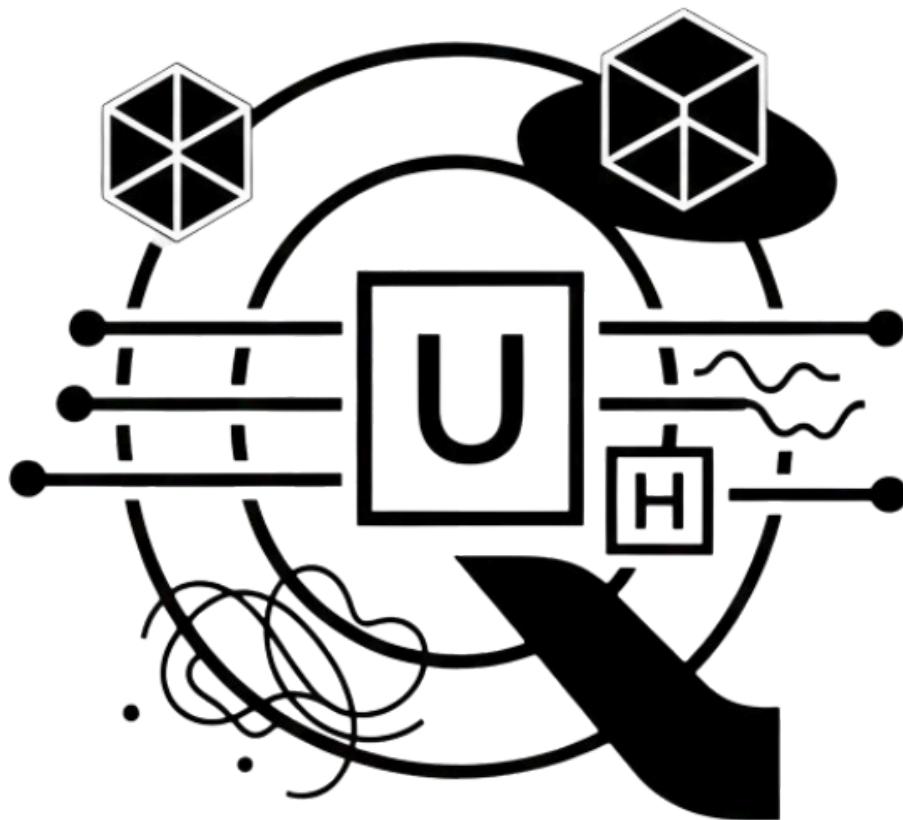


2025

Fundamentos de la Computación Cuántica

Segunda Entrega de Cursada



Alumno

Roumec, Iñaki: iroumec@alumnos.exa.unicen.edu.ar.

Profesores

Leiva, Lucas: lleiva@labset.exa.unicen.edu.ar.

Vázquez, Martín: martin.o.vazquez@gmail.com.

Introducción.....	3
Resolución de los Incisos.....	4
A) Full-Adder con Operandos de 4 Qubits.....	4
Full-Adder.....	4
Full-Adder-4.....	6
B) Deutsch-Jozsa - Función Constante Igual a 1.....	10
C) Deutsch-Jozsa - Función Balanceada.....	13
D) Deutsch-Jozsa - Función No Balanceada Ni Constante.....	15
Conclusiones.....	17

Introducción

A lo largo de este informe, se presentan los resultados correspondientes a los ejercicios solicitados por la cátedra de la materia optativa Fundamentos de la Computación Cuántica como parte de la segunda entrega de la cursada. Estos son:

- a. Ejercicio 10 del TP de Operaciones (Parte I).
- b. Ejemplo 1 del ejercicio 1 del TP de Paralelismo Cuántico (Deutsch-Jozsa, función constante igual a 1).
- c. Ejemplo 3 del ejercicio 1 del TP de Paralelismo Cuántico (Deutsch-Jozsa, función balanceada).
- d. Ejemplo 4 del ejercicio 1 del TP de Paralelismo Cuántico (Deutsch-Jozsa, función no balanceada ni constante).

Para ello, se utilizó el *framework* Qiskit, junto al lenguaje de programación Python.

El código fuente correspondiente a la entrega, junto a sus pasos de ejecución, puede hallarse en la sección de *releases* en el siguiente repositorio de GitHub: <https://github.com/iroumec/Quantum-Computing/releases>.

Aclaración: los resultados de los histogramas presentados en el informe pueden variar levemente entre las ejecuciones del código, debido a la naturaleza de las simulaciones. No obstante, estas variaciones no invalidan de ninguna manera las conclusiones presentadas en este informe.

Resolución de los Incisos

A) Full-Adder con Operandos de 4 Qubits

Este inciso corresponde al ejercicio 10 del TP de operaciones (parte I), cuya consigna es la siguiente:

Basándose en el ejercicio 9, implemente en el simulador un sumador completo entre dos operandos de tamaño 4 qubits. Utilice subcircuitos.

Por consiguiente, para resolverlo, debió resolverse primero el ejercicio 9 en Qiskit, cuya consigna se presenta a continuación:

Implemente un sumador completo con operandos de un qubit (enfoque clásico) basándose en computación reversible clásica, mediante el uso de compuertas C-NOT y Toffoli (CC-NOT).

Full-Adder

A continuación, se presenta el circuito planteado en Qiskit para el FA:

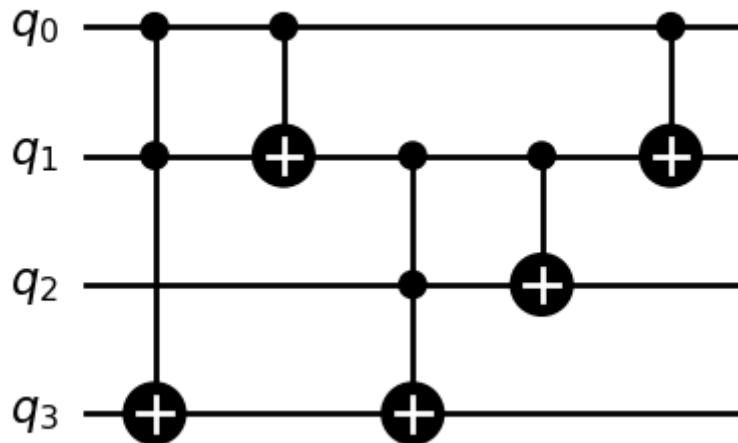


Imagen 1: circuito del FA.

Para corroborar su funcionamiento, se le añadieron medidores y se colocaron en superposición las entradas, de forma que luego puedan ser corroborados sus resultados, para todas las combinaciones, mediante una simulación:

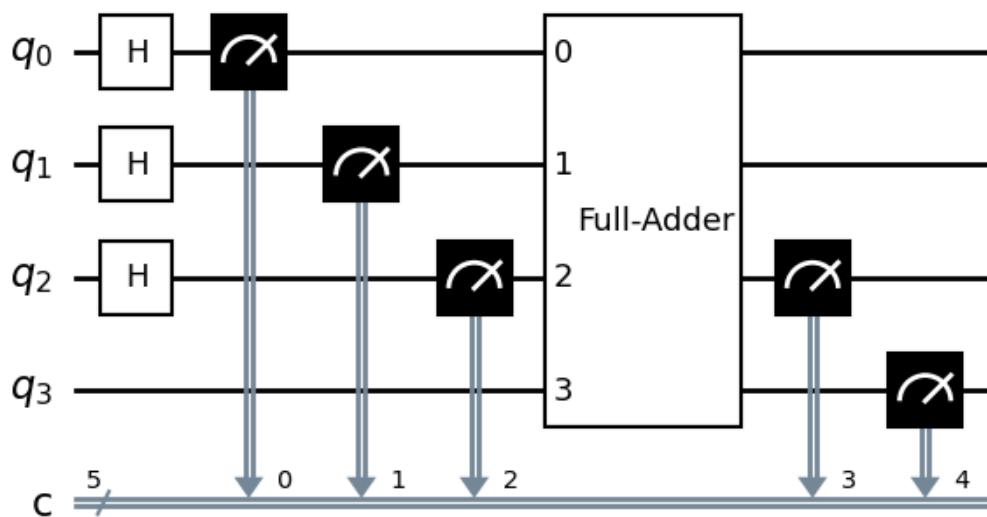


Imagen 2: circuito del FA con mediciones añadidas.

Con las mediciones añadidas, se realizó una simulación de 1024 *shots*. A continuación, se presentan los resultados obtenidos mediante un histograma:

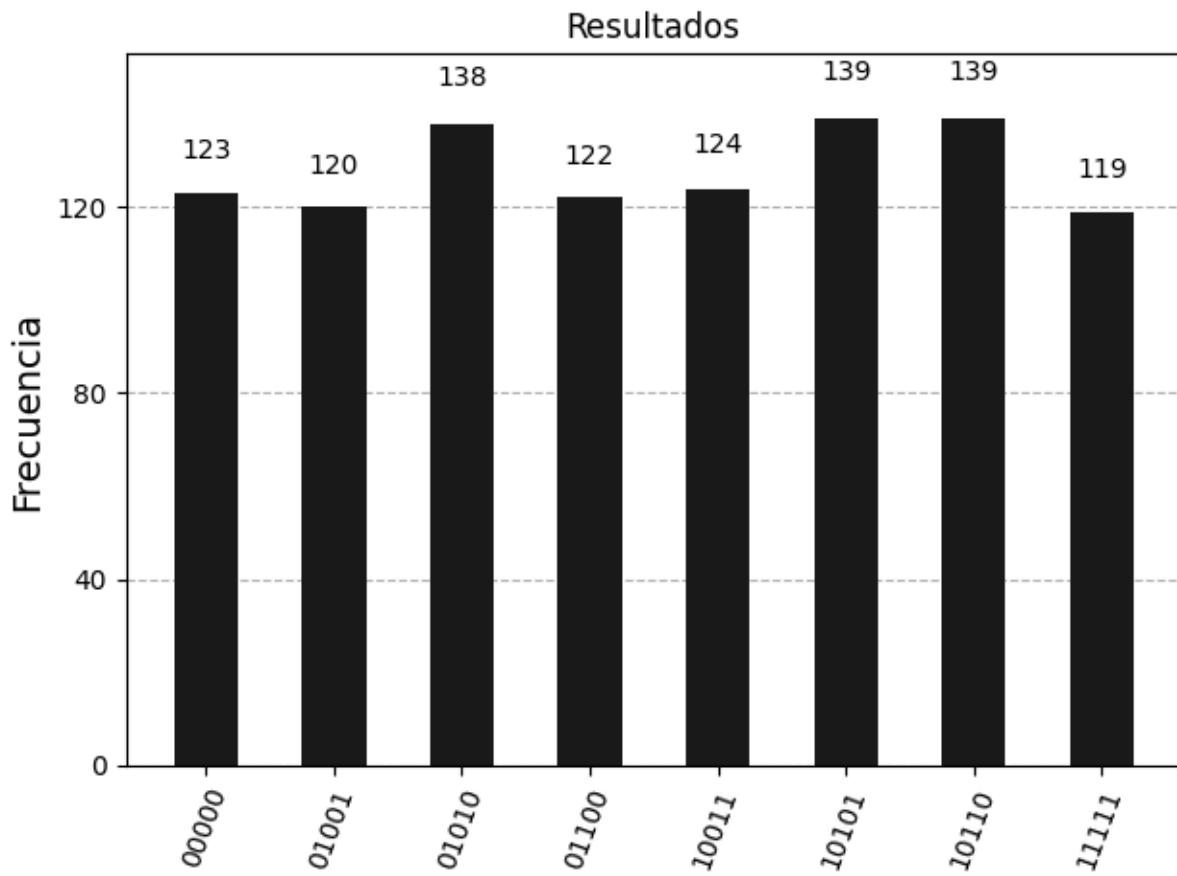


Imagen 3: histograma correspondiente a la simulación del circuito del FA. La correspondencia, en orden, del bitstring es: cout-s-cin-b-a.

Como puede observarse, un histograma no es la mejor forma de mostrar el correcto funcionamiento del circuito. Este se adjunta ya que forma parte del enunciado de la entrega. Sin embargo, se consideró más apropiado para este inciso mostrar los resultados mediante una tabla. Esta no solo facilita la lectura del *bitstring* devuelto por el simulador, sino que, además, indica si el resultado obtenido es correcto, calculando, para ello, con los operadores de bits de Python, la salida esperada y comparándola con la obtenida:

Tabla de Verdad - Full-Adder							
A	B	Cin	S	Cout	Frecuencia	¿Correcto?	
0	1	1	0	1	128	SÍ	
0	1	0	1	0	126	SÍ	
1	1	0	0	1	132	SÍ	
0	0	1	1	0	123	SÍ	
1	0	0	1	0	124	SÍ	
1	1	1	1	1	117	SÍ	
0	0	0	0	0	115	SÍ	
1	0	1	0	1	159	SÍ	

Imagen 4: tabla correspondiente a la simulación del circuito del FA.

Como puede observarse en la tabla, el funcionamiento del FA es el esperado y, por lo tanto, puede proseguirse con su conversión en un subcircuito para implementar así el FA4.

Full-Adder-4

A continuación se plantea el circuito correspondiente al FA4, utilizando como subcircuito el FA creado anteriormente. Cabe aclarar que la implementación planteada del circuito difiere con la realizada en el anterior entrega, la cual utilizaba *swaps* para no alterar los resultados de los operandos. Este cambio fue realizado considerando la recomendación de los profesores.

Previo a la implementación, se indica la interpretación tomada de los qubits. Esta se tomó de la siguiente manera por considerarse la forma más intuitiva:

- q_0, q_1, q_2 y q_3 representan a A_0, A_1, A_2 y A_3 , respectivamente.
- q_4, q_5, q_6 y q_7 representan a B_0, B_1, B_2 y B_3 , respectivamente.
- q_8 representa a *Cin* al comienzo del circuito y, a *S0* luego del primer FA.
- q_9 es $|0\rangle$ al comienzo del circuito y representa a *Cout0* al final del primer FA y, a *S1* al final del segundo FA.
- q_{10} es $|0\rangle$ al comienzo del circuito y representa a *Cout1* al final del segundo FA y, a *S2* al final del tercer FA.
- q_{11} es $|0\rangle$ al comienzo del circuito y representa a *Cout2* al final del tercer FA y, a *S3* al final del cuarto FA.

- q_{12} es $|0\rangle$ al comienzo del circuito y representa a $Cout3$ (acarreo final) al final del cuarto FA.

Con lo anterior en cuenta, se realizó el planteo del circuito:

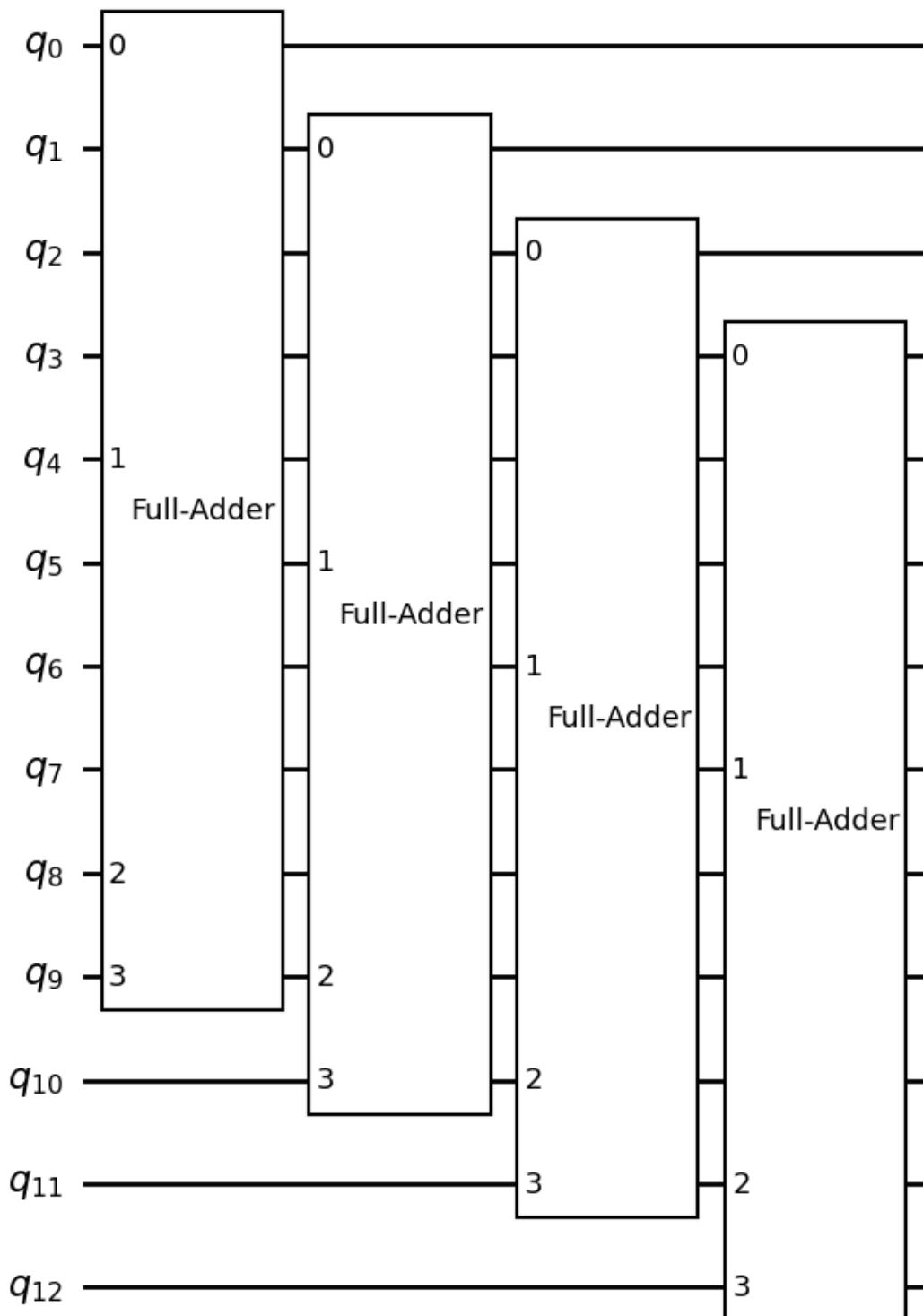


Imagen 5: circuito del FA4 usando subcircuitos FA.

Posteriormente, se añadieron al circuito las mediciones correspondientes para realizar la simulación. Se utilizaron 14 bits clásicos, permitiendo así medir cada una de las entradas y cada una de las salidas (el bit adicional, con respecto al número de qubits, 13, se debe a que, en q_8 , es igual de importante medir Cin como $S0$).

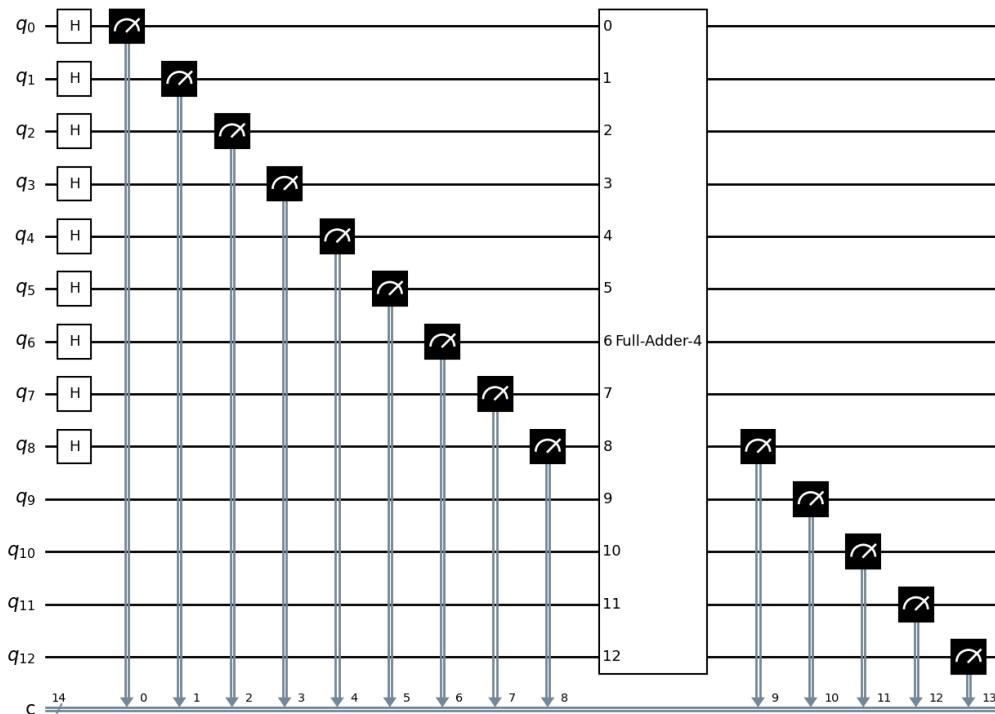


Imagen 6: circuito del FA4 con mediciones añadidas.

Con las mediciones colocadas, se realizó una simulación de 10240 *shots*. A continuación, se presentan los resultados mediante un histograma:

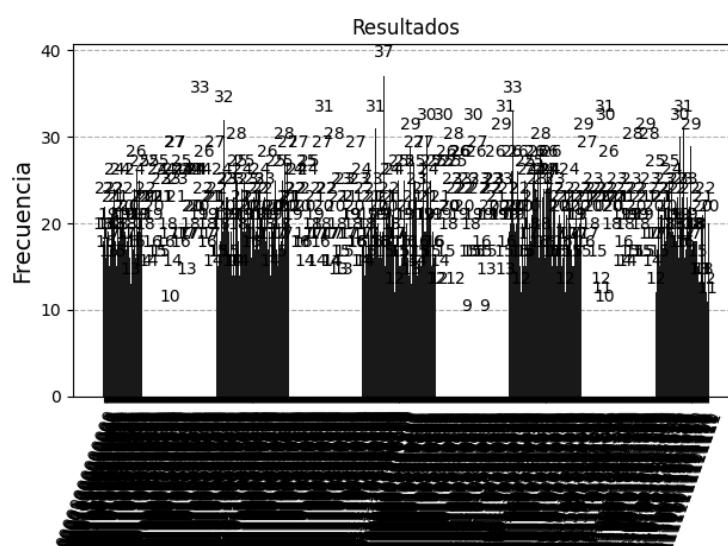


Imagen 7: histograma correspondiente a la simulación del circuito del FA4.

Similar al caso del FA, **un histograma no es apropiado para mostrar los resultados de la simulación**, mucho menos dada la gran cantidad de posibilidades que abarca un FA4. Se intentó ajustar sus parámetros, con la intención de hacerlo más legible, sin embargo, resultó una tarea imposible. Además, la lectura de los *bitstring* no resultaría sencilla. Por consiguiente, aunque resulte totalmente ilegible, este se incluye para cumplir con las pautas del informe.

Dada la ilegibilidad del histograma, la comprobación de que el algoritmo funciona correctamente se presenta en formato de tabla. Esta última, en cada fila, muestra cada una de las posibles combinaciones y sus resultados. Adicionalmente, en la última columna, para cada combinación, se calculó su valor esperado y se corroboró con el valor obtenido en la simulación.

Tabla de Verdad - Sumador de 4 qubits							
A3...A0	B3...B0	Cin	S3...S0	Cout	Frecuencia	¿Correcto?	
0 0 0 0	0 0 0 0	0	0 0 0 0	0	17	SÍ	
0 0 0 1	0 0 0 0	0	0 0 0 1	0	20	SÍ	
0 0 0 0	0 0 0 1	0	0 0 0 1	0	13	SÍ	
0 0 0 0	0 0 0 0	1	0 0 0 1	0	24	SÍ	
0 0 1 0	0 0 0 0	0	0 0 1 0	0	19	SÍ	
0 0 0 1	0 0 0 1	0	0 0 1 0	0	15	SÍ	
0 0 0 0	0 0 1 0	0	0 0 1 0	0	23	SÍ	
0 0 0 1	0 0 0 0	1	0 0 1 0	0	18	SÍ	
0 0 0 0	0 0 0 1	1	0 0 1 0	0	8	SÍ	
0 0 1 1	0 0 0 0	0	0 0 1 1	0	20	SÍ	
0 0 1 0	0 0 0 1	0	0 0 1 1	0	20	SÍ	
0 0 0 1	0 0 1 0	0	0 0 1 1	0	15	SÍ	
0 0 0 0	0 0 1 1	0	0 0 1 1	0	16	SÍ	
0 0 1 0	0 0 0 0	1	0 0 1 1	0	20	SÍ	
0 0 0 1	0 0 0 1	1	0 0 1 1	0	20	SÍ	
0 0 0 0	0 0 1 0	1	0 0 1 1	0	21	SÍ	
0 1 0 0	0 0 0 0	0	0 1 0 0	0	18	SÍ	
0 0 1 1	0 0 0 1	0	0 1 0 0	0	17	SÍ	
0 0 1 0	0 0 1 0	0	0 1 0 0	0	21	SÍ	
0 0 0 1	0 0 1 1	0	0 1 0 0	0	24	SÍ	
...							
Total de mediciones: 10240							
Configuraciones únicas: 512							
Precisión: 100.00%							

Imagen 8: tabla acortada correspondiente a la simulación del circuito del FA4.

La tabla presenta solo unas pocas combinaciones. Esto, dado que el código se incluye con la entrega, con el objetivo de no extender innecesariamente el informe: colocar la tabla completa ocuparía más de cinco hojas. Si desea visualizarla completa, puede ejecutar la *notebook*, accediendo al código adjunto al comienzo del informe.

Como puede observarse en las estadísticas de la tabla, la precisión del circuito es del 100 %, lo que indica que el FA4 fue implementado correctamente.

B) Deutsch-Jozsa - Función Constante Igual a 1

Este inciso corresponde al ejemplo 1 del ejercicio 1 del TP de paralelismo, el cual se muestra a continuación:

Función constante

Ejemplo 1: $f(x_2, x_1, x_0) = 1$

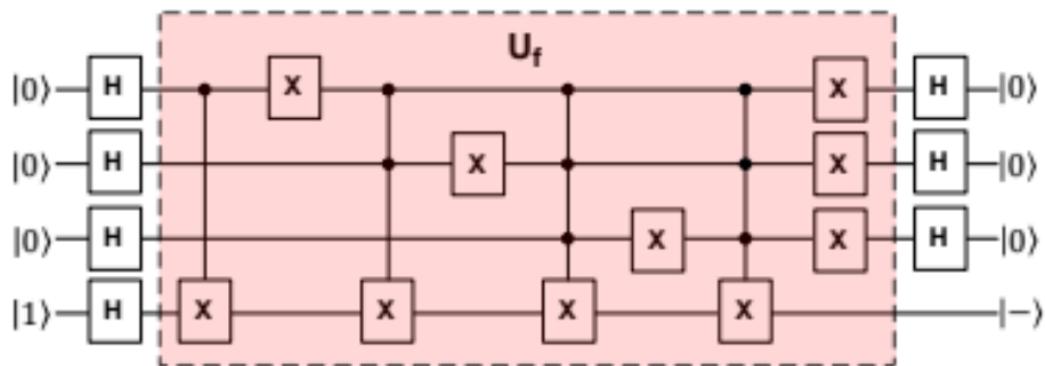


Imagen 9: ejemplo 1 del ejercicio 1 del TP de paralelismo cuántico.

Como primer paso, se construyó el circuito de Deutsch-Jozsa. Este se planteó de forma que sea reutilizable, ya que los demás incisos también lo requieren.



Imagen 10: circuito correspondiente al algoritmo de Deutsch-Jozsa para funciones de 3 entradas. La primera barrera solo tiene el objetivo de evitar el reordenamiento automático que realiza Qiskit. Negar el qubit auxiliar es necesario para inicializarlo en $|1\rangle$, ya que en Qiskit, por defecto, todos los qubits comienzan en $|0\rangle$.

Con el circuito de Deutsch-Jozsa planteado, se construyó el circuito de la función (oráculo en el contexto del algoritmo) del enunciado:

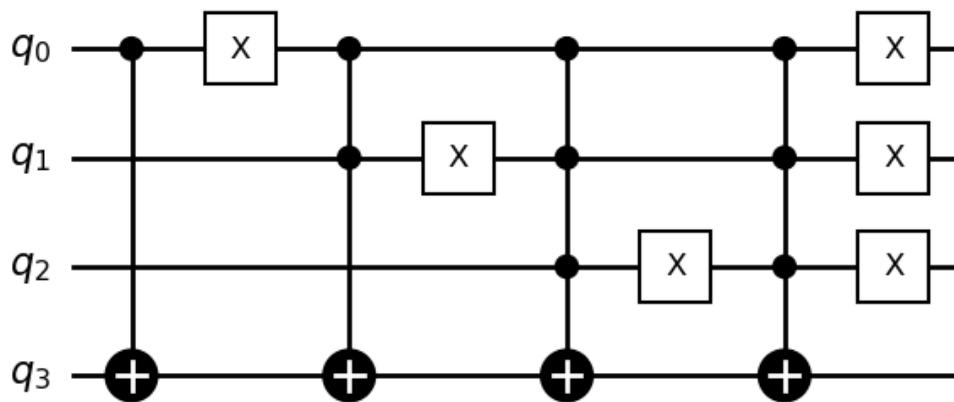


Imagen 11: circuito de la función del ejemplo 1 del ejercicio 1 del TP de paralelismo cuántico.

Aplicando el algoritmo de Deutsch-Jozsa al oráculo, se obtuvo el siguiente circuito:

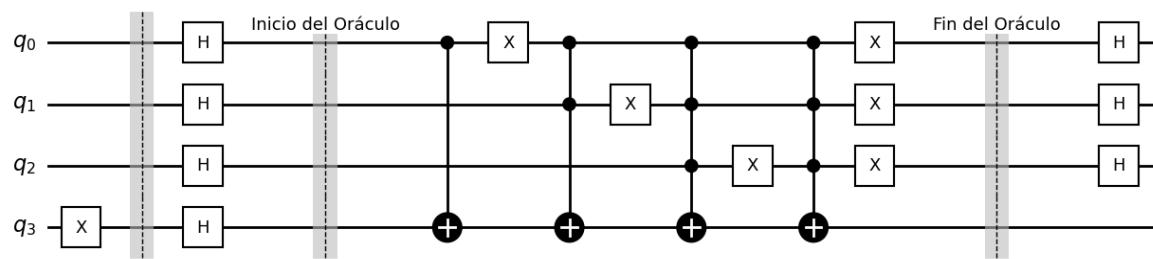


Imagen 12: aplicación del algoritmo de Deutsch-Jozsa al oráculo definido anteriormente.

Se le añadieron las mediciones correspondientes para realizar la simulación. En Deutsch-Jozsa, no se debe medir el qubit auxiliar, ya que no aporta información sobre si la función es constante o balanceada. Solo importan los qubits de entrada. El circuito resultante es el siguiente:

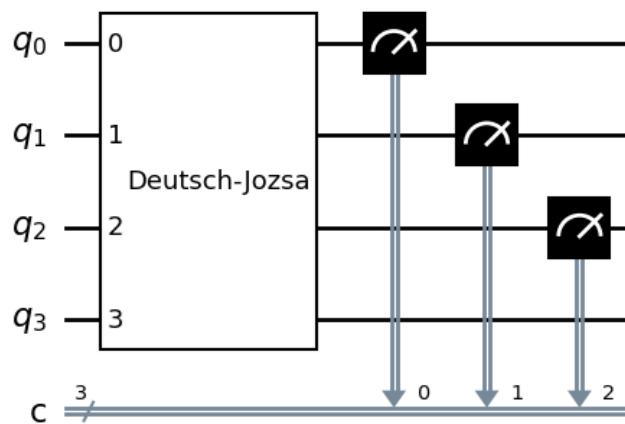


Imagen 13: aplicación de las mediciones al circuito anterior.

Posteriormente, con las mediciones ya colocadas, se realizó una simulación de 1024 *shots*. Los resultados obtenidos se presentan en formato de histograma a continuación:

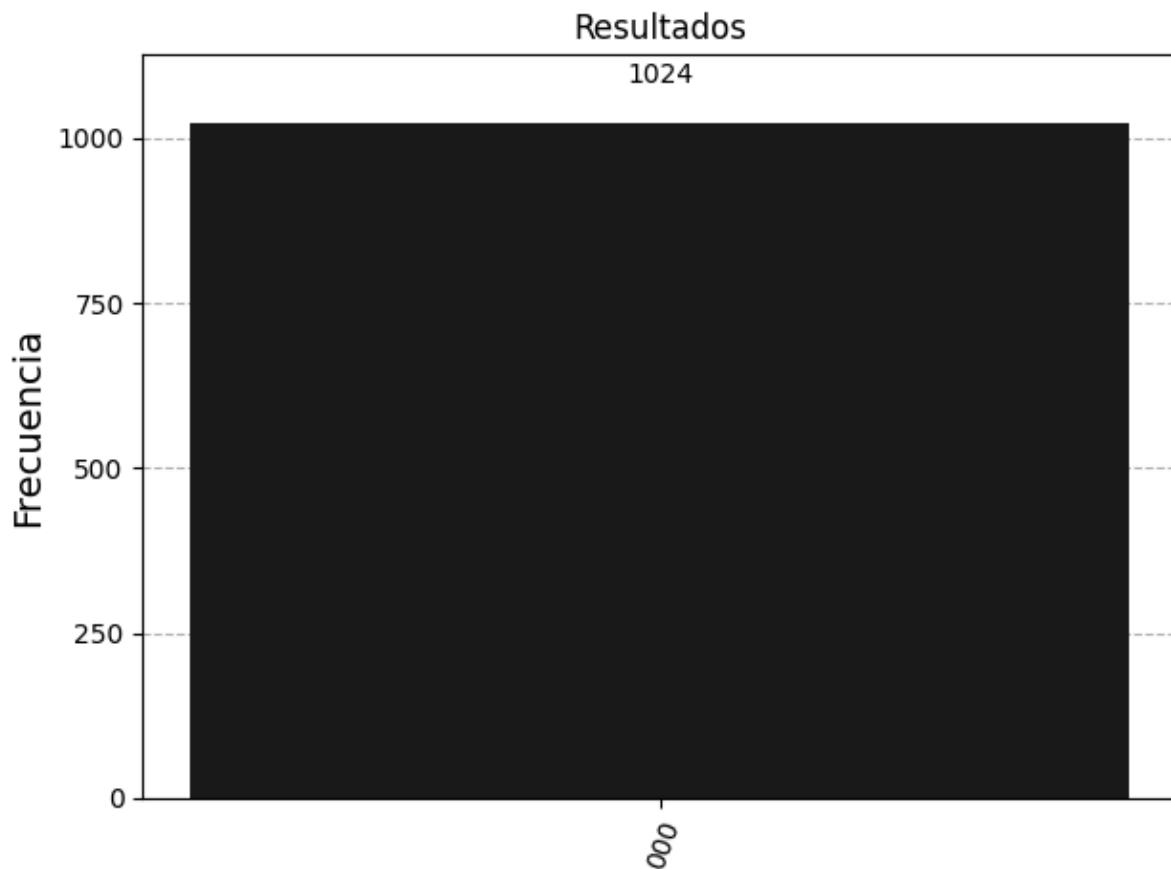


Imagen 14: histograma correspondiente a la simulación del circuito anterior.

El algoritmo de Deutsch-Jozsa establece que:

- Si la función es constante, toda la probabilidad colapsa en un solo estado básico: el $|000\rangle$.
- Por otro lado, si la función es balanceada, la probabilidad se reparte entre cualquier combinación de estados básicos **exceptuando** el $|000\rangle$.

Por consiguiente, como todas las mediciones colapsan únicamente en el estado básico $|000\rangle$, queda demostrado que la función es constante.

C) Deutsch-Jozsa - Función Balanceada

Este inciso corresponde al ejemplo 3 del ejercicio 1 del TP de paralelismo, el cual se muestra a continuación:

Función balanceada

Ejemplo 3: $f(x_2, x_1, x_0) = x_2\bar{x}_1 + x_1\bar{x}_0$

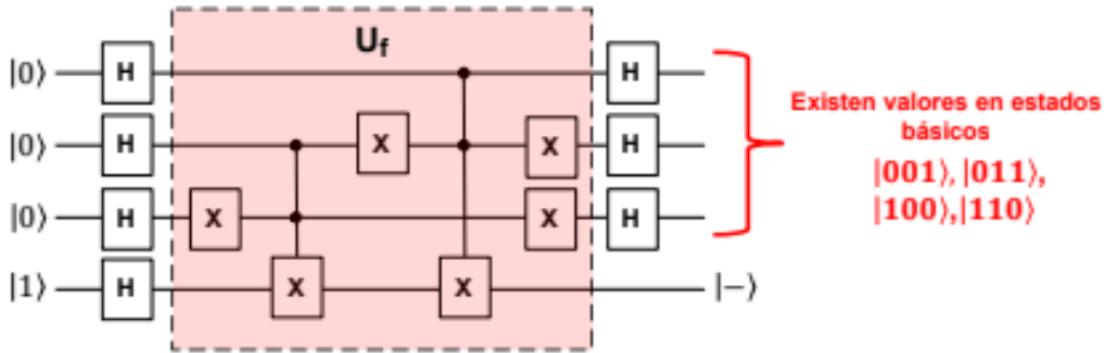


Imagen 15: ejemplo 3 del ejercicio 1 del TP de paralelismo cuántico.

Debido a que previamente ya se planteó el algoritmo de Deutsch-Jozsa, solo se requirió plantear el oráculo de la función:

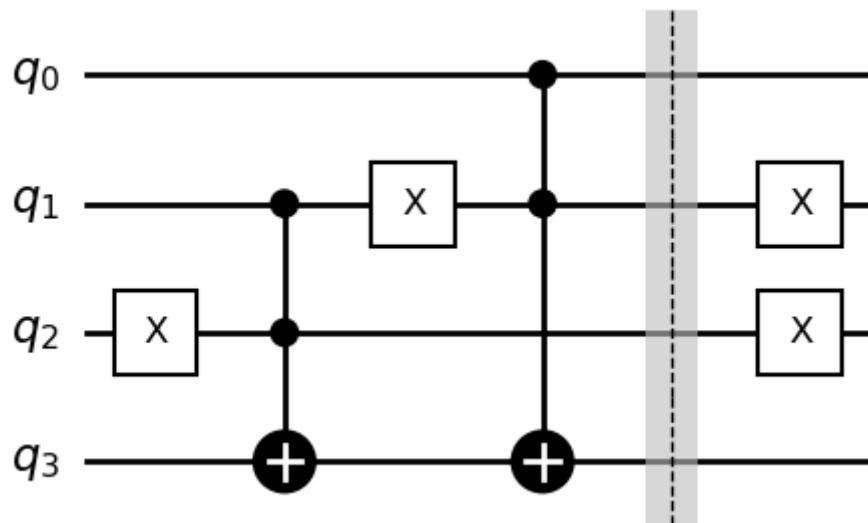


Imagen 16: circuito de la función del ejemplo 3 del ejercicio 1 del TP de paralelismo cuántico. La barrera fue colocada para evitar el reordenamiento automático realizado por Qiskit.

Aplicando el algoritmo de Deutsch-Jozsa al oráculo planteado, se obtuvo el siguiente circuito:

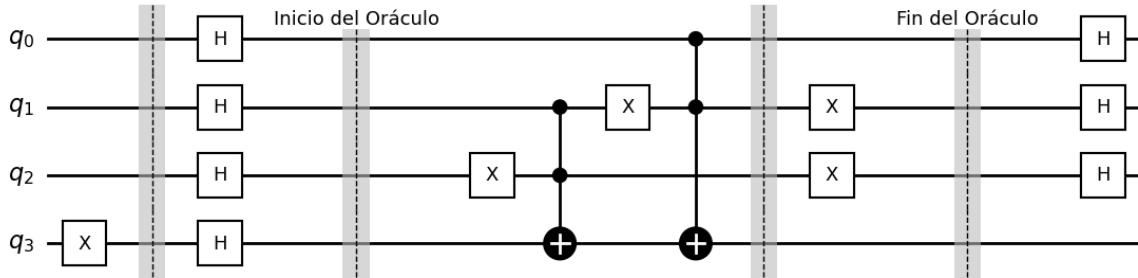


Imagen 17: aplicación del algoritmo de Deutsch-Jozsa al oráculo definido anteriormente.

Luego, se añadieron las mediciones correspondientes para realizar la simulación (el circuito resultante, debido al uso de un subcircuito, es igual al del inciso anterior, por lo que no se incluye su imagen). Posteriormente, con las mediciones agregadas, se realizó la simulación. A continuación, se presentan, en formato de histograma, los resultados obtenidos:

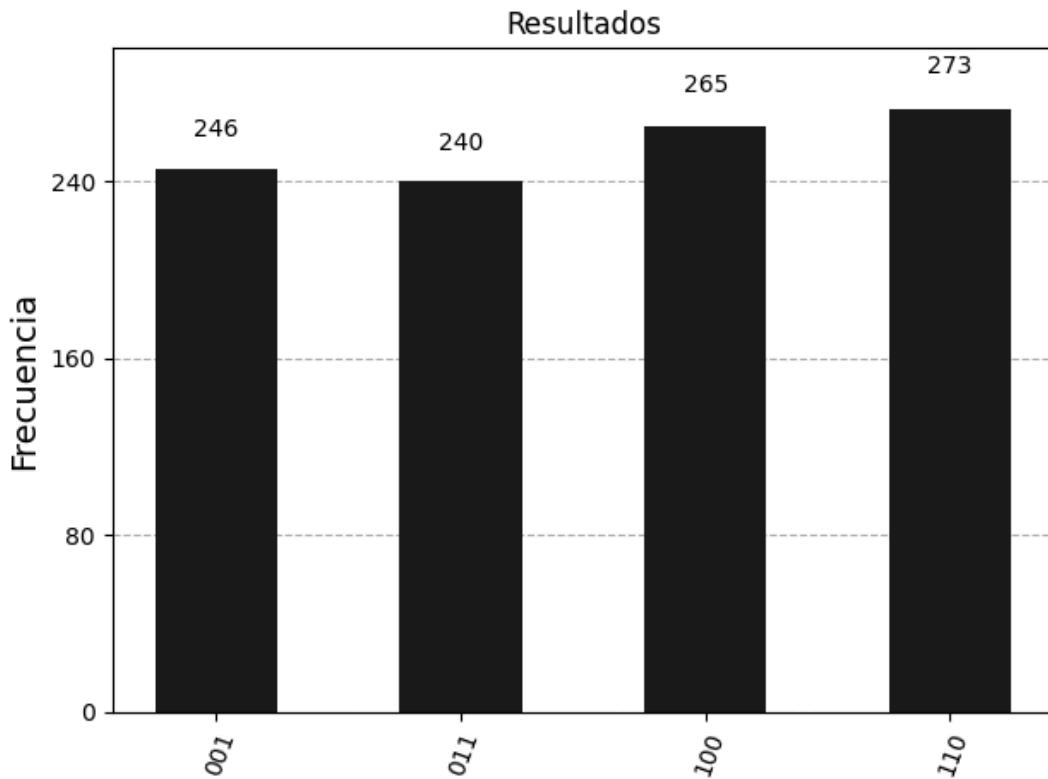


Imagen 18: histograma correspondiente a la simulación del circuito con 1024 shots.

El algoritmo de Deutsch-Jozsa establece que:

- Si la función es constante, toda la probabilidad colapsa en un solo estado básico: el $|000\rangle$.
- Por otro lado, si la función es balanceada, la probabilidad se reparte entre cualquier combinación de estados básicos **exceptuando** el $|000\rangle$.

Como puede observarse, los resultados presentados coinciden con la segunda definición. Por consiguiente, queda demostrado que la función es balanceada.

D) Deutsch-Jozsa - Función No Balanceada Ni Constante

Este inciso corresponde al ejemplo 4 del ejercicio 1 del TP de paralelismo, el cual se muestra a continuación:

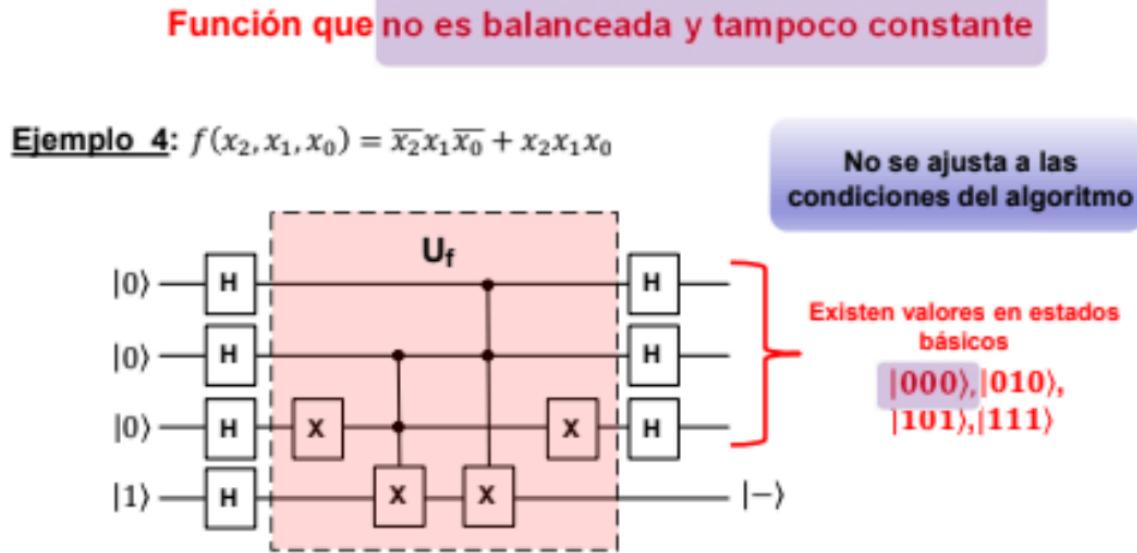


Imagen 19: ejemplo 4 del ejercicio 1 del TP de paralelismo cuántico.

Debido a que previamente ya se planteó el algoritmo de Deutsch-Jozsa, solo se requiere plantear el oráculo de la función:

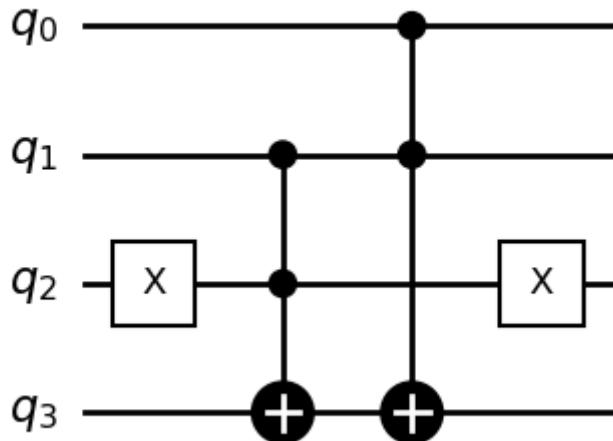


Imagen 20: circuito de la función del ejemplo 4 del ejercicio 1 del TP de paralelismo cuántico.

Aplicando el algoritmo de Deutsch-Jozsa al oráculo, se obtuvo el siguiente circuito:

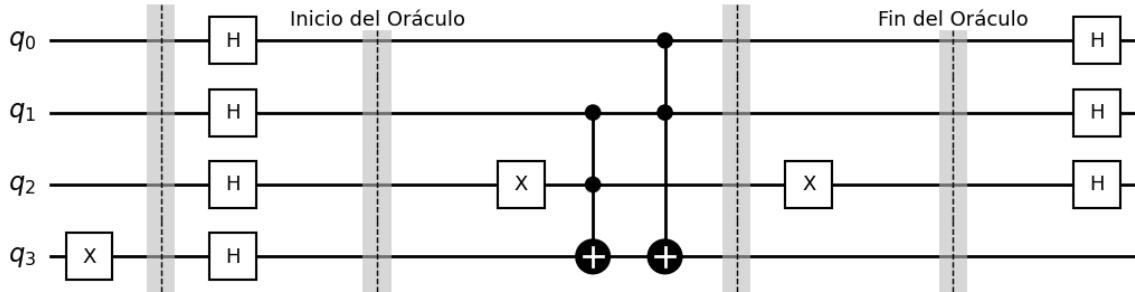


Imagen 21: aplicación del algoritmo de Deutsch-Jozsa al oráculo. La barrera adicional fue colocada para evitar el reordenamiento realizado por Qiskit, lo que provocaba que el circuito no sea igual al del ejemplo.

Luego, se añadieron las mediciones correspondientes para realizar la simulación (el circuito resultante, debido al uso de un subcircuito, es igual al del inciso B, por lo que no se incluye su imagen). Posteriormente, con las mediciones agregadas, se realizó la simulación. A continuación, se presentan, en formato de histograma, los resultados obtenidos:

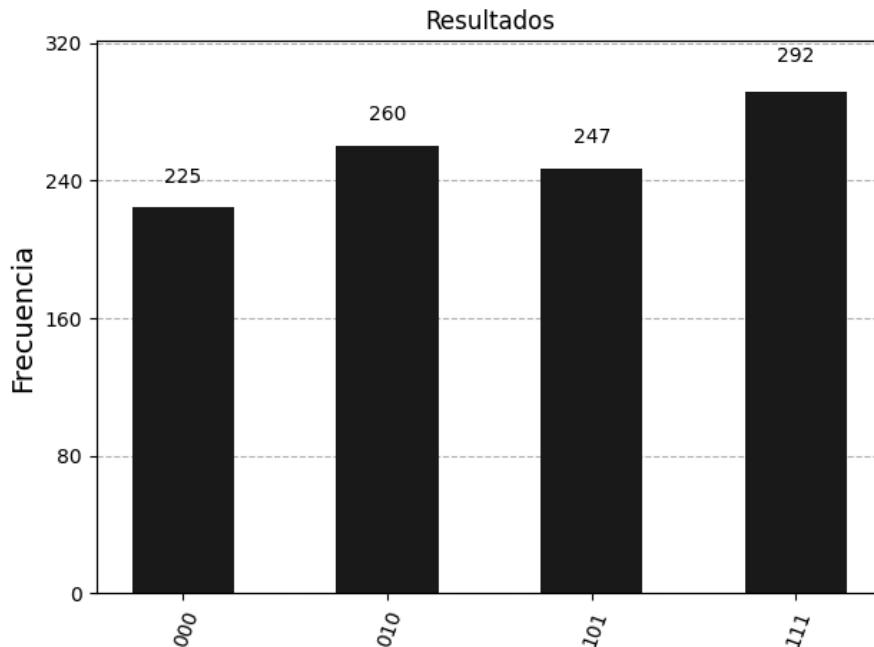


Imagen 22: histograma correspondiente a la simulación del circuito con 1024 shots.

El algoritmo de Deutsch-Jozsa establece que:

- Si la función es constante, toda la probabilidad colapsa en un solo estado básico: el $|000\rangle$.
- Por otro lado, si la función es balanceada, la probabilidad se reparte entre cualquier combinación de estados básicos **exceptuando** el $|000\rangle$.

Como puede observarse, la función no cumple con los requisitos del algoritmo. Los estados básicos que se miden son variados e incluyen al $|000\rangle$. Por consiguiente, queda demostrado que la función no es balanceada ni constante.

Conclusiones

A lo largo de la resolución de la entrega, no solo se asentaron los conocimientos vistos en la materia, sino que se refrescaron las habilidades de programación en Python, adquiridas durante varias de las materias de la carrera. Adicionalmente, se sumó al conjunto de herramientas un nuevo *framework*: Qiskit, el cual resultó intuitivo y fácil de usar.