Universidad del País Vasco
Euskal Herriko Unibertsitatea

ZIENTZIA ETA TEKNOLOGIA FAKULTATEA
FACULTAD DE CIENCIA Y TECNOLOGÍA

Gradu Amaierako Lana / Trabajo Fin de Grado
Fisikako Gradua / Grado en Física

Computación cuántica:
Tests de entrelazamiento y algoritmo de búsqueda de Grover en dispositivos de IBM Quantum de 5 cúbits

Egilea/ Autora:
Raquel Nicolás del Álamo
Zuzendaria/ Director:
Jesús Etxebarria Ecenarro

Leioa, 2021ko ekainaren 16a / Leioa, 16 de junio de 2021

Índice

1 Introducción **1**

2 Fundamento teórico **3**

2.1 El cúbit . . . 3

2.1.1 Medidas en otras bases . . . 5

2.2 Sistemas de varios cúbits . . . 5

2.3 Computación cuántica . . . 6

2.3.1 Puertas lógicas para estados de un cúbit . . . 6

2.3.2 Puertas lógicas para estados de dos cúbits . . . 7

2.4 Representación de circuitos . . . 7

2.4.1 Puertas lógicas para *N* cúbits en *IBM Quantum Experience* . . . 8

3 Dispositivos empleados **9**

3.1 IBM Quantum Experience . . . 9

3.2 Simulador . . . 10

4 Entrelazamiento cuántico **11**

4.1 Versión de Mermin de las desigualdades de Bell . . . 12

4.1.1 Circuito cuántico para implementar el dispositivo de Mermin . . . 13

4.1.2 Resultados experimentales . . . 14

4.2 Desigualdades de Mermin-Klyshko . . . 15

4.2.1 Resultados de la implementación . . . 18

4.3 Conclusiones . . . 19

5 Algoritmo de Grover **21**

5.1 El algoritmo . . . 21

5.2 Implementación en *IBM Quantum Experience* . . . 22

5.2.1 Resultados de la implementación en 2 cúbits . . . 23

5.2.2 Resultados de la implementación en 3 cúbits . . . 25

5.2.3 Resultados de la implementación en 4 cúbits . . . 26

5.2.4 Conclusiones . . . 28

5.3 Simulador . . . 29

5.3.1 Resultados . . . 32

5.3.2 Conclusiones 32

6 Conclusiones globales **33**

Bibliografía **34**

A Valores y vectores propios de los operadores de Mermin **36**

A.1 Definición 36

A.2 Matrices asociadas a los operadores 37

A.2.1 Estados de 1 cúbit 37
A.2.2 Estados de 2 cúbits 37
A.2.3 Estados de 3 cúbits 37
A.2.4 Estados de 4 cúbits 38
A.2.5 Estados de 5 cúbits 39

B Oráculo de Grover **41**

B.1 2 cúbits 41

B.2 3 cúbits 41

B.3 4 cúbits 42

C Algoritmo de Grover: circuito completo **43**

C.1 3 cúbits 43

C.2 4 cúbits 43

Capítulo 1

Introducción

A principios del siglo XX tuvo lugar una de las revoluciones científicas más importantes para la física contemporánea: la revolución cuántica. La incapacidad de las teorías clásicas para explicar el espectro de emisión de los cuerpos negros, el colapso de los átomos en el modelo de Rutherford o el efecto fotoeléctrico pusieron de manifiesto la necesidad de una nueva teoría capaz de explicar todos estos fenómenos. Las carencias de la teoría clásica sirvieron como precedente para que, en la década de 1920, se postulara la teoría cuántica.

Aproximadamente dos décadas después, en 1936, Alan Turing desarrolló un modelo de máquina computacional denominada *máquina de Turing* [1]. La máquina de Turing es un dispositivo abstracto que representa una computadora programable; fue la idea precursora de la ciencia de la computación y de la informática moderna. Poco tiempo después de la publicación del artículo de Turing, comenzaron a desarrollarse los primeros ordenadores.

La computación cuántica engloba muchos de los avances de la teoría cuántica y la ciencia computacional; surgió en 1980, cuando Paul Benioff describió el funcionamiento de una máquina de Turing cuántica [2]. Consiste en un paradigma computacional radicalmente distinto al de la computación clásica: se basa en algunas de las propiedades fundamentales de los sistemas cuánticos, tales como el principio de superposición y el fenómeno de entrelazamiento.

En la actualidad, el aumento de la potencia de las computadoras se debe en gran parte a la progresiva miniaturización de los circuitos integrados. Sin embargo, la dificultad para disipar el calor generado en los procesadores y la creciente importancia de los efectos cuánticos a las escalas a las que se trabaja hoy en día[1] están mermando de manera alarmante nuestra capacidad para reducir el tamaño de los componentes electrónicos.

La computación cuántica se postula como una de las vías más prometedoras para seguir aumentando la capacidad computacional de los ordenadores, más aún en una sociedad cada vez más dependiente de dicha tecnología. El interés de la comunidad científica por realizar avances en la disciplina es innegable; no obstante, nos encontramos ante el reto mayúsculo que supone la construcción de ordenadores cuánticos verdaderamente eficientes. La mayoría de los avances hasta la fecha han sido en el ámbito teórico; y, de acuerdo con ellos, los ordenadores cuánticos han demostrado ser más potentes que los ordenadores clásicos para la resolución de ciertos tipos de problemas. Dos de los ejemplos más relevantes para argumentar el fervor que han causado los progresos en la teoría de la información cuántica son el algoritmo de Grover

[1] Algunas estructuras electrónicas tienen un tamaño menor que $10\mathrm{nm}$, que corresponde a la longitud de una cadena de 40 átomos, aproximadamente.

[3] y el algoritmo de Shor [4]. El primero es un algoritmo de búsqueda capaz de encontrar una entrada en una base de N datos desordenados en tan solo $O(\sqrt{N})$ iteraciones, mientras que en el caso clásico serían necesarias $O(N)$. El segundo consiste en un algoritmo de factorización en números primos en tiempo polinomial, clásicamente no existe ningún algoritmo de este tipo.

Pese a que los ordenadores cuánticos se encuentran en su etapa inicial de desarrollo, en 2016, la compañía *IBM* liberó el acceso a algunos de sus procesadores cuánticos a través de *IBM Quantum Experience*. Se trata aún de ordenadores de potencia muy limitada; todavía no es posible realizar con ellos tareas complejas.

El objetivo de este trabajo es llevar a cabo la implementación de una serie de circuitos y algoritmos de complejidad ascendente en los ordenadores cuánticos de *IBM*, a fin de realizar un análisis sobre la fiabilidad de estos dispositivos y la caracterización de algunas de sus propiedades.

El trabajo se ha dividido en dos bloques. En primer lugar, utilizaremos los ordenadores cuánticos como dispositivo experimental para probar el teorema de Bell. Asimismo, comprobaremos la violación de las desigualdades de Mermin-Klyshko para analizar la calidad del entrelazamiento entre cúbits proporcionado por las máquinas de *IBM*.

En segundo lugar, se realizará la implementación del algoritmo de búsqueda de Grover de acuerdo a dos paradigmas diferentes. Se han construido una serie de circuitos ejecutables en *IBM Quantum Experience*, que permiten la implementación del algoritmo de Grover en 3 y 4 cúbits. De acuerdo con los resultado obtenidos en este análisis, realizaremos una comparativa entre el rendimiento teórico y el rendimiento real de dos de los ordenadores cuánticos disponibles en *IBM Quantum Experience*, denominados *IBMQ Santiago* e *IBMQ Melbourne*. Paralelamente, se ha simulado el funcionamiento de un ordenador cuántico en uno clásico, utilizando *Mathematica* [5]. El fin de este último estudio consiste en comprender el funcionamiento del algoritmo y su fiabilidad en función del número de cúbits involucrados en el proceso.

Capítulo 2

Fundamento teórico

En este capítulo se presentan los conceptos fundamentales para comprender el análisis realizado en este trabajo. Comenzaremos por introducir el sistema más sencillo: el cúbit, y examinaremos algunas de sus propiedades más relevantes. A continuación, analizaremos los aspectos fundamentales de los sistemas de varios cúbits y las bases de la computación cuántica: las puertas lógicas cuánticas y los circuitos cuánticos.

2.1. El cúbit

En la teoría de la información clásica la unidad mínima de información es el *bit*, o dígito binario. Consiste en sistema que puede estar en dos estados: 0 o 1; aunque los dígitos pueden representar infinidad de condiciones, por ejemplo: verdadero o falso, encendido o apagado, etc.

En la teoría de la información cuántica, la unidad mínima de información es el *cúbit* (bit cuántico). El cúbit representa un espacio de Hilbert de dimensión dos, análogo a un sistema de espín $1/2$; donde el operador *medida* tiene dos estados propios: $|0\rangle$ y $|1\rangle$. La diferencia esencial entre el bit clásico y el cúbit es que el estado descrito por un cúbit, $|\Psi\rangle$, es una *superposición lineal* de los estados $|0\rangle$ y $|1\rangle$. De esta forma, siendo a y b dos números complejos, el estado $|\Psi\rangle$ vendrá descrito por:

$$|\Psi\rangle = a\,|0\rangle + b\,|1\rangle \tag{2.1}$$

Si recurrimos a la representación matricial, los estados $|0\rangle$ y $|1\rangle$ corresponden a

$$|0\rangle \doteq \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle \doteq \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.2}$$

y el estado del sistema puede reescribirse:

$$|\Psi\rangle \doteq \begin{pmatrix} a \\ b \end{pmatrix} \tag{2.3}$$

Sin embargo, esta representación no es única: pueden definirse otras bases, todas ellas compuestas por una pareja de estados ortonormales. Una de las más comunes es la *base de Hadamard*:

$$\mathfrak{B}_{\mathrm{H}} = \left\{ |+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\ |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \right\} \tag{2.4}$$

De acuerdo con los principios de la mecánica cuántica, al realizar una medida sobre un sistema cuántico la función de onda colapsa a alguno de los estados propios del operador. En el caso de un cúbit, el resultado de la medida del mismo será alguno de los estados de la *base computacional*, $\mathfrak{B}_{\mathrm{C}} = \{|0\rangle, |1\rangle\}$, cada uno de ellos con una cierta *probabilidad*.

La probabilidad de encontrar cada uno de los dos posibles estados viene dada por el módulo al cuadrado de a y b, de modo que éstos reciben el nombre de *amplitud de probabilidad:*

$$P(|0\rangle) = |a|^2 \tag{2.5}$$
$$P(|1\rangle) = |b|^2 \tag{2.6}$$

Dado que $|a|^2$ y $|b|^2$ representan probabilidades, el estado $|\Psi\rangle$ debe estar normalizado:

$$|a|^2 + |b|^2 = 1 \tag{2.7}$$

Es útil reescribir la expresión (2.1) como

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle \tag{2.8}$$

Los ángulos θ y φ definen un punto en la *esfera de Bloch*, donde cada punto de la superficie corresponde a un estado del sistema. Esta representación resulta útil para visualizar la acción de los distintos operadores, o *puertas lógicas* (Sección 2.3.1), sobre los estados.

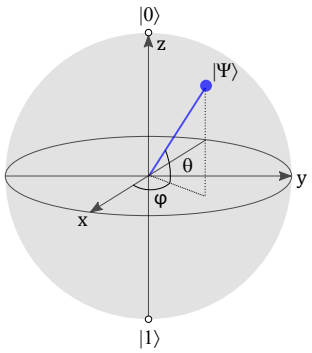


Figura 2.1: Representación de un estado $|\Psi\rangle$ en la esfera de Bloch.

Podemos definir una *base propia* del operador medida para cada dirección en la esfera de Bloch, determinada por los parámetros θ y φ:

$$\mathfrak{B}_{\hat{n}} = \left\{ |+\rangle_{\hat{n}} := \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, |-\rangle_{\hat{n}} := -\sin\frac{\theta}{2}e^{-i\varphi}|0\rangle + \cos\frac{\theta}{2}|1\rangle \right\} \tag{2.9}$$

También es conveniente definir la base propia del operador medida en el eje Y:

$$\mathfrak{B}_{\mathrm{Y}} = \left\{ |+\rangle_{\mathrm{Y}} := \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), |-\rangle_{\mathrm{Y}} := \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \right\} \tag{2.10}$$

2.1.1. Medidas en otras bases

Tal y como hemos visto en la Sección 2.1, al medir el estado de un cúbit, $|\Psi\rangle = a\,|0\rangle + b\,|1\rangle$, se obtienen los estados $|0\rangle$ o $|1\rangle$, con probabilidades definidas por el módulo de sus amplitudes al cuadrado.

Sin embargo, la elección de la base no es única. De forma general, dados dos estados ortonormales $|a\rangle$ y $|b\rangle$, el estado del cúbit puede representarse como $|\Psi\rangle = \alpha\,|a\rangle + \beta\,|b\rangle$. Si realizamos ciertas transformaciones sobre el cúbit antes de realizar la medida, podremos medir el sistema en la base $\mathfrak{B} = \{|a\rangle, |b\rangle\}$. El resultado será $|a\rangle$ con probabilidad $|\alpha|^2$ y $|b\rangle$ con probabilidad $|\beta|^2$.

Para poder realizar este procedimiento, debemos identificar los estados de $\mathfrak{B}$ con los de la base computacional. Antes de realizar la medida, será necesario realizar una transformación tal que $|a\rangle \to |0\rangle$ y $|b\rangle \to |1\rangle$. En general los estados $|a\rangle$ y $|b\rangle$ corresponden a los estados $|+\rangle_{\hat{n}}$ y $|-\rangle_{\hat{n}}$, y la transformación previa a la medida es tal que $|+\rangle_{\hat{n}} \to |0\rangle$ y $|-\rangle_{\hat{n}} \to |1\rangle$.

2.2. Sistemas de varios cúbits

Un bit puede representar únicamente dos estados, para representar más información utilizamos combinaciones de bits. A partir de la concatenación de dos bits, surgen cuatro estados diferentes: 00, 01, 10 y 11.

Similarmente, de la combinación de dos cúbits surge un espacio con cuatro estados propios: $|00\rangle, |01\rangle, |10\rangle$ y $|11\rangle$. En este caso, el par de cúbits se encontrará en una superposición lineal de los 4 estados propios de la base, $\mathfrak{B}_{\mathrm{C}} = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$.

Consideremos un estado $|\Psi\rangle_{\text{inicial}} = a_{00}|00\rangle + a_{01}|01\rangle + a_{10}|10\rangle + a_{11}|11\rangle$. Al realizar una medida sobre el *primer* cúbit, resultando en el estado $|q_1\rangle \to |0\rangle$, el estado final pasará a ser

$$|\Psi\rangle_{\text{final}} = \frac{a_{00}|00\rangle + a_{01}|01\rangle}{\sqrt{|a_{00}|^2 + |a_{01}|^2}} \tag{2.11}$$

Esta propiedad es fundamental para dar cuenta de los fenómenos poco intuitivos que se dan cuando los cúbits están entrelazados (véase Capítulo 4). Para comprender la relevancia de esta característica, consideremos ahora un estado $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Si medimos el primer cúbit, entonces el estado del segundo pasará de ser una mezcla estadística de estados $|0\rangle$ y $|1\rangle$ al $50\,\%$, a ser un estado puro $|1\rangle$. En general, la medida de uno de los cúbits influye sobe el otro.

Otra de las ventajas de los sistemas cuánticos es que con muy pocos cúbits puede representarse una gran cantidad de estados. Esta característica hace que la memoria alcanzable por los ordenadores cuánticos sea muy superior a la de los ordenadores clásicos. En un sistema de N cúbits, un estado se define por N amplitudes de probabilidad complejas, cada una dada por 2 números reales. Si un número real equivale a 8 bytes de almacenamiento, el número de bytes de memoria será $2^N \cdot 2 \cdot 8 = 2^{N+4}$. Para 28 cúbits el almacenamiento logrado es de 4GB.

Por otra parte, mientras que los ordenadores clásicos solo pueden evaluar una determinada función $f(x)$ para un único valor de x cada vez, el *paralelismo cuántico* permite a los ordenadores cuánticos evaluar $f(x)$ para varios valores x *simultáneamente* [6]. Gracias a esta propiedad, la capacidad de los ordenadores cuánticos para realizar operaciones en paralelo crece de forma exponencial con el número de cúbits.

2.3. Computación cuántica

Los ordenadores cuánticos manipulan los cúbits de acuerdo con los *circuitos cuánticos*, y las operaciones realizadas se representan por medio de *puertas lógicas cuánticas*. La única restricción [6] para construir puertas lógicas cuánticas es que las transformaciones realizadas sobre los cúbits sean *unitarias* para preservar la norma de los estados y puedan representar la evolución física de los mismos, es decir, que verifiquen:

$$\hat{U}^\dagger = \hat{U}^{-1} \tag{2.12}$$

A continuación se enumeran algunas de las puertas lógicas cuánticas más comunes, junto con algunas otras consideraciones fundamentales para la implementación de circuitos cuánticos en los ordenadores de *IBM Quantum Experience*.

2.3.1. Puertas lógicas para estados de un cúbit

Las operaciones realizadas sobre un único cúbit están representadas por matrices unitarias de dimensión 2 [7].

- **Puerta de Hadamard:** La puerta de Hadamard es una de las puertas lógicas cuánticas más utilizadas. Su representación en forma matricial es

  $$\hat{H} \doteq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{2.13}$$

  Al actuar sobre alguno de los estados de la base computacional, transforma estos estados en los estados de la base de Hadamard, y viceversa.

  $$\hat{H}|0\rangle = |+\rangle \qquad \hat{H}|1\rangle = |-\rangle \qquad \hat{H}|+\rangle = |0\rangle \qquad \hat{H}|-\rangle = |1\rangle \tag{2.14}$$

  En la representación de la esfera de Bloch, la acción de la puerta de Hadamard puede visualizarse como el intercambio de los ejes X y Z.

- **Puerta de cambio de fase:** La puerta de cambio de fase cambia la fase relativa entre las amplitudes del estado en la base computacional, de forma que la probabilidad de medida de los estados $|0\rangle$ y $|1\rangle$ queda invariante. Su matriz asociada es:

  $$\hat{P}_\theta \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{2.15}$$

  Las puertas $\hat{T}$ y $\hat{T}^\dagger$ utilizadas en en la Sección 5.2 son un caso particular de $\hat{P}_\theta$ con $\theta = \frac{\pi}{4}$ y $\theta = -\frac{\pi}{4}$, respectivamente. La puerta S es equivalente a $\hat{P}_\theta$ con $\theta = \frac{\pi}{2}$.

  $$\hat{T} \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix} \qquad \hat{T}^\dagger \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{pmatrix} \qquad \hat{S} \doteq \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{2.16}$$

- **Puerta NOT:** La puerta NOT, también conocida como la matriz de Pauli $\hat{\sigma_x}$ o $\hat{X}$ para mayor simplicidad, transforma el estado $|0\rangle$ en $|1\rangle$ y viceversa. Es equivalente a una rotación de π radianes en torno al eje X en la esfera de Bloch.

  La matriz asociada a la transformación es

  $$\hat{X} \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2.17}$$

- **Puertas de Pauli Y,Z:** Las puertas de Pauli $\hat{Y}$, $\hat{Z}$ también son rotaciones de π radianes en torno a los ejes Y y Z, respectivamente, en la esfera de Bloch.

  Sus matrices asociadas son:

$$\hat{Y} \doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \hat{Z} \doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.18}$$

- **Rotaciones:** Las puertas de rotación corresponden a rotaciones de un ángulo θ en torno a los ejes X ($\hat{R}_x$), Y ($\hat{R}_y$) y Z ($\hat{R}_Z$) en la esfera de Bloch. Corresponden a las matrices:

$$\hat{R}_X \doteq \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \qquad \hat{R}_y \doteq \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \qquad \hat{R}_z \doteq \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix} \tag{2.19}$$

2.3.2. Puertas lógicas para estados de dos cúbits

Para la implementación de los algoritmos computacionales los cúbits no evolucionan de forma aislada: existen puertas que involucran varios cúbits, y que son esenciales para el diseño de los circuitos.

- **Puerta CNOT:** La puerta *CNOT* o *puerta NOT controlada*, involucra a dos cúbits: un cúbit de *control* y otro que será el *objetivo*. Al utilizar una puerta CNOT se aplicará la puerta NOT al cúbit objetivo únicamente si el cúbit de control se encuentra en el estado $|1\rangle$, si no, el sistema queda invariante.
- **Puertas controladas:** De forma general, podemos añadir controles a las puertas lógicas de un cúbit, de forma que la operación sobre el cúbit objetivo se realice si y solo si el o los cúbits de control se encuentran en el estado $|1\rangle$.

2.4. Representación de circuitos

Es conveniente realizar diagramas con las secuencias de puertas lógicas empleadas para el diseño de los circuitos, puede verse un ejemplo en la Figura 2.2. En los diagramas empleados, cada línea horizontal representa un cúbit. En el lado izquierdo se indica cuál es el estado inicial de cada uno de ellos y al final de cada línea se indica la medida del sistema.

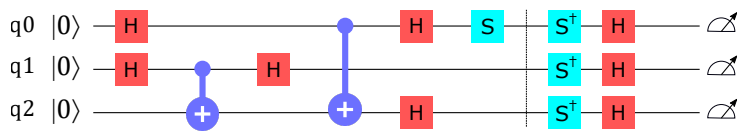
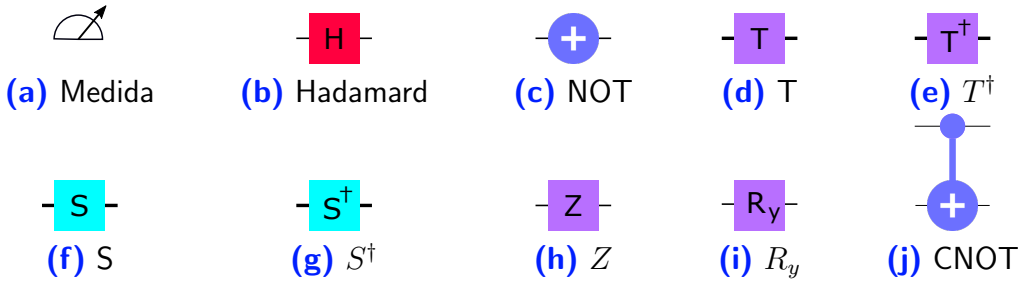


Figura 2.2: Ejemplo de diagrama de un circuito cuántico.

Cada puerta lógica lleva asociado un símbolo, que puede verse en la Tabla 2.1.

Tabla 2.1: Representación de las puertas lógicas utilizadas en los circuitos.



(a) Medida **(b)** Hadamard **(c)** NOT **(d)** T **(e)** $T^\dagger$

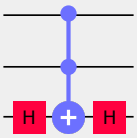
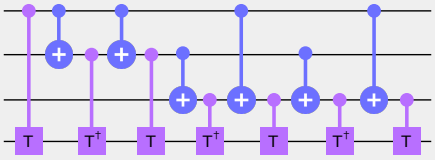
(f) S **(g)** $S^\dagger$ **(h)** Z **(i)** R_y **(j)** CNOT

2.4.1. Puertas lógicas para N cúbits en IBM Quantum Experience

Para poder implementar puertas lógicas en más de 2 cúbits en ordenadores cuánticos como los de *IBM Quantum Experience* es necesario descomponer estas operaciones en otras que requieran únicamente uno o dos cúbits. La construcción de puertas que requieran más de dos cúbits y que tengan una alta fiabilidad puede ser una labor compleja.

Para los algoritmos implementados en este trabajo en los Capítulos 4 y 5 ha sido necesario construir las puertas CZ, CCZ y CCCZ. La C indica que se añade un cúbit de control; las puertas CZ, CCZ y CCCZ son dos puertas Z (véase Sección 2.3.1) con 1, 2 y 3 controles, respectivamente. La operación se realiza sobre el cúbit objetivo si y solo si *todos* los cúbits de control están en el estado $|1\rangle$. Para generar la puerta CCCZ se ha recurrido a la propuesta de Mc Gettrick y Murphy [8]. Los circuitos empleados para cada una de ellas de muestran en la Tabla 2.2.

Tabla 2.2: Equivalencia de las puertas CZ, CCZ, CCCZ usadas para la implementación del algoritmo de Grover en *IBM Quantum Experience*.

Puerta	Equivalencia
 Puerta CZ	Disponible en *IBM Quantum Experience*
 Puerta CCZ	
 Puerta CCCZ	

Capítulo 3

Dispositivos empleados

Este trabajo emplea dos paradigmas computacionales: los ordenadores cuánticos de *IBM Quantum Experience* [9] y el simulador clásico propuesto por D. Candela en [10].

3.1. IBM Quantum Experience

Con el propósito de implementar los circuitos en dispositivos cuánticos reales, hemos recurrido a *IBM Quantum Experience*. Se trata una plataforma en línea que pone a disposición de los usuarios una serie de sistemas cuánticos reales de acceso libre. Los circuitos pueden programarse a través del interfaz del *Quantum Composer*, o en *Qiskit*, un entorno de *Python* definido específicamente para el diseño de circuitos cuánticos.

Todos los sistemas de *IBMQ* emplean cúbits superconductores. Las propiedades cuánticas de los cúbits son extremadamente frágiles y sensibles a pequeñas variaciones del campo electromagnético y temperatura, por ello, deben permanecer aislados y a temperaturas extremadamente pequeñas, de alrededor de $15\ \mathrm{mK}$. Por otro lado, existen varios factores influyentes sobre la eficiencia de los sistemas: el número de cúbits, la arquitectura del ordenador, etc. Es importante enfatizar que un mayor número de cúbits no implica directamente un mejor rendimiento. Debido a esto, es conveniente definir una magnitud *universal* que permita evaluar el rendimiento e índice de error de los ordenadores cuánticos. IBM emplea el *volumen cuántico* con este propósito [11].

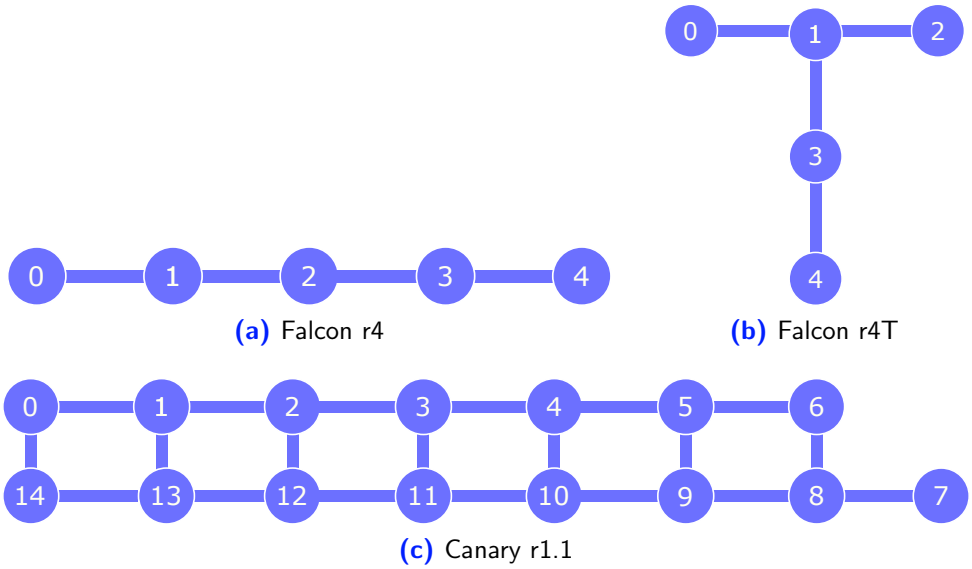
En relación a la implementación de circuitos, deben considerarse ciertos aspectos. El primero es que las operaciones realizables en *IBM Quantum Experience* son limitadas. Por ello, si el circuito a realizar requiere de alguna operación no listada en el interfaz, deberá descomponerse de forma que pueda realizarse con las puertas lógicas disponibles. Asimismo, una vez enviado el circuito para su implementación, el propio software realiza un proceso de *transpilaje*, modificando el diseño inicial del usuario. El objetivo de esto es adaptar el circuito a las características del dispositivo empleado, optimizando los resultados. Teniendo en cuenta lo anterior, para el análisis realizado en este trabajo se han tomado todos los cúbits como equivalentes, ya que el usuario no tiene control sobre cuál de los cúbits *físicos* se realizan las operaciones.

Para este trabajo se han utilizado los ordenadores cuánticos *IBMQ Santiago*, *IBMQ Vigo*, *IBMQ Melbourne* y el simulador *IBM QASM*. Sus propiedades más relevantes se recogen en la Tabla 3.1.

En los dispositivos, no todos los cúbits están conectados entre sí, y esto podría influir sobre los resultados de la implementación de los circuitos. La topología de conexión de cúbits de los sistemas empleados en este trabajo se representa en la Figura 3.1. La lista completa de dispositivos de *IBMQ* y sus propiedades puede verse en [12].

Tabla 3.1: Dispositivos de *IBM Quantum* empleados en el trabajo.

Nombre	Número de cúbits	Procesador	Volumen cuántico
Simulador			
ibm_qasm_simulator	32	-	-
Sistemas cuánticos			
ibmq_santiago	5	Falcon r4	32
ibmq_melbourne	16	Canary r1.1	8
ibmq_vigo	5	Falcon r4T	-



(a) Falcon r4 **(b)** Falcon r4T

(c) Canary r1.1

Figura 3.1: Mapa de los procesadores de *IBM Quantum* empleados en el trabajo.

3.2. Simulador

El simulador construido implementa los circuitos cuánticos en ordenadores clásicos usando cálculo matricial: la información sobre los estados se almacena en vectores de 2^N dimensiones, donde N es el número de cúbits del sistema, y las operaciones se representan por medio de matrices unitarias $2^N \times 2^N$. En este trabajo se ha desarrollado un simulador utilizando el software *Mathematica* [5].

Capítulo 4

Entrelazamiento cuántico

El entrelazamiento es una propiedad característica de algunos estados cuánticos cuyas propiedades son de suma relevancia, ya que las transformaciones realizadas sobre alguno de los cúbits alteran instantáneamente el estado del resto de los constituyentes del sistema, incluso si éstos se encuentran separados por largas distancias.

Tras del desarrollo de la teoría cuántica de la información se ha reforzado el interés por el estudio de los sistemas entrelazados: han demostrado ser la clave de la potencia de las máquinas cuánticas. Asimismo, resulta ser un recurso fructífero para la asistencia de ciertos procesos de transmisión de datos; siendo esencial para el *teleporte cuántico* [13], la *codificación superdensa* [14] o la *criptografía cuántica* [15]. Es crucial para un buen rendimiento de los ordenadores cuánticos que éstos sean capaces de generar y mantener sus cúbits entrelazados; no obstante, se trata de una propiedad extremadamente delicada.

Un sistema está entrelazado cuando el estado que lo describe no puede descomponerse en el producto de sus constituyentes. Por ejemplo, en un sistema de 2 cúbits:

$$|\Psi\rangle \neq |q_1\rangle \otimes |q_2\rangle. \tag{4.1}$$

La forma general del estado del sisema será

$$|\Psi\rangle = \sum_{i,j} A_{ij} \, |q_1\rangle_i \otimes |q_2\rangle_j, \tag{4.2}$$

donde los índices i y j recorren los estados propios de cada cúbit y A_{ij} es la amplitud de probabilidad del estado $|q_1\rangle_i \otimes |q_2\rangle_j \equiv |ij\rangle$.

El objetivo de este capítulo es realizar un análisis sobre el entrelazamiento proporcionado por los sistemas de *IBM Quantum Experience*. El estudio se ha limitado a dos dispositivos: los ordenadores *IBMQ Santiago* e *IBMQ Vigo*, ambos de 5 cúbits.

En primer lugar, comprobaremos experimentalmente la violación de las desigualdades de Bell [16], modelizando el dispositivo descrito por N. D. Mermin en «*Bringing home the atomic world: Quantum mysteries for any-body*» [17].

Por otra parte, las desigualdades de *Mermin-Klyshko* [18], [19] definen una frontera entre los distintos niveles de entrelazamiento; gracias a ellas, podemos inferir el número de cúbits entrelazados en una cierta configuración, además de descartar las *teorías locales* [20]. A través de estas desigualdades, analizaremos el entrelazamiento en los dispositivos de *IBM Quantum Experience*.

4.1. Versión de Mermin de las desigualdades de Bell

Tras la publicación del artículo de Bell [16], se han propuesto otros muchos tipos de desigualdades [20] a fin de estudiar los sistemas entrelazados y establecer un límite entre las teorías locales y la mecánica cuántica.

En el artículo [17], Mermin describe un dispositivo sencillo que sirve para visualizar las implicaciones del teorema de Bell. El dispositivo (véase Figura 4.1) consta de dos detectores que pueden configurarse en 3 posiciones: 1, 2 y 3. Además, cada detector posee dos bombillas: una verde y una roja. Entre ambos detectores hay una caja. Al pulsar el botón, emergen de la caja 2 partículas hacia los detectores. La caja y los detectores están aislados, de forma que ningún componente puede influir sobre el resto.

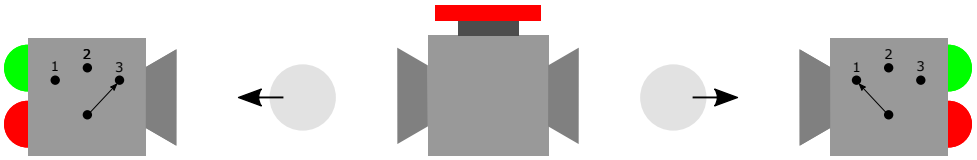


Figura 4.1: Dispositivo propuesto por Mermin para visualizar las desigualdades de Bell.

Tras realizar varias medidas modificando las configuraciones de las cajas se obtienen los siguientes resultados: si la configuración de ambos detectores es la misma (11, 22 o 33) el color de la bombilla que se enciende es la misma siempre en ambos detectores; si la configuración de los detectores no coincide, el color de la bombilla encendida es distinto un $75\,\%$ de las veces, y el mismo un $25\,\%$ de ellas. Estos valores son incompatibles con las teorías locales, tal y como veremos a continuación.

Debemos enfatizar que tanto los detectores como la caja de la que emergen las partículas están aislados y, por hipótesis, no hay ninguna acción a distancia entre las cajas. Por ello, cabe esperar que exista alguna propiedad del sistema que determine cuál será el color de la luz encendida para que las interacciones entre los dispositivos sean locales. Esta explicación es la única posible para dar cuenta de por qué si la posición de ambos detectores es la misma, entonces el color de las dos bombillas coincide.

Supongamos que las partículas emergen en un *estado* RVR: si el detector se coloca en la posición 1, siempre se encenderá la luz roja; para la posición 2, la verde; y para la 3, la roja. En este contexto, las partículas pueden estar en 8 estados diferentes: RRR, VRR, RVR, RRV, RVV, VRV, VVR y VVV.

En cuanto a los detectores, hay 9 configuraciones posibles: 11, 22, 33, 12, 21, 13, 31, 23 y 32. Si dividimos las configuraciones de los detectores en dos clases tenemos 3 en las que los detectores se colocan en la misma posición (11, 22, 33) y 6 en las que la posición es diferente (12, 21, 13, 31, 23, 32).

Aunque los detectores no revelan nunca el estado por completo, ya que solo medimos dos de las posiciones, podemos extraer conclusiones relevantes. Supongamos que los detectores se colocan en posiciones diferentes y que las partículas emergen en un estado RRV. Para las configuraciones de los detectores 12 y 21 la luz encendida en ambos detectores será roja. En el resto de casos: 13, 31, 23 y 32, la luz encendida será de distinto color. Como hay un total de 6 configuraciones en las que la posición de los detectores es diferente, la probabilidad de que la luz encendida sea del mismo color en ambos detectores es de $1/3$, y la probabilidad de que sea distinta, de $2/3$.

Puede hacerse una argumentación análoga para los estados VRR, RVR, RVV, VRV y VVR. En los estados RRR y VVV la luz encendida será del mismo color para cualquier configuración de los detectores. Teniendo en cuenta todos los estados posibles en los que pueden emerger las partículas, la probabilidad de que la luz encendida sea del mismo color debe ser de *al menos* $1/3$. En general, la probabilidad de que el color de la luz coincida será mayor que $1/3$, a menos que las partículas nunca estén en los estados RRR y VVV.

Experimentalmente, la luz encendida es del mismo color $1/4$ de las veces y distinto $3/4$, siendo este resultado incompatible con el modelo basado en la localidad considerado. Para explicar el resultado, debemos abandonar las teorías locales y recurrir a la mecánica cuántica.

El dispositivo de Mermin puede realizarse en los ordenadores cuánticos, modelando las partículas que emergen de la caja como dos cúbits en el estado $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Las configuraciones de los detectores corresponden a la medida de los cúbits en la base computacional, y en 2 bases rotadas $\pm 120°$ en torno al eje Y (véase Ecuación 2.19). Ambos detectores encienden la luz verde al medir el estado $|0\rangle$ y la roja al medir el estado $|1\rangle$.

De acuerdo con la teoría cuántica, la probabilidad de medida del mismo estado en ambos cúbits es proporcional a $\cos^2(\theta/2)$, donde $\theta/2$ es el ángulo relativo entre las bases de medida. Recordemos que para cualquier dirección en la esfera de Bloch puede definirse una base propia:

$$\mathfrak{B}_{\hat{n}} = \left\{ |+\rangle_{\hat{n}} := \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, |-\rangle_{\hat{n}} := -\sin\frac{\theta}{2}e^{-i\varphi}|0\rangle + \cos\frac{\theta}{2}|1\rangle \right\}. \tag{4.3}$$

Si invertimos la relación se tiene:

$$|0\rangle = \cos\frac{\theta}{2}|+\rangle_{\hat{n}} - e^{i\varphi}\sin\frac{\theta}{2}|-\rangle_{\hat{n}}, \qquad |1\rangle = \sin\frac{\theta}{2}e^{-i\varphi}|+\rangle_{\hat{n}} + \cos\frac{\theta}{2}|-\rangle_{\hat{n}}. \tag{4.4}$$

Partimos de un estado $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Al medir el estado $|0\rangle$ para alguno de los cúbits, entonces el estado total pasará a ser $|\Psi\rangle' = |00\rangle$. Si ahora medimos el otro cúbit en una determinada dirección $\hat{n}$, la probabilidad de medida de $\{|+\rangle_{\hat{n}}, |-\rangle_{\hat{n}}\}$ depende de el módulo de su amplitud de probabilidad al cuadrado. Como el estado del segundo cúbit es $|0\rangle$, la probabilidad de medir $|+\rangle_{\hat{n}}$ será $P(|+\rangle_{\hat{n}}) = \cos^2\theta/2$, que, en la base computacional corresponde al estado $|0\rangle$. De forma análoga puede comprobarse que si se mide inicialmente $|1\rangle$, entonces $P(|-\rangle_{\hat{n}}) = \cos^2\theta/2$.

Si ambos cúbits se miden en la misma base, $\theta = 0°$, siempre mediremos el mismo estado en ambos. Sin embargo, cuando la medida de los cúbits se realice en distintas bases, $\theta = 240°$, la probabilidad de medir el mismo valor es $\cos^2 120° = 0{,}25$; y de medir distinto, $\mathrm{sen}^2\, 120° = 0{,}75$. Este resultado concuerda con el funcionamiento del dispositivo descrito por Mermin en [17].

4.1.1. Circuito cuántico para implementar el dispositivo de Mermin

El circuito empleado para realizar el sistema se representa en la Figura 4.2. Comenzamos por generar el estado, $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, aplicando una puerta de Hadamard al cúbit q_0 y posteriormente una puerta CNOT. Para modificar la base de medida, colocamos una puertas de rotación en torno al eje Y, R_y. Para que el ángulo entre las bases de medida sea de $\pm 120°$, el parámetro de la rotación será $4\pi/3$ [21].

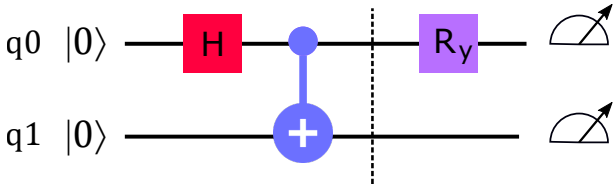


Figura 4.2: Circuito para comprobar la violación de las desigualdades de Bell.

Podemos además comprobar que la probabilidad de medida del mismo valor es efectivamente proporcional a $\cos^2(\theta/2)$. Para ello, basta con implementar el mismo circuito modificando el valor del ángulo de rotación, θ, de la puerta R_y.

4.1.2. Resultados experimentales

Se ha implementado el circuito de la Figura 4.2 en *IBM Quantum Experience*. El ordenador en el que se ha realizado el experimento es el *IBMQ Santiago*, de 5 cúbits, ejecutando un total de 8192 iteraciones para cada experimento.

En el primer experimento, los datos recogidos concuerdan con la descripción del dipositivo de Mermin: cuando las bases de medida forman $120°$ en el $72{,}86\,\%$ de las veces el estado medido es distinto, y coincide en solo el $27{,}14\,\%$ de ellas. Cuando la medida se realiza en la misma base, se mide el mismo estado en un $97{,}13\,\%$ de los casos. Las discrepancias con los valores teóricos podrían deberse a que, por un lado, el entrelazamiento entre cúbits no es perfecto y, por otra parte, las puertas lógicas no son totalmente fiables e introducen cierto error en el sistema. Sin embargo, hemos comprobado que el comportamiento está de acuerdo con el del dispositivo de Mermin, y por lo tanto está lejos de poder ser tratado como un sistema clásico.

En el segundo experimento se ha analizado la correlación entre medidas en distintas bases. Cambiando el valor de rotación de la puerta R_y, se obtienen los resultados recogidos en la Figura 4.3. En este caso, el numero de implementaciones se ha reducido a 1000 para cada valor del ángulo. Observamos que la forma de la curva se ajusta bastante bien a la curva cosenoidal predicha por la teoría cuántica, marcada en gris.

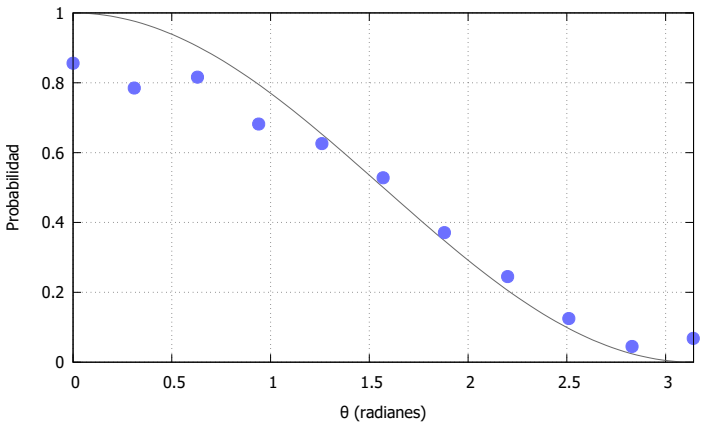


Figura 4.3: Evolución de la proporción de veces en las que se mide el mismo estado en ambos cúbits en función del ángulo de rotación de la puerta R_y.

4.2. Desigualdades de Mermin-Klyshko

Las desigualdades de Bell solo son aplicables a sistemas de dos partículas. Mermin extendió el resultado a sistemas con un mayor número de constituyentes, a través de la definición de los *polinomios de Mermin-Klyshko* (MK) [18], se trata de una generalización de las desigualdades de Bell. Las desigualdades MK definen una frontera entre las teorías locales y los sistemas entrelazados. Son importantes ya que, además de permitir comprobar que el sistema se comporta de acuerdo a las leyes de la mecánica cuántica, también sirven para determinar el *número de cúbits entrelazados* del sistema.

Para comprender el análisis realizado, tomaremos como ejemplo un sistema de 3 cúbits descrito por un estado $|\Psi\rangle$. Para realizar una clasificación de acuerdo con el entrelazamiento, debemos considerar todas las descomposiciones posibles de $|\Psi\rangle$. Si el estado puede descomponerse como el producto de sus constituyentes, $|\Psi\rangle = |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle$, el sistema no estará entrelazado, y podrá describirse por medio de teorías locales. Si tenemos dos cúbits entrelazados y uno independiente, podemos descomponer el estado como $|\Psi\rangle = |q_i \leftrightarrow q_j\rangle \otimes |q_k\rangle$ [1]; denominaremos a este caso 2QM. Finalmente, en caso de que las 3 partículas estén entrelazadas, $|\Psi\rangle = |q_1 \leftrightarrow q_2 \leftrightarrow q_3\rangle \neq |q_1\rangle \otimes |q_2\rangle \otimes |q_3\rangle$, diremos que el sistema es 3QM.

Las desigualdades MK involucran la medida de dos observables a_j y a'_j, que pueden realizarse sobre cada uno de los cúbits del sistema analizado: $j = 1, .., N$. En nuestro caso, los observables a_j y a'_j se asocian con la medida de los cúbits en dos bases diferentes: la base de Hadamard, o X (Ec. 2.4), para a_j y la base Y (Ec. 2.10) para a'_j. Para medir en la base X basta con disponer una puerta de Hadamard antes de realizar la medida. Para medir en la base Y, se coloca una puerta $\mathrm{S}^\dagger$ (Ec. 2.16) seguida de una puerta de Hadamard.

Los polinomios MK pueden definirse a partir la regla de recurrencia de la Ecuación 4.5, donde $M_1 = X_1$ y los M'_n se obtienen a partir de los M_n intercambiando los X_j por Y_j, y viceversa.

$$M_n = \frac{1}{2}\left(M_{n-1}\left(X_n + Y_n\right) + M'_{n-1}\left(X_n - Y_n\right)\right), \tag{4.5}$$

Las expresiones de los polinomios MK para sistemas de 2,3,4 y 5 cúbits se recogen en las Ecuaciones (4.6 - 4.9).

$$M_2 = \frac{1}{2}\left(X_1X_2 + Y_1X_2 + X_1Y_2 - Y_1Y_2\right) \tag{4.6}$$

$$M_3 = \frac{1}{2}\left(Y_1X_2X_3 + X_1Y_2X_3 + X_1X_2Y_3 - Y_1Y_2Y_3\right) \tag{4.7}$$

$$\begin{aligned} M_4 =& \frac{1}{4}[-(X_1X_2X_3X_4) - (Y_1Y_2Y_3Y_4) + (Y_1X_2X_3X_4 + X_1Y_2X_3X_4 \\ &+ X_1X_2Y_3X_4 + X_1X_2X_3Y_4) + (Y_1Y_2X_3X_4 + Y_1X_2Y_3X_4 \\ &+ Y_1X_2X_3Y_4 + X_1Y_2X_3Y_4 + X_1X_2Y_3Y_4 + X_1Y_2Y_3X_4) \\ &- (Y_1Y_2Y_3X_4 + Y_1Y_2X_3Y_4 + Y_1X_2Y_3Y_4 + X_1Y_2Y_3Y_4)] \end{aligned} \tag{4.8}$$

$$\begin{aligned} M_5 =& \frac{1}{4}[-(X_1X_2X_3X_4X_5) + (X_1X_2X_3Y_4Y_5 + X_1X_2Y_3X_4Y_5 + X_1Y_2X_3X_4Y_5 \\ &+ Y_1X_2X_3X_4Y_5 + X_1X_2Y_3Y_4X_5 + X_1Y_2X_3Y_4X_5 + Y_1X_2X_3Y_4X_5 \\ &+ X_1Y_2Y_3X_4X_5 + Y_1X_2Y_3X_4X_5 + Y_1Y_2X_3X_4X_5) - (X_1Y_2Y_3Y_4Y_5 \\ &+ Y_1X_2Y_3Y_4Y_5 + Y_1Y_2X_3Y_4Y_5 + Y_1Y_2Y_3X_4Y_5 + Y_1Y_2Y_3Y_4X_5)] \end{aligned} \tag{4.9}$$

[1] Las flechas "$\leftrightarrow$" indican que los cúbits están entrelazados.

Hay que entender las expresiones anteriores como la suma de los valores esperados de las distintas secuencias de operadores.

$$X_1...X_n = \langle X_1...X_n \rangle \tag{4.10}$$

Las expresiones (4.6 - 4.9) pueden simplificarse al tener en cuenta ciertas condiciones de simetría. Si consideramos que todas las permutaciones de una cierta secuencia de operadores son equivalentes; por ejemplo, $\langle \mathrm{X_1X_2Y_3} \rangle = \langle \mathrm{X_1Y_2X_3} \rangle = \langle \mathrm{Y_1X_2X_3} \rangle$, los polinomios de Mermin-Klyshko se reducen a:

$$M_2 = \frac{1}{2}(\langle \mathrm{XX} \rangle + 2\langle \mathrm{XY} \rangle - \langle \mathrm{YY} \rangle) \tag{4.11}$$

$$M_3 = \frac{1}{2}\left(3\langle \mathrm{XXY} \rangle - \langle \mathrm{YYY} \rangle\right) \tag{4.12}$$

$$M_4 = \frac{1}{4}\left(-\langle \mathrm{XXXX} \rangle + 4\langle \mathrm{YXXX} \rangle + 6\langle \mathrm{YYXX} \rangle - 4\langle \mathrm{YYYX} \rangle - \langle \mathrm{YYYY} \rangle\right) \tag{4.13}$$

$$M_5 = \frac{1}{4}\left(-\langle \mathrm{XXXXX} \rangle + 10\langle \mathrm{YYXXX} \rangle - 5\langle \mathrm{YYYYX} \rangle\right) \tag{4.14}$$

Hemos identificado los operadores X e Y con la medida de los cúbits en las bases X e Y. Haciendo una analogía con la medida del espín en un sistema de espín $1/2$, asociaremos los valores 1 y -1 a la medida de los estados $|0\rangle$ y $|1\rangle$, respectivamente. Como los sistemas son de varios cúbits, el valor asociado al estado total será el producto del valor asociado a cada cúbit. Para visualizar esto, consideraremos el caso más sencillo. Cuando tenemos un sistema de dos cúbits hay cuatro medidas posibles: $|00\rangle, |10\rangle, |01\rangle$ y $|00\rangle$. El valor asociado a cada estado es:

$$00 \to (1)(1) = 1 \qquad 01 \to (1)(-1) = -1$$
$$10 \to (-1)(1) = -1 \qquad 11 \to (-1)(-1) = 1$$

Entonces, el valor esperado de una determinada secuencia de operadores, AB, será:

$$\langle AB \rangle = (1)\, P\left(|00\rangle\right) + (-1)\, P(|01\rangle) + (-1)\, P(|10\rangle) + (1)\, P(|11\rangle)$$

donde P denota la probabilidad de medida de cada estado.

El límite dado por las teorías locales (TL) para el valor de todos los polinomios M_N es 1, es decir, $M_N^{TL} \leq 1$. Si se construyen los operadores matricialmente de acuerdo con la descripción cuántica (QM), pueden obtenerse sus valores y vectores propios (véase Anexo A). El valor propio máximo de cada operador da el límite cuántico para el valor de los M_N, en general, $M_N^{QM} \leq 2^{(n-1)/2}$. Para maximizar el valor experimental de los polinomios, previo a la medida el sistema debe estar en el estado propio asociado al valor propio máximo. Los valores y estados propios de cada M_N se representan en la Tabla 4.1.

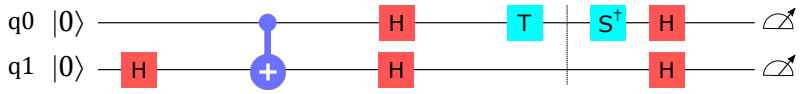
Si el valor de los M_N es mayor (en valor absoluto) que el establecido por las teorías locales, $M_N \geq 1$, entonces el sistema está entrelazado. Cuando en un sistema de n partículas hay m entrelazadas, con $1 \leq m \leq n$, el límite superior viene dado por $2^{(m-1)/2}$. Si $M_N \geq 2^{(m-1)/2}$, entonces hay *al menos* $(m+1)$ partículas entrelazadas.

Para calcular el valor esperado de los M_N en *IBM Quantum Experience*, partimos de la construcción los estados de la Tabla 4.1, de forma que se maximice el valor de los M_N obtenido.

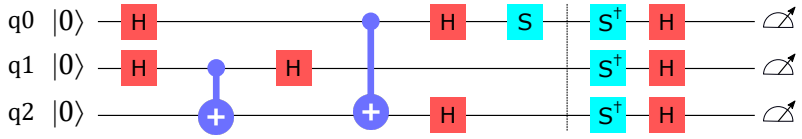
Tabla 4.1: Valores límite de los polinomios y estados que maximizan su valor. Se ha tomado el valor absoluto en ambos límites.

Polinomio	Límite local	Límite cuántico	Estado
M_2	1	$\sqrt{2}$	$\frac{1}{\sqrt{2}}(\|00\rangle + e^{\pi i/4}\|11\rangle)$
M_3	1	2	$\frac{1}{\sqrt{2}}(\|000\rangle + i\|111\rangle)$
M_4	1	$2\sqrt{2}$	$\frac{1}{\sqrt{2}}(\|0000\rangle + e^{7\pi i/4}\|1111\rangle)$
M_5	1	4	$\frac{1}{\sqrt{2}}(\|0000\rangle + \|1111\rangle)$

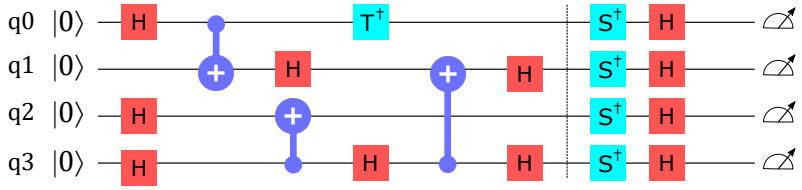
Los circuitos se recogen en la Figura 4.4. La construcción de los estados está delimitada en la figura por una raya vertical. Después se realizan las transformaciones requeridas para medir el cúbit según el eje X o Y en cada caso. Recordemos que una puerta $S^\dagger$ seguida de una puerta H corresponde a la medida en la base Y, y una única puerta H a la medida en X. Los circuitos representados recogen únicamente la medida de una de las configuraciones: YX para M_2, YYY para M_3, YYYY para M_4 e YYYYX para M_5; para medir el resto de configuraciones habrá que modificar la disposición de las puertas $S^\dagger$ y H tras la línea vertical.



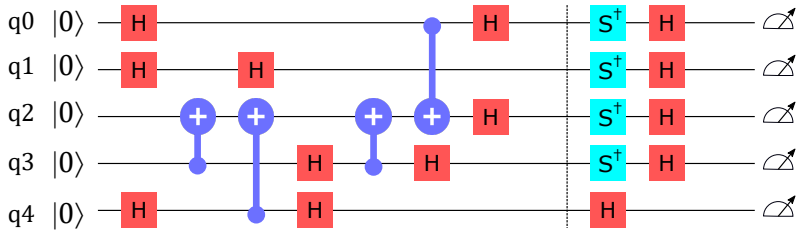
(a) Circuito para la medida de M2



(b) Circuito para la medida de M3



(c) Circuito para la medida de M4



(d) Circuito para la medida de M5

Figura 4.4: Circuitos para el cálculo del valor de los polinomios de Mermin-Klyshko

4.2.1. Resultados de la implementación

A continuación se presentan los resultados obtenidos de la implementación de los circuitos en la Figura 4.4. Se han realizado 8192 iteraciones de cada uno de los circuitos en las máquinas *IBMQ Santiago* e *IBMQ Vigo*. Primero explicaremos en detalle el procedimiento seguido para calcular el valor numérico de todos los polinomios, y se discutirán los resultados obtenidos para M_2. Seguidamente, se presentarán los datos obtenidos para todos los M_N calculados, y se interpretarán los resultados determinando así el tipo de entrelazamiento de cada sistema.

En el caso de 2 cúbits, tras la medida pueden obtenerse 4 estados: $00, 01, 10, 11$. Los estados $00, 11$ se asocian al valor 1, los estados $10, 01$ contribuirán al valor -1. La proporción de veces en las que se ha medido cada estado y el valor esperado de los observables analizados en la Tabla 4.2. Hemos considerado el error de una distribución multinómica: $\delta p = \sqrt{p(1-p)/N}$, que para $N = 8192$ da un error $O(\delta p) = 10^{-2}$ [18].

Tabla 4.2: Resultados de la medida de M2.

XX		**XY**		**YY**	
$\langle XX \rangle$ =0,66		$\langle XY \rangle$ =0,64		$\langle YY \rangle$ = -0,67	
Estado	Proporción	Estado	Proporción	Estado	Proporción
00	0,46	00	0,43	00	0,11
01	0,09	01	0,07	01	0,41
10	0,07	10	0,11	10	0,43
11	0,39	11	0,39	11	0,06

Para calcular el valor esperado de los operadores, multiplicamos cada uno de los valores posibles $(-1, 1)$ por la proporción de veces en las que se ha medido *algún* estado con la paridad asociada, como ejemplifica la Ecuación 4.15 para $\langle XX \rangle$. Finalmente, la Ecuación 4.16 permite calcular la medida M_2.

$$\begin{aligned} \langle XX \rangle &= (1)\,(P(00) + P(11)) + (-1)\,(P(10) + P(01)) \\ &= (0{,}46 + 0{,}39) - (0{,}09 + 0{,}07) = 0{,}69 \end{aligned} \tag{4.15}$$

$$M_2 = \frac{1}{2}\left(\langle \mathrm{XX} \rangle + 2\langle \mathrm{YX} \rangle - \langle \mathrm{YY} \rangle\right) = 1{,}32 \tag{4.16}$$

Puede extenderse el procedimiento y calcular el valor obtenido experimentalmente para los polinomios de Mermin-Klyshko de orden superior. Los resultados pueden verse en la Tabla 4.3.

En todos los casos se supera el límite dado por las teorías locales, por lo que todos los estados son entrelazados. En vista a los resultados obtenidos, podemos además inferir el *número de cúbits entrelazados* en los sistemas de *IBM Quantum*. Para 2 cúbits, el sistema es 2QM, aunque los resultados de *IBMQ Santiago* se acercan más al valor teórico. Cuando el sistema está compuesto por 3, vemos que el ordenador *IBMQ Vigo* supera el límite de las teorías 2QM, por lo que el entrelazamiento es 3QM. Contrariamente, el ordenador de Santiago está en el límite de 2QM, lo que indica que hay al menos 2 cúbits entrelazados, pero al estar justamente en la frontera entre 2QM y 3QM no podemos decir que haya 3 entrelazados. Para 4

cúbits, el ordenador de Vigo consigue entrelazar al menos 3, aunque esta vez el valor obtenido es muy cercano al límite de 3QM. Santiago, de nuevo, entrelaza al menos 2. Si el sistema consta de 5 cúbits, Vigo logra entrelazar al menos 4, aunque de nuevo la diferencia entre el valor obtenido y el límite de las teorías 4QM es muy pequeña, tan solo del $1{,}8\,\%$. El ordenador de Santiago, de nuevo, no sobrepasa el límite de 3QM, por lo que solo podemos deducir que entrelaza al menos 2.

Tabla 4.3: Resultados implementación en *IBM Santiago*.

	5QM	4QM	3QM	2QM	Local	IBMQ Vigo	IBMQ Santiago
$\boldsymbol{M_2}$	-	-	-	$\sqrt{2}$ ($\approx$ 1,41)	1	1,17	1,32
$\boldsymbol{M_3}$	-	-	2	$\sqrt{2}$	1	1,50	1,41
$\boldsymbol{M_4}$	-	$2\sqrt{2}$ ($\approx$ 2,83)	2	$\sqrt{2}$	1	2,09	1,80
$\boldsymbol{M_5}$	4	$2\sqrt{2}$	2	$\sqrt{2}$	1	2,88	1,78

4.3. Conclusiones

A partir de la implementación de los circuitos de la Figura 4.4 en los ordenadores de *IBM Quantum Experience*, se ha comprobado experimentalmente la violación de las desigualdades de Mermin-Klyshko.

Los estados iniciales se han preparado de forma que maximizaran el valor de los polinomios. Sin embargo, los valores experimentales se alejan del valor teórico a medida que aumenta el número de cúbits del sistema analizado. Esto supone un revés para la eficiencia de los dispositivos de *IBMQ*, ya que el entrelazamiento es la herramienta más potente de los sistemas cuánticos. Muchos de los algoritmos cuánticos propuestos están basados en las propiedades de los sistemas entrelazados; si el entrelazamiento proporcionado por las máquinas no cumple los requisitos necesarios, los algoritmos no pueden implementarse con éxito.

Existen varias razones que podrían explicar los resultados obtenidos. El entrelazamiento es una propiedad extremamente delicada y construir y mantener los sistemas entrelazados es una tarea complicada. Además, a medida que aumenta el tamaño de los sistemas, los circuitos para construir los estados de partida aumentan su complejidad. Las puertas lógicas introducen cierto error en el estado, por ello, a medida que aumenta el número de operaciones del circuito, aumenta también el error introducido por las mismas.

Cabe también destacar que hay una clara diferencia entre los resultados de *IBMQ Vigo* e *IBMQ Santiago*: los resultados del primer dispositivo son considerablemente mejores que los del segundo. La forma en la que los cúbits están conectados podría explicar por qué ocurre esto (véase Figura 3.1): el ordenador de Vigo tiene una arquitectura en forma de T, que establece un mayor número de conexiones entre cúbits. En este sentido, la arquitectura de las computadoras podría ser un factor a tener en cuenta a la hora de seleccionar en qué dispositivo implementar un determinado circuito.

El hecho de que no todos los cúbits estén conectados entre sí explica también por qué la calidad del entrelazamiento proporcionado por los sistemas empeora a medida que aumenta el número de cúbits activos.

No obstante, a pesar de que los resultados obtenidos están lejos de ser óptimos, podemos apreciar una clara mejoría con respecto a los obtenidos por Alsina et al. en [18]. En este artículo, publicado en 2016, se realiza una caracterización similar a la de este trabajo. Se trata de la primera violación experimental de las desigualdades de Mermin para cuatro y cinco cúbits. La definición de los polinomios MK es ligeramente distinta, excluye una constante de normalización que en este trabajo se ha incluido para que el valor de los distintos límites (Local, 2QM, 3QM y 4QM) no cambie, a pesar de aumentar el orden del polinomio.

En el trabajo de Alsina et al., aunque las desigualdades se violan en todos los casos, se observa un claro deterioro de los resultados a medida que aumenta el número de cúbits. De hecho, para cinco cúbits la relevancia estadística del resultado es pequeña para descartar la localidad, ya que $M_5 = 1{,}01 \pm 0{,}01$. En nuestro análisis, los resultados obtenidos sí que sirven para descartar en todos los casos las teorías locales. Con todo esto, podemos inferir una clara mejoría de los sistemas de *IBMQ* a lo largo de los últimos cinco años.

Capítulo 5

Algoritmo de Grover

Una de las propiedades más atractivas de la computación cuántica es su extraordinaria eficiencia para resolver ciertos tipos de problemas. Un ejemplo de ello es el algoritmo de búsqueda propuesto en el año 1996 por Lov K. Grover [3].

El algoritmo de Grover es un algoritmo de búsqueda cuántico en una secuencia de N datos desordenados. En el caso clásico, el número de evaluaciones promedio necesarias para encontrar una entrada en una secuencia de N datos es de $O(N)$. El algoritmo de Grover reduce el número de operaciones requeridas a $O(\sqrt{N})$, esto se debe al principio de superposición en la mecánica cuántica. Mientras que en el algoritmo clásico es necesario consultar todas las entradas una a una hasta dar con la correcta, en el caso cuántico podemos definir un *oráculo* capaz de consultar todas las entradas a la vez.

El objetivo de esta sección es comprobar la eficiencia del algoritmo de Grover al implementarlo en los sistemas de *IBMQ*. Los sistemas cuánticos reales de los que disponemos tienen un número de cúbits y volumen cuántico reducidos. Por ello, la implementación se realizará en sistemas de 2, 3 y 4 cúbits, que permiten listas con 4, 8 y 16 entradas respectivamente.

Aunque en sistemas pequeños como los que van a analizarse en este trabajo la diferencia entre las máquinas clásicas y las cuánticas no es demasiado significativa, en bases de datos lo suficientemente grandes la diferencia se vuelve mayúscula.

Los circuitos de *IBM Quantum Experience* se vuelven complejos al aumentar el número de cúbits, por lo que es complicado aumentar el tamaño del sistema. Para comprobar la eficiencia del algoritmo en sistemas más grandes, se ha desarrollado un simulador utilizando el software *Mathematica*. Para ello, se han seguido las indicaciones de D. Candela en [10], utilizando cálculo matricial como herramienta para simular los circuitos cuánticos.

5.1. El algoritmo

Supongamos que tenemos una lista con N entradas, y queremos localizar una entrada determinada, $\omega \equiv |\psi_k\rangle$ utilizando el algoritmo de Grover. El algoritmo se divide en los cuatro pasos descritos en la Figura 5.1, y que serán explicados a continuación.

1. **Inicialización**: De entrada no tenemos ningún tipo de información sobre la localización del elemento que nos interesa, por lo que tomamos el estado inicial como una superposición uniforme de todos los elementos $|\Psi\rangle = \frac{1}{\sqrt{N}} \sum_i |\psi_i\rangle$.

Si en esta situación realizáramos una medida sobre el sistema, obtendríamos cualquiera de los elementos de la base $\{|\psi_n\rangle\}$ con la misma probabilidad.

2. **Oráculo**: El algoritmo de Grover permite amplificar la amplitud de probabilidad del elemento de la base buscado. Para ello, se utiliza un *oráculo*, $\hat{O}$, que *marca* el estado buscado realizando un cambio de fase sobre la amplitud del mismo. Las amplitudes del resto de estados quedan inalteradas.

$$\begin{cases} \hat{O}|\psi_i\rangle = & |\psi_i\rangle & \text{si } i \neq k \\ \hat{O}|\psi_i\rangle = & -|\psi_i\rangle & \text{si } i = k \end{cases} \tag{5.1}$$

3. **Difusión**: Se define un operador de difusión que modifica las amplitudes de todos los estados con respecto a la amplitud media. La amplitud del estado marcado estará invertida tras la aplicación del oráculo. Por ello, su amplitud aumentará y la del resto de estados se verá disminuida.

4. **Medida**: Tras la aplicación sucesiva de los pasos 2 y 3, se mide el estado del sistema. El número óptimo de iteraciones es $\frac{\pi}{4}\sqrt{N}$ [10]. Si no se realizan las suficientes iteraciones, la amplitud del estado no se habrá amplificado lo suficiente y es probable que obtengamos como resultado un elemento diferente al buscado. Asimismo, si sobrepasamos el número óptimo de iteraciones, la eficiencia del algoritmo disminuirá.

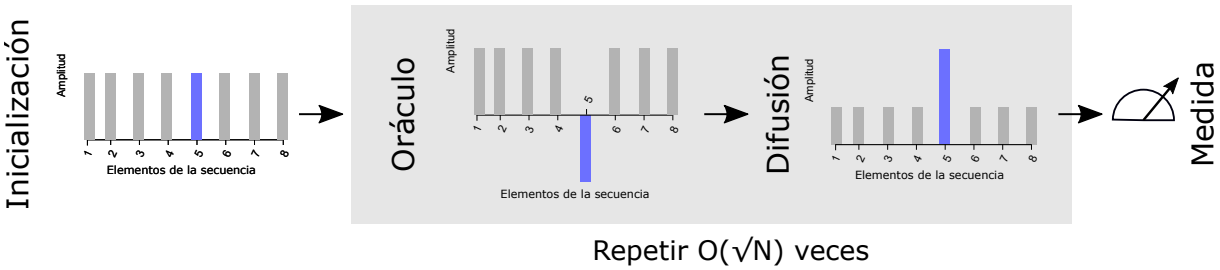


Figura 5.1: Esquema del procedimiento de amplificación en el algoritmo de Grover.

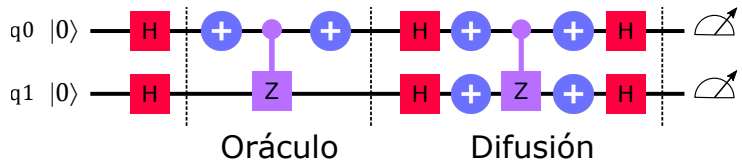
5.2. Implementación en IBM Quantum Experience

Hemos implementado el algoritmo de Grover en los dispositivos de *IBMQ*, en $N = 2, 3$ y 4 cúbits. Los circuitos empleados se muestran en la Figura 5.2.

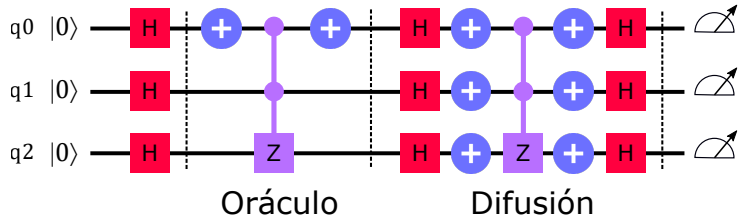
El cambio en la disposición de las puertas NOT del oráculo modifica el estado amplificado. Las configuraciones de la Figura 5.2 corresponden a los estados 10, 110 y 1110; el listado completo de la distribución de las puertas NOT y sus estados asociados se recogen en el Apéndice B. Los bloques correspondientes al oráculo y el operador de difusión deben repetirse aproximadamente $\frac{\pi}{4}\sqrt{N}$ veces: para 2 cúbits 2 veces, 2 veces también para 3, y 3 veces para 4 cúbits.

Los circuitos empleados para construir las puertas CCZ y CCCZ pueden verse en la Tabla 2.2 de la Sección 2.4.1.

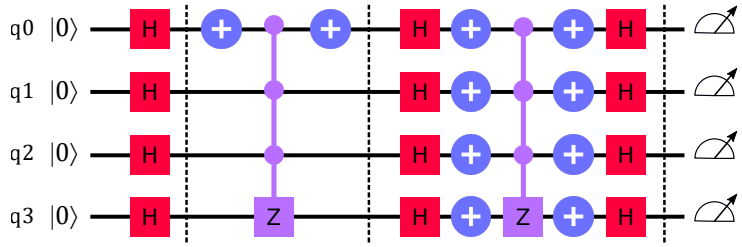
Para poder realizar un análisis comparativo de los dispositivos, los circuitos para $N = 2, 3$ y 4 cúbits se han implementado en el *QASM Simulator* y los ordenadores *IBM Santiago* e *IBM Melbourne*, las características de los dispositivos pueden verse en el Capítulo 3.



(a) Circuito para $N = 2$ cúbits.



(b) Circuito para $N = 3$ cúbits.



(c) Circuito para $N = 4$ cúbits.

Figura 5.2: Circuitos para la implementación del algoritmo de Grover en IBMQ.

El circuito completo para $N = 3$ y 4 cúbits puede verse en el Apéndice C.

5.2.1. Resultados de la implementación en 2 cúbits

A continuación se presentan los resultados obtenidos tras la implementación del circuito en la Figura 5.2. Se comparará el número de veces en las que el resultado coincide con el estado marcado en cada caso. En todos los casos se han realizado 8192 iteraciones del algoritmo.

En el caso del *Simulador QASM*, se encuentra el estado marcado en un $(100 \pm 0)\,\%$ de las veces, en promedio. Se ha considerado el error cuadrático medio, de acuerdo con la ecuación:

$$\Delta x = \sqrt{\sum_{i=1}^{n} \frac{(\bar{x} - x_i)^2}{n(n-1)}}, \tag{5.2}$$

donde $\bar{x}$ es el valor promedio de la medida, el índice i recorre todas las medidas x_i y n es el tamaño de la muestra, en nuestro caso, el número de estados de la base del sistema.

En la Figura 5.9 se representa el número de medidas de cada estado para cada configuración del oráculo. En el eje vertical aparecen listadas todas las configuraciones del circuito, y en el eje horizontal todos los resultados posibles. A lo largo de cada fila horizontal, para un cierto estado marcado, aquellos estados que hayan sido medidos un mayor número de veces tendrán asociada una casilla roja o anaranjada. Aquellos que hayan sido medidos menos veces, tendrán una casilla morada o azul. La escala de colores correspondiente al conteo puede consultarse a la derecha del gráfico. Si el ordenador encuentra el estado marcado, entonces deberá coincidir

con el estado medido un mayor número de veces. En estas circunstancias, los elementos de la diagonal de la Figura 5.3 deberán aparecer marcados por colores cálidos, y el resto, en colores fríos.

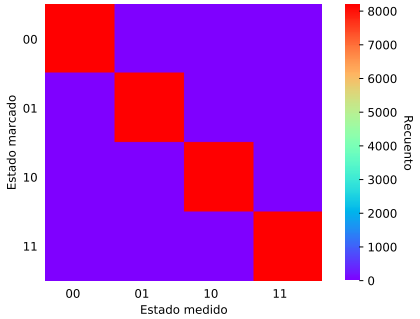
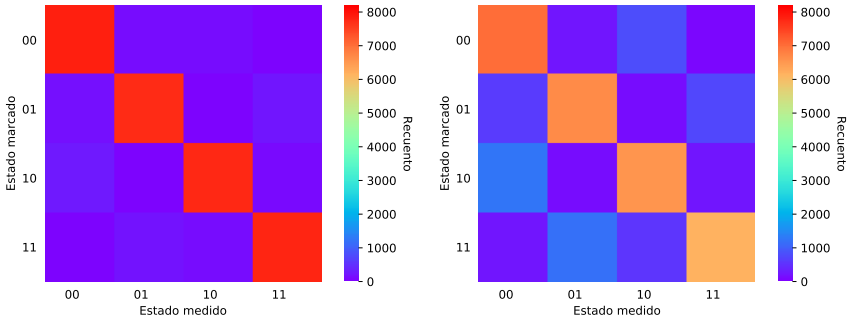


Figura 5.3: Resultados en N=2 cúbits del simulador QASM.

En cuanto a los dispositivos cuánticos reales, la eficiencia se reduce. El ordenador de *IBMQ Santiago* encuentra el estado correcto un $(95{,}0 \pm 0{,}2)\,\%$ de las veces, e *IBMQ Melbourne* tiene una eficiencia aún menor: acierta solo el $(81 \pm 2)\,\%$ de las veces. En la Figura 5.4 pueden verse los resultados para *IBMQ Santiago* e *IBMQ Melbourne*.



(a) *IBMQ Santiago* **(b)** *IBMQ Melbourne*

Figura 5.4: Resultados de la implementación del algoritmo de Grover en estados de 2 cúbits.

En la Figura 5.5 pueden observarse las diferencias entre la eficiencia del simulador y los ordenadores para cada configuración del oráculo.

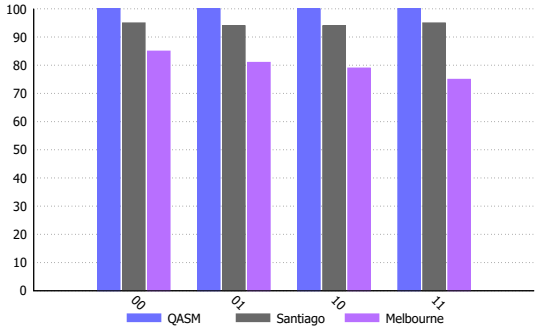


Figura 5.5: Comparativa de la eficiencia del algoritmo de Grover en 2 cúbits.

5.2.2. Resultados de la implementación en 3 cúbits

Se ha repetido el mismo procedimiento que en la Sección 5.2.1, ejecutando nuevamente los circuitos un total de 8192 veces para cada configuración del circuito.

En este caso, el simulador encuentra el estado marcado en el $(94{,}50 \pm 0{,}08\,\%)$ de las veces. El resultado empeora respecto a la implementación en estados de 2 cúbits, aunque en la Figura 5.6 puede comprobarse que en todos los casos la diferencia de veces en la que se mide el estado correcto sigue siendo notable.

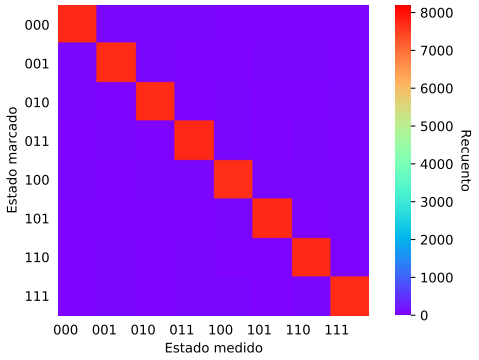
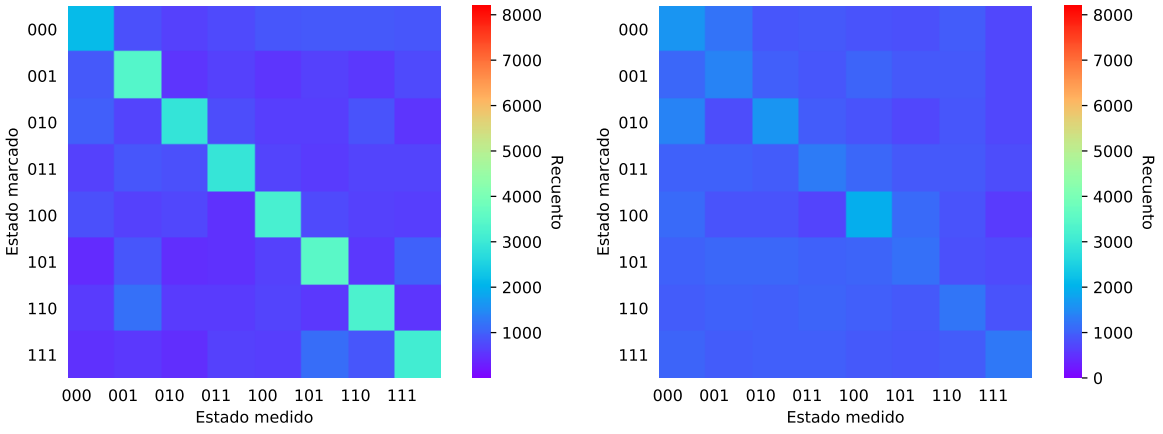


Figura 5.6: Resultados del simulador QASM para estados de 3 cúbits.

En el caso de los sistemas cuánticos reales, la reducción de la eficiencia es aún más evidente. *IBMQ Santiago* completa la búsqueda con éxito en un $(37 \pm 1)\,\%$ de las veces, e *IBMQ Melbourne* tiene una eficiencia muy baja, de tan solo un $(18 \pm 1)\,\%$. Si el estado se colocase en una superposición de todos los estados, siendo la medida totalmente aleatoria, la probabilidad de medida de cada estado es del $12{,}5\,\%$ en un sistema de 3 cúbits. Por lo tanto, la diferencia entre una medida aleatoria y los resultados de *IBMQ Melbourne* es muy pequeña, del $5{,}5\,\%$.

En la Figura 5.7 se evidencia la diferencia entre los resultados obtenidos para el simulador y los dos ordenadores. Puede además apreciarse que, mientras que para *IBMQ Santiago* aún se destacan los elementos marcados (corresponden a las casillas en la diagonal), para *IBMQ Melbourne* la diferencia entre éstos y el resto de estados es muy ligera.



(a) *IBMQ Santiago* **(b)** *IBMQ Melbourne*

Figura 5.7: Resultados de la implementación del algoritmo de Grover en estados de 2 cúbits.

Finalmente, el histograma comparando la eficiencia de los dispositivos para cada estado marcado puede verse en la Figura 5.8.

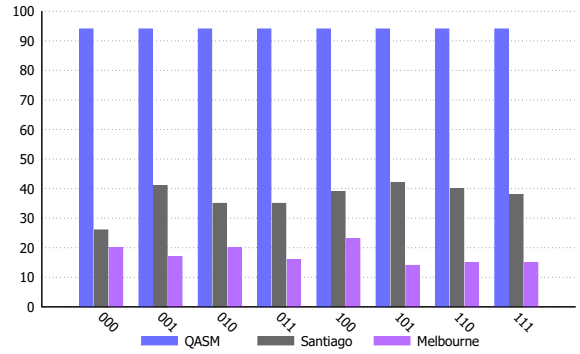


Figura 5.8: Comparativa entre los dispositivos para 3 cúbits.

5.2.3. Resultados de la implementación en 4 cúbits

Para 4 cúbits, el *Simulador QASM* se encuentra el estado marcado en un $(91{,}8 \pm 0{,}6)\,\%$ de las veces, en promedio. De nuevo, la eficiencia del algoritmo se reduce al aumentar el número de cúbits del sistema.

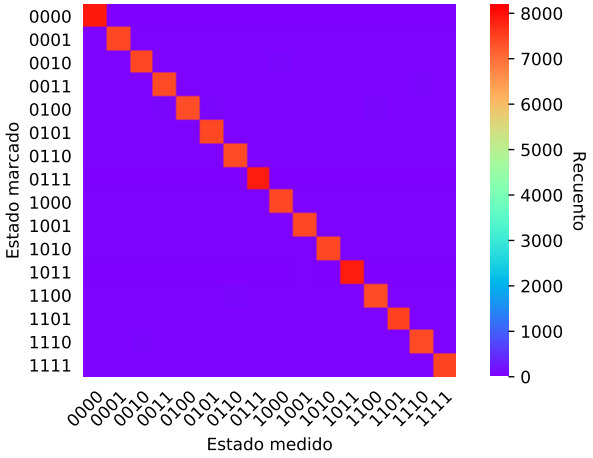
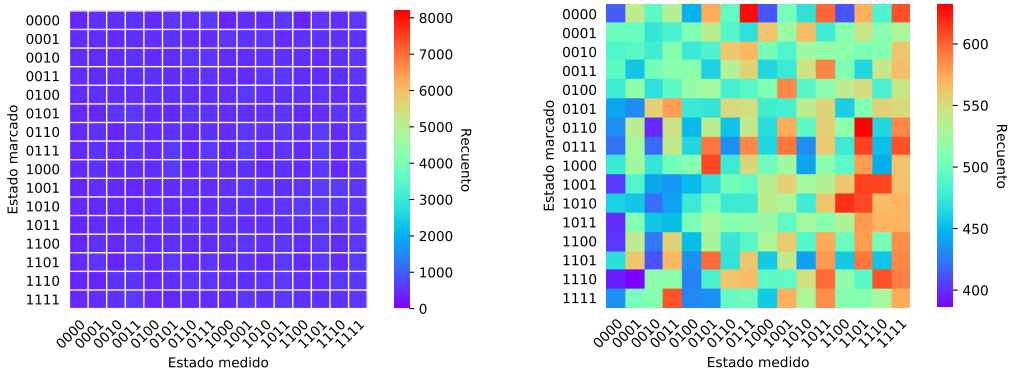


Figura 5.9: Resultados del simulador QASM Simulator

En el caso de los sistemas cuánticos reales, *IBMQ Santiago* y *IBMQ Melbourne* la fiabilidad del algoritmo cae de forma aún más abrupta.

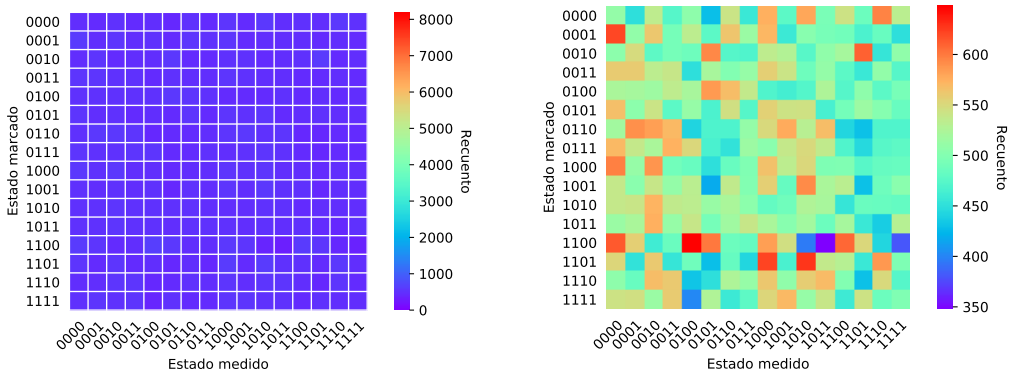
En *IBMQ Santiago*, se obtiene el estado correcto solo un $(6{,}3 \pm 0{,}1)\,\%$ de las veces, el resultado es aparentemente aleatorio. Al considerar una superposición uniforme de estados la probabilidad de medida de cada uno de ellos en un sistema con 4 cúbits es del $6{,}25\,\%$. En contraposición con el simulador (véase Figura 5.9), en la Figura 5.10a la distribución de colores es uniforme, lo que implica que todos los estados son medidos con probabilidad similar. La Figura 5.10b se ha reescalado de forma que puedan observarse mejor las diferencias en el conteo de los estados. No se observa ningún patrón similar al de la Figura 5.9, en la que aparecen en rojo las casillas en la diagonal. Aunque aparentemente hay estados que aparecen con mayor frecuencia que otros, por ejemplo 0000 frente a 1111, las variaciones son demasiado pequeñas como para considerarlas relevantes.



(a) IBMQ Santiago **(b)** Ampliación IBMQ Santiago

Figura 5.10: Resultados del sistema *IBMQ Santiago*.

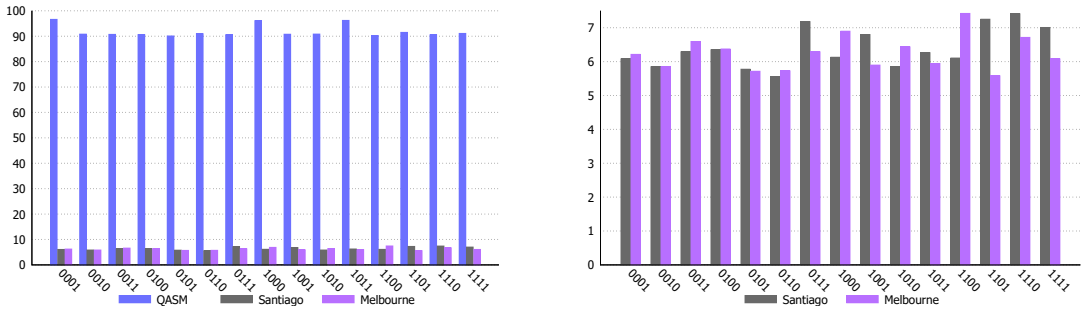
La Figura 5.11 representa los resultados obtenidos en *IBMQ Melbourne*, muy similares a los de *IBMQ Santiago*: el porcentaje promedio de veces en la que el estado marcado y el medido coinciden en este caso es del $(6{,}2 \pm 0{,}1)\,\%$, el mismo que si la medida fuera aleatoria.



(a) IBMQ Melbourne **(b)** Ampliación IBMQ Melbourne

Figura 5.11: Resultados de sistema *IBMQ Melbourne*.

Finalmente, la Figura 5.12a comparar la fiabilidad de los sistema: mientras que la diferencia entre los resultados del simulador *QASM* y los ordenadores es muy marcada, no hay diferencia significativa entre los ordenadores de Santiago y Melbourne (véase 5.12b).



(a) Simulador y ordenadores. **(b)** Ampliación.

Figura 5.12: Comparativa entre los dispositivos.

5.2.4. Conclusiones

A partir de los resultados obtenidos, podemos concluir que los sistemas cuánticos reales no están aún lo suficientemente desarrollados como para implementar el algoritmo de Grover con éxito. La fiabilidad del simulador *QASM* desciende ligeramente con el número de cúbits, pero en los computadores cuánticos el deterioro de los resultados es mucho mayor: en el caso más extremo, para 4 cúbits, la probabilidad de encontrar el estado correcto es la misma que si la medida fuera totalmente aleatoria.

Para comprender este resultado, debemos tener en cuenta la complejidad del circuito. Al aumentar el tamaño del sistema, es necesario realizar una gran cantidad de operaciones sobre los cúbits, especialmente si se quiere realizar el número óptimo de iteraciones del oráculo y el algoritmo de difusión. Esto, además, aumenta el tiempo de ejecución del circuito.

El incremento de las puertas lógicas cuánticas involucradas en el circuito conlleva un aumento del error introducido por las mismas. Los circuitos de 3 y 4 cúbits requieren de la aplicación de una puerta CCZ y CCCZ, respectivamente. Conseguir realizar esta operación correctamente es complicado ya que no todos los cúbits están conectados entre sí, por lo que dicha puerta podría ser también una importante fuente de error.

Para subsanar ese problema, podría ampliarse el circuito haciendo uso de cúbits auxiliares y empleando la construcción de la Figura 5.13 [6]. Sin embargo, el resultado apenas mejora ya que igualmente es necesario implementar el algoritmo en el ordenador de *IBMQ Melbourne*, cuyo volumen cuántico es muy reducido.

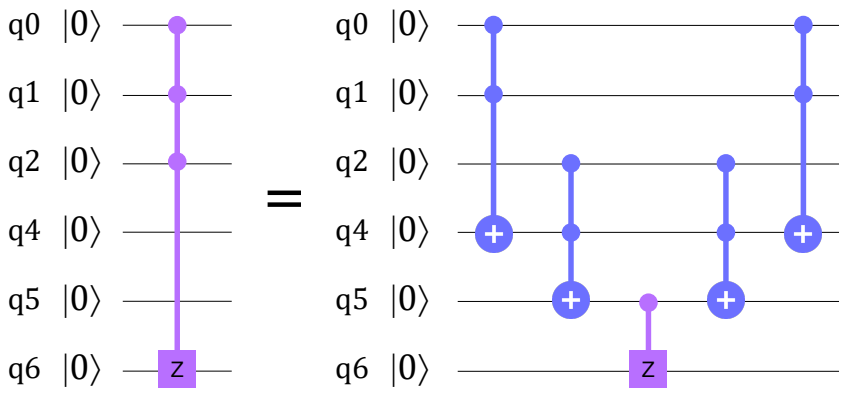


Figura 5.13: Alternativa para la construcción de la puerta CCCZ.Los cúbits 4 y 5 funcionan como cúbits auxiliares para poder realizar la operación. Esta construcción puede generalizarse para cualquier operación C^nU.

Por otra parte, el aumento en el tiempo de ejecución del algoritmo también empeora el resultado debido a la *decoherencia cuántica*. La decoherencia hace referencia a la perturbación de ciertas propiedades del sistema debido a su interacción con el entorno. Aunque los cúbits se mantengan aislados y refrigerados a una temperatura de $15\ \mathrm{mK}$ para minimizar estos efectos, aún tienen una influencia notoria en los circuitos. La decoherencia se presenta de dos formas. Con el transcurso del tiempo los cúbits en el estado más energético, $|1\rangle$, decaen al de menos energía, $|0\rangle$, este fenómeno se denomina *relajación energética*. Asimismo, la fase relativa entre los estados $|0\rangle$ y $|1\rangle$ de un estado en superposición varía con el tiempo, introduciendo un *desfase*.

5.3. Simulador

Al implementar el algoritmo de Grover en los ordenadores de *IBM Quantum*, hemos comprobado que incluso en las condiciones óptimas, dadas por el simulador, el algoritmo falla en un $8{,}2\,\%$ de las veces (en 4 cúbits). Podemos preguntarnos por qué supone una ventaja este algoritmo, si tiene un porcentaje de error tan elevado. La realidad es que el nivel de confianza del mismo aumenta con el tamaño de la base de datos considerada; el objetivo de esta sección es, por medio de un simulador, comprobar cómo evoluciona la fiabilidad del algoritmo a medida que aumenta el tamaño del sistema.

Nos hemos basado en el artículo de D. Candela, «*Undergraduate computational physics projects on quantum computing*» [10]. Siguiendo las indicaciones de la publicación, se ha desarrollado un programa que permite simular en un ordenador convencional el algoritmo de búsqueda de Grover. Para ello, hemos utilizado cálculo matricial. Los programas se han escrito haciendo uso del software *Mathematica*. Los sistemas simulados corresponden a 3, 4, 5, 6, 7 y 8 cúbits.

Comencemos por el caso más sencillo. Simulamos un sistema con $N = 3$ cúbits, por lo que el número de elementos de la base del sistema es $2^3 = 8$. Los estados cuánticos están determinados por la amplitud de probabilidad de los elementos de la base, $\{|\psi_i\rangle\}$. En consecuencia, el estado puede registrarse en un vector de dimensión 8,

$$|\Psi\rangle = \sum_i a_i \, |\psi_i\rangle \equiv (a_1 \; a_2 \; a_3 \; a_4 \; a_5 \; a_6 \; a_7 \; a_8)^{\mathrm{T}} \tag{5.3}$$

donde cada elemento a_i es la amplitud del elemento i de la base. Cabe recordar que los a_i son números complejos.

Los operadores vendrán representados por matrices unitarias de dimensión 8×8.

Podemos generalizar la definición de aquellas puertas cuánticas que actuan esobre un único cúbit para obtener su representación matricial en espacios de dimensión 2^N. La matriz $2^N \times 2^N$ de un operador $\hat{U}$ aplicado sobre el cúbit i es:

$$\hat{U}_{2^N}^{(i)} = \mathbb{I}_2^{(1)} \otimes \mathbb{I}_2^{(2)} \otimes ... \otimes \hat{U}_2^{(i)} \otimes ... \otimes \mathbb{I}_2^{(N)} \tag{5.4}$$

En un sistema de $N = 3$ cúbits, la puerta de Hadamard aplicada a cada uno de los cúbits toma la forma:

$$\hat{H}_8^{(1)} = \hat{H}_2^{(1)} \otimes \mathbb{I}_2^{(2)} \otimes \mathbb{I}_2^{(3)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix} \tag{5.5}$$

$$\hat{H}_8^{(2)} = \mathbb{I}_2^{(1)} \otimes \hat{H}_2^{(2)} \otimes \mathbb{I}_2^{(3)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{pmatrix} \tag{5.6}$$

$$\hat{H}_8^{(3)} = \mathbb{I}_2^{(1)} \otimes \mathbb{I}_2^{(2)} \otimes \hat{H}_2^{(3)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \tag{5.7}$$

Estas matrices pueden generarse en *Mathematica* utilizando el comando `KroneckerProduct[Matriz1, Matriz2]`.

Si queremos aplicar una puerta de Hadamard a los 3 cúbits simultáneamente es más sencillo consturir la puerta

$$\begin{aligned} \hat{H}_8^{(1),(2),(3)} =& \hat{H}_8^{(1)} \ \hat{H}_8^{(2)} \ \hat{H}_8^{(3)} = \hat{H}_2^{(1)} \otimes \hat{H}_2^{(2)} \otimes \hat{H}_2^{(3)} = \\ =& \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}. \end{aligned} \tag{5.8}$$

El circuito para implementar el algoritmo de Grover en el simulador se muestra en la Figura 5.14.

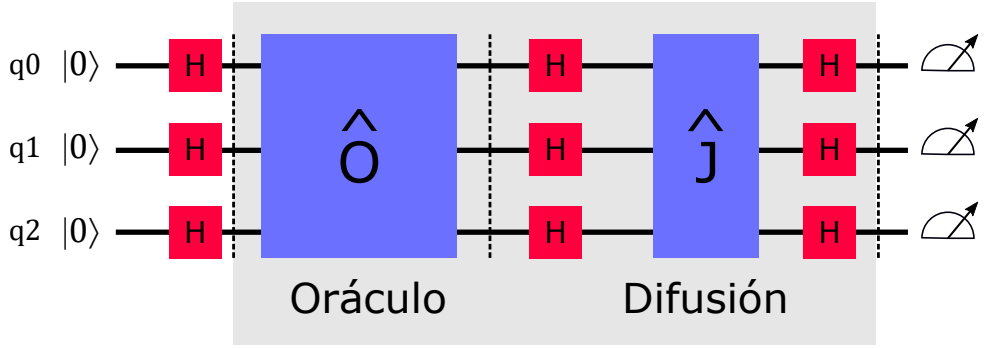


Figura 5.14: Esquema del algoritmo de Grover en el simulador.

Las matrices asociadas al operador *oráculo* y el operador de difusión de Grover se definen a continuación.

La matriz del *oráculo* es una matriz diagonal con todos los elementos de la diagonal igual a 1 a excepción del correspondiente al elemento de la base buscado. Por ejemplo, en caso de buscar el tercer elemento de la base, la matriz asociada al operador será:

$$\hat{\mathrm{O}} = \mathrm{diag}\,(1, 1, -1, 1, 1, 1, 1, 1) \tag{5.9}$$

El operador $\hat{J}$ del algoritmo de difusión de Grover es una matriz diagonal con todos sus elementos igual a uno a excepción del primero, que es igual a -1:

$$\hat{\mathrm{J}} = \mathrm{diag}\,(-1, 1, 1, 1, 1, 1, 1, 1) \tag{5.10}$$

Debemos definir también un procedimiento para realizar la medida del estado del sistema. La probabilidad de medida de un estado de la base viene determinada por el módulo al cuadrado de su amplitud. Para realizar la medida, se define un vector con el módulo de las amplitudes al cuadrado. A continuación, se define un vector de probabilidad acumulada, que definirá un rango de probabilidad entre 0 y 1 para cada estado. Se genera un número aleatorio entre 0 y 1, y se comprueba a qué intervalo del vector de probabilidad acumulada pertenece. El intervalo en el que se encuentre dicho número será el resultado de la medida. El código completo puede consultarse en el Código 5.1.

Código 5.1: Programa de *Mathematica* para simular el algoritmo de Grover en 3 cúbits.

```
Resultado = 3 (*Selecciona el estado marcado*)
Oracle = IdentityMatrix[8]
Oracle[[Resultado, Resultado]] = -1
J = IdentityMatrix[8]
J[[1, 1]] = -1
GDO = HadamardMatrix[8] . J . HadamardMatrix[8] (*Operador de difusion
   de Grover*)
Grover = GDO . Oracle (* Primero se aplica el oraculo, despues el
   operador de difusion *)
Resultado={0,0,0,0,0,0,0,0} (*Recuento*)
Do[
Psi={1,0,0,0,0,0,0,0};
Psi=HadamardMatrix[8] . Psi;
Do[ Psi = Grover.Psi, 2]; (* Dos iteraciones *)
(* Medida *)
Prob=Table[Abs[Psi[[i]]^2],{i,1,8}];
Do[Prob[[i]]=Prob[[i-1]]+Prob[[i]],{i,2,8}];
x=RandomReal[1];
Indice=FirstPosition[Prob,SelectFirst[Prob,#>x&]];
Resultado[[Indice]]=Resultado[[Indice]]+1
,10000] (* Numero de iteraciones del programa *)
Print[Resultado]
```

Podemos extender el programa a sistemas con un mayor número de cúbits modificando la dimensión de las matrices y el número de iteraciones del algoritmo. En este trabajo, se ha extendido a sistemas de 4, 5, 6, 7 y 8 cúbits.

5.3.1. Resultados

En la Figura 5.15 pueden verse los resultados obtenidos. En el eje de ordenadas se representa el número de cúbits del sistema simulado, y en el vertical la eficiencia del algoritmo. El número de iteraciones del código ha sido de 1000 en cada caso.

Atendiendo a los resultados obtenidos, hemos comprobado que cuando el número de cúbits aumenta, aumenta también la fiabilidad del algoritmo. Al extender el Código 5.1 a sistemas de 4, 5, 6 y 7 cúbits, hemos comprobado que la fiabilidad aumenta hasta un $99{,}7\,\%$ para sistemas de 5 cúbits.

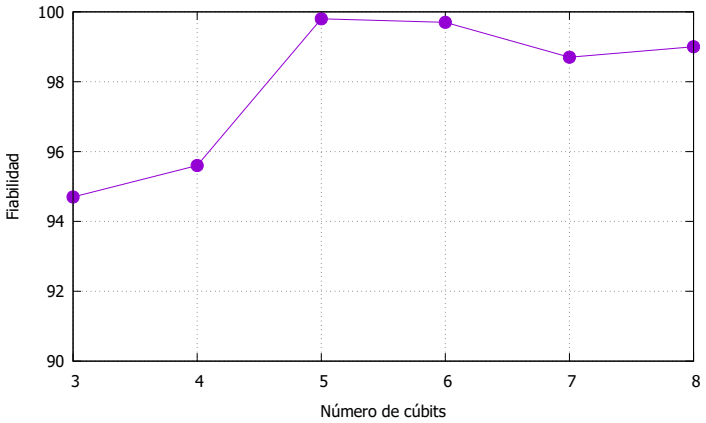


Figura 5.15: Fiabilidad del algoritmo en función del número de cúbits del sistema.

5.3.2. Conclusiones

El simulador programado es útil para pruebas de concepto, pero, sin embargo, sus aplicaciones son limitadas. A medida que aumenta el tamaño del sistema, el tamaño de las matrices aumenta de forma exponencial, incrementando el tiempo de ejecución del algoritmo y la memoria de almacenamiento requerida. A partir de 8 cúbits el tiempo de ejecución aumenta significativamente, ya que las matrices son de dimensión 256×256.

Capítulo 6

Conclusiones globales

El objetivo de este trabajo era caracterizar las propiedades de los sistemas cuánticos de *IBM Quantum Experience*, centrándonos en el análisis del entrelazamiento y la implementación del algoritmo de Grover.

Para el análisis del entrelazamiento proporcionado por las computadoras, hemos comenzado por la prueba experimental del teorema de Bell. Después, hemos realizado una serie de circuitos cuánticos que han permitido probar la violación de las desigualdades de *Mermin-Klyshko* en los dipositivos de *IBMQ Vigo* e *IBMQ Santiago*, ambos de 5 cúbits. A pesar de que los resultados están lejos de ser óptimos, se aprecia una clara mejoría con respecto a los obtenidos en 2016 por Alsina et al. [18]. Los sistemas no consiguen entrelazar sus 5 cúbits correctamente, pero las teorías locales quedan descartadas en estos dispositivos. Las diferencias entre los resultados obtenidos en este trabajo y los obtenidos por Alsina et al. en 2016 son solo una muestra de los recientes avances y el esfuerzo realizado por la comunidad científica y el sector de la ingeniería en la disciplina de la computación cuántica.

En relación al algoritmo de Grover, hemos explicado su funcionamiento y las ventajas del algoritmo frente a los algoritmos de búsqueda clásicos, ya que reduce significativamente el número de operaciones requeridas para dar con el resultado. Por medio de su simulación en un computador clásico, hemos comprobado que el algoritmo tiene una alta fiabilidad para sistemas grandes. En cuanto a la implementación del algoritmo de búsqueda de Grover en los sistemas cuánticos de *IBMQ*, se han implementado los circuitos para 2, 3 y 4 cúbits. A medida que aumenta el tamaño del sistema analizado, se aprecia un claro deterioro de los resultados, especialmente para los sistemas cuánticos reales: no son aún capaces de ejecutar el algoritmo con éxito. El deterioro de los resultados del simulador indica que debemos asimismo mejorar el diseño de los circuitos, ideando alternativas eficientes para la realización de las puertas lógicas que requieran 3 cúbits o más.

A pesar de sus limitaciones, cabe destacar que *IBM Quantum Experience* es una buena herramienta para iniciarse en el campo de la computación cuántica. El interfaz de usuario es sencillo de utilizar y existe gran cantidad documentación de fácil acceso que sirve como apoyo en el proceso de aprendizaje, especialmente en la etapa inicial.

Con todo esto, podemos concluir que la capacidad computacional de los dispositivos analizados es aún muy limitada, y no permiten realizar con ellos tareas complejas. Aún son necesarios más avances en el desarrollo de los ordenadores cuánticos para explotar todo su eventual potencial y poder implantarlos a escala global.

Bibliografía

[1] A. M. Turing, «On Computable Numbers, with an Application to the Entscheidungsproblem,» *Proceedings of the London Mathematical Society*, vol. s2-42, n.º 1, págs. 230-265, 1937, DOI: 10.1112/plms/s2-42.1.230.

[2] P. Benioff, «The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines,» *Journal of Statistical Physics*, vol. 22, págs. 563-591, 1980, DOI: 10.1007/BF01011339.

[3] L. K. Grover, «A Fast Quantum Mechanical Algorithm for Database Search,» en *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ép. STOC '96, Association for Computing Machinery, 1996, 212–219, ISBN: 0897917855, DOI: 10.1145/237814.237866.

[4] P. W. Shor, «Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,» *SIAM Journal on Computing*, vol. 26, n.º 5, 1484–1509, 1997, ISSN: 1095-7111, DOI: 10.1137/s0097539795293172.

[5] W. R. Inc., *Mathematica, Version 12.2 (Student edition)*, Champaign, IL, 2021, dirección: https://www.wolfram.com/mathematica.

[6] M. A. Nielsen e I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th. USA: Cambridge University Press, 2011, ISBN: 1107002176.

[7] *Operations glossary*, Último acceso: Junio 2021, dirección: https://quantum-computing.ibm.com/composer/docs/iqx/operations_glossary#operations-glossary.

[8] M. Mc Gettrick y B. Murphy, «Simulation of the CCC-Not Quantum Gate,» *Technical Report NUIG-IT-061002*, dirección: https://www.researchgate.net/publication/238508097_Simulation_of_the_CCC-Not_Quantum_Gate.

[9] *IBM Quantum*, Último acceso: Junio 2021, dirección: https://quantum-computing.ibm.com/.

[10] D. Candela, «Undergraduate computational physics projects on quantum computing,» *American Journal of Physics*, vol. 83, n.º 8, págs. 688-702, 2015, DOI: 10.1119/1.4922296.

[11] Qiskit, *What Is Quantum Volume, Anyway?* Último acceso: Junio 2021, dirección: https://medium.com/qiskit/what-is-quantum-volume-anyway-a4dff801c36f.

[12] *Quantum services*, Último acceso: Junio 2021, dirección: https://quantum-computing.ibm.com/services?systems=all.

[13] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres y W. K. Wootters, «Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels,» *Phys. Rev. Lett.*, vol. 70, págs. 1895-1899, 13 mar. de 1993, DOI: 10.1103/PhysRevLett.70.1895.

[14] C. H. Bennett y S. J. Wiesner, «Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states,» *Phys. Rev. Lett.*, vol. 69, págs. 2881-2884, 20 nov. de 1992, DOI: 10.1103/PhysRevLett.69.2881.

[15] A. K. Ekert, «Quantum cryptography based on Bell's theorem,» *Phys. Rev. Lett.*, vol. 67, págs. 661-663, 6 ago. de 1991, DOI: 10.1103/PhysRevLett.67.661.

[16] J. S. Bell, «On the Einstein Podolsky Rosen paradox,» *Physics Physique Fizika*, vol. 1, págs. 195-200, 3 nov. de 1964, DOI: 10.1103/PhysicsPhysiqueFizika.1.195.

[17] N. D. Mermin, «Bringing home the atomic world: Quantum mysteries for anybody,» *American Journal of Physics*, vol. 49, n.º 10, págs. 940-943, 1981, DOI: 10.1119/1.12594.

[18] D. Alsina y J. I. Latorre, «Experimental test of Mermin inequalities on a five-qubit quantum computer,» *Physical Review A*, vol. 94, n.º 1, 2016, ISSN: 2469-9934, DOI: 10.1103/physreva.94.012314.

[19] D. Collins, N. Gisin, S. Popescu, D. Roberts y V. Scarani, «Bell-Type Inequalities to Detect Truen-Body Nonseparability,» *Physical Review Letters*, vol. 88, n.º 17, 2002, ISSN: 1079-7114, DOI: 10.1103/physrevlett.88.170405.

[20] N. Brunner, D. Cavalcanti, S. Pironio, V. Scarani y S. Wehner, «Bell nonlocality,» *Reviews of Modern Physics*, vol. 86, n.º 2, 419–478, abr. de 2014, ISSN: 1539-0756, DOI: 10.1103/revmodphys.86.419.

[21] R. Ross, «Computer simulation of Mermin's quantum device,» *American Journal of Physics*, vol. 88, n.º 6, págs. 483-489, 2020, DOI: 10.1119/10.0000833.

Apéndice A

Valores y vectores propios de los operadores de Mermin

En este apéndice se describe el procedimiento a seguir para la construcción de los operadores de Mermin en *Mathematica*, y la obtención de sus valores y vectores propios.

A.1. Definición

Recordemos la definición recursiva de los operadores de Mermin, partiendo de $M_1 = X_1$:

$$M_n = \frac{1}{2}\left(M_{n-1}\left(X_n + Y_n\right) + M'_{n-1}\left(X_n - Y_n\right)\right), \tag{A.1}$$

Los M'_N se obtienen a partir de los M_N, cambiando los operadores X por Y, y viceversa.

Mediante la regla de recurrencia de la Ecuación A.1, podemos construir en *Mathematica* de forma sencilla las matrices $2^N \times 2^N$ asociadas a cada operador, como ilustra el ejemplo del Código A.1. Los valores y vectores propios se obtienen con el comando `Eigensystem`. Si solo se desea obtener los valores propios basta con el comando `Eigenvalues`, y para los vectores propios, `Eigenvectors`.

Código A.1: Programa de *Mathematica* para simular el algoritmo de Grover en 3 cúbits.

```
(*Definicion de operadores X e Y *)
X = {{0, 1}, {1, 0}}
Y = {{0, -I}, {I, 0}}
(*Definicion de los operadores, los MN' pasan a estar denominados como
   MNp *)
M1 = X
M1p = Y
(*M2*)
M2 = KroneckerProduct[M1,(X+Y)/2]+[M1p, KroneckerProduct[(X-Y)/2]
M2p = KroneckerProduct[M1p,(X+Y)/2]+KroneckerProduct[M1,(Y-X)/2]
Eigensystem[M2]
(*M3*)
M3 = KroneckerProduct[M2,(X+Y)/2]+[M2p, KroneckerProduct[(X-Y)/2]
M3p = KroneckerProduct[M2p,(X+Y)/2]+KroneckerProduct[M2,(Y-X)/2]
Eigensystem[M3]
(*M4*)
M4 = KroneckerProduct[M3,(X+Y)/2]+[M3p, KroneckerProduct[(X-Y)/2]
M4p = KroneckerProduct[M3p,(X+Y)/2]+KroneckerProduct[M3,(Y-X)/2]
```

```
Eigensystem[M4]
(*M5*)
M5 = KroneckerProduct[M4,(X+Y)/2]+[M4p, KroneckerProduct[(X-Y)/2]
M5p = KroneckerProduct[M4p,(X+Y)/2]+KroneckerProduct[M4,(Y-X)/2]
Eigensystem[M5]
```

A.2. Matrices asociadas a los operadores

Los resultados se han obtenido con el código de *Mathematica* en Código A.1.

A.2.1. Estados de 1 cúbit

$$M_1 \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad M_1' \doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{A.2}$$

- Valores propios: $\{-1, 1\}$
- Vectores propios: $\left\{\left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right), \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\right\}$

A.2.2. Estados de 2 cúbits

$$M_2 \doteq \begin{pmatrix} 0 & 0 & 0 & 1-i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1+i & 0 & 0 & 0 \end{pmatrix} \quad M_2' \doteq \begin{pmatrix} 0 & 0 & 0 & -1-i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1+i & 0 & 0 & 0 \end{pmatrix} \tag{A.3}$$

- Valores propios: $(-\sqrt{2}, \sqrt{2}, 0, 0)$
- Vectores propios: $\left\{\left(-\frac{1-i}{2}, 0, 0, \frac{1}{\sqrt{2}}\right), \left(\frac{1-i}{2}, 0, 0, \frac{1}{\sqrt{2}}\right), (0, 0, 1, 0), (0, 1, 0, 0)\right\}$

A.2.3. Estados de 3 cúbits

$$\mathrm{M}_3 \doteq \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2i \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} M_3' \doteq \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{A.4}$$

- Valores propios: $\{-2, 2, 0, 0, 0, 0, 0, 0\}$
- Vectores propios:

$\{(\frac{i}{\sqrt{2}}, 0, 0, 0, 0, 0, 0, \frac{1}{\sqrt{2}}), (-\frac{i}{\sqrt{2}}, 0, 0, 0, 0, 0, 0, \frac{1}{\sqrt{2}}), (0, 0, 0, 0, 0, 0, 1, 0), (0, 0, 0, 0, 0, 1, 0, 0),$
$(0, 0, 0, 0, 1, 0, 0, 0), (0, 0, 0, 1, 0, 0, 0, 0), (0, 0, 1, 0, 0, 0, 0, 0), (0, 1, 0, 0, 0, 0, 0, 0)\}$

A.2.4. Estados de 4 cúbits

$$
\mathsf{M}_4 \doteq \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2-2i \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2+2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{A.5}
$$

$$
\mathsf{M'}_4 \doteq \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2+2i \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2-2i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{A.6}
$$

- Valores propios: $\left\{-2\sqrt{2}, 2\sqrt{2}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\right\}$
- Vectores propios:

$$
\begin{aligned}
\{ &\left(\frac{1+i}{2}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \frac{1}{2}\right), \\
&\left(-\frac{1+i}{2}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \frac{1}{\sqrt{2}}\right), \\
&(0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0), (0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0), \\
&(0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0), (0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0), \\
&(0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0), (0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0), \\
&(0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0), (0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0), \\
&(0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0), (0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0), \\
&(0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0), (0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0), \\
&(0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0), (0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)\}
\end{aligned}
$$

A.2.5. Estados de 5 cúbits

$$
\mathsf{M}_5 \doteq \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{A.7}
$$

$$
\mathsf{M'}_5 \doteq \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -4i & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 4i & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{A.8}
$$

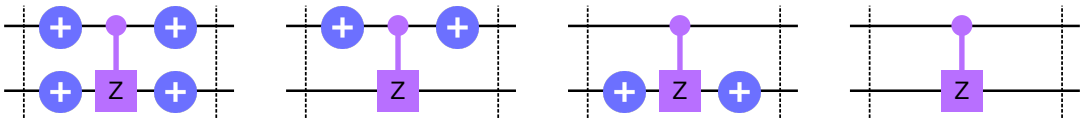
- Valores propios: $\{-4, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$
- Vectores propios:

$$\{\left(\frac{1}{\sqrt{2}}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \frac{1}{\sqrt{2}}\right),$$
$$\left(-\frac{1}{\sqrt{2}}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \frac{1}{\sqrt{2}}\right),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),$$
$$(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)\}$$

Apéndice B

Oráculo de Grover

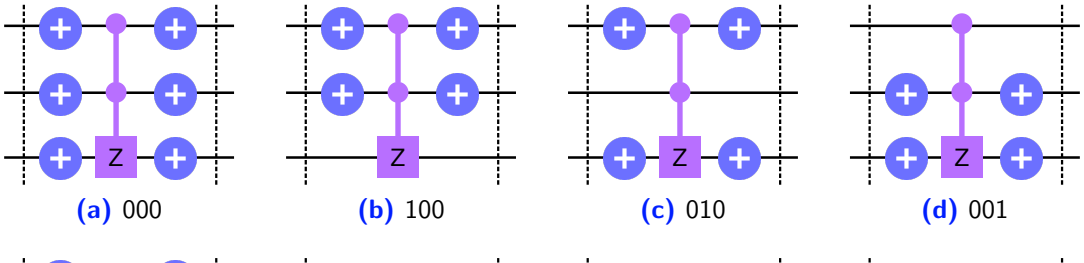
B.1. 2 cúbits



(a) 00 **(b)** 10 **(c)** 01 **(d)** 11

Figura B.1: Circuitos del oráculo de Grover en $N = 2$ qubits.

B.2. 3 cúbits

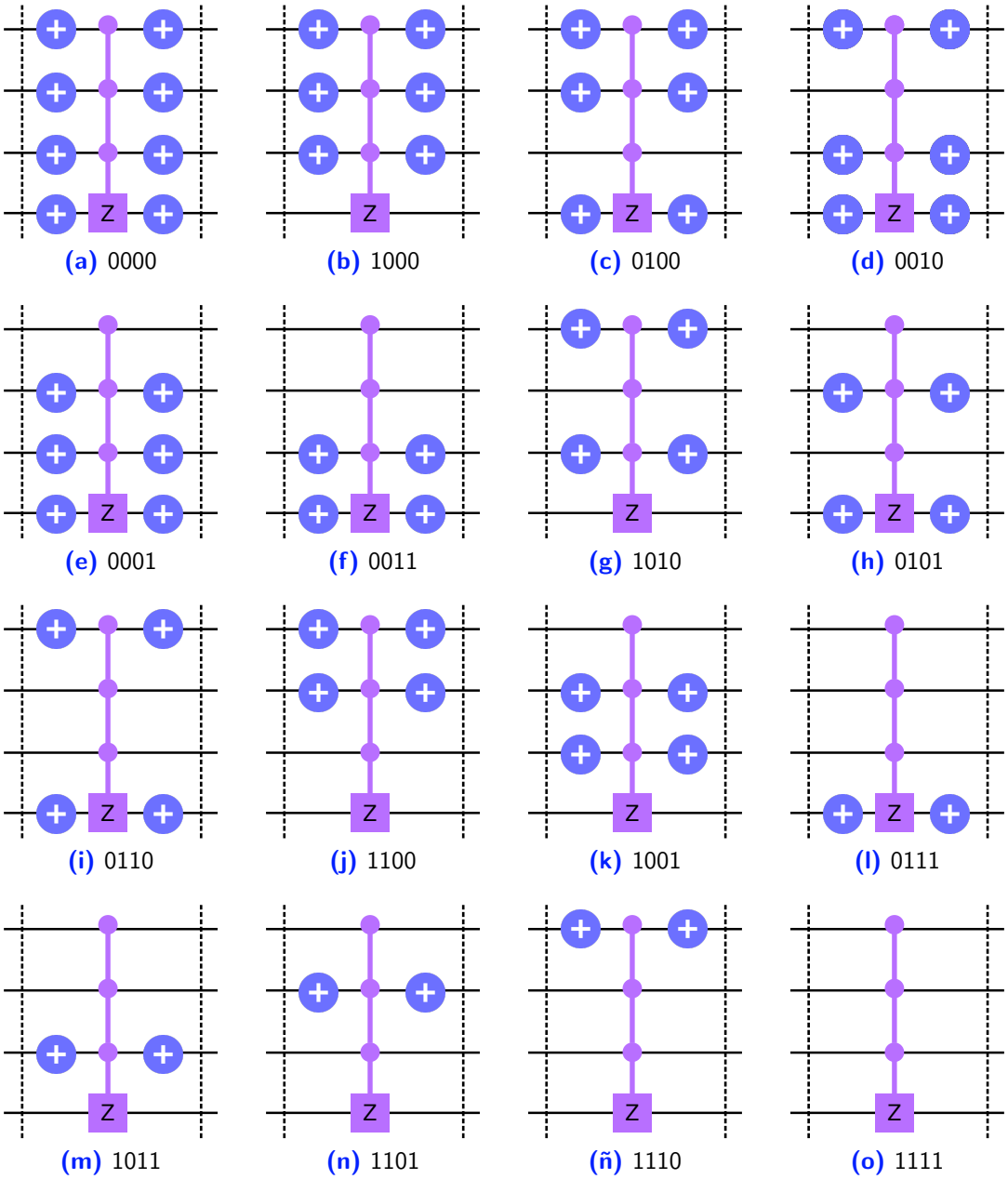


(a) 000 **(b)** 100 **(c)** 010 **(d)** 001

(e) 110 **(f)** 101 **(g)** 011 **(h)** 111

Figura B.2: Circuitos del oráculo de Grover en $N = 3$ qubits.

B.3. 4 cúbits



(a) 0000 **(b)** 1000 **(c)** 0100 **(d)** 0010

(e) 0001 **(f)** 0011 **(g)** 1010 **(h)** 0101

(i) 0110 **(j)** 1100 **(k)** 1001 **(l)** 0111

(m) 1011 **(n)** 1101 **(ñ)** 1110 **(o)** 1111

Figura B.3: Circuitos del oráculo de Grover en $N = 4$ qubits.

Apéndice C

Algoritmo de Grover: circuito completo

C.1. 3 cúbits

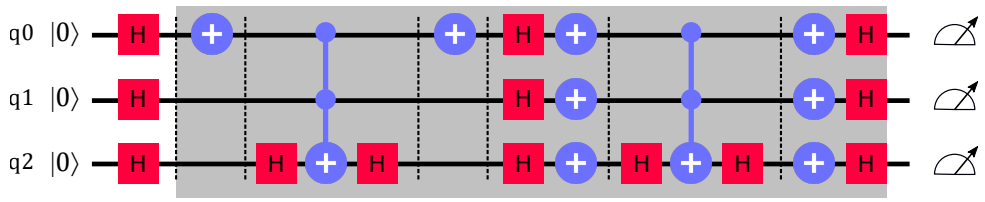


Figura C.1: Circuito completo para 3 cúbits. La zona sombreada debe repetirse 2 veces.

C.2. 4 cúbits

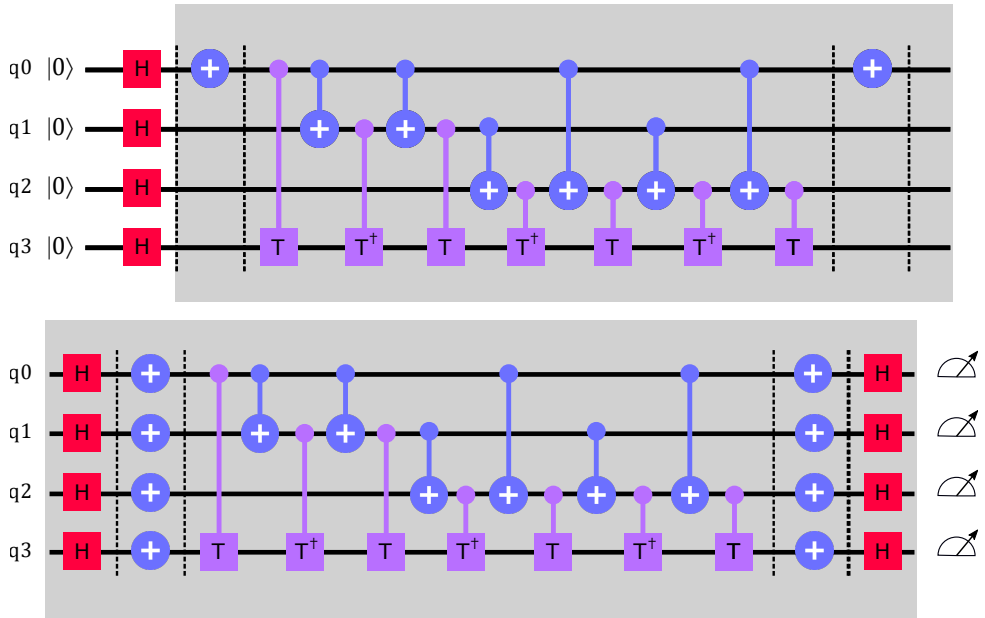


Figura C.2: Circuito completo para 4 cúbits. La zona sombreada debe repetirse 3 veces.