

Universidad Nacional de Educación a Distancia
Facultad de Ciencias
Departamento de Física Interdisciplinar



Memoria del Trabajo de Fin de Grado en Física
ESTUDIO DE ALGORITMOS CUÁNTICOS Y RELACIÓN DE
LOS MISMOS CON LA FÍSICA

Raúl Osuna Sánchez-Infante
Tutor: Víctor Alberto Fairén Le Lay
Curso 2020/21

**DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO,
PARA LA DEFENSA DEL TRABAJO FIN DE GRADO**

Fecha: 25/06/2021

Quien se suscribe:

Autor: **RAÚL OSUNA SÁNCHEZ-INFANTE**
D.N.I.: **72.738.833-E**

Hace constar que es la autor(a) del trabajo:

Título completo del trabajo.

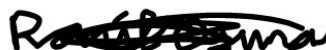
**ESTUDIO DE ALGORITMOS
CUÁNTICOS Y RELACIÓN DE LOS
MISMOS CON LA FÍSICA**

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y la elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores, se han referenciado debidamente en el texto de dicho trabajo.

DECLARACIÓN:

- ✓ Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente.
- ✓ Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.
- ✓ No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.

Fdo.



Resumen

Este trabajo estudia el estado del arte de la computación cuántica, tras una detallada introducción en base matemática, computacional y física, haciendo especial hincapié en ésta última. Una vez introducidos los conceptos, se centra en la resolución de algunos tipos de problemas mediante algoritmos cuánticos, que harán uso de ciertas propiedades especiales de la mecánica cuántica, como son la superposición, el entrelazamiento y la interferencia, que aplicadas a estos problemas resultan en una gran mejora, a veces incluso exponencial, de la velocidad de resolución de los mismos. Por último, la base del trabajo se centra en la implementación de algunos ejemplos de estos algoritmos, como son el algoritmo de Deutsch-Josza para la demostración de la viabilidad de estos algoritmos cuánticos, así como el algoritmo de Grover, que tiene una aplicación real como es la búsqueda en una lista desestructurada. Se emplea el lenguaje Python para la codificación de los algoritmos, ya que la existencia en el mismo de la librería Qiskit permite implementar fácilmente los circuitos cuánticos necesarios. Se analizarán los resultados tanto teóricos como aplicados a hardware cuántico real, a cuyo acceso existe la posibilidad hoy en día gracias a servicios en la nube. A modo de corolario, se discutirán las limitaciones de estos algoritmos y posibles soluciones a las mismas.

Abstract

This document studies the state of the art of quantum computing, after a detailed introduction of some mathematic, computational and physic bases, insisting specially on the latter. Once the concepts have been introduced, the work will be focused in the resolution of certain kinds of problems by quantum algorithms, which will make use of certain special properties of quantum mechanics, such as superposition, entanglement and interference, which applied to these problems, result in a great speed up of the resolution times, even exponentially sometimes. Last but not the least, the base of this study is targeted towards the implementation of a few examples of such algorithms, namely Deutsch-Josza algorithm in order to demonstrate the viability of quantum algorithms, together with Grover's search algorithm, which has a real application for the search in an unstructured list. Python language will be used to code the algorithms, as its library Qiskit allows to easily implement all the needed quantum circuits. The theoretical results will be studied, together with its application to real quantum hardware, whose use was possible thanks to the existance of tools in the cloud that allowed the access to such a hardware. As a summary, the limitations of these algorithms will be presented and possible solutions suggested.

Índice

1. Introducción	5
1.1. Fundamentos matemáticos y lógicos. Computación clásica	5
1.1.1. Trigonometría	5
1.1.2. Historia de la computación clásica [2]	5
1.1.3. Álgebra de Boole y puertas lógicas clásicas [4]	5
1.1.4. Vectores, números complejos y matrices	6
1.2. Bases físicas, fundamentos y particularidades de la computación cuántica	6
1.2.1. Historia de la computación cuántica [7]	6
1.2.2. Analogía con la computación clásica	7
1.3. Mecánica cuántica y aplicación a algoritmos [18]	10
1.3.1. Principios de la mecánica cuántica	10
1.3.2. Sistemas cuánticos de dos valores y medición. Entrelazamiento cuántico.	11
1.3.3. Postulados de la mecánica cuántica [24]	12
1.3.4. El qubit y la esfera de Bloch [?, 25]	13
1.4. Fundamentos computacionales [18]	14
1.4.1. Niveles de abstracción. Computación clásica vs cuántica	14
1.4.2. El modelo de circuito cuántico	16
1.4.3. Formulación matemática de los circuitos cuánticos	17
1.4.4. Algoritmos	20
2. Objetivos	22
3. Métodos y material	22
3.1. Lenguaje Python para la programación de los algoritmos.	22
3.2. Librería Qiskit [33]	22
3.3. IBM Quantum Experience [34]	22
3.4. Control de versiones Github [35]	22
3.5. Procesador de textos LyX [36]	23
4. Resultados	23
4.1. Algoritmo de Deutsch-Josza [23]	23
4.1.1. Caso constante	23
4.1.2. Caso balanceado	23
4.1.3. Definición del oráculo	24
4.2. Algoritmo de Grover [23]	26
4.2.1. Aplicación a dos qubits [37]	26
4.2.2. Aplicación a tres qubits [38]	28
5. Discusión	31
5.1. Discusión de los errores en el hardware real	31
5.2. Limitaciones de los algoritmos cuánticos [23, 40, 41]	32
6. Conclusiones	32
7. Anexos	33
7.1. Código fuente	33
8. Agradecimientos	33
9. Bibliografía	33

1. Introducción

1.1. Fundamentos matemáticos y lógicos. Computación clásica

En esta sección se presenta un conjunto de conceptos matemáticos y lógicos de una forma muy resumida. En caso de ser necesario un mayor grado de detalle, se recomienda consultar la bibliografía referenciada en cada sección o subsección, ya que la mayoría de ellos se han empleado únicamente como herramientas. La descripción detallada de cada una de esas herramientas incrementaría la densidad del trabajo y podría retirar la atención del tema fundamental del mismo.

1.1.1. Trigonometría

Debido a la dualidad onda-materia observada en la mecánica cuántica por De Broglie [?], a menudo se tratará con ondas. Por ello un dominio básico de la trigonometría es importante para tratar con dichas ondas. Estos conocimientos son básicos en un grado de física (incluso antes del comienzo del mismo) y se consideran dominados, por lo que no se entrará en más detalle al respecto.

1.1.2. Historia de la computación clásica [2]

La computación se basa en el empleo de una máquina con la que realizar ciertos cálculos, mediante los cuales se obtendrán unas salidas a partir de unas entradas.

Se considera a menudo a la «Máquina de Turing» como la primera computadora (u ordenador, como se suele denominar en España), diseñada para descifrar las comunicaciones alemanas durante la Segunda Guerra Mundial (1939).

Posteriormente, en 1946 en los Estados Unidos, se desarrolló «ENIAC» (Electronic Numerical Integrator And Computer). Se componía de tubos de vacío y fue el primer ordenador de uso general. Ocupaba el espacio completo de una enorme habitación y tenía como tarea calcular tablas de tiro de artillería.






Desde esos primeros desarrollos hasta la actualidad, esta computación, a la que se referirá en adelante como «computación clásica» ha experimentado un desarrollo increíble. Sin embargo, existe la creencia que dicho desarrollo exponencial según la ley de Moore [3], que establece que cada dos años se duplica el número de transistores en un microprocesador, parece estar llegando a un límite de saturación. Al necesitarse más y más transistores en el mismo espacio, la escala en la que se trabaja es cada vez menor, llegándose al punto donde los efectos cuánticos tienen una influencia que deja de ser despreciable.

1.1.3. Álgebra de Boole y puertas lógicas clásicas [4]

Los ordenadores clásicos trabajan con un sistema binario, como podría ser: «sí» ó «no», «corriente» ó «no corriente», «tensión nula» ó «tensión a cierto valor», o cualquier par de valores que se pueda imaginar. Como resumen, se asignará a este par de valores la numeración de «0» ó «1». Esta forma de representación no es muy conveniente para los humanos a la hora de representar números, ya que se tiende a pensar de una forma decimal. Resulta por tanto importante saber manejar esta representación binaria, operar con ella y transformar cuando sea necesario a o desde formato decimal. La ventaja de implementar los ordenadores con un lenguaje binario radica en la facilidad para construir el hardware y la mayor velocidad de las operaciones como resultado.

- Lógica de Boole: se emplearán las operaciones «AND» (que corresponde con el producto binario), «OR» (que corresponde con la suma binaria) y «NOT» (que no es más que un cambio de bit). Dichas operaciones se pueden combinar, produciendo otras más complejas como la «NAND» (NOT+AND), «NOR» (NOT+OR). Existe una operación adicional, conocida como «OR exclusivo» ó «XOR». Todas estas operaciones tienen sus puertas lógicas correspondientes, que se construyen a partir de transistores. Tanto la representación como la construcción se pueden consultar en la bibliografía adjunta [17]. En la figura 1 se adjunta una tabla descriptiva, que también añade lo que se conoce como «tabla de la verdad», que no es más que el análisis de la salida en función de la(s) entrada(s).

- The package **Truth Tables** and **Boolean Algebra** set out the basic principles of logic.

Name	Graphic Symbol	Boolean Algebra	Truth Table															
AND		$F = A \cdot B$ Or $F = AB$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = A + B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$F = \bar{A}$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	
NAND		$F = \overline{A \cdot B}$ Or $F = \overline{AB}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = \overline{A + B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																

the symbols, algebra signs and the truth table for the gates

Figura 1: Puertas lógicas: nombres, símbolos, álgebra booleana y tablas de la verdad [5]

- Universalidad de la computación: con este término se hace referencia a que se podrá implementar el ordenador adecuado para cada cálculo únicamente a partir de ciertas puertas lógicas.
- Reversibilidad [6]: con esta propiedad se analiza si la puerta lógica (y con ello, el ordenador como conjunto de puertas lógicas) preserva la información original (reversible) o no (no reversible). Este concepto cobra importancia en la computación cuántica y será analizado al introducir los postulados de la mecánica cuántica en la sección 1.3.3.

1.1.4. Vectores, números complejos y matrices

Como se explicará en la siguiente sección, al pasar de una representación unidimensional de la computación clásica con los bits, a una bidimensional con los qubits, resultan de especial utilidad los vectores y los números complejos para dicha representación. Existe una formulación matemática (de especial complejidad), que si bien no ha sido necesaria para codificar los algoritmos, está presente y en ocasiones resulta útil para poder tener otro punto de vista, especialmente en el caso de disponer de una capacidad de cálculo matemático superior a la de la comprensión de los fenómenos físicos. En dichos casos, una primera aproximación matemática permite comprender mejor los fenómenos físicos (aunque la intuición haga pensar que el estudio del fenómeno físico puede ser más sencillo, no lo es para estos casos tan difíciles).

Por tanto, los vectores son la representación matemática de los qubits, los cuales pueden tener componentes complejas. Las matrices (también con elementos complejos) son simplemente la representación matemática de las puertas cuánticas.

1.2. Bases físicas, fundamentos y particularidades de la computación cuántica

1.2.1. Historia de la computación cuántica [7]

A modo de brevísima introducción, se enumeran algunos de los hitos de la computación cuántica.

- En 1981, Richard Feynman propuso un marco para la simulación de la evolución de los sistemas cuánticos.
- En 1984, Peter Shor demuestra que los ordenadores cuánticos son capaces de factorizar eficientemente números enteros de gran magnitud. Esta factorización es empleada en la actualidad para diversos mecanismos de cifrado. El solo hecho de poder realizar dichas factorizaciones mucho más eficientemente, pondría en jaque los mecanismos actuales de cifrado, tan empleados hoy en día en cualquier ámbito de la computación, en cualquier parte de Internet o incluso en archivos almacenados de una forma desconectada.

- En 1998 tuvo lugar la primera demostración de un algoritmo cuántico. Se trataba de un ordenador cuántico NMR (resonancia magnética nuclear) de 2 qubits.
- En 2012 se aumenta progresivamente el número de qubits, añadiéndose asimismo algoritmos de detección de errores.
- En 2017 se ponen a disposición ordenadores cuánticos en la nube, dando la posibilidad de acceso a mucha más gente. Un ejemplo de esto sería el empleado en ciertas secciones de los algoritmos desarrollados en este trabajo, y que se detallan en la sección 5.
- En septiembre de 2019, Google anunció haber demostrado la supremacía cuántica (postulado que afirma que un ordenador cuántico puede resolver ciertos problemas para los cuales no existe actualmente solución en una cantidad finita de tiempo [9]) en el Financial Times [10]. Esta demostración fue publicada posteriormente en Nature [11], apenas un mes después (y por cierto, con un español en el equipo, Sergio Boixo), no sin cierta controversia con IBM, que argumenta que los resultados de dichos estudios podrían alcanzarse con supercomputadores actuales (siendo suyo el más potente hoy en día, el Summit) [12, 13, 14]. Sin embargo, en diciembre del mismo año también se habría demostrado la supremacía cuántica en el USTC (China) [15], siendo ya dos fuentes distintas las que confirmarían su existencia.

1.2.2. Analogía con la computación clásica

El hecho de haber introducido anteriormente la computación clásica tenía un fundamento, que no era otro que comparar ciertas partes análogas de la computación cuántica con la actual (clásica).

En primer lugar, la unidad de información en la computación cuántica pasa a ser el qubit. Si anteriormente en la computación clásica se utilizaban los bits, «0» y «1», ahora se pasa a emplear los qubits o bits cuánticos, tomando los mismos los valores de $|0\rangle$ y $|1\rangle$, donde se ha empleado la notación bra-ket formulada por Dirac [16], expresando los qubits como un «ket», que no es otra cosa que un vector columna, información que es conocida de los temarios de Física Cuántica.

Un concepto importante de que representan los qubits es el de la superposición de estados. Si bien un bit sólo puede tomar el valor de «0» ó «1», un qubit puede hallarse en cualquier superposición de ambos estados, tomando la forma:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Sóloamente al medir dicho qubit $|\psi\rangle$ se medirá uno de dichos valores (se tiende a expresar este concepto diciendo que el qubit «colapsa» a un valor de «0» ó «1» al realizar la medición). Los valores de las constantes α y β , que pueden ser complejos, tienen una representación física, y no es otra que el hecho de que el cuadrado de sus respectivos módulos coincide con la probabilidad de que la medición del qubit colapse al valor del qubit al que preceden (multiplican). Por tanto se cumple:

$$|\alpha|^2 + |\beta|^2 = 1$$

Una superposición equiprobable de ψ estaría dada por la expresión:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

Mientras que una distribución con distintas probabilidades (75 % y 25 %) sería la siguiente:

$$|\psi_2\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$$

Las puertas lógicas de la computación clásica también tienen su analogía en el mundo cuántico, y se denominan, como no podría ser de otra manera «puertas cuánticas». A continuación enunciaremos las puertas cuánticas básicas, así como algunas adicionales que se han empleado en los algoritmos de este trabajo:

- Puerta X o cambio de qubit: esta puerta cambia su qubit de entrada al opuesto en la salida.

- Puerta Z o cambio de fase: deja inalterada la salida si el qubit es $|0\rangle$, y añade un signo negativo si la entrada es $|1\rangle$
- Puerta H de Hadamard: crea una superposición de estados equiprobables, de forma que:

$$H(|0\rangle) = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \quad (1)$$

$$H(|1\rangle) = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \quad (2)$$

Estas puertas se representan simplemente como una rectángulo con la letra correspondiente.

En la figura se pueden ver las puertas enunciadas, así como otros ejemplos. Además, se añade la matriz correspondiente para cada puerta, que tendrá importancia si se quiere hacer una representación matemática del circuito.














Operator	Gate(s)	Matrix
Pauli-X (X)	 	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP	 	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Figura 2: Puertas cuánticas, símbolos y matrices asociadas [17]

Destacar que las puertas se aplican a cada estado de una posible superposición, y por tanto:

$$X(|\psi\rangle) = X(\alpha|0\rangle + \beta|1\rangle) = \alpha(X|0\rangle) + \beta(X|1\rangle) = \alpha|1\rangle + \beta|0\rangle$$

Una puerta interesante y que se ha empleado en este trabajo es la CNOT, (not controlado). Esta puerta, que toma como entradas dos qubits, denomina qubit de control al de la parte superior, y qubit objetivo (target) al inferior. Si el qubit de control es 0, la puerta no hace nada al qubit objetivo, pero si es 1, el valor del bit de objetivo cambiará. El bit de control siempre se mantiene igual a su salida.

La computación cuántica hace uso extensivo de tres propiedades principales de la física/mecánica cuántica que se detallarán más adelante, siendo las mismas:

1. Superposición
2. Entrelazamiento
3. Interferencia

Se puede introducir ya el efecto de alguna de estas propiedades, como la superposición (que ya se ha explicado con la puerta H) y la interferencia: el resultado de un experimento «clásico» de lanzar una moneda al aire dos veces nos dará un 50 % de probabilidad para cada uno de los resultados, cara y cruz. En el mundo cuántico, tras aplicar dos veces una superposición equiprobable (puerta H), sobre un qubit con estado inicial $|0\rangle$, el resultado final va a ser el mismo qubit $|0\rangle$ (con una probabilidad del 100 % por tanto). Es decir, los estados implicados en una superposición cuántica se pueden cancelar o amplificar, lo cual es análogo al experimento de la rendija estudiado en óptica [19], y de hecho tiene la misma explicación, al estar tratando también con una onda según el principio de dualidad onda-materia. Con este experimento de la doble puerta de Hadamard además se puede apreciar una característica adicional: la reversibilidad, de la que ya se había hablado anteriormente. En la computación cuántica, es necesario que las puertas sean reversibles (como se explicará en la sección 1.3.3 de postulados de la mecánica cuántica).

A modo de breve introducción respecto al entrelazamiento, decir que se trata de una correlación entre varios qubits, donde el estado de un qubit, depende del estado de otro qubit. A modo de explicación sencilla para dos qubits, partiendo del estado:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

Se observa que la probabilidad de medir cada uno de los pares $|00\rangle$ y $|11\rangle$ es la misma y del 50 %. Pero si se mide solamente uno de los qubits, el primero, al conocer su valor, inmediatamente se sabe el valor del otro qubit (que en este caso, será idéntico).

Para la creación del estado cuántico de entrelazamiento de este ejemplo, ψ_1 , el circuito sería tan sencillo como partir de dos qubits inicializados a cero, aplicar una puerta H al primero de ellos y posteriormente una puerta CNOT en la que el qubit de control sería también el primero (al que se le había aplicado la puerta H). Esto se puede ver en la figura 3.

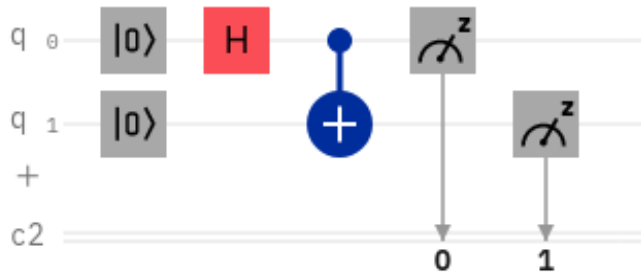


Figura 3: Circuito que implementa el entrelazamiento cuántico para 2 qubits

A partir de estas tres propiedades se podrían buscar más aplicaciones a la computación cuántica, las cuales no serán de estudio en este trabajo, siendo algunas de ellas simplemente enunciadas en la siguiente lista:

1. Teleportación cuántica
2. Criptografía cuántica (BB84)
3. Codificación superdensa

Por último, y en este caso esto es algo idéntico a la computación cuántica, destacar que para la codificación de 2^N posibles estados, será necesario disponer de N qubits.

1.3. Mecánica cuántica y aplicación a algoritmos [18]

1.3.1. Principios de la mecánica cuántica

La mecánica cuántica trata de describir objetos a pequeña escala (microscópica, del orden del tamaño de los átomos). Esto es incompatible con las observaciones que percibimos en el día a día, ya que muestra efectos que escapan de nuestra intuición, tales y como las tres propiedades expuestas en el apartado anterior (superposición, entrelazamiento e interferencia). La computación cuántica tratará de hacer uso de propiedades como las anteriores para resolver ciertos problemas computacionales que los ordenadores clásicos son incapaces de resolver a día de hoy. Es por ello necesario tener una base de mecánica cuántica. Sin embargo, resulta curioso fijarse que no es necesario conocer las particularidades de los semiconductores si se quiere emplear la computación tradicional.

A modo de introducción, se ha de recordar que la materia es dual, tal y como se ha comentado anteriormente y como expuso De Broglie [?]. Las partículas y las ondas son entidades distintas: las partículas tienen una posición y momento definidos, y son discretas. Cuando las partículas colisionan, interaccionan según los postulados conocidos de la física clásica (choques elásticos o inelásticos). Sin embargo, las ondas no tienen una posición definida, y dejan de ser discretas para pasar a ser continuas. Cuando dos ondas interaccionan (chocan) no rebotan como en un choque elástico, sino que «interfieren», pudiendo ser esta interferencia en los casos más extremos «constructiva» (una onda con el doble de amplitud) o «destruktiva» (la onda resultante se anula), tal y como se puede apreciar en el experimento de la doble rendija [19], ampliamente conocido. La parte ondulatoria de la luz se puede demostrar con este experimento, mientras que la parte corpuscular se demuestra a partir del efecto fotoeléctrico [20].

Demostrada esta dualidad de la luz, se puede aplicar la misma a cualquier partícula, sin más que aplicar el experimento de la doble rendija a un haz de electrones: si los mismos fueran sólo materia, cabría esperar simplemente dos franjas en la pantalla del experimento, sin embargo se verá un patrón de interferencia con franjas de distinta intensidad, exactamente igual que el mismo experimento realizado con un haz de luz. El resultado de que las partículas sean también ondas es importante, ya que implica que no van a tener una posición fija. En su lugar, estarán definidas por una función de onda, habitualmente denominada como $\psi(x, t)$, que no es capaz de determinar la posición exacta de la misma, sino que dará una estimación, o dicho de otra forma, una función de probabilidad para la posición (y esto mismo ocurre para otras magnitudes, como podría ser el momento). Los qubits se comportan como ondas, y los estados de superposición de los mismos pueden interferir los unos con los otros. Un ejemplo de aplicación sería el de amplificar la probabilidad de la respuesta correcta en algoritmo de búsqueda (como el de Grover que se estudiará en este trabajo), dando lugar a una aceleración cuántica en la búsqueda de la misma.

Para la representación matemática de estas ondas (o mejor dicho, la función de probabilidad de su posición) se empleará el llamado «operador Hamiltoniano», que permitirá encontrar la energía de los estados de energía permitidos de esta función de onda. Este Hamiltoniano, en este contexto, se trata de un operador cuántico que da información acerca de la energía de un estado cuántico, o dicho de otra forma: $\hat{H}|\psi\rangle = E|\psi\rangle$. Los autovalores de esta expresión corresponden a soluciones permitidas para los valores de energía que se pueden tomar en la ecuación de ondas. Esta energía dispone de una componente cinética y otra potencial. La forma más fácil de interpretar estas componentes es asociarlas al mundo clásico, donde la primera se asocia a la energía debida a una velocidad y otra debida a la gravedad (por la altura/distancia a la que se encuentra respecto a la base de referencia, como podría ser la Tierra). En el caso que nos atañe, la energía cinética es debida a la velocidad de las partículas (electrones) y la potencial no sería debida a la gravedad, sino a energía potencial eléctrica dependiente de la distancia al núcleo en este caso. La forma de la energía sería la siguiente: $E = \frac{p^2}{2m} + V(x)$, en donde la variable p se refiere al momento de una partícula, estando relacionado el mismo con la velocidad mediante la fórmula $p = mv$.

Mediante la ecuación de Schrödinger [20, 21] es posible describir cómo un estado cuántico (o el estado de un qubit, que es en lo que se centra este trabajo) cambia con el tiempo en función de sus valores de energía:

$$i\frac{\hbar}{2\pi}\frac{\partial}{\partial t}|\psi\rangle = \hat{H}|\psi\rangle$$

En la expresión anterior, \hbar se refiere a la constante de Planck ($\hbar = 6,626 \cdot 10^{-34} J \cdot s$) e i es la unidad imaginaria. A menudo se utiliza en la expresión anterior la constante de Planck reducida, o constante de Dirac, a fin de simplificar la expresión, y conocida como: $\hbar = \frac{\hbar}{2\pi} = 1,055 \cdot 10^{-34} J \cdot s$.

Esta ecuación de Schrödinger permite saber cómo se comporta el hardware cuántico empleado, así como conocer cómo operarlo. Permite asimismo diseñar e implementar puertas cuánticas. Resulta por último interesante nombrar

que la mecánica cuántica se puede aplicar a otros campos distintos a los de la computación cuántica, tales y como serían otros campos de la física, la química o incluso campos a priori tan distintos como la economía.

1.3.2. Sistemas cuánticos de dos valores y medición. Entrelazamiento cuántico.

Se refieren a sistemas cuánticos compuestos con dos valores, como podrían ser los qubits o el spin de un electrón. Para el qubit, se podría poner como ejemplo el caso de superposición antes expuesto, de $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, a diferencia de un posible sistema de tres niveles, como podría ser $|\psi'\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle$. El motivo de elegir con únicamente dos niveles y no, por ejemplo, tres, es que se puede asignar muy fácilmente un nivel a 0 y el otro a 1, es algo a lo que ya se está acostumbrado de los sistemas clásicos y resulta fácil de controlar. Las propiedades algebraicas de estos sistemas son mucho más simples, y por último y no menos importante, estos sistemas de dos niveles han sido estudiados durante casi un siglo.

¿Cómo se conoce que el spin de un electrón es un ejemplo de un sistema cuántico de dos valores? Para responder a esta pregunta, se utilizará el experimento de Stern-Gerlach [22]. En este experimento se usaban imanes, los cuales creaban un campo magnético, y se contraponía el resultado de hacer pasar pequeños imanes a través de estos imanes (caso clásico) respecto a hacerlo con electrones (caso cuántico). Resumiendo el experimento, y con ánimo de no complicar en exceso el estudio, se llegaba a la conclusión que mientras que en el caso clásico de los imanes, los mismos se desviaban siguiendo un espectro continuo de opciones, en el caso de los electrones el resultado sólo tomaba dos opciones distintas (como por ejemplo «arriba» o «abajo»).

Una aplicación práctica interesante que surge a partir de estos niveles de dos niveles no es otra que la propiedad introducida anteriormente conocida como «entrelazamiento». Este entrelazamiento cuántico se define como la correlación cuántica entre dos o más objetos, en los cuales cada uno de sus estados depende del estado del otro (o de los otros). Anteriormente ya se expuso un ejemplo de estado entrelazado con dos qubits, otra opción sería la siguiente:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Si se mide el primer qubit, inmediatamente se conoce el valor del segundo. El entrelazamiento cuántico tiene la propiedad de que los qubits entrelazados se mantendrán así si se aplica un operador únicamente a uno de ellos. Por ejemplo, en el estado anterior, si se aplica una puerta X (recuérdese que esta puerta tiene como resultado el qubit opuesto al de la entrada) al primer qubit, el estado continua entrelazado:

$$|\psi'\rangle = \frac{1}{\sqrt{2}}(|11\rangle + |00\rangle)$$

Es inmediato ver cómo si se mide únicamente uno de los dos qubits (cualquiera), el valor del otro sigue siendo conocido inmediatamente.

Al conjunto de posibles estados entrelazados con dos qubits se les conoce como estados de Bell [23]:

- $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = |+\rangle$
- $|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$
- $|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) = |-\rangle$
- $|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

El entrelazamiento cuántico tiene aplicaciones reales, sin embargo no se ha empleado en ninguno de los algoritmos implementados en este trabajo. Alguno de los ejemplos serían:

- Teleportación cuántica
- Criptografía cuántica (BB84)
- Codificación superdensa

1.3.3. Postulados de la mecánica cuántica [24]

Resulta interesante proponer un marco de trabajo matemático para la mecánica cuántica, mediante los siguientes postulados:

1. Un estado cuántico se representa mediante un «ket» en el espacio de estados.
 - a) Analogía clásica: se puede crear una función clásica para expresar el movimiento.
 - b) Consecuencia: la superposición de dos estados cuánticos cualesquiera no es más que otro estado cuántico.
2. Los observables clásicos se introducen en la mecánica cuántica empleando operadores. Específicamente, cualquier observable (propiedad que pueda ser medida) de un sistema físico se describe mediante un operador que actúa sobre los kets de estado.
 - a) Analogía clásica: en la descripción del movimiento, se puede medir (exactamente) la velocidad, dirección, posición, energía, momento... En el caso cuántico, se aplica un operador a la función de onda para obtener esos valores. Por ejemplo, para medir la energía cinética, se aplicaría el operador correspondiente a la función de onda $|\psi\rangle$. El empleo del verbo «aplicar», se refiere a la operación de producto escalar.
 - b) Consecuencia: es posible relacionar observables que pueden ser medidos al mundo cuántico.
3. El resultado de la medida de un observable con un operador \hat{A} será únicamente un autovalor de \hat{A} .
 - a) Analogía clásica: resulta complicado establecer una, ya que esto se trata de un concepto exclusivamente cuántico. Podríamos pensar en una cinta caminadora, habitual en los gimnasios, en la que velocidad se introduce en intervalos de 0,5 km/h. Las velocidades posibles sólo tomarían los valores de 0km/h, 0,5km/h, 1km/h, 1,5km/h, etc...
 - b) Consecuencia: la multiplicación matriz/vector se simplifica a una multiplicación escalar vector: $A\vec{v} = \lambda\vec{v}$, y se puede determinar los posibles resultados de una medición de un observable.
4. Al realizar una medida de un observable con un operador \hat{A} en un estado genérico de $|\psi\rangle$, la probabilidad de obtener un cierto autovalor a_i corresponde al cuadrado del producto escalar de $|\psi\rangle$ con el correspondiente autovalor: $|\langle a_i|\psi\rangle|^2$.
 - a) Analogía clásica: al realizar la medida de la velocidad en el caso anterior, cada uno de los resultados tiene una cierta probabilidad ya establecida.
 - b) Consecuencia: es posible conocer la probabilidad de medir cada una de los posibles valores de una medida.
5. Inmediatamente después de medir un observable A con un valor a_n , el estado del sistema es el autovalor normalizado $|a_n\rangle$.
 - a) Analogía clásica: antes de medir la velocidad de la caminadora, ésta no era conocida. Sin embargo, una vez medida, la velocidad queda establecida a ese valor.
 - b) Consecuencia: el valor medido en un estado de superposición «colapsa» (o dicho de otra manera, «queda establecido como») a un estado concreto una vez realizada la medida.
6. Los estados cuánticos generalmente evolucionan con el tiempo. Esta evolución temporal preserva la normalización del estado y es descrita mediante la expresión: $|\psi(t)\rangle = \hat{U}(t)|\psi(0)\rangle$, para cierto operador unitario \hat{U} .
 - a) Analogía clásica: cambiando la caminadora estática por un corredor real, conocida la posición final y las condiciones de la carrera (quién es el corredor, qué tiempo hacía, etc), se podría estimar el punto de comienzo de dicha carrera.
 - b) Consecuencia: la evolución temporal del estado cuántico es reversible.

En la computación cuántica estos postulados permiten:

- Dar instrucciones sobre cómo operar el hardware cuántico.
- Predecir qué ocurrirá tras una medida en el circuito cuántico.
- Calcular la probabilidad de obtener la respuesta adecuada basándose en el conjunto de puertas cuánticas en las que se ejecuta el algoritmo

1.3.4. El qubit y la esfera de Bloch [?, 25]

El qubit es la unidad básica para la construcción de ordenadores cuánticos. Se trata de un sistema de dos niveles y puede ser cualquier superposición de ambos valores (es decir, que puede ser 0 y 1 a la vez, con distintas probabilidades para cada valor). Resulta muy conveniente una representación del qubit mediante números complejos, asignando las partes reales e imaginarias, a los valores de los dos niveles, $|0\rangle$ y $|1\rangle$. La representación de un número complejo se puede hacer mediante notación cartesiana, con partes reales e imaginarias separadas, o mediante forma polar, representando módulo y fase.

El estado de un qubit se puede representar también como un punto en la «esfera de Bloch», que no es más que una esfera de radio unidad y que se muestra en la figura 4. El equivalente al «polo Norte» (como analogía respecto de la Tierra) sería el estado $|0\rangle$, y el polo Sur, el $|1\rangle$, que son los únicos puntos posibles para un bit clásico. A partir de la notación polar, y similarmente al uso de las coordenadas esféricas, se puede expresar un estado cualquiera como:

$$|\Psi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2) \cdot e^{i\varphi}|1\rangle \quad (3)$$

La explicación de por qué se pasa de dos dimensiones (notación polar) a tres, se basa en que en el qubit, las constantes que expresan el porcentaje de superposición, α y β , pueden ser complejas, y la variable φ se refiere a la diferencia de fase entre ambas constantes. Resulta importante mencionar que en la computación cuántica esta fase relacionada con la variable φ no va a tener ningún significado físico y se puede ignorar.

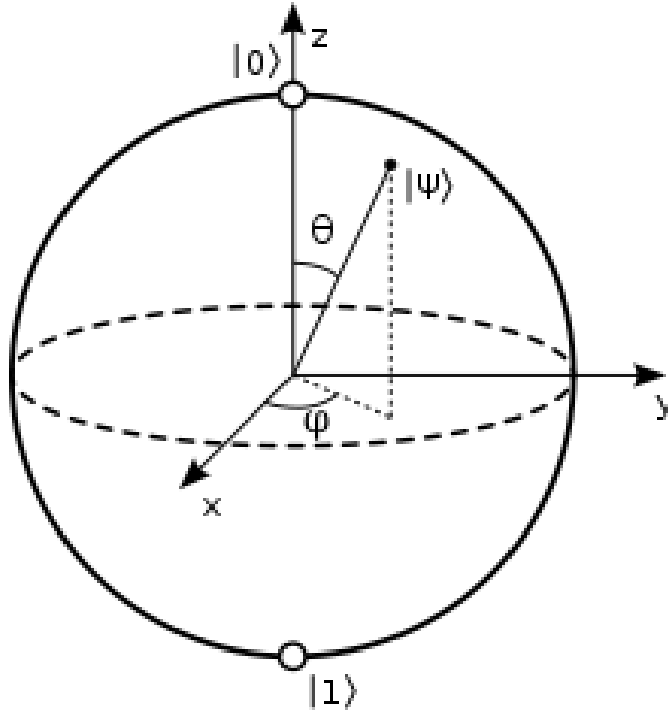


Figura 4: Esfera de Bloch [25]

Algunos puntos interesantes de la esfera de Bloch, además de los anteriormente expuestos de los polos, serían los puntos correspondiente a $|+\rangle$ y $|-\rangle$, que corresponden a sustituir los valores de $\theta = \pi/2$ y $\varphi = 0, \pi$ en la ecuación

de la Esfera de Bloch (3). Gráficamente serían los puntos anterior (eje X) y posterior (eje -X) del ecuador de la esfera de Bloch.

La relación de las puertas cuánticas con la esfera de Bloch consiste en asignar un punto destino en la esfera de Bloch a un punto origen. Estas puertas cuánticas se pueden representar mediante matrices, como ya se introdujo anteriormente. Ejemplos simples serían las puertas X, Y y Z, conocidas también como operadores de Pauli. Dichas puertas, interpretadas geométricamente, no hacen otra cosa que realizar una rotación respecto al eje correspondiente de 180° . Por ejemplo, aplicando una puerta de Pauli-X al estado $|0\rangle$ se obtendrá $|1\rangle$ (rotación de 180° respecto al eje X), o al aplicar al mismo estado original una puerta Pauli-Z se obtendrá el punto original, mediante una rotación de 180° respecto del eje Z.

Estas rotaciones de 180° pueden ser realmente de cualquier valor de ángulo, y estarían representadas por unas puertas distintas.

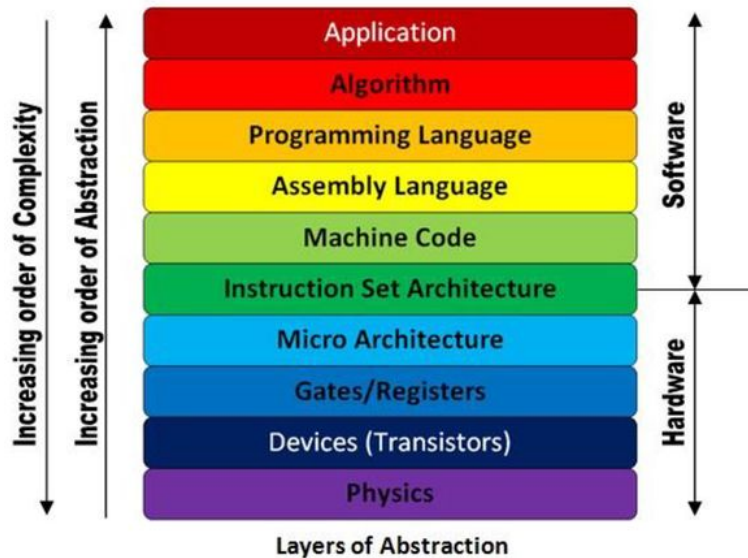
Es ahora cuando se puede volver al concepto de «universalidad», que si se recuerda no es otra cosa que la posibilidad de representar cualquier operación cuántica como una cierta combinación de puertas. Las puertas universales son: CNOT, H, S y T, siendo S una rotación de $\pi/2$ respecto del eje Z, $S^2 = Z$, y T una rotación una rotación de $\pi/4$ también respecto del eje Z, con $T^2 = S$.

1.4. Fundamentos computacionales [18]

1.4.1. Niveles de abstracción. Computación clásica vs cuántica

Antes de plantearse cómo funciona la computación cuántica, resulta útil hacerse la misma pregunta para la computación clásica. Un gráfico permite hacer un gran resumen y situarse de una manera sencilla, tal y como se puede ver en la figura 5:

Abstract View of a Computer



8

Figura 5: Niveles de abstracción de la computación clásica [26]

Visto de otra forma más compacta, existen tres componentes principales (tal y como se puede ver en la división de la derecha): el software, el hardware y la interconexión de los mismos, que también se podría llamar «compilador clásico» (un compilador es un traductor de un lenguaje de programación, o en ciertos casos, ensamblador, a código máquina que es ejecutado sobre el hardware mediante una serie de instrucciones implementadas en el mismo).

En el caso de la computación cuántica, al menos hoy en día, se siguen empleando ordenadores clásicos en la pila de abstracción. Así, la capa del software estaría dividida (de arriba a abajo), en software cuántico y software clásico. El compilador clásico no se sustituye, y el hardware clásico pasaría a estar dividido de nuevo en dos partes al igual que el software: un hardware clásico de control, que controlaría la última parte, el hardware cuántico. Se hará a continuación una breve descripción de cada una de las cinco partes enunciadas, de una forma ordenada:

1. Hardware cuántico: se trata del ordenador cuántico en sí. Similarmente a como sucedía en los primeros días de la computación clásica, este hardware ocupa mucho espacio y se encuentra en habitaciones gigantes. Suele necesitar además mantenerse a temperaturas muy bajas, cercanas al cero absoluto, para proporcionar estabilidad a los qubits de cara al posible ruido. Existen diversas tecnologías empleadas en las diferentes partes del hardware cuántico, que no son parte de este estudio. Se pueden nombrar algunas de las mismas, tal y como serían la resonancia magnética nuclear, los iones atrapados, la superconducción, los centros de diamantes NV, la fotónica o los átomos neutros, entre otros.
2. Hardware clásico de control (del hardware cuántico): por ejemplo, los qubits superconductores son controlados

por medio de secuencias de pulsos de microondas. Al introducir diferentes formas y longitudes para los pulsos, se generan puertas cuánticas distintas. Una sala de control de hardware cuántico no tiene mucha diferencia con su análogo de los años 50, salvando la distancia. O poniendo una analogía más habitual para el público general, una de las primeras centralitas telefónicas.

3. **Compilador:** en este caso, no hay gran diferencia. El compilador empleado va a seguir siendo un compilador clásico, ya que la capa anterior es la que se encarga de traducir al mundo cuántico.
4. **Software clásico:** se trata del lenguaje de programación junto a cualquier tipo de librerías empleado. En el caso de este trabajo, se empleará el lenguaje Python. Asimismo, existen librerías de una gran utilidad que son utilizadas a menudo, tal y como las librerías SciPy o Numpy, a fin de simplificar la programación de ciertas operaciones matemáticas y científicas.
5. **Software cuántico:** aquí es donde entra en especial la librería Qiskit, mediante la cual se puede codificar en código la implementación de puertas cuánticas. Existen otras opciones, como la implementación gráfica con las herramientas de la web de IBM Quantum Experience. Aunque éstas se han explorado, no se usaron para la codificación de los algoritmo de este trabajo. En la sección de métodos y material se detalla todo esto con más detención.

1.4.2. El modelo de circuito cuántico

Se trata de un concepto teórico empleado en la computación cuántica y consta de tres componentes fundamentales: estados, puertas y medidas. Los postulados de este modelo se pueden resumir en los siguiente puntos:

- Los estados cuánticos se reducen a qubits, dados por la expresión: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, o puesto en forma de vector: $|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$
- Los estados cuánticos múltiples se representan como cadenas de qubits: $|0101000111\dots\rangle$
- Las operaciones cuánticas (matrices) son las puertas de la computación cuántica.
- Al contrario que para las medidas clásicas, que son deterministas, las mediciones cuánticas son probabilísticas-
- Para comprobar la distribución probabilística de un experimento en la práctica, es necesario realizar las mediciones en numerosas ocasiones a fin de obtener una estadística.

En cuanto a las componentes enunciadas, se puede destacar los siguiente de ellas:

- **Estados**
 - Se pasa del concepto clásico de «interruptor» al de «dial», por ejemplo el mando con el que se controla la tensión de un osciloscopio, para el mundo cuántico.
- **Puertas**
 - En el apartado (1.1.3) se hizo referencia a las puertas clásicas, que no son más que una representación de cierto hardware que se puede implementar en su mayoría mediante transistores. Es más, como se ha podido ver en la primera sección de este punto de fundamentos computacionales, todos los programas escritos son compilados hasta obtener lenguaje máquina, que simplemente ejecuta estas operaciones lógicas booleanas en cadenas de bits. Como se vio anteriormente, cada puerta tiene una relación entrada/salida dada por una tabla de la verdad.
 - Para el mundo cuántico, estas puertas se representan matemáticamente como matrices. Las operaciones afectando a un único qubit se pueden representar como rotaciones en la esfera de Bloch, y estas operaciones no son otra cosa que las puertas cuánticas. Por supuesto, existen también puertas que afectan a múltiples qubits al mismo tiempo.

- Las puertas en un circuito se representan por recuadros (cajas), habitualmente con un símbolo de puerta asociado, tal y como H, X, Y, Z, etc...
- Medidas
 - En el mundo clásico no se suele hablar demasiado de medidas para estos sistemas deterministas. Esto es debido a que, corrección de errores aparte, el resultado de la medición es conocido a priori. Por ejemplo, el resultado de la medición a la salida de una puerta NOT, cuando la entrada es 0, será de 1 en un 100 % de las ocasiones (obviando un cierto error en los aparatos de medición).
 - Sin embargo, en el mundo cuántico, la medición es probabilística. Por ejemplo, el resultado a la salida de una puerta H cuando la entrada es $|0\rangle$, será a veces $|0\rangle$, a veces $|1\rangle$. Es necesario realizar un estudio de probabilidad, mediante la aplicación de la ley de los grandes números, para poder llegar a un porcentaje realista. En el caso del ejemplo, será de un 50 %-50 %.
 - La medición se representa con un símbolo de medición mostrado como una flecha en un dial. Se verán ejemplos en la demostración de los algoritmos implementados.
 - A partir de estas mediciones, se mostrará un diagrama con las probabilidades de cada posible resultado.

Se puede destacar que hasta ahora se ha hablado de dos analogías para caracterizar los qubits: diales y la esfera de Bloch. Se puede discutir que la definición de la esfera de Bloch constituiría entonces una complicación innecesaria, cuando parece posible caracterizar los qubits mediante una analogía más sencilla. El origen de esta complicación viene de comienzos del siglo XX. Por aquel entonces, había una gran expectación respecto a la computación analógica, que empleaba, curiosamente, diales. Sin embargo, para la década de los 60 se abandonó esta idea debido principalmente a la incapacidad para corregir errores debidos al ruido. Este problema se trató extensamente en las décadas de los 40 y 50 con la introducción de la teoría de la información [27], por Claude Shannon y el desarrollo de un sistema clásico de detección/corrección de errores [28], por Richard Hamming, todo esto ya de vuelta a nuestro mundo digital.

Análogamente, en los 60, Peter Shor desarrolló un estudio similar hasta dar con un método de corrección cuántica de errores [29]. Shor fue más conocido por el algoritmo de factorización [29] que lleva su nombre, el cual proporciona una aceleración exponencial en la factorización de números en factores primos, la cual es muy importante en el campo de la criptografía, ya que el principal algoritmo de cifrado, conocido como RSA [30], se basa en que la factorización de números muy grandes en factores primos es especialmente complicada computacionalmente hablando. Sin embargo, fue su trabajo en la corrección de errores cuántica, incluyendo la introducción del código Shor, la que aseguró que los ordenadores cuánticos funcionaran en el mundo real. Esta corrección de errores permite cierta resiliencia de los qubits ante ciertos tipos de ruido. Sin el desarrollo de esta teoría, el desarrollo de los ordenadores cuánticos probablemente no habría seguido adelante y habría muerto de la misma forma que los ordenadores analógicos en el mundo clásico.

Tras estos conceptos es cuando se puede entender la introducción de la esfera de Bloch. Los qubits se pueden seguir asemejando a diales, pero no a uno, sino a dos: un dial que controla el nivel de superposición de los estados $|0\rangle$ y $|1\rangle$ (eje vertical), y un segundo dial (eje horizontal) que controla la diferencia de fase entre los mismos en dicha superposición.

1.4.3. Formulación matemática de los circuitos cuánticos

Existe la posibilidad de formular los circuitos cuánticos de una manera totalmente matemática. De esta forma, es posible calcular sobre el papel la probabilidad de cada qubit en la medición, en función de la entrada y las puertas que haya por el camino.

- Puertas de un único qubit: se ha de recordar que los qubits son vectores complejos, y las puertas cuánticas son matrices, también con elementos complejos. La concatenación de diversas puertas cuánticas equivale al producto matricial, en donde el orden importa (el producto de matrices no es conmutativo). Los circuitos se leen de derecha a izquierda, poniendo primero las matrices de las puertas del final. En el último paso, se pondrá el vector correspondiente a la entrada. El valor medido a la salida vendrá dado por la superposición obtenida al realizar el producto de las matrices por el vector correspondiente. Un ejemplo sería el añadido en la figura

6, en donde se inicializa un qubit a 0, posteriormente se añade una puerta X de Pauli (en este nomenclatura se representa mediante un signo +) y a continuación, una puerta H de Hadamard. Con los conocimientos hasta el momento, se sabe que la salida medida será la correspondiente al qubit $|-\rangle$ (uno de los estados Bell mencionados anteriormente), pero la pregunta es si se puede llegar a esta expresión matemáticamente:



Figura 6: Circuito de ejemplo para 1 qubit

Operando matemáticamente, el valor a la salida será:

$$H \cdot X \cdot |0\rangle = \sqrt{0,5} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \sqrt{0,5} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |-\rangle$$

Se pueden hacer algunas observaciones:

1. Un conjunto de puertas A, B, C dispuestas consecutivamente (y en ese orden), equivalen a otra puerta, cuya matriz D estaría dada por la expresión $D=C \cdot B \cdot A$.
 2. Resulta de utilidad el empleo de la matriz identidad, I, cuando se quiere representar «que no hay nada», sólo que el «cable» continúa. Esto resultará de especial relevancia en el siguiente apartado.
 3. Por defecto se suponen los qubits de los circuitos inicializados a $|0\rangle$.
- Puertas con varios qubits: si se dispone de un circuito con n qubits, la formulación ya no es tan sencilla como en el ejemplo anterior. Existirían dos opciones de análisis: o bien resolver cada qubit individualmente, o bien resolver el circuito conjuntamente. El primer método únicamente será válido cuando todas las puertas empleadas sean puertas de un único qubit. Sin embargo, existen puertas que utilizan varios qubits de entrada, tales y como la CNOT, SWAP, etc. Es más, estas puertas tienen asociadas matrices de dimensiones distintas a las 2x2 vistas hasta ahora (en los ejemplos anteriores, 4x4). La resolución matemática mediante el segundo método, analizando el circuito como un conjunto, necesitará el empleo de productos tensoriales. En el caso de la computación cuántica será necesario preocuparse únicamente de los productos tensoriales en una (qubits) y dos (puertas cuánticas) dimensiones. Sin entrar en demasiados detalles matemáticos, se ha de recordar que el producto tensorial de dos matrices, A y B, con dimensiones (n x m) y (p x q), será otra matriz de dimensiones (n·p x m·q), y en este caso los valores de n, m, p y q podrán ser cualquier valor natural (no ocurre como en el producto de matrices, que sólo es posible para ciertas dimensiones de las matrices asociadas). Además, en general el producto tensorial de dos matrices, tal y como ocurre con el producto de matrices, tampoco va a ser conmutativo. A la hora de resolver los circuitos, se subdividirá el mismo en productos tensoriales y productos corrientes de matrices. Los qubits de entrada vendrán dados por el producto tensorial de cada uno de los qubits, y el circuito equivalente a las puertas cuánticas en cada «paso», se obtiene con el producto tensorial de cada una de las puertas en el mismo «paso». Mostremos un ejemplo, en el que hay un único «paso», en la figura 7:

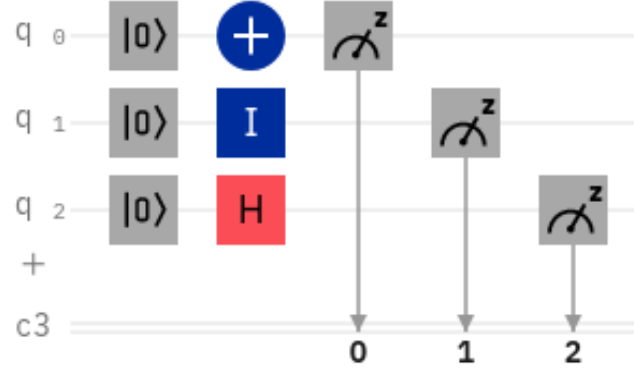


Figura 7: Circuito de ejemplo para varios qubits y un único «paso»

Matemáticamente, se calcularía primero el vector asociado a los qubits de entrada, de la forma:

$$|\psi\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ 0 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Y la matriz resultante para las puertas cuánticas de este único «paso» sería la siguiente (y aquí es donde se entiende el uso de la puerta identidad, con la matriz identidad asociada).

$$X \otimes I \otimes H = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \sqrt{0,5} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \dots = \sqrt{0,5} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Y ahora, realizando el producto matricial, se obtiene el estado resultante:

$$|\phi\rangle = (X \otimes I \otimes H) \cdot |\psi\rangle = \sqrt{0,5} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

La interpretación de la respuesta es que la probabilidad está distribuida al 50 % (recuérdese que la probabilidad es el módulo de la amplitud al cuadrado) de los estados quinto y sexto, que corresponden a $|100\rangle$ y $|101\rangle$ (se

empieza a contar por $|000\rangle$). Sin darnos cuenta se ha pasado también de la medición de un único qubit, a la medición de varios simultáneamente.

Otro ejemplo, esta vez con varios «pasos», en la figura 8:



Figura 8: Circuito de ejemplo para varios qubits y varios «pasos»

El estado $|\phi\rangle$ medido a la salida será:

$$|\phi\rangle = T_2 \cdot T_1 \cdot |\psi\rangle = (H \otimes I) \cdot (CNOT) (|0\rangle \otimes |0\rangle) = \left[\sqrt{0,5} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right] \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \dots = \sqrt{0,5} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Se ha empleado la notación T_n para representar cada uno de los «pasos» de la evolución temporal. Se ha de resaltar que en este método, el orden de los productos (tanto tensoriales como matriciales) es el habitual de nuestra lectura (de arriba a abajo y de izquierda a derecha), es decir, ya no se invierte el orden de las matrices para el producto matricial. En este caso se llega a la respuesta que los estados $|00\rangle$ y $|11\rangle$ son equiprobables a la salida (o lo que es lo mismo, la salida corresponde al estado de Bell $|+\rangle$). Si bien esta notación matemática presenta cierta complejidad operacional, en algunas ocasiones puede resultar práctica a fin de orientarse correctamente.

1.4.4. Algoritmos

Un algoritmo es un conjunto de instrucciones que ayudan a calcular la respuesta de un problema. Se podría hacer una analogía con una receta de cocina, o las instrucciones de montaje de un mueble: se tienen unas entradas (alimentos o piezas), y mediante ciertas instrucciones, se llega a las salidas deseadas (una comida cocinada o un mueble montado). Ejemplos de algoritmos reales y prácticos podrían ser las recomendaciones de nuestra plataforma de streaming de contenido multimedia favorita (música o vídeo), las rutas proporcionadas por el navegador de nuestro coche o teléfono móvil, o la obtención de resultados mostrada por un buscador de Internet. Un ejemplo muy habitual en cualquier curso básico de programación sería un algoritmo de búsqueda de un elemento en una lista, y será precisamente eso lo que se implementará de una forma cuántica en el algoritmo de Grover.

Resulta interesante tener una métrica para poder medir la eficiencia de estos algoritmos. Se podría pensar en algo tan simple como medir el tiempo de ejecución, seguramente se disponga de herramientas fácilmente accesibles para ello (por ejemplo, el comando «time» del sistema operativo GNU/Linux). Sin embargo esta idea presenta una serie de problemas que no la hacen óptima:

- Depende del hardware: no es lo mismo ejecutar un algoritmo en una supercomputadora, que en un ordenador doméstico.
- Depende de la dificultad en la distribución de las entradas: no se tardará lo mismo si el elemento a buscar estaba en la primera posición de la lista, que si estuviera en la última posición de millones.

- Depende de las entradas: no es lo mismo realizar una búsqueda de un elemento en una lista de 100 posibilidades, que en una de millones.

Sin embargo, científicamente se ha desarrollado una métrica adecuada para realizar comparaciones entre algoritmos. Ésta se conoce como «big O notation» [31], que en algunas fuentes se traduce en español como «cota superior asintótica». Los problemas antes expuestos se resuelven de la siguiente manera:

- Se mide el número de operaciones, no el tiempo empleado
- Se toma el peor caso posible («worst-case scenario», en el caso de una búsqueda en una lista, el elemento a buscar va a estar siempre el último)
- Se expresa la salida como función del tamaño de la entrada. La notación va a ser de la forma « $O(x)$ », siendo x la entrada (de ahí el nombre inglés de «big O notation». En el caso del ejemplo de la búsqueda, la complejidad sería lineal y tomaría precisamente la forma « $O(n)$ », siendo n el número de entradas (para 1000 entradas, serían necesarias 1000 operaciones).

La complejidad de los algoritmos se ordena de la siguiente manera, de menor a mayor:

- Constante: $O(1)$ (realmente sería cualquier número constante)
- Logarítmica: $O(\log(n))$
- Raíz cuadrada: $O(\sqrt{n})$
- Lineal: $O(n)$
- Cuadrática: $O(n^2)$
- Exponencial: $O(2^n)$
- Factorial: $O(n!)$

Cuando se esté calculando la complejidad de un algoritmo, los términos más arriba de la lista se pueden despreciar frente a los inferiores, hasta quedarnos con uno sólo de ellos. Por supuesto, pueden cambiar los exponentes del caso cuadrático (por ejemplo ser un 3 y pasaría a llamarse «cúbica»), o la base de la exponencial. En esos casos, se tomaría el mayor valor y se despreciarían los demás, constantes incluidas (una complejidad de $O(3n^3 + 8n^2 + 0,5n)$ se mostraría como $O(n^3)$).

A la hora de diseñar algoritmos, interesa obtener la mejor opción posible en la notación recién introducida. Por ejemplo, en lugar de una búsqueda lineal, en el mundo clásico ya se diseñó algo conocido como búsqueda binaria [32], cuando los valores de la lista están ordenados (se va dividiendo la lista en mitades). De esta forma, se pasa de una complejidad de $O(n)$ a otra de $O(\log(n))$, obteniéndose una aceleración del algoritmo. Se buscará esto mismo en los algoritmos cuánticos.

Existe la creencia de que la computación cuántica podría ayudar en muchos de los problemas cuya complejidad (expresada en la forma de esta notación) es muy grande, tal y como la creación de rutas eficientes en un mapa, algoritmos de optimización, simulaciones químicas, diseño de medicamentos, etc...

Por último, se enuncian algunos algoritmos cuánticos, así como algún detalle técnico de su implementación y sus objetivos en el cuadro 1 (se detallarán más detenidamente los algoritmos implementados en la sección de resultados):

Cuadro 1: Algoritmos cuánticos

Algoritmo cuántico	Característica técnica	Objetivo
Deutsch-Josza	Oráculo	Algoritmo de prueba
Algoritmo de Shor	QFT (transformada de Fourier cuántica)	Factorización de números
Algoritmo de búsqueda de Grover	Oráculo + Amplificación de amplitud	Búsqueda en una lista desordenada

2. Objetivos

En este trabajo de fin de grado se buscan diversos objetivos, tales como:

- Estudio de diversos ejemplos de algoritmos cuánticos.
- Comparación de los mismos con los algoritmos clásicos equivalentes, en términos como velocidad y eficiencia.
- Implementación práctica personal de un algoritmo ya existente (Deutsch-Josza).
- Implementación personal y ampliación de un algoritmo ya existente (Grover para 3 qubits, implementando la búsqueda de cualquier posible solución dentre de las $2^3 = 8$ posibles, incluyendo soluciones dobles).
- Ejecución de los algoritmos tanto en simuladores cuánticos, como en ordenadores cuánticos reales.

3. Métodos y material

Para el desarrollo de este trabajo se han empleado las siguiente herramientas:

3.1. Lenguaje Python para la programación de los algoritmos.

El lenguaje de programación empleado es python. No sólo por ser muy habitual para cualquier tipo de algoritmo hoy en día, ni por su sencillez, sino principalmente por la librería ya implementada existente, que se trata en el siguiente punto.

3.2. Librería Qiskit [33]

Permite, entre otras cosas, la implementación de puertas cuánticas de una manera simple mediante código. Las posibilidades con esta librería son infinitas, y para cualquier consulta concreta lo mejor es hacer referenciar a la bibliografía añadida aquí. Se entiende que los detalles de Qiskit quedan un poco al margen de la Física y por ello se intenta centrar el trabajo en ella, tomando esta librería como lo que es, una herramienta de la que se hace uso para un fin, como podrían ser las Matemáticas en tantos campos de la Física.

3.3. IBM Quantum Experience [34]

IBM ofrece la posibilidad de acceder a hardware cuántico real mediante una cuenta gratuita, la cual permite ejecutar código en dicho hardware (lo que sería una analogía de implementar las puertas cuánticas deseadas en dicho hardware).

Una vez creada la cuenta, el acceso a ella en el código del programa no tiene mayor complicación y está detallado en la documentación de IBM referenciada aquí. Se ha de crear un token de acceso (que sustituye a la contraseña y se guarda en el propio equipo, en caso de ejecutarse el programa en otro equipo, haría falta el mismo token. O bien se podría modificar el código para usar la cuenta del usuario que desee ejecutarlo, tras haber guardado su propio token). En el caso de que no se opte por visualizar los resultados en hardware cuántico real, no es necesario configurar ninguna cuenta de IBM, ya que esta parte del código no se llegará a ejecutar y no daría por tanto ningún error.

3.4. Control de versiones Github [35]

A la hora de codificar un programa de cualquier tipo, resulta muy útil el empleo de un software para control de versiones. Es más, este mismo software se puede usar para cualquier tipo de archivo, siendo especialmente útil cuando dicho archivo está en texto plano, ya que permite comparar versiones de una manera muy fácil e intuitiva, el ejemplo serían las fuentes de L^AT_EX (para el PDF correspondiente ya no tiene tanta utilidad al no estar éste en texto plano).

3.5. Procesador de textos L_AT_EX [36]

Como se introdujo en el punto anterior, esta memoria se ha escrito en L^AT_EX. Sin embargo, si no se tiene el conocimiento de dicho lenguaje, la curva de aprendizaje puede ser algo lenta al principio. Resulta mucho más fácil el empleo de un procesador de textos que haga uso de L^AT_EX, sin tener que codificar el mismo. Esto es lo que logra el software libre L_AT_EX. Este documento es el segundo empleado para realizar un texto de características similares y el resultado es muy satisfactorio.

4. Resultados

4.1. Algoritmo de Deutsch-Josza [23]

El algoritmo de Deutsch-Josza resuelve un problema que no tiene ningún objetivo práctico, salvo demostrar la existencia de problemas cuya resolución a partir de un algoritmo cuántico es más eficiente que el método análogo correspondiente mediante computación tradicional.

El algoritmo trata de hallar si una función cualquiera $f(x)$ es constante o balanceada. Esto quiere decir, para una entrada de un único bit, se quiere hallar si su salida depende de la entrada (balanceada) o no (constante).

La forma más fácil y gráfica de visualizar cada uno de los casos es mediante una tabla de la verdad, habitual en la electrónica digital. Dicha tabla podría tomar dos formas en cada uno de los dos casos, como se muestra en los siguientes apartados

4.1.1. Caso constante

Como el propio nombre indica, la salida será siempre 0 ó 1, independientemente de la entrada

Cuadro 2: Tablas de la verdad para el caso constante

x	f(x)	x	f(x)
0	0	0	1
1	0	1	1

4.1.2. Caso balanceado

La salida se alterna según sea la entrada 0 ó 1:

Cuadro 3: Tablas de la verdad para el caso balanceado

x	f(x)	x	f(x)
0	0	0	0
1	1	1	1

Se puede observar para ambos casos, que se necesitan 2 evaluaciones de la función para saber si la misma es constante o balanceada. Generalizando, para n bits, serían necesarias 2^n evaluaciones (o visto de otra manera, analizar todas las combinaciones posibles de bits de entrada). La complejidad del algoritmo clásico sería por tanto de $O(2^n)$.

Para la evaluación de la función mediante un algoritmo cuántico, que se espera que sea más eficiente, se conoce que todas las puertas cuánticas empleadas han de ser reversibles, como se estudió anteriormente. Analizando $f(x)$, se concluye rápidamente que ésta no es reversible. Se debe evitar este problema mediante la introducción de un oráculo (vocablo inglés, en español se podría denominar también como «caja negra»). Dicho oráculo tendrá como entradas dos qubits y como salidas otros dos qubits. Las entradas serán denominadas como q_0 y q_1 , y las salidas serán simplemente q_0 y la operación XOR de q_1 con $f(x)$. En el mundo cuántico, la puerta XOR no existe, sin embargo sí que hay un equivalente, que no es otro que la puerta CNOT.

El algoritmo emplea dos de las propiedades del mundo cuántico antes introducidas y que tanto se emplean para esta clase de algoritmos: la superposición y la interferencia (en este caso no se hace uso de la tercera propiedad, el entrelazamiento cuántico).

Esquemáticamente, el algoritmo para construcción del circuito cuántico y que se ha codificado en python como se puede ver en el código anexo, es el siguiente:

1. Definir dos qubits, «x» e «y», o «q0» y «q1».
2. Invertir «y» o «q1» para tener un valor de 1 en dicho qubit (los qubit se inicializan por defecto a 0).
3. Aplicar superposición mediante la puerta H de Hadamard a ambos qubits.
4. Aplicar el oráculo.
5. Aplicar de nuevo puertas H a ambos qubits, mediante lo cual se estará aplicando interferencia para poder medir el resultado.
6. Medir la salida. El valor de q0 codifica el tipo de función, significando 0 que dicha función es constante y 1 que es balanceada.

Se puede observar como el oráculo sólo se aplica una vez. Las puertas Hadamard tienen una complejidad computacional despreciable respecto al oráculo, una vez que se escala el algoritmo. Se ha pasado de una complejidad exponencial, a una constante $O(1)$, siendo por tanto la mejora exponencial.

4.1.3. Definición del oráculo

La definición detallada del oráculo daría para otro estudio completo por separado. Tanto en este caso del algoritmo de Deutsch-Josza como en el siguiente de Grover, no se detallarán en demasía estas «cajas negras» y simplemente se hará uso de ellas, suponiendo conocida su forma. A la hora de codificar, la forma del oráculo se implementa conocido el tipo de función, balanceada o constante (esto es en realidad la solución del problema). Sin embargo, no se vuelve a consultar esta solución para obtener la respuesta. Esto permite centrarse en el algoritmo, tema principal de estudio de este trabajo. Para este caso de Deutsch-Josza, se conoce que para la entrada de dos qubits, se deja el primero de ellos igual y para el segundo se efectúa una XOR del mismo con la función a evaluar $f(x)$. Se analizará si hay que hacer algún procesamiento adicional para cada uno de los cuatro casos enunciados en el punto anterior, los dos de la función constante y los dos de la función balanceada.

Con esta explicación, se procede a analizar las tablas de la verdad para los cuatro casos. Recuérdese que una de las salidas del oráculo es inmediata y coincide con el valor del primer qubit, por lo que no se muestra en las tablas para simplificar. Se estudia primero la construcción del oráculo para el caso de $f(x)$ balanceada, como se muestra en el cuadro 4:

Cuadro 4: Tablas de la verdad del oráculo para el caso constante

x(q0)	y(q1)	f(x)	y XOR f(x)	x(q0)	y(q1)	f(x)	y XOR f(x)
0	0	0	0	0	0	1	1
0	1	0	1	0	1	1	0
1	0	0	0	1	0	1	1
1	1	0	1	1	1	1	0

Como se puede ver, en el primero de los casos la salida corresponde con $y(q0)$, y en el segundo de ellos, su valor opuesto. La implementación mediante puertas cuánticas sería la que se puede observar en la figura 9. Por simplicidad, se muestra todo el circuito y no sólo el oráculo, que correspondería a lo que hay entre los separadores (barreras) segundo y tercero:

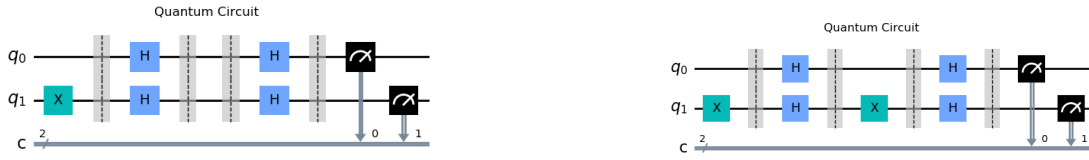


Figura 9: Circuitos cuánticos para el caso de una función constante

Las probabilidades obtenidas para cada uno de los casos se pueden ver en las gráficas de la figura 10 (las gráficas están en el mismo orden que los circuitos anteriores).

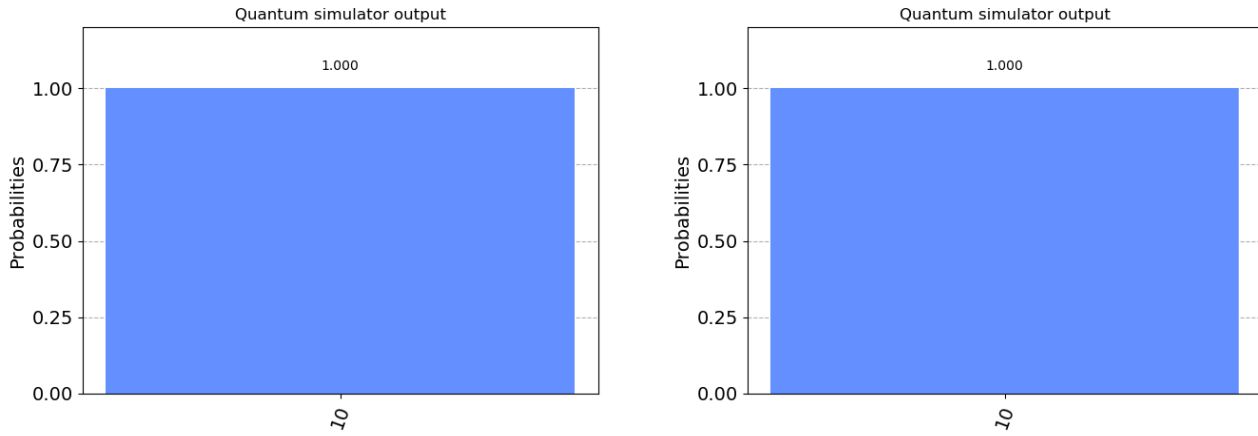


Figura 10: Probabilidades medidas para los circuitos cuánticos en el caso de una función constante

Desarrollando el caso análogo para una función balanceada, se llega a los valores mostrados en el cuadro 5:

Cuadro 5: Tablas de la verdad del oráculo para el caso balanceado			
x(q0)	y(q1)	f(x)	y XOR f(x)
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

La implementación utilizará las mencionadas puertas cuánticas CNOT, como se puede mostrar en la figura 11:



Figura 11: Circuitos cuánticos para el caso de una función balanceada

Y se pueden añadir las probabilidades calculadas, tal y como se hizo en el caso anterior, como se muestra en la figura 12, de nuevo siguiendo el mismo orden que el elegido para los circuitos:

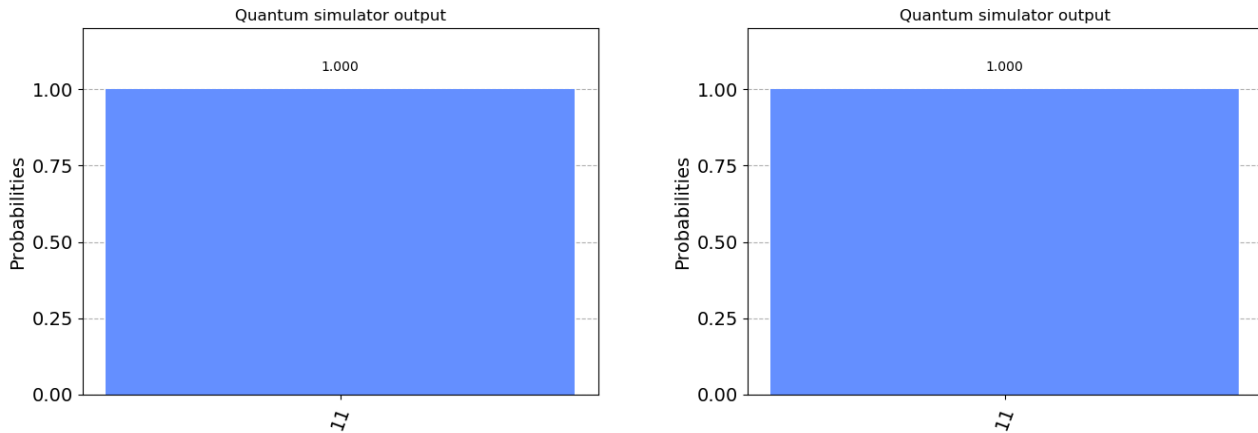


Figura 12: Probabilidades medidas para los circuitos cuánticos en el caso de una función balanceada

4.2. Algoritmo de Grover [23]

El algoritmo fue propuesto por Lov Grover en 1996. Se trata de un algoritmo para hallar un elemento en una lista, problema muy habitual en computación y cuya resolución clásica se estudia en cualquier curso básico de programación. Esto mismo se puede aplicar no sólo para búsquedas de una base de datos, sino para problemas de optimización (que también se basan en búsquedas de posibles soluciones), o estadísticas tales como la media, mediana, etc... Resulta intuitivo pensar que la complejidad de un algoritmo clásico de búsqueda en una lista, buscando elemento por elemento de la misma, es de $O(n)$. Se asume que la lista no tiene estructura y se ha tomado el peor caso posible (worst case scenario).

El algoritmo cuántico a estudiar reduce la complejidad al $O(\sqrt{N})$, produciendo una mejora cuadrática. Análogamente al caso estudiado anteriormente de Deutsch-Josza, el algoritmo se vuelve a basar en los principios de superposición e interferencia para acelerar la búsqueda. Se necesitarán $\log_2 N$ qubits. Esquemáticamente, los pasos del algoritmo serían los siguientes:

1. Inicialización de los qubits a 0.
2. Aplicación de una puerta H de Hadamard a cada uno de los qubits (superposición).
3. Aplicación de un oráculo a los qubits, y aplicación del algoritmo de difusión de Grover para amplificar la solución (interferencia).
4. Medición.

Los puntos 2 y 3 se pueden repetir, para los casos en los que el algoritmo precise de más de una iteración. Esto se verá con más claridad en alguno de los casos de la implementación.

4.2.1. Aplicación a dos qubits [37]

En primer lugar, se asigna igual probabilidad a cualquiera de los elementos de la lista, ya que no se presupone ninguna solución más probable que otra, lo cual se consigue mediante la superposición (segundo punto).

La mayor complejidad del algoritmo estará por tanto en el punto tercero. Respecto al oráculo («caja negra»), para este caso lo que hará será simplemente multiplicar la amplitud de la probabilidad (que no es más que la raíz cuadrada de la probabilidad) de la solución por un valor de (-1). Recuérdese que el oráculo conoce la solución (como se introdujo anteriormente, la construcción real del oráculo es un campo complicado en el que se están realizando muchos estudios al respecto y fuera del alcance de este trabajo). Analizando matemáticamente, y para un ejemplo

de cuatro elementos en la lista (necesitándose por tanto dos qubits), con el elemento a buscar en la cuarta posición,

se necesitaría pasar del vector de amplitudes de probabilidad: $\frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, a este otro vector: $\frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$

Este se conseguiría aplicando una puerta cuya matriz fuera: $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$, matriz que coincide con la puerta

Z controlada (CNOT). Para el caso genérico de una posición distinta a la cuarta, se deberán manipular debidamente los qubits para obtener con la misma matriz el -1 en otro punto de la diagonal principal, sin más que aplicar puertas X de Pauli (equivalente a la puerta NOT de los ordenadores clásicos) antes y después de la puerta Z controlada. Esta implementación se puede ver en el código.

Operando en el ejemplo anterior:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

A partir de aquí, sólo faltaría el operador de difusión de Grover, que es el que se encarga de reflejar las amplitudes respecto de la media (produciendo por tanto una amplificación de la respuesta buscada). La media de las amplitudes ahora es de 1. Si se refleja la amplitud respecto a esa nueva media, las tres primeras componentes del vector van a

resultar nulas y la última la unidad, es decir, se pasa del vector $\frac{1}{2} \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$, al vector: $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$.

En este caso se puede ver como la medición nos daría la solución con una probabilidad del 100 % tras una única ejecución del algoritmo (recuérdese que \sqrt{N} es un valor esperado para N suficientemente grande). En la figura (13) se puede ver el circuito cuántico correspondiente. Se han separado mediante barreras o separadores cada uno de los pasos del algoritmo: paso 2 al principio (los qubits ya están inicializados a cero), primera parte del paso 3 (oráculo) tras la siguiente barrera, algoritmo de difusión después de la siguiente barrera y por último la medición tras la última barrera (que se ha dibujado como doble en este caso):

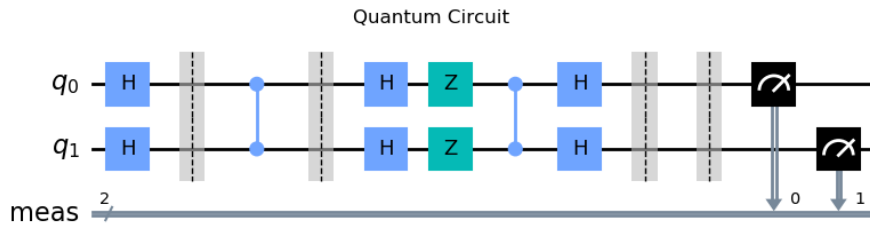


Figura 13: Circuito para el algoritmo cuántico de Grover con 2 qubits y solución «11»

Se adjunta también la distribución de probabilidades de la medición, que en este caso es del 100 %, como se puede ver en la figura 14:

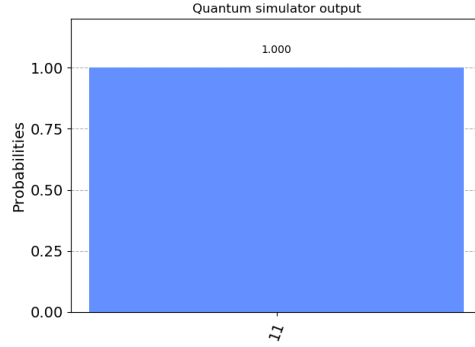


Figura 14: Probabilidad medida para el algoritmo cuántico de Grover con 2 qubits y solución «11»

Esta particularización para únicamente dos qubits se puede encontrar en muchos estudios (véase la bibliografía de este apartado). Por ello, y queriendo ampliar el algoritmo a un caso más complicado, se procede a implementar el mismo para tres qubits, lo que equivaldría a la búsqueda en una lista de ocho elementos.

4.2.2. Aplicación a tres qubits [38]

La base es la misma que en el caso anterior. Se estudiarán dos casos distintos, el primero de ellos con una única solución (un único elemento a buscar), y el segundo con dos elementos distintos de la lista a buscar.

Para el primero de los casos, en la bibliografía se puede ver como la probabilidad de hallar el valor adecuado es de:

$$p = \left(\left[\frac{N-2t}{N} + \frac{2(N-t)}{N} \right] \frac{1}{\sqrt{N}} \right)^2 = \left(\frac{5}{4\sqrt{2}} \right)^2 = 0,78125 \quad (4)$$

En donde el parámetro «t» corresponde al número de soluciones a buscar. Comparado con el caso clásico, el mismo valor de probabilidad sería:

$$p' = \frac{t}{N} + \frac{N-t}{N} \cdot \frac{t}{N-1} = \frac{1}{8} + \frac{7}{8} \cdot \frac{1}{7} = 0,25$$

Se adjunta el circuito resultante para la medición del valor «000» en la figura 15. En el mismo se puede apreciar, gracias a la separación mediante barreras, cada una de las partes del algoritmo: en primer lugar, la superposición (antes de la primera barrera). A continuación, la parte del oráculo, hasta la siguiente barrera. Seguidamente, el algoritmo de difusión de Grover, hasta la doble barrera final, tras la cual se produce la medición.

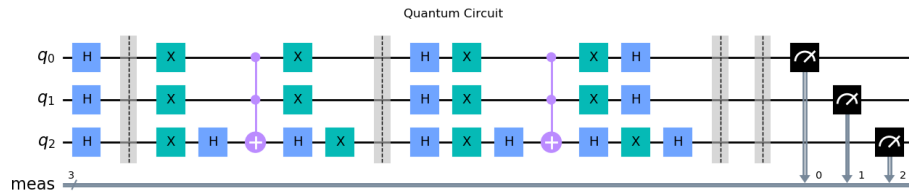


Figura 15: Circuito para el algoritmo cuántico de Grover con 3 qubits y solución «000»

Para este caso también se adjunta la probabilidad medida, como se puede ver en la figura 16:

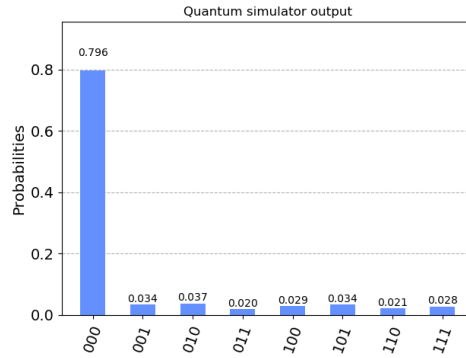


Figura 16: Probabilidad medida para el algoritmo cuántico de Grover con 3 qubits y solución «000»

Obviamente la probabilidad para el caso cuántico repitiendo el algoritmo una segunda vez mejorará, al tener $2 < \sqrt{N} < 3$. En el circuito cuántico correspondiente de la figura 17 se puede apreciar como hay una parte duplicada que corresponde a la segunda iteración. En este caso se está tratando de hallar la solución «111». Gracias a los separadores resulta fácil observar que la parte del oráculo y la del algoritmo de difusión están repetidas una vez, produciéndose la medición después de esta repetición.

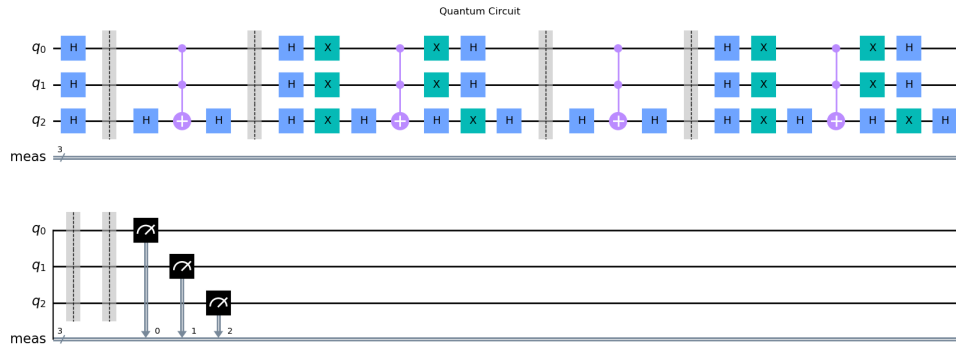


Figura 17: Circuito para el algoritmo cuántico de Grover con 3 qubits, doble iteración y solución «111»

Y efectivamente, la probabilidad aumenta en este caso:

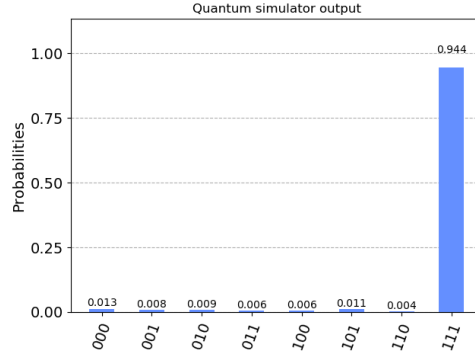


Figura 18: Probabilidad medida para el algoritmo cuántico de Grover con 3 qubits, doble iteración y solución «111»

Si se repitiera una tercera vez, el valor pasaría a empeorar (lo cual no parece intuitivo a primera vista, pero es debido a la parte del algoritmo de difusión de Grover, que refleja las amplitudes de probabilidad respecto de la media, tras invertir previamente la del valor a buscar, tal y como se comentó en el punto anterior).

Para el caso de dos soluciones a buscar, $t = 2$, sustituyendo en (4), se obtiene $p = 1$ (probabilidad de encontrar al menos una de las dos soluciones), mientras que el mismo caso para el algoritmo clásico obtendría $p' = 0,464$. Por ello, en este caso no se ha permitido al algoritmo ejecutarse más de una vez.

Se puede ver un ejemplo de la ejecución del algoritmo en la figura 19:

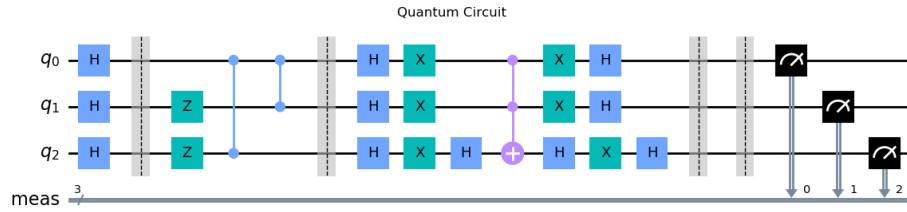


Figura 19: Circuito para el algoritmo cuántico de Grover con 3 qubits y soluciones «010» y «100»

Y las probabilidades asociadas:

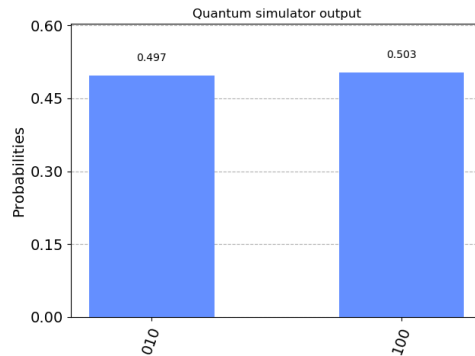


Figura 20: Probabilidad medida para el algoritmo cuántico de Grover con 3 qubits y soluciones «010» y «100»

Todos estos resultados han sido implementados en un simulador cuántico, esto es, una situación ideal y teórica.

Adicionalmente, se le ha añadido la opción al programa implementado de ejecutar los distintos algoritmos en hardware cuántico real, gracias al empleo de una cuenta gratuita en el programa «IBM Quantum Experience». Al habilitar esta opción, lo único que se hace es añadir una gráfica adicional con el cálculo de probabilidades para este caso. Se ha de destacar que esta parte de la simulación tardará bastante más en ejecutarse, ya que el acceso al hardware cuántico es limitado y dicha simulación se añadirá a una cola de ejecución. El programa irá informando de la situación en la cola, y tras unos minutos estarán disponibles los resultados, que se discutirán en la siguiente sección.

5. Discusión

5.1. Discusión de los errores en el hardware real

Se estudia el caso del algoritmo de Grover para 2 qubits (como se podría haber estudiado cualquier otro). Se añade la opción para ejecución en hardware cuántico real. En primer lugar, se muestran los resultados teóricos, ya conocidos de la sección anterior (figura 14). Tras unos pocos minutos, se obtiene también la gráfica con la distribución de probabilidades del experimento realizado en dicho hardware cuántico, que se muestra en la figura:

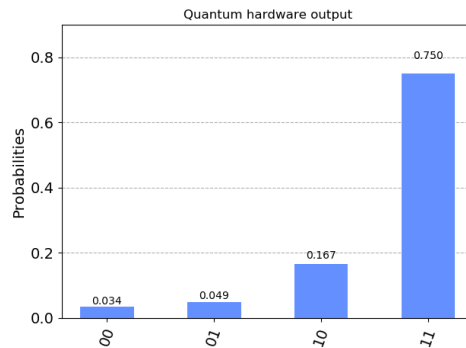


Figura 21: Probabilidad medida para el algoritmo cuántico de Grover con 2 qubits y solución «11», ejecutado en hardware cuántico real.

Parece lógico pensar que, en caso de existir errores en los resultados, sea más fácil que cambie un único qubit (resultados «01» ó «10»), que el caso de que ambos cambien a la vez (resultado «00»).

Cabe plantearse si algoritmos como los estudiados en este trabajo se pueden aplicar de una manera realista en la actualidad, y de no ser así, si se podrá en un futuro. Se puede adelantar, que hoy en día no se puede aplicar el algoritmo de Grover para una cantidad de datos significativos. Existen diversos problemas para ello, tales como:

- Insuficiencia de qubits (no se podrá aplicar una búsqueda en una lista demasiado grande)
- Ruido en los qubits (los resultados no serán fieles a la realidad)
- Imperfecciones en las puertas cuánticas (las salidas tras cada puerta no son las esperadas).

Al comenzar a estudiar los algoritmos, la mayoría de los resultados se obtuvieron en un simulador cuántico «ideal» (perfecto). Este caso no existe en la realidad, por lo que antes o después se planteará la pregunta de cómo elegir un equipo cuántico y cómo saber cómo de bueno es el mismo. Gracias a los criterios de DiVincenzo [39], esto resulta más fácil. Dicho criterio establece cinco condiciones para establecer la calidad de un ordenador cuántico:

1. Un sistema físico escalable y con qubits bien caracterizados. Esto es, qubits de un sistema con dos valores ($|0\rangle$ y $|1\rangle$), y cuyas propiedades (como podrían ser los valores de energía) están bien definidas. Un qubit con valor de $|1\rangle$ decae a $|0\rangle$ con el tiempo, según una gráfica de una exponencial caracterizada por un valor de tiempo de relajación, T_1 : $p(|1\rangle) = K \cdot e^{-t/T_1}$. La construcción de buenos ordenadores cuánticos se basa en tratar

de aislar los qubits todo lo posible, mediante el empleo de mejores materiales y reducción los mecanismos de pérdidas.

2. Posibilidad de inicializar el estado de los qubits (a uno de los valores expuestos en el punto anterior, siendo $|0\rangle$ el valor más habitual).
3. Largos periodos de decoherencia (tiempo en el que el qubit preserva sus propiedades), mucho mayores que los tiempos de operación de las puertas cuánticas. Esto se caracteriza mediante un tiempo de decoherencia, llamado T_2 y que se puede relacionar con el tiempo de relajación mediante la expresión: $\frac{1}{T_2} = \frac{1}{2T_1} + \frac{1}{T_\phi}$, siendo T_ϕ el tiempo de desfase. El resumen de estos tres puntos es tratar de aislar los qubits del entorno, bien reduciendo el ruido del mismo, bien haciendo a los qubits menos susceptibles a este ruido.
4. Empleo de un conjunto universal de puertas cuánticas, tales como H, S, T, CNOT. El control de la respuesta de dichas puertas es complicado. En la teoría una puerta va a suponer una transformación en la esfera de Bloch a un punto en concreto (véase en la sección correspondiente en la introducción). Sin embargo, al tener dicha esfera un número infinito de puntos, las posibilidades de error son también infinitas. Resumidamente, hay mucho margen de error. Para cuantificar la calidad de las puertas cuánticas se empleará la fidelidad de puerta, que no es otra cosa que el complementario de la tasa de error: $f = 1 - \varepsilon$, siendo ε la tasa de error. La medición de la fidelidad es un campo de activa investigación. Se podría pensar en una tasa de fidelidad $f = 0,995$ como una buena cifra. Sin embargo, suponiendo un circuito con 100 puertas con esta tasa de fidelidad, la fidelidad global sería de $f = 0,995^{100} = 0,6058 = 60,58\%$. Peor aún, para 200 puertas: $f = 0,995^{200} = 0,367 = 36,7\%$. Con «tan sólo» mejorar esta tasa de fidelidad de puerta a $f = 0,999$, los cálculos serían ahora: $f = 0,999^{100} = 90,48\%$ o $f = 0,999^{200} = 81,86\%$. Y un circuito cuántico con 200 puertas no es nada poco habitual.
5. Una capacidad de medida específica para qubits (si la medición no es adecuada, los resultados no valen para nada).

5.2. Limitaciones de los algoritmos cuánticos [23, 40, 41]

Actualmente los algoritmos cuánticos aún presentan grandes limitaciones. Por ejemplo, el mayor número factorizado a día de hoy mediante el algoritmo de Shor es únicamente 35. Asimismo, los errores que fueron presentados en el apartado anterior, resultan insalvables en la práctica cuando el circuito cuántico crece. Es necesaria la implementación práctica de corrección de errores. En el futuro será necesario disponer de un número mayor de qubits, sin ruido (irrealizable, pero sí se podría conseguir que éste fuera lo suficientemente pequeño), o aún mejor, qubits para cuyos errores se implemente una corrección. Sin embargo las posibilidades actuales son las que son, y se dispone de pocos qubits, con ruido, y una fidelidad de puertas imperfecta. Es por ello que han surgido teorías para mezclar ordenadores cuánticos con tradicionales, a fin de obtener un mejor resultado. La idea sería medir unos resultados imperfectos mediante nuestro hardware cuántico, y procesar los mismos con un ordenador tradicional. Los resultados obtenidos serán introducidos de nuevo al hardware cuántico, esperando obtener un mejor resultado. A este tipo de algoritmos mixtos se les conoce como algoritmos variacionales, y ejemplos de los mismos podrían ser el VQE (variational quantum eigensolver), empleado para sistemas continuos como podrían ser simulaciones químicas, o el QAOA (quantum approximate optimization algorithms), empleados para optimización combinatoria. Por supuesto, el detalle de este tipo de algoritmos es muy extenso y queda fuera del estudio de este trabajo.

6. Conclusiones

El campo de la computación cuántica se encuentra aún en sus primeros pasos, una época análoga a los primeros años tras el diseño del transistor en la computación clásica. Demostraciones recientes como la de la supremacía cuántica expuestas anteriormente demostrarían que se está avanzando en la dirección correcta, e incluso es posible ver este resultado en pequeños algoritmos de una gran simplicidad, tales y como los estudiados en este trabajo en detalle, como serían los algoritmos de Deutsch-Josza y el de búsqueda de Grover, así como otros enunciados a lo largo de este documento, como serían el de factorización de Shor. Parece claro que la computación cuántica no

viene a sustituir a la computación tradicional, ya que como se ha insistido, esta computación cuántica presenta utilidad para resolver únicamente cierto tipo de problemas, en los cuales se aprovechan las propiedades extrañas para nosotros de la computación cuántica, como por ejemplo la superposición, el entrelazamiento y la interferencia, que resultan útiles para este tipo de problemas, de tal forma que se obtiene una aceleración en la obtención de una solución, reduciendo por tanto su complejidad. Seguirán por tanto existiendo algoritmos para los cuales el empleo de ordenadores tradicionales siga siendo válido. Es más, como se expuso en el apartado anterior, cabe incluso la posibilidad de que ordenadores tradicionales y cuánticos trabajen juntos para resolver cierto tipo de problemas.

7. Anexos

7.1. Código fuente

Disponible en Github: <https://github.com/raulillo82/TFG-Fisica-2021>

Listado de archivos:

```
license.txt  
d-j.py  
grover.py
```

Es necesario instalar previamente las librerías empleadas (qiskit o matplotlib, entre otras). Esto depende del sistema operativo empleado, por lo que no se detallará en esta memoria el procedimiento, ya que en primer lugar, el objetivo primario es la discusión de los algoritmos en sí en relación a la física, y no la implementación informática de los mismos. En segundo lugar, el haber adjuntado una serie de capturas a lo largo del apartado anterior permite perfectamente analizar los resultados.

Al ejecutar los programas sin parámetros, se nos mostrarán unas instrucciones, indicando la cantidad de parámetros y la forma de los mismos, en función del caso particular que se quiera estudiar.

8. Agradecimientos

La idea de este trabajo de fin de grado nació al cursar durante el curso 2020/21 el programa «Introduction to Quantum Computing with IBM Quantum», de «Qubit x Qubit» [18]. Se han tomado partes de dicho material a modo de resumen teórico, y la idea de los algoritmos a plantear estaba propuesta en él. Por ejemplo, se hacía una implementación básica del algoritmo de Grover para 2 qubits con una solución fija, y en este trabajo se ha generalizado para cualquier solución. En el caso de 3 qubits, sólo se estudiaba el caso de solución única con una única iteración y con una solución concreta. Tras consultar bibliografía adicional, se implementó para cualquier posible solución única en una o dos iteraciones, así como para una solución doble, tal y como se ha explicado en la sección de resultados.

9. Bibliografía

Referencias

- [1] Louis de Broglie Nobel Lectur. eobel Lecture, December 12, 1929 The Wave Nature of the Electron.
- [2] History of computers - A Timeline (<https://www.youtube.com/watch?v=pBiVyEfZVUU>)
- [3] Moore's law: http://large.stanford.edu/courses/2012/ph250/lee1/docs/Excepts_A_Conversation_with_Gordon_Moore.pdf
- [4] Álgebra de Boole. Javier Borge Holthoefer (<http://serbal.pntic.mec.es/~cmunoz11/boole.pdf>)
- [5] IT 1: Boolean Algebra and Digital Logic. Maecel Lesther (<https://maecellesther.wordpress.com/2017/10/02/it-1-boolean-algebra-and-digital-logic/>)

- [6] RevComp - The Reversible and Quantum Computing Research Group Reversible/Adiabatic Chips: Tick, Flat-Top, XRAM, Pendulum <https://www.cise.ufl.edu/research/revcomp/>
- [7] Forbes: 27 Milestones In The History Of Quantum Computing <https://www.forbes.com/sites/gilpress/2021/05/18/27-milestones-in-the-history-of-quantum-computing/>
- [8] Nuclear Magnetic Resonance Quantum Computing (NMRQC) <http://web.physics.ucsb.edu/~msteffen/nmrqc.htm>
- [9] Why I Called It ‘Quantum Supremacy’ John Preskill, Quanta Magazine <https://www.quantamagazine.org/john-preskill-explains-quantum-supremacy-20191002/>
- [10] «Google claims to have reached quantum supremacy». Financial Times, Sept 2019 (<https://www.ft.com/content/b9bb4e54-dbc1-11e9-8f9b-77216ebe1f17>)
- [11] Arute, Frank; Arya, Kunal; Babbush, Ryan; Bacon, Dave; Bardin, Joseph C.; Barends, Rami; Biswas, Rupak; Boixo, Sergio et al. (2019-10). «Quantum supremacy using a programmable superconducting processor». *Nature* (en inglés) 574 (7779): 505-510. ISSN 1476-4687. doi:10.1038/s41586-019-1666-5. Consultado el 25 de octubre de 2019 (<https://www.nature.com/articles/s41586-019-1666-5>)
- [12] "What the Google vs. IBM debate over quantum supremacy means | ZDNet". www.zdnet.com (<https://www.zdnet.com/google-amp/article/what-the-google-v-ibm-debate-over-quantum-means/>)
- [13] "On "Quantum Supremacy"". IBM Research Blog. 2019-10-22. Retrieved 2019-10-24 (<https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/>)
- [14] "Google Claims To Achieve Quantum Supremacy — IBM Pushes Back". NPR.org. Retrieved 2019-10-24 (<https://www.npr.org/2019/10/23/772710977/google-claims-to-achieve-quantum-supremacy-ibm-pushes-back>)
- [15] Zhong, Han-Sen; Wang, Hui; Deng, Yu-Hao; Chen, Ming-Cheng; Peng, Li-Chao; Luo, Yi-Han; Qin, Jian; Wu, Dian; Ding, Xing; Hu, Yi; Hu, Peng (2020-12-03). "Quantum computational advantage using photons". *Science*. 370 (6523): 1460–1463. arXiv:2012.01625. Bibcode:2020Sci...370.1460Z. doi:10.1126/science.abe8770 (inactive 31 May 2021). ISSN 0036-8075. PMID 33273064 (<https://science.sciencemag.org/content/early/2020/12/02/science.abe8770>)
- [16] Dirac, P. A. M. (1939). "A new notation for quantum mechanics". *Mathematical Proceedings of the Cambridge Philosophical Society*.
- [17] Wikipedia: Quantum logic gate (https://en.wikipedia.org/wiki/Quantum_logic_gate#/media/File:Quantum_Logic_Gates)
- [18] Qubit x Qubit: Programs (<https://www.qubitbyqubit.org/programs>)
- [19] Óptica, Eugéne Hecht.
- [20] Física cuántica. Sánchez del Río, Carlos
- [21] Física cuántica I. Alvarellos Bermejo, José Enrique; García González, Pablo; García Sanz, José Javier
- [22] Stern and Gerlach: How a Bad Cigar Helped Reorient Atomic Physics. Bretislav Friedrich and Dudley Herschbach <https://web.archive.org/web/20070929091315/http://www.physicstoday.org/vol-56/iss-12/p53.html>
- [23] Quantum Computation and quantum information. Nielsen, Michael A.; Chuang, Isaac L.
- [24] Physics 8.05 Sep 12, 2007. Supplementary notes on Diract notation, quantum states. R. L. Jaffe. <http://web.mit.edu/8.05/handouts/jaffe1.pdf>
- [25] Bloch, Felix (Oct 1946). "Nuclear induction". *Phys. Rev.* 70 (7–8): 460–474 (<https://journals.aps.org/pr/abstract/10.1103/PhysRev.70.460>)

- [26] What is the role of ISA (Instruction Set Architecture) in the comp arch abstraction stack (<https://electronics.stackexchange.com/questions/353915/what-is-the-role-of-isa-instruction-set-architecture-in-the-comp-arch-abstract>)
- [27] Shannon, C.E. (1948), "A Mathematical Theory of Communication", Bell System Technical Journal, 27, pp. 379–423 & 623–656, July & October, 1948. (<http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf>)
- [28] Hamming, Richard Wesley (1950). "Error detecting and error correcting codes" (https://calhoun.nps.edu/bitstream/handle/10945/46756/Hamming_1982.pdf;jsessionid=5E608A71A351333E226F6D34F4)
- [29] Good Quantum Error-Correcting Codes Exist A. R. Calderbank, Peter W. Shor. (<https://arxiv.org/abs/quant-ph/9512032>)
- [30] Menezes, Alfred; van Oorschot, Paul C.; Vanstone, Scott A. (October 1996). Handbook of Applied Cryptography
- [31] Knuth, Donald (1997). "1.2.11: Asymptotic Representations". Fundamental Algorithms. The Art of Computer Programming. 1 (3rd ed.). Addison-Wesley
- [32] Bentley, Jon (2000). Programming pearls (2nd ed.). Addison-Wesley.
- [33] Qiskit homepage (<https://qiskit.org/>)
- [34] IBM Quantum (<https://quantum-computing.ibm.com/>)
- [35] Github (<https://github.com>)
- [36] Lyx (<https://www.lyx.org/>)
- [37] Qiskit reference: Grover's Algorithm (<https://qiskit.org/textbook/ch-algorithms/grover.html>)
- [38] Nature: Complete 3-Qubit Grover search on a programmable quantum computer (<https://www.nature.com/articles/s41467-017-01904-7>)
- [39] DiVincenzo, David P. (2000-04-13). "The Physical Implementation of Quantum Computation". Fortschritte der Physik. 48 (9–11): 771–783. (<https://arxiv.org/abs/quant-ph/0002077>)
- [40] Karbach, M.; Müller, G. (September 10, 1998), Introduction to the Bethe ansatz I. Computers in Physics 11 (1997), 36-43. (<http://www.phys.uri.edu/~gerhard/MSS/ms82.pdf>)
- [41] Moll, Nikolaj; Barkoutsos, Panagiotis; Bishop, Lev S.; Chow, Jerry M.; Cross, Andrew; Egger, Daniel J.; Filipp, Stefan; Fuhrer, Andreas; Gambetta, Jay M.; Ganzhorn, Marc; Kandala, Abhinav; Mezzacapo, Antonio; Müller, Peter; Riess, Walter; Salis, Gian; Smolin, John; Tavernelli, Ivano; Temme, Kristan (2018). "Quantum optimization using variational algorithms on near-term quantum devices". Quantum Science and Technology. (<https://arxiv.org/abs/1710.01022>)