle: dict<string, int>
def oracle_union(oracles):
```

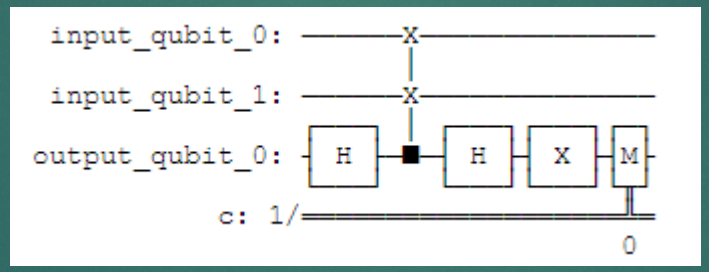
Probando Múltiples Inputs con Superposicion



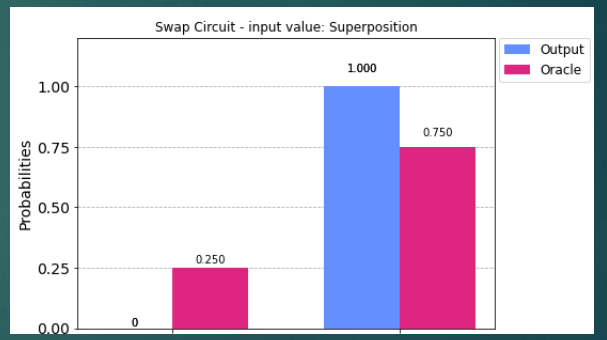



```
Circuit passed evaluation.
```






```
Circuit failed evaluation! There is some error.
Circuit returned {'1': 1024}
```



Conclusiones y trabajo a futuro

- Hemos generado una herramienta para testear programas cuánticos de forma automática.
- Tiene dos modos:
  - Modo clásico: Requiere inputs
  - Modo cuántico: Prueba todos los inputs a la vez

- Encontrar la forma de calcular el oráculo conjunto
- Calcular la matriz del circuito y comprobar si se corresponde

Quantum programs testing tool

Miembros:

- **Alfredo Ibias Martínez**
- **David Presa**
- **Javier Parra**
- **Ilya Lapshin**