

# IMPLEMENTACIÓN DE ALGORITMO DE DEUTSCH Y DE DEUTSCH-JOZSA

*GitHub:*

<https://github.com/anndr0/CNYT-Deutsch-Deutsch-Jozsa>

Ana María Durán Burgos  
ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO  
[ana.duran-b@mail.escuelaing.edu.co](mailto:ana.duran-b@mail.escuelaing.edu.co)



**23 DE NOVIEMBRE**

*Este reporte se entrega para cumplir con los requisitos parciales del curso CNYT:  
Computación Cuántica- 2022-1*

## TABLA DE CONTENIDO

IMPLEMENTACIÓN DE ALGORITMO DE DEUTSCH Y DE DEUTSCH-JOZSA .....	1
1 INTRODUCCIÓN .....	3
2 ALGORITMO DE DEUTSCH-JOZSA .....	3
2.1 ALGORITMO DE DEUTSCH.....	3
2.2 PROBLEMA.....	3
2.3 IMPLEMENTANDO LAS FUNCIONES EN EL COMPUTADOR CUÁNTICO .....	5
2.3.1 PRIMERA FUNCIÓN ( <i>cruzados</i> ) .....	5
2.3.2 SEGUNDA FUNCIÓN ( <i>iguales</i> ) .....	8
2.3.3 TERCERA FUNCIÓN ( <i>a uno</i> ) .....	11
2.3.4 CUARTA FUNCIÓN ( <i>a cero</i> ) .....	14
2.4 IMPLEMENTANDO EL ALGORITMO DE DEUTSCH EN UN COMPUTADOR CUÁNTICO .....	17
2.4.1 PRIMERA FUNCIÓN ( <i>cruzados</i> ) .....	17
2.4.2 SEGUNDA FUNCIÓN ( <i>iguales</i> ) .....	18
2.4.3 TERCERA FUNCIÓN ( <i>a uno</i> ) .....	19
2.4.4 CUARTA FUNCIÓN ( <i>a cero</i> ) .....	20
3 ALGORITMO DE DEUTSCH-JOZSA .....	21
3.1 PROBLEMA.....	21
3.2 IMPLEMENTANDO LAS FUNCIONES EN EL COMPUTADOR CUÁNTICO .....	22
3.2.1 PRIMERA FUNCIÓN BALANCEADA .....	22
3.2.2 SEGUNDA FUNCIÓN BALANCEADA .....	23
3.2.3 TERCERA FUNCIÓN BALANCEADA.....	24
3.2.4 FUNCIÓN CONSTANTE .....	25
3.3 IMPLEMENTANDO EL ALGORITMO DE DEUTSCH-JOSZA EN UN COMPUTADOR CUÁNTICO.....	26
3.3.1 PRIMERA FUNCIÓN BALANCEADA .....	26
3.3.2 SEGUNDA FUNCIÓN BALANCEADA .....	26
3.3.3 TERCERA FUNCIÓN BALANCEADA.....	27
3.3.4 FUNCIÓN CONSTANTE .....	27
4. CONCLUSIONES .....	28
5. BIBLIOGRAFÍA .....	28

# 1 Introducción

En este reporte se ejecutan los algoritmos Deutsch y Deutsch-Jozsa y se prueba su comportamiento utilizando varios ejercicios sugeridos. Estos algoritmos surgen para resolver el problema de encontrar o adivinar la "caja negra" de una función receptora (balanceada o constante) y en este reporte, por medio de gráficas y pruebas lograremos comprobar el correcto funcionamiento de estos identificando si una función es constante o balanceada. Si se define un tipo de función, se debe tener en cuenta que en el desarrollo de estos algoritmos se utilizan varias puertas como Hadamard. Su función principal es llevar los qubits entrantes a lo que se llama una superposición cuántica.

## 2 Algoritmo de Deutsch-Jozsa

[1] “En el circuito que implementa el algoritmo Deutsch-Jozsa se define el estado inicial como sigue: simulación de sistemas cuánticos y resolución de sistemas de ecuaciones lineales (Montanaro, 2016). Entre esos problemas existe uno que consiste en lo siguiente: dada una función binaria, y asumiendo que la misma puede ser únicamente constante o balanceada, distinguir cada caso. Un computador clásico debe evaluar la función  $2^{n-1} + 1$  veces, para una función con  $n$  bits de entrada, con el fin de llevar a cabo esta tarea.

Se presenta el algoritmo de Deutsch-Jozsa para tener una visión del poder de cómputo que tiene la computación cuántica, este poder de cómputo se basa en el paralelismo, el cual se caracteriza por el crecimiento exponencial, en relación con el número de qubits, de la capacidad de realizar operaciones de manera simultánea.”

### 2.1 Algoritmo de Deutsch

“Es un algoritmo cuántico, diseñado para ejecutar sobre un computador cuántico, en este nos dan una función cuántica (que para nosotros es una caja negra)  $f(x_1, x_2)$  que toma 2 bits de entrada  $x_1, x_2$ , y devuelve un valor binario  $f(x_1, x_2)$ .”

### 2.2 Problema

El problema surge al querer implementar una función  $f$ , tal que  $f: \{0,1\} \rightarrow \{0,1\}$  que sea capaz de decir si dicha función es Balanceada o Constante,

donde:

1.  $f$  es balanceada si  $f(1) \neq f(0)$
2.  $f$  es constante si  $f(1) = f(0)$

Entonces, graficamos las funciones para un mejor entendimiento de la información que se nos presenta, relacionar y crear patrones que se ven evidenciados en las siguientes 4 funciones.

- ***Dibujo de función***

- ***Matriz correspondiente***

Para crear la matriz correspondiente a la función debemos tener en cuenta que las columnas 00, 01, 10, 11 representarán las entradas del circuito y las filas serán el resultado de esta entrada.

- ***Circuito correspondiente***

Ya que está planteada una matriz que represente la función dada, se debe encontrar un patrón que nos permitirá crear un circuito.

- ***Resultados de las 4 pruebas***

Se deben realizar 4 pruebas a cada una de las funciones para mostrar el efecto que modela la función propuesta, haciendo que las entradas sean:

$$q[0] = |0\rangle, q[1] = |0\rangle$$

$$q[0] = |0\rangle, q[1] = |1\rangle$$

$$q[0] = |1\rangle, q[1] = |0\rangle$$

$$q[0] = |1\rangle, q[1] = |1\rangle$$

## 2.3 IMPLEMENTANDO LAS FUNCIONES EN EL COMPUTADOR CUÁNTICO

### 2.3.1 PRIMERA FUNCIÓN (cruzados)

a) Dibujo de función



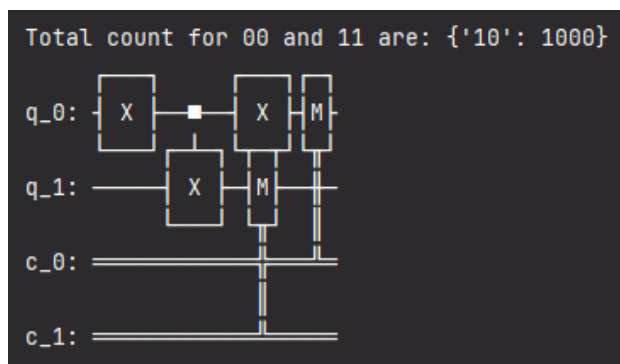
b) Matriz correspondiente

	00	01	10	11
00	0	1	0	0
01	1	0	0	0
10	0	0	1	0
11	0	0	0	1

c) Circuito correspondiente

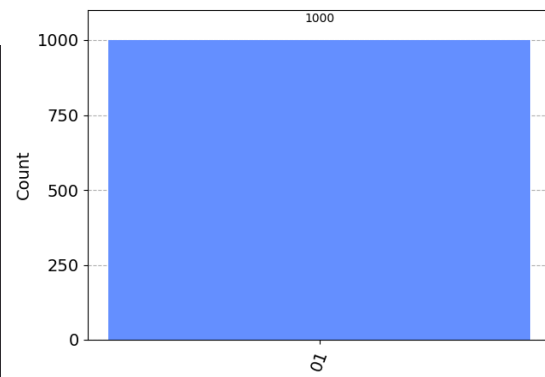
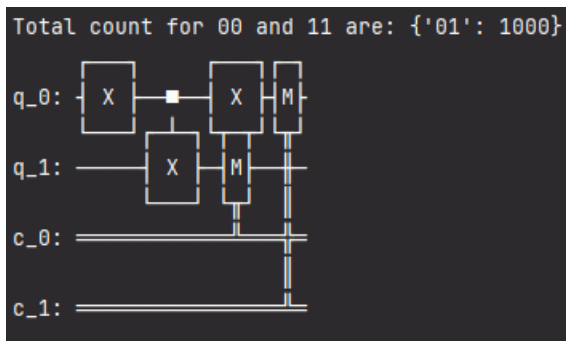
El primer alambre se mantiene igual, pero en el segundo es la negación de la segunda entrada. Solo se niega el segundo alambre si el primero tiene un cero como entrada.

Se modela con un CNOT controlado, pero negando el bit antes y después de la lectura.

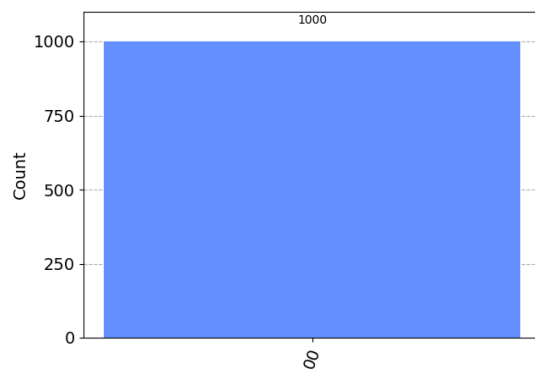
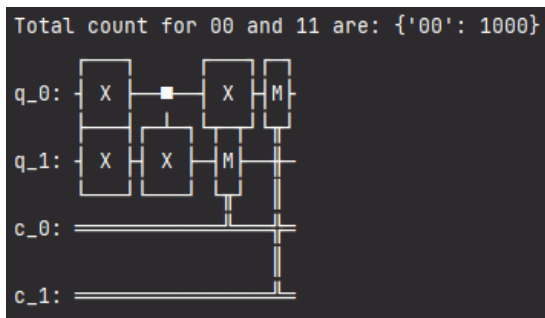
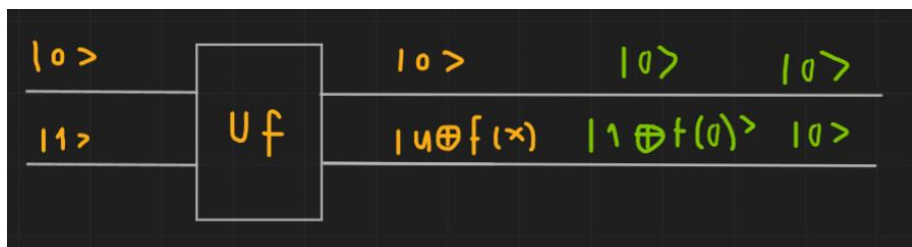


d) Resultados

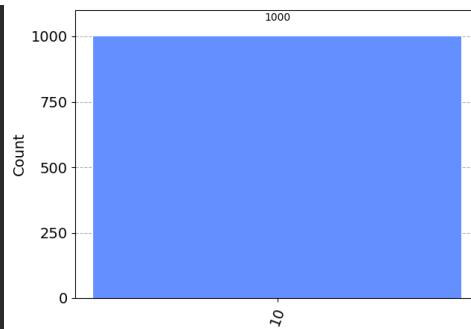
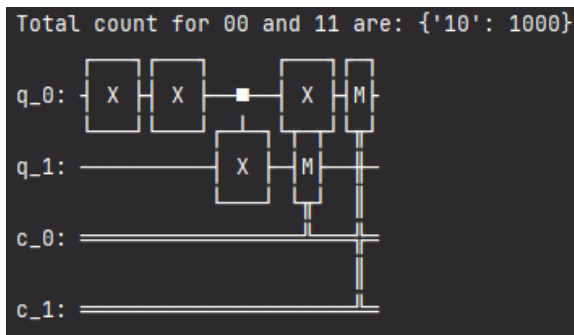
♥ Entrada  $|0\rangle, |0\rangle$



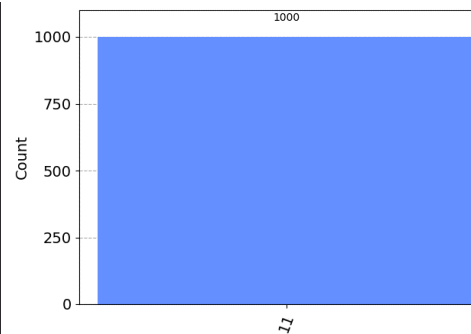
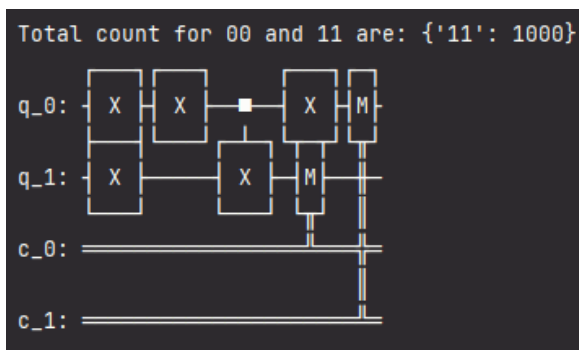
♥ Entrada  $|0\rangle, |1\rangle$



♥ Entrada  $|1\rangle, |0\rangle$

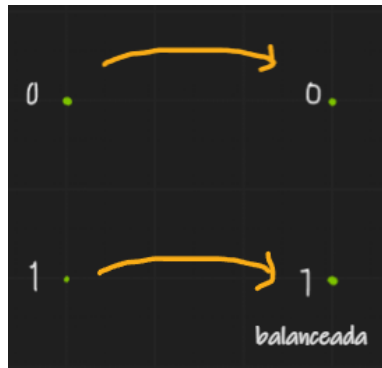


♥ Entrada  $|1\rangle, |1\rangle$



### 2.3.2 SEGUNDA FUNCIÓN (iguales)

a) Dibujo de función

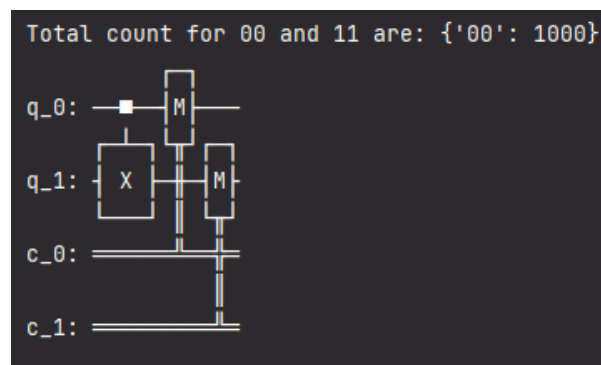


b) Matriz correspondiente

	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	0	1
11	0	0	1	0

c) Circuito correspondiente

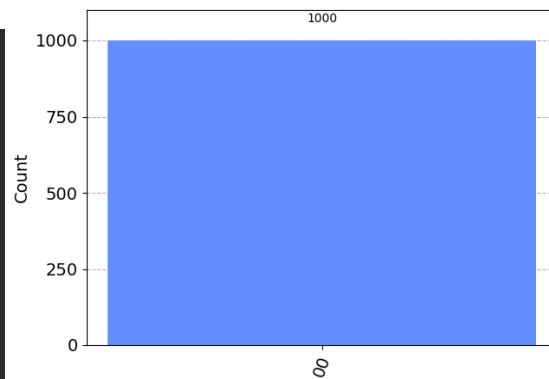
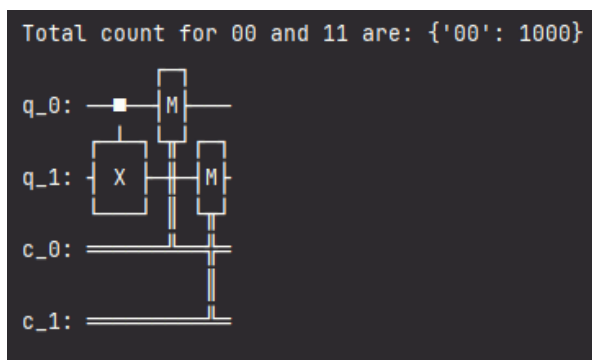
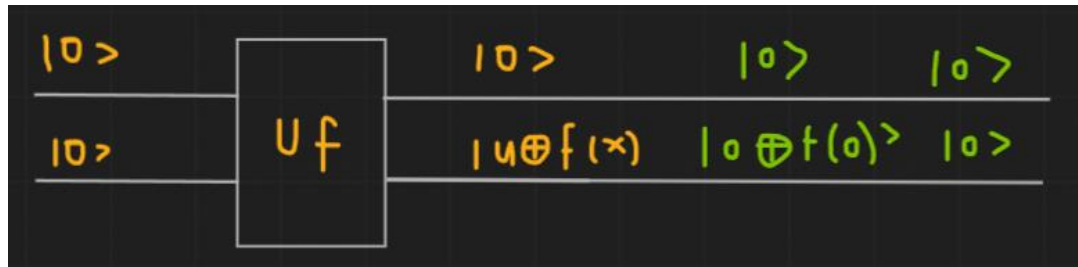
El primer alambre se mantiene igual, el segundo alambre cambia si el primero es un uno, entonces utilizamos un CNOT controlado.



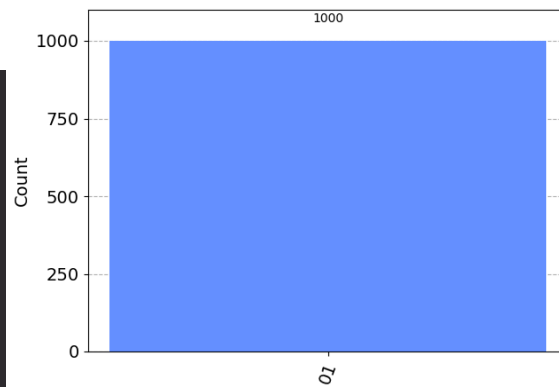
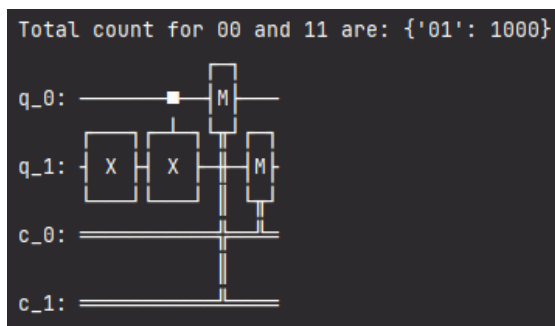
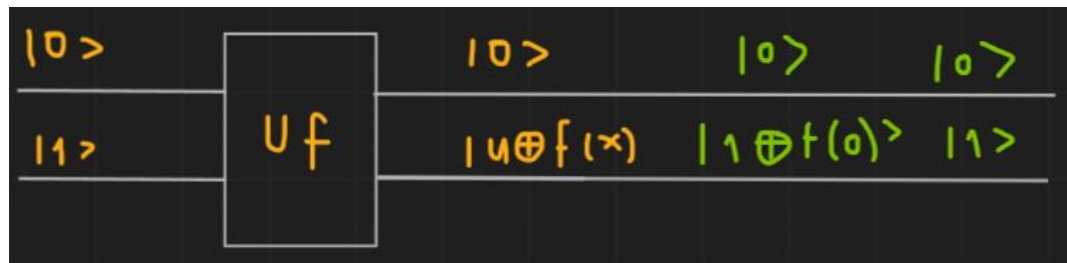


d) Resultados

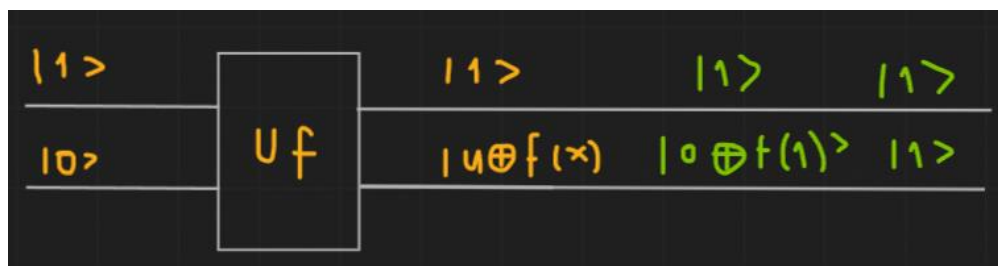
♥ Entrada  $|0\rangle, |0\rangle$



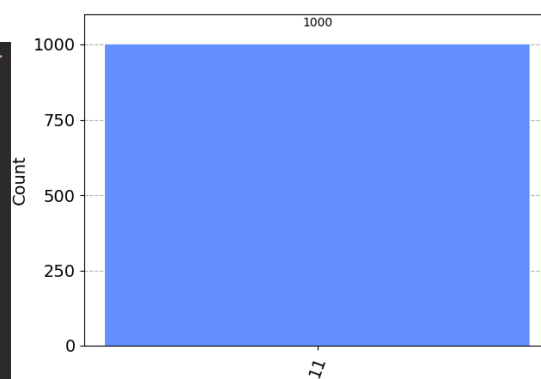
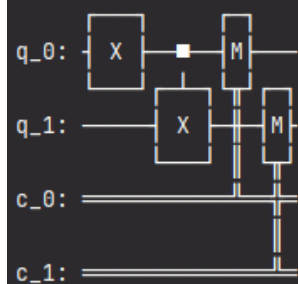
♥ Entrada  $|0\rangle, |1\rangle$



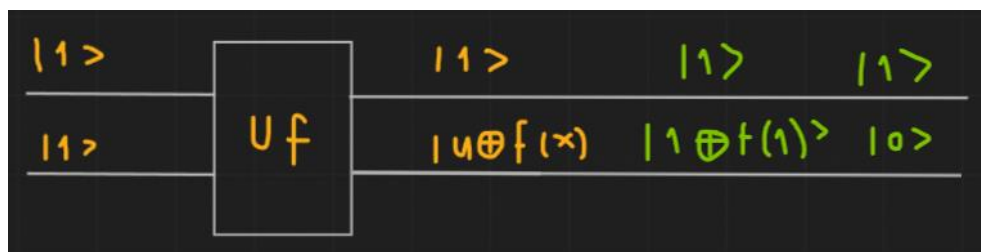
♥ Entrada  $|1\rangle, |0\rangle$



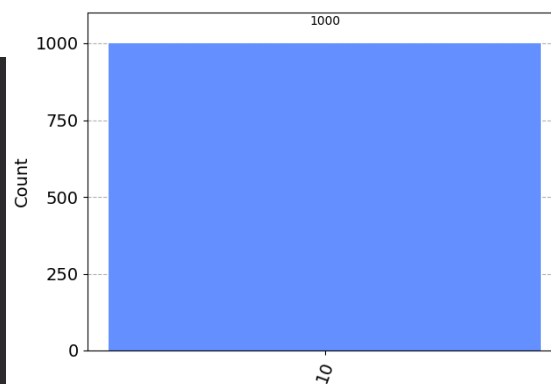
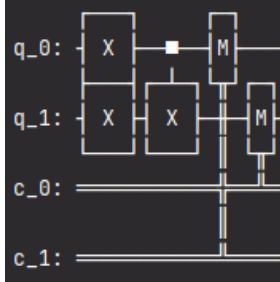
Total count for 00 and 11 are: {'11': 1000}



♥ Entrada  $|1\rangle, |1\rangle$



Total count for 00 and 11 are: {'10': 1000}



### 2.3.3 TERCERA FUNCIÓN (a uno)

a) Dibujo de función

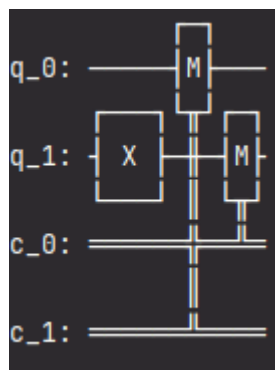


b) Matriz correspondiente

	00	01	10	11
00	0	1	0	0
01	1	0	0	0
10	0	0	0	1
11	0	0	1	0

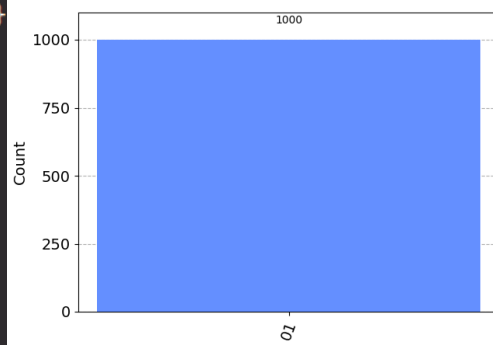
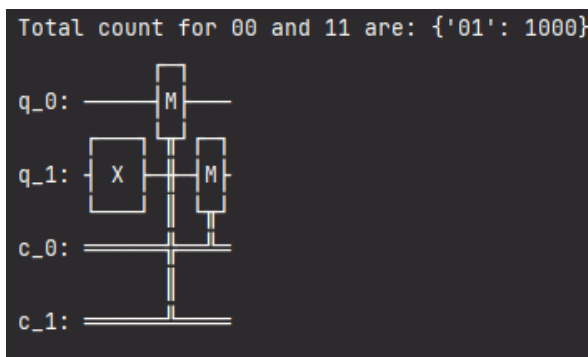
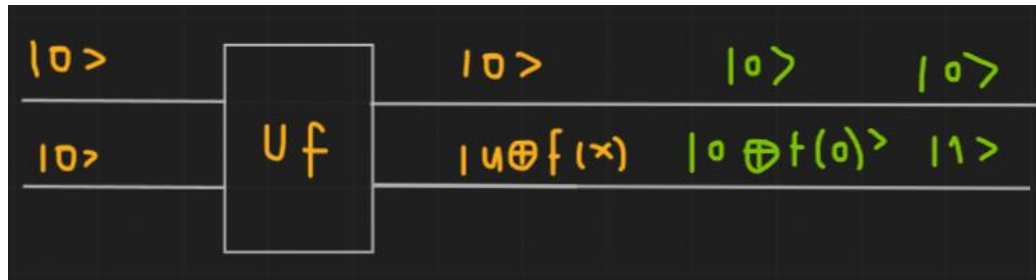
c) Circuito correspondiente

El primer alambre se mantiene igual, el segundo alambre se niega.

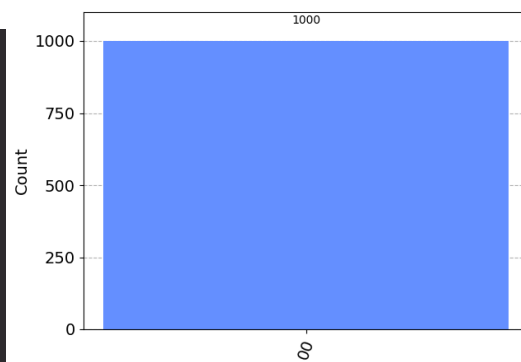
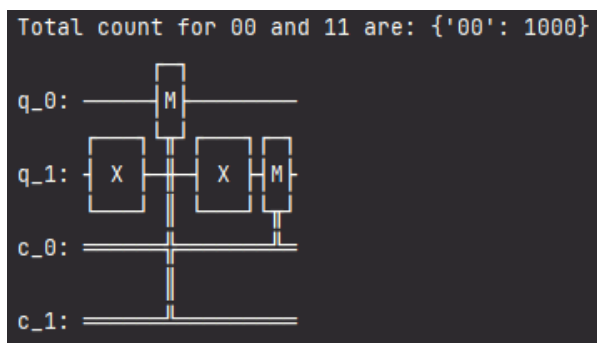
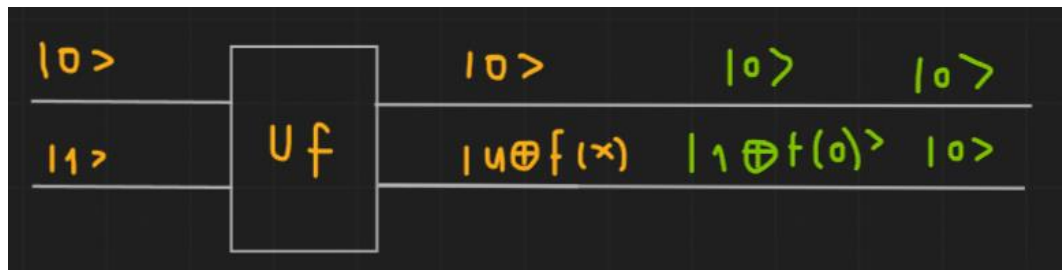


#### d) Resultados

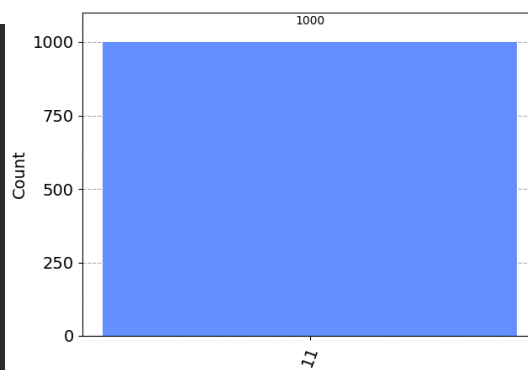
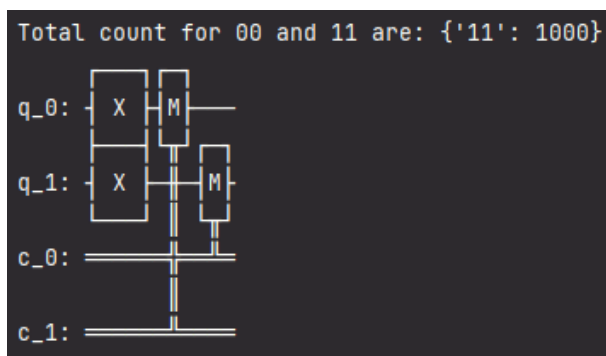
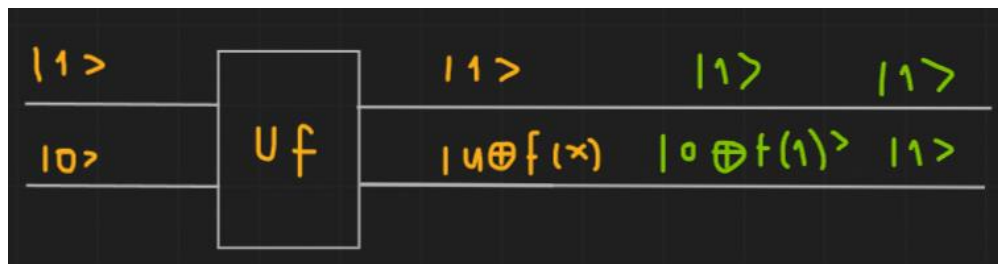
♥ Entrada  $|0\rangle, |0\rangle$



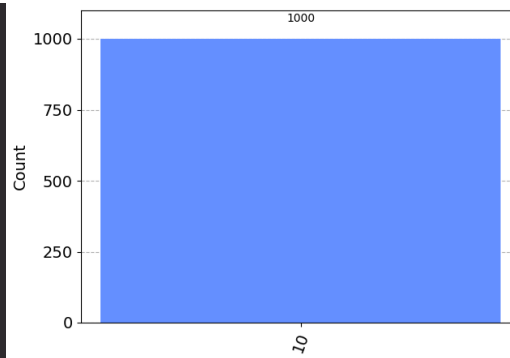
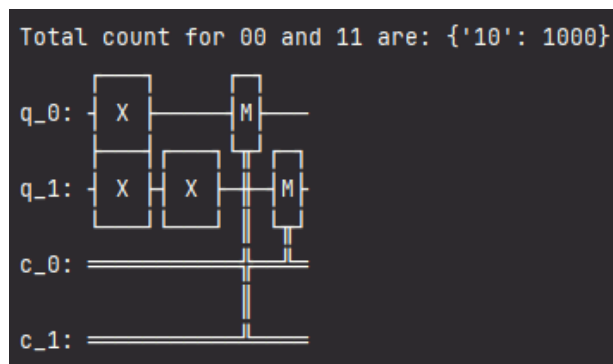
♥ Entrada  $|0\rangle, |1\rangle$



♥ Entrada  $|1\rangle, |0\rangle$

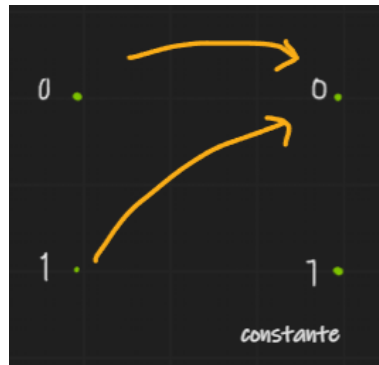


♥ Entrada  $|1\rangle, |1\rangle$



### 2.3.4 CUARTA FUNCIÓN (a cero)

a) Dibujo de función

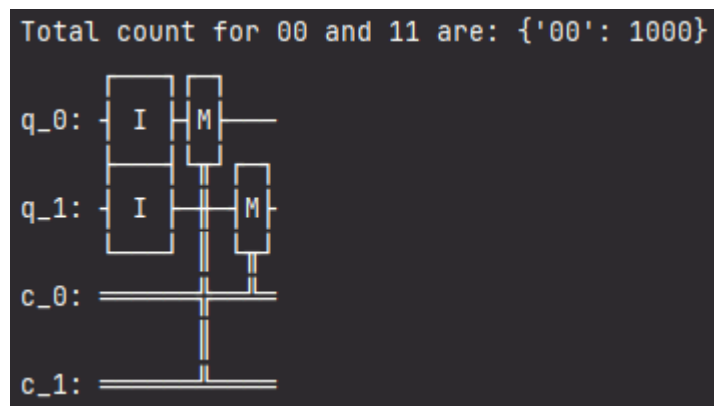


b) Matriz correspondiente (*Identidad*)

	00	01	10	11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

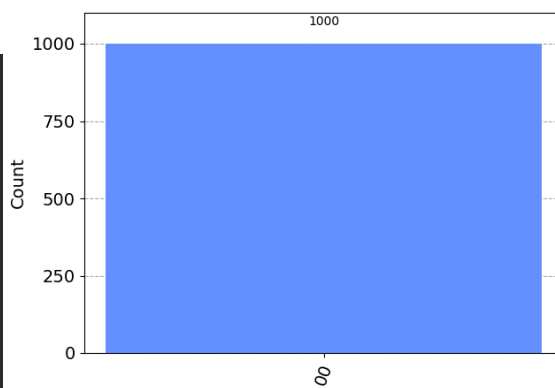
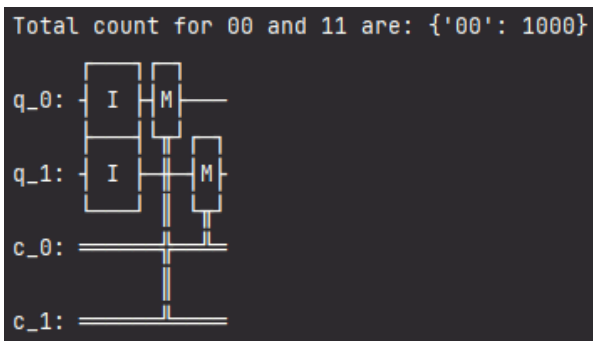
c) Circuito correspondiente

El primer alambre se mantiene igual, el segundo alambre se niega.

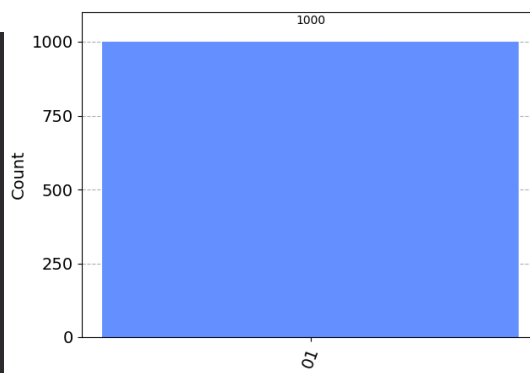
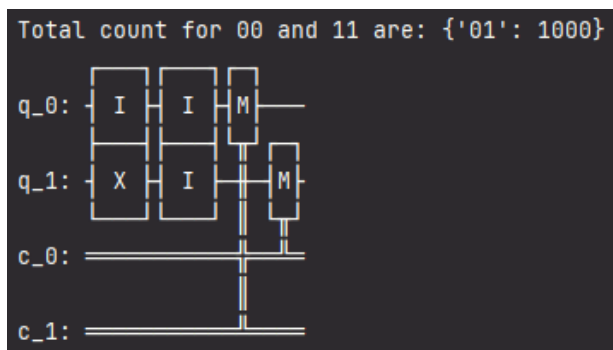
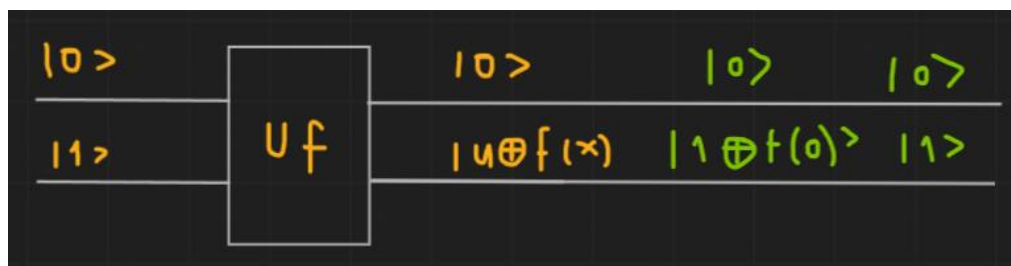


#### d) Resultados

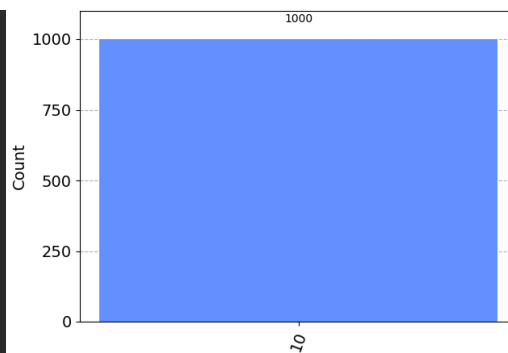
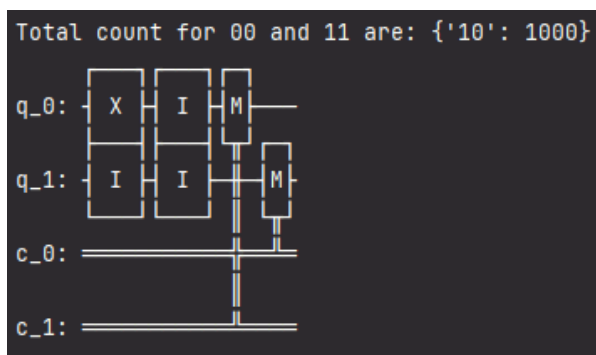
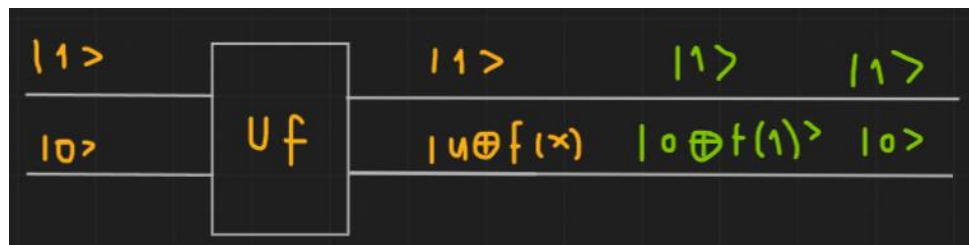
♥ Entrada  $|0\rangle, |0\rangle$



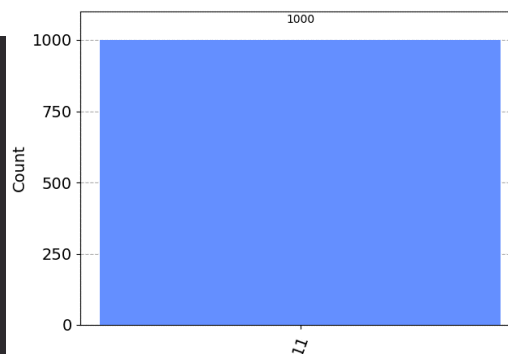
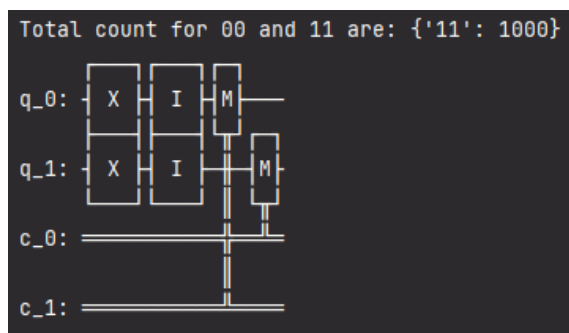
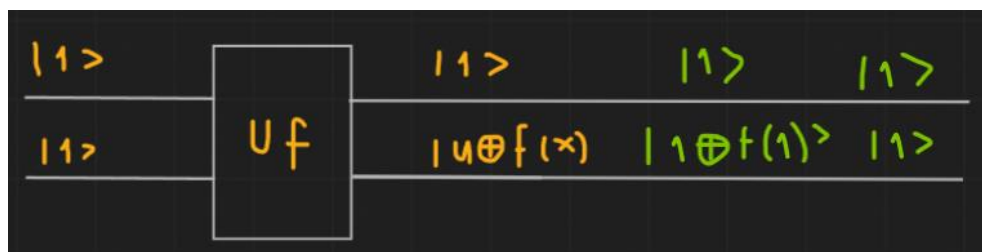
♥ Entrada  $|0\rangle, |1\rangle$



♥ Entrada  $|1\rangle, |0\rangle$



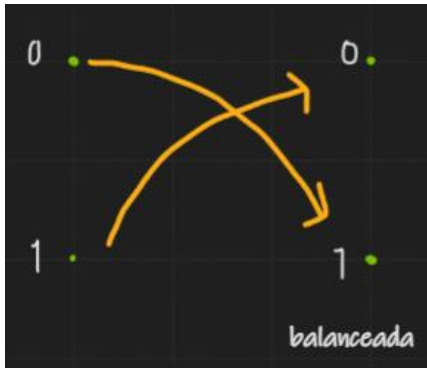
♥ Entrada  $|1\rangle, |1\rangle$



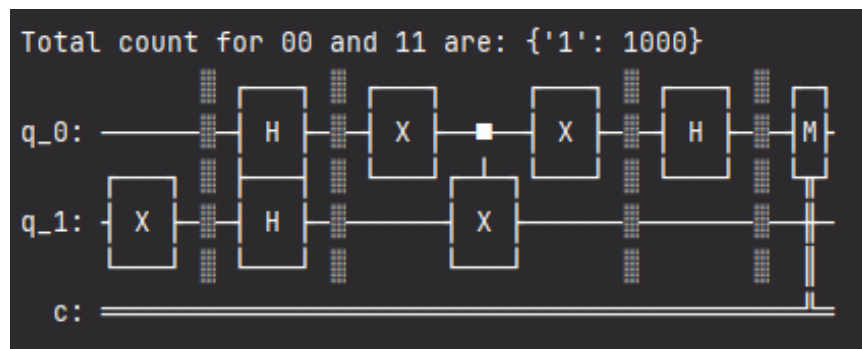


## 2.4 IMPLEMENTANDO EL ALGORITMO DE DEUTSCH EN UN COMPUTADOR CUÁNTICO

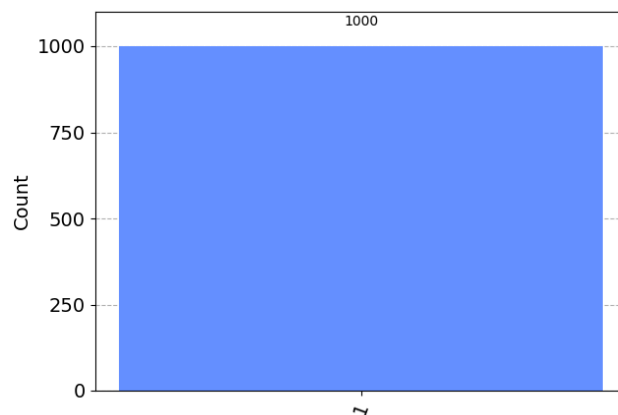
### 2.4.1 PRIMERA FUNCIÓN (cruzados)



#### Circuito

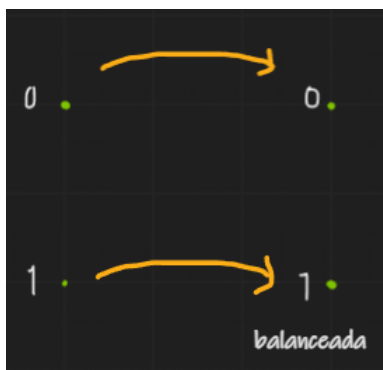


#### Resultados

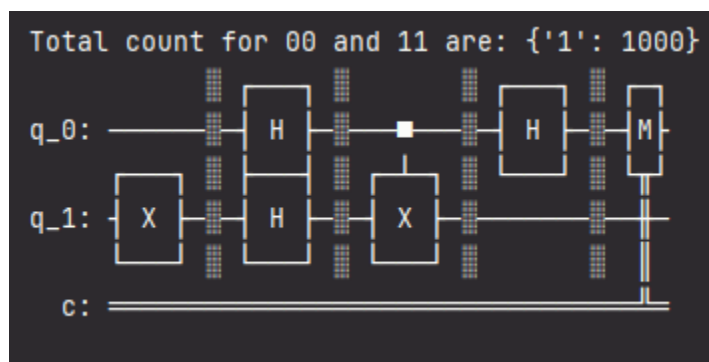


El 100% de las veces el resultado es  $|1\rangle$ , por lo que la función es **BALANCEADA**,  $f(0)!$   
 $= f(1)$ .

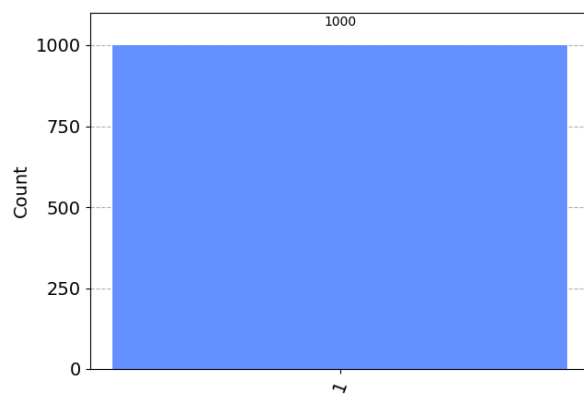
## 2.4.2 SEGUNDA FUNCIÓN (iguales)



### Circuito



### Resultados

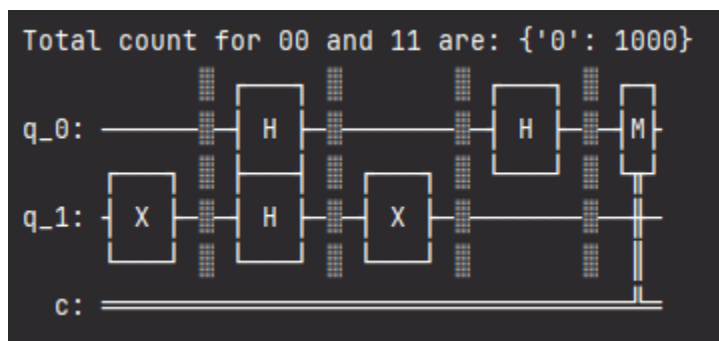


El 100% de las veces el resultado es  $|1\rangle$ , por lo que la función es **BALANCEADA**,  $f(0)!$   
 $= f(1)$

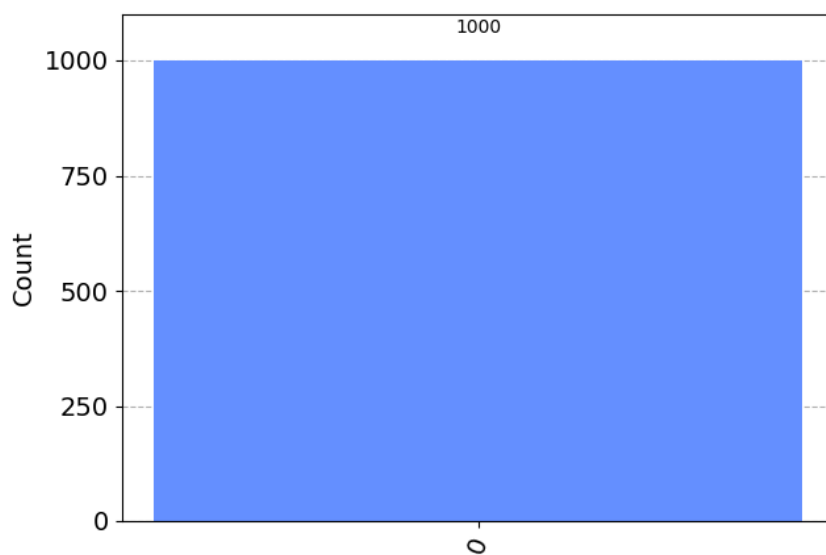
### 2.4.3 TERCERA FUNCIÓN (a uno)



#### Circuito



#### Resultados

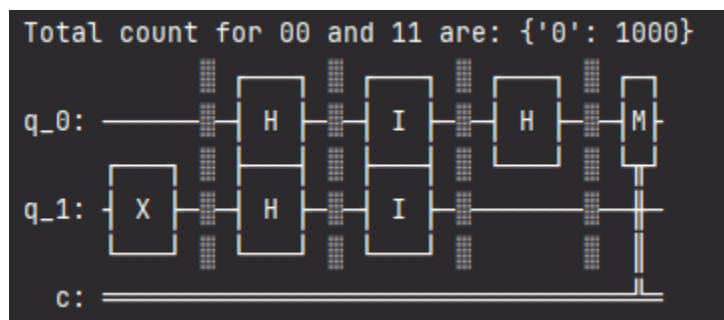


El 100% de las veces el resultado es  $|0\rangle$ , por lo que la función es **CONSTANTE**,  $f(0) = f(1)$ .

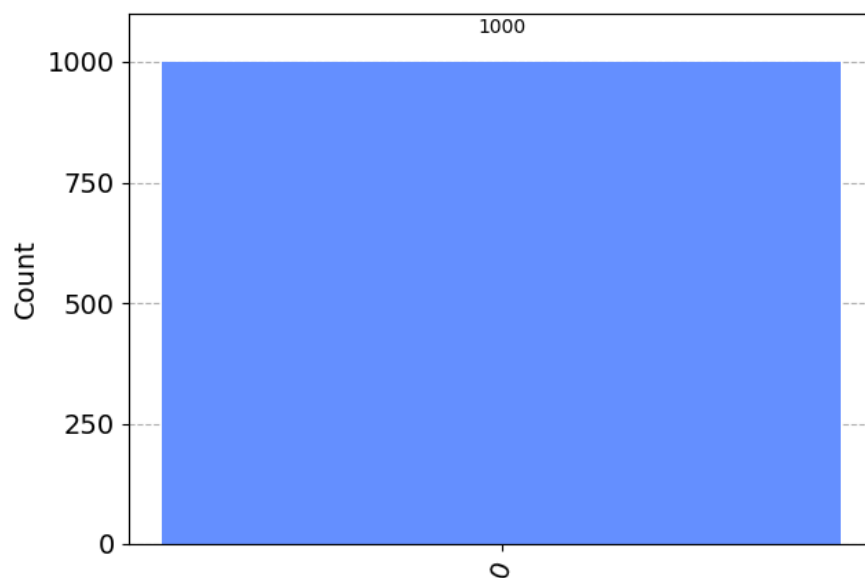
## 2.4.4 CUARTA FUNCIÓN (a cero)



### Circuito



### Resultados



El 100% de las veces el resultado es  $|0\rangle$ , por lo que la función es **CONSTANTE**,  $f(0) = f$

(1).

### 3 Algoritmo de Deutsch-Jozsa

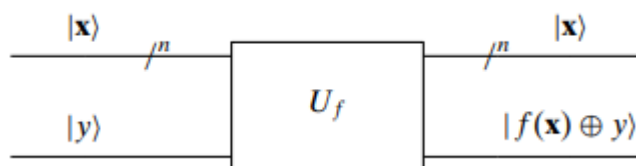
[1] “En el circuito que implementa el algoritmo Deutsch-Jozsa se define el estado inicial como sigue: simulación de sistemas cuánticos y resolución de sistemas de ecuaciones lineales (Montanaro, 2016). Entre esos problemas existe uno que consiste en lo siguiente: dada una función binaria, y asumiendo que la misma puede ser únicamente constante o balanceada, distinguir cada caso. Un computador clásico debe evaluar la función  $2^{n-1} + 1$  veces, para una función con  $n$  bits de entrada, con el fin de llevar a cabo esta tarea.

Se presenta el algoritmo de Deutsch-Jozsa para tener una visión del poder de cómputo que tiene la computación cuántica, este poder de cómputo se basa en el paralelismo, el cual se caracteriza por el crecimiento exponencial, en relación con el número de cubits, de la capacidad de realizar operaciones de manera simultánea.”

#### 3.1 Problema

Este algoritmo nos dice si una función  $f$ , tal que  $f: \{0,1\} \rightarrow \{0,1\}$ , es **balanceada** o **constante**. Es **constante** si la salida es  $0^n$  y **balanceada** en cualquier otro caso.

*Algoritmo*

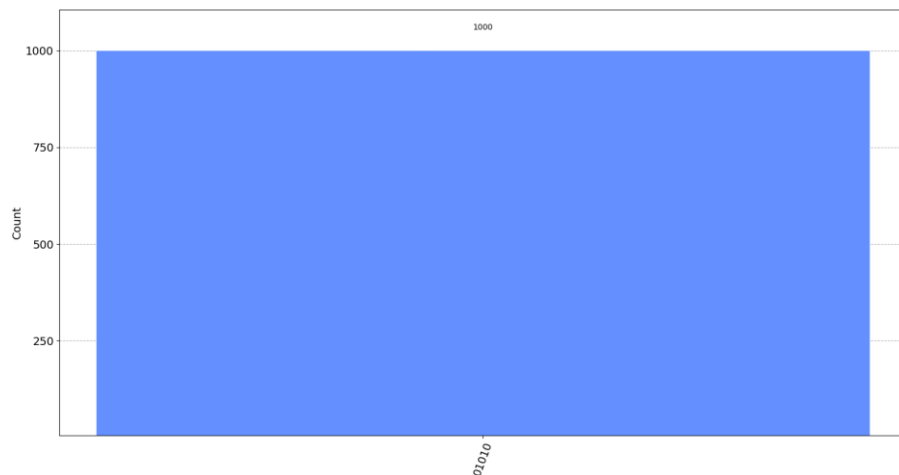
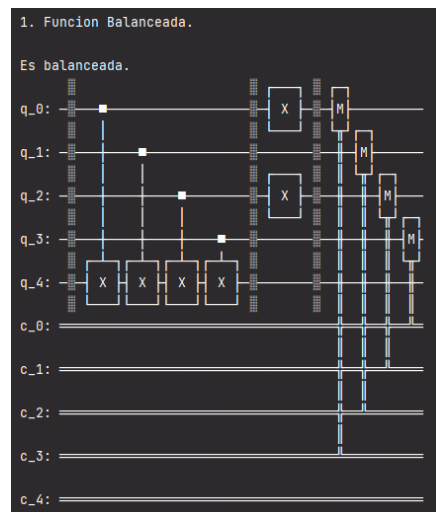


## 3.2 IMPLEMENTANDO LAS FUNCIONES EN EL COMPUTADOR CUÁNTICO

Se hacen 4 funciones con  $n = 4$  para comprobar el funcionamiento del algoritmo.

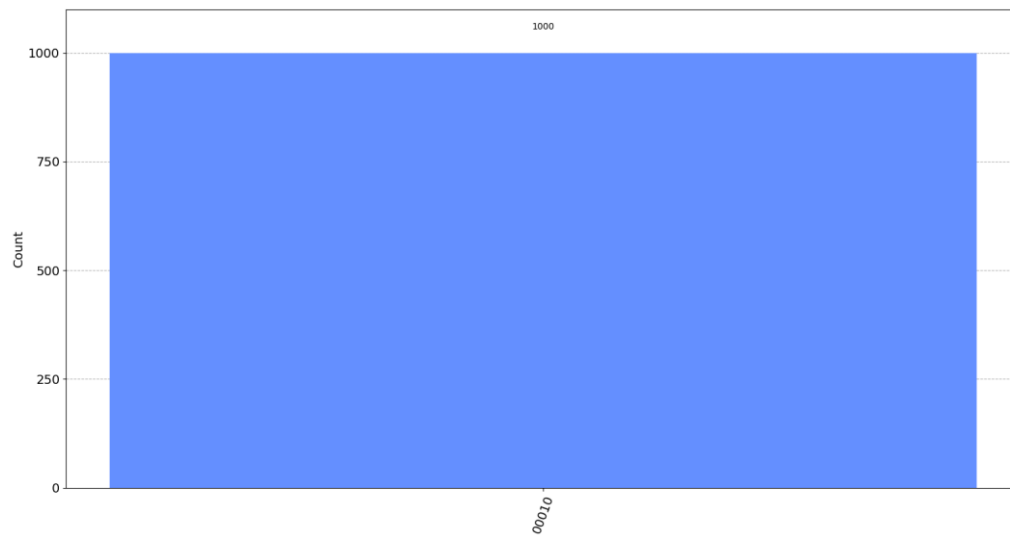
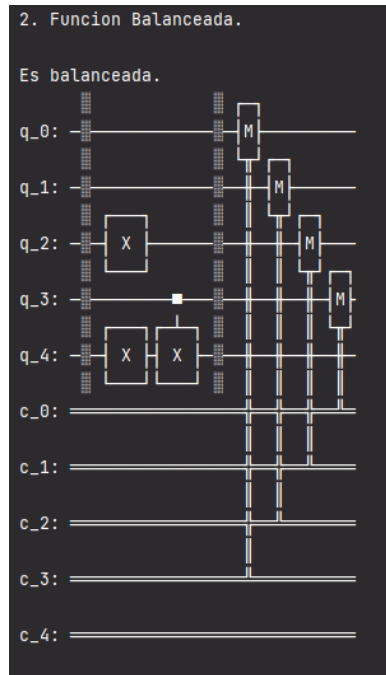
### 3.2.1 PRIMERA FUNCIÓN BALANCEADA

a) Circuito correspondiente:

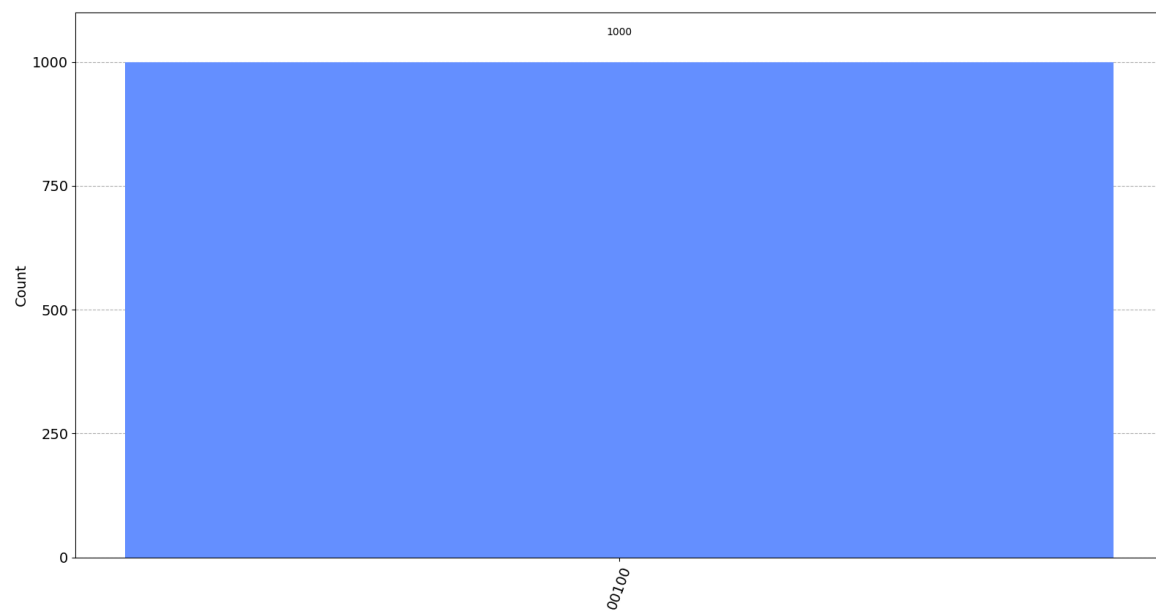
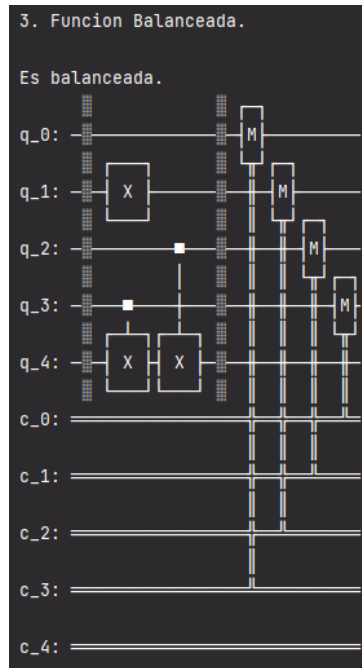


### 3.2.2 SEGUNDA FUNCIÓN BALANCEADA

a) Circuito correspondiente

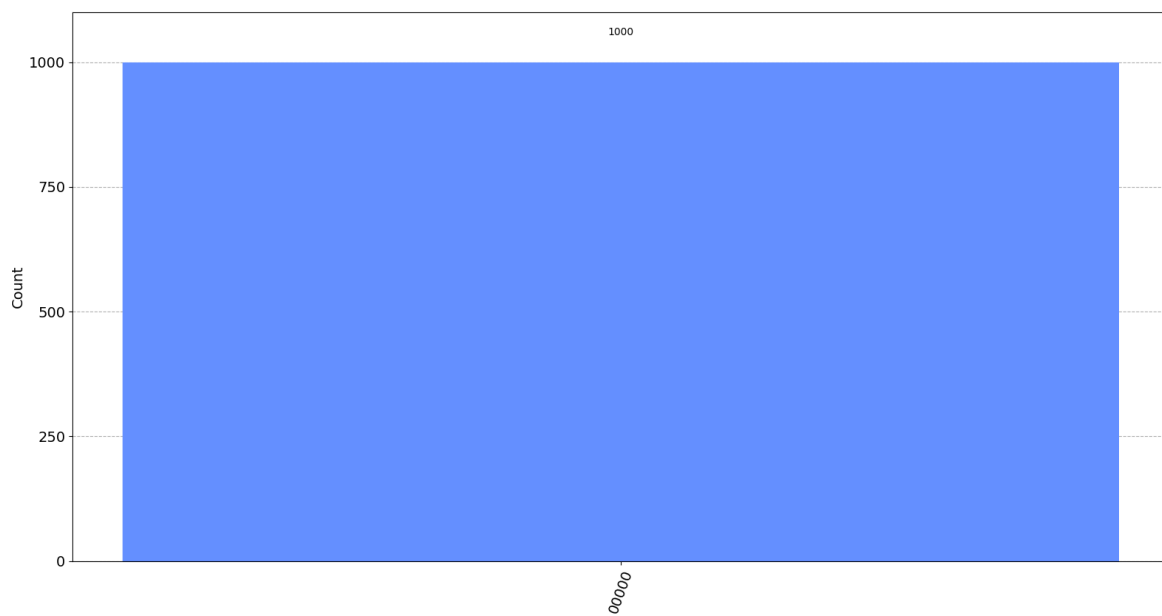
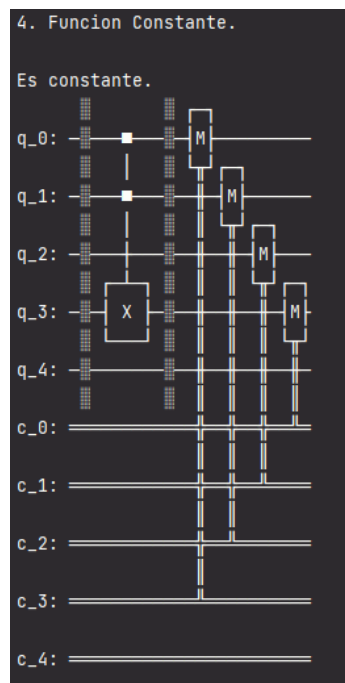


### 3.2.3 TERCERA FUNCIÓN BALANCEADA





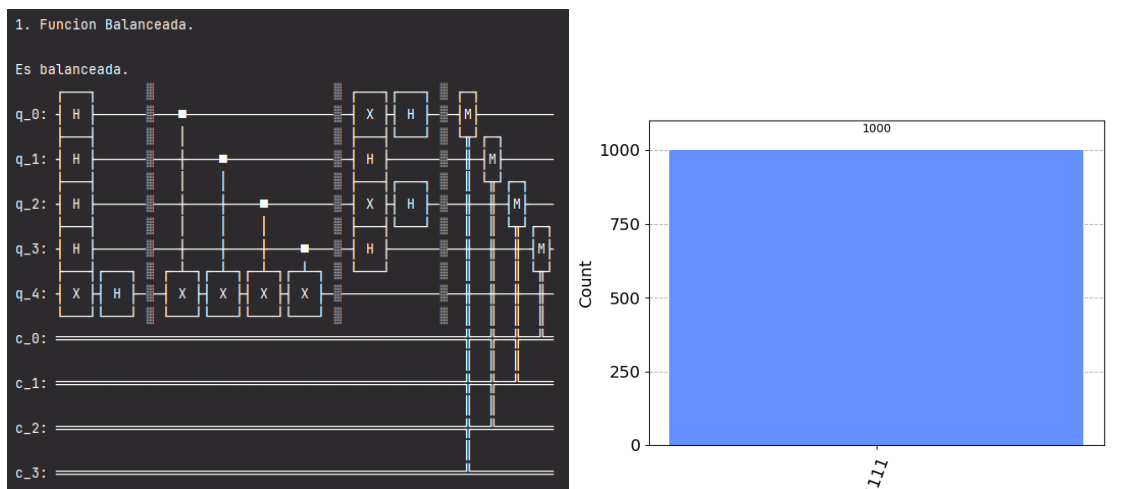
### 3.2.4 FUNCIÓN CONSTANTE



### 3.3 IMPLEMENTANDO EL ALGORITMO DE DEUTSCH-JOSZA EN UN COMPUTADOR CUÁNTICO

#### 3.3.1 PRIMERA FUNCIÓN BALANCEADA

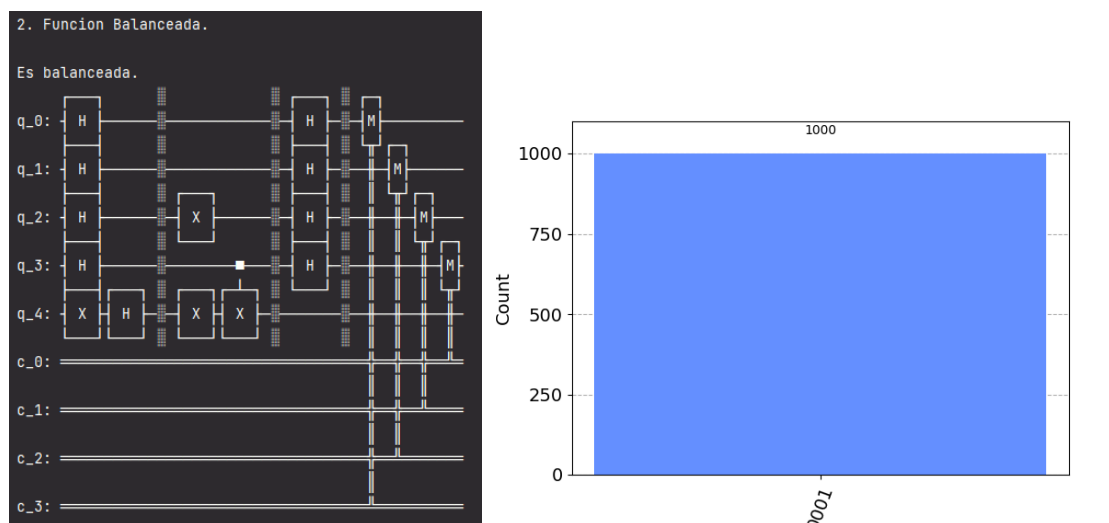
Circuito Deutsch-Jozsa correspondiente:



Gráfica resultante indica que el 100 % de las veces el valor es diferente de cero, por lo tanto, podemos concluir que es una función balanceada.

#### 3.3.2 SEGUNDA FUNCIÓN BALANCEADA

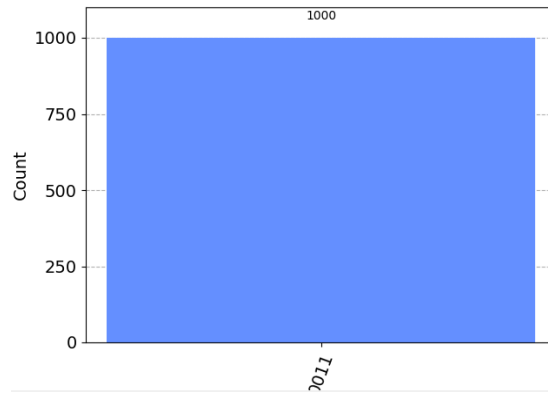
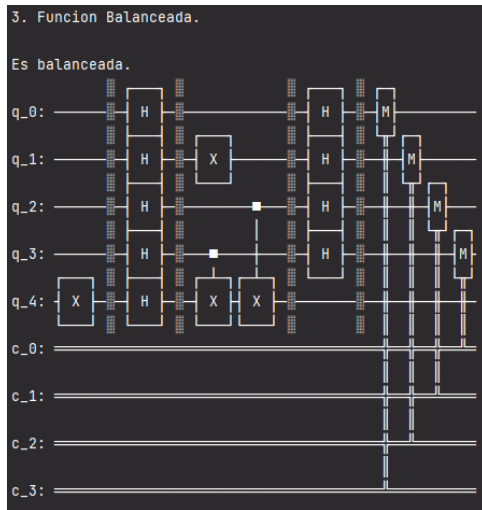
Circuito Deutsch-Jozsa correspondiente:



Gráfica resultante indica que el 100 % de las veces el valor es diferente de cero, por lo tanto, podemos concluir que es una función balanceada.

### 3.3.3 TERCERA FUNCIÓN BALANCEADA

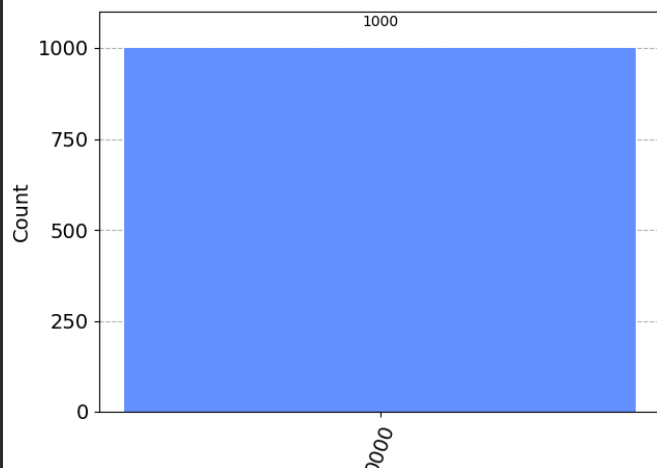
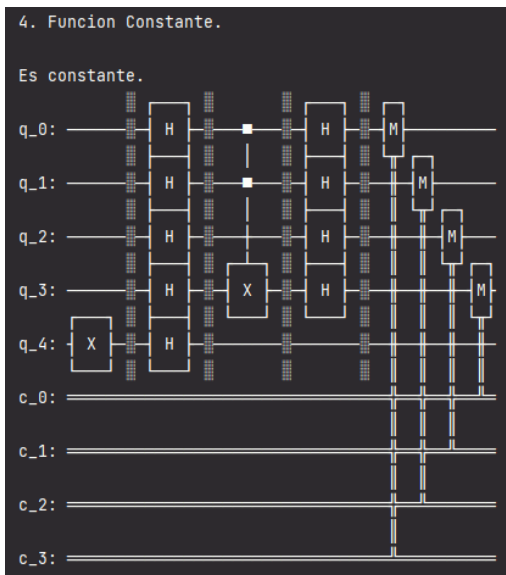
Circuito Deutsch-Jozsa correspondiente:



Gráfica resultante indica que el 100 % de las veces el valor es diferente de cero, por lo tanto, podemos concluir que es una función balanceada.

### 3.3.4 FUNCIÓN CONSTANTE

Circuito Deutsch-Jozsa correspondiente:



Gráfica resultante indica que el 100 % de las veces el valor es diferente de uno, por lo tanto, podemos concluir que es una función constante.

## 4. Conclusiones

En este reporte pusimos a prueba la computación cuántica con el objetivo final de aprender y experimentar el potencial de la tecnología cuántica. En primer lugar, se mostraron las funciones binarias que nos ayudan a entender cómo funciona la herramienta qiskit, luego se introdujo el concepto del algoritmo Deutsch el cual se probó y validó haciendo uso de gráficos, tablas y de código. Estas funciones se utilizaron en el algoritmo Deutsch-Jozsa para determinar si eran funciones constantes o balanceadas. Por último, el desarrollo de este reporte me ayudó a comprender cómo la ciencia cuántica puede ayudarnos a resolver problemas en diferentes áreas que las computadoras clásicas aún no pueden resolver.

## 5. Bibliografía

- [1] Franco, R. (2021). Deutsch-Jozsa Algorithm: a look at the power of quantum computing. *Reportes Científicos de La FACEN*, 12(2), 83–87.  
<https://doi.org/10.18004/rcfacen.2021.12.2.83>
- [2] Francisco, D., Iemmi, C., & Ledesma, S. (2006). *SIMULACIÓN ÓPTICA DEL ALGORITMO CUÁNTICO DE DEUTSCH*. ResearchGate; unknown.  
[https://www.researchgate.net/publication/237641214\\_SIMULACION\\_OPTICA\\_DEL\\_ALGORITMO\\_CUANTICO\\_DE\\_DEUTSCH](https://www.researchgate.net/publication/237641214_SIMULACION_OPTICA_DEL_ALGORITMO_CUANTICO_DE_DEUTSCH)
- [3] *Introducción a la computación cuántica: qubits, superposición y más*. (2021, August 22). ICHI.PRO. <https://ichi.pro/es/introduccion-a-la-computacion-cuantica-qubits-superposicion-y-mas-43440270856698>
- [4] *Algoritmo de Deutsch ~ Conceptos matemáticos básicos de Computación Cuántica*. (2020). Docirs.cl.  
[https://www.docirs.cl/algoritmo\\_de\\_deutsch.asp](https://www.docirs.cl/algoritmo_de_deutsch.asp)

