uc3m | Universidad Carlos III de Madrid

Doble Grado en Ingeniería informática y Administración de empresas

Curso académico 2018-2019

Trabajo Fin de Grado

"Computación cuántica: Aplicaciones prácticas que la computación clásica no puede solucionar"

Ignacio Kleinman Ruiz

Tutores:
Gonzalo Génova Fuster

Francisco J. Gálvez Ramírez

Madrid – 22 de septiembre de 2019

Resumen: La computación cuántica es una tecnología que se encuentra a pleno auge. El potencial que tiene puede suponer un cambio de paradigmas todos los aspectos de la sociedad actual. Existen muchos casos prácticos donde los algoritmos y computadores cuánticos son mejores a los clásicos. No obstante, aún no se han logrado cálculos lo suficientemente grandes como para demostrar la supremacía cuántica. A pesar de ello, algunos de los computadores cuánticos funcionales hoy en día ya son eficaces ante la resolución de ciertos problemas altamente complejos (es el caso de los problemas de optimización o de búsqueda). De este modo, la computación cuántica se perfila como un camino prometedor en diversos campos de trabajo como la simulación de sistemas químicos, el aprendizaje automático, la gestión del tráfico de las ciudades, entre otros. En este trabajo se lleva a cabo un repaso del estado actual de la tecnología cuántica, así como sus perspectivas futuras.

Palabras clave: Computación cuántica; clases de complejidad; supremacía cuántica; sistemas adiabáticos; temple cuántico, ingeniería de software cuántico.

[En] Quatum Computing: Practical applications that classical computing cannot solve.

Abstract: Quantum computing is a booming technology. Its potential supposes a paradigm shift in all aspects of today's society. There are many practical cases where algorithms and quantum computers are better than the classic ones. However, calculations large enough to demonstrate quantum supremacy have not yet been achieved. Despite this, some functional quantum computers today are already effective in solving certain highly complex problems (such as optimization or search problems). Thus, quantum computing is emerging as a promising path in various fields of work such as simulation of chemical systems, automatic learning, traffic management of cities, among others. This paper reviews the current state of quantum technology and its prospects.

Keywords: Quantum computing; complexity classes; quantum supremacy; adiabatic systems; quantum annealing, quantum software engineering.

Índice General

Índice de ilustraciones

Índice de Tablas

1. Introducción y objetivos

1.1. Contextualización

La teoría de la computación comenzó a desarrollarse antes de la invención de los primeros ordenadores. Figuras como Turing, Boole o Von Neumann empezaron a desarrollar las primeras máquinas de cálculo desde principios del siglo XX, así como las primeras teorías de la computación, entre las que se encuentran el álgebra de Boole y la teoría de autómatas (O'Regan, G., 2008). Inicialmente, estas máquinas podían resolver únicamente problemas simples y deterministas, es decir, aquellos problemas que siempre generan un mismo resultado con una entrada conocida.

No fue hasta finales de los años cuarenta y, sobre todo, los cincuenta cuando los primeros ordenadores vieron la luz. Estas computadoras eran enormes y complejas máquinas que únicamente podían resolver sencillos cálculos. A medida que fueron pasando los años el rápido avance de la tecnología permitió desarrollar computadoras con mayor capacidad de cálculo lo que permitió resolver problemas más complejos. Al mismo tiempo, la tecnología fue sofisticándose más a medida que el tamaño de los componentes se reducía.

En los años 70, S. Cook llevó más allá la teoría de autómatas. Comenzó a distinguir entre aquellos problemas que podían ser solucionados y aquellos que no, mediante la computación existente basada en, lo que hoy se conoce como, computación clásica (Cook, S. A., 1971), (Cook, S. A., 1973). A medida que se clasificaban más problemas en relación con su complejidad, muchos de estos se agruparon en un conjunto de problemas que no parecían tener una solución fácil.

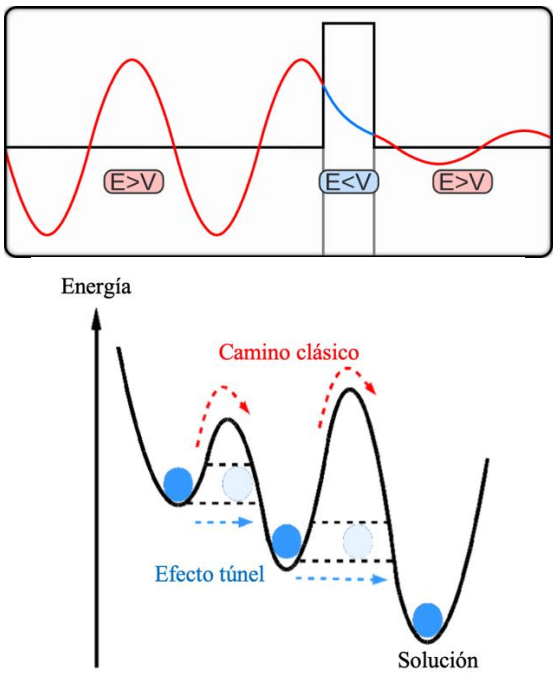
De tal modo, el rápido avance de la computación desembocó en la creación de circuitos más pequeños y rápidos que eran capaces de resolver problemas más complejos y grandes. Aun así, a pesar de estos avances tecnológicos, ciertos problemas seguían sin tener una solución eficiente, es decir, no aquellos que no pueden ser resueltos en tiempo polinomial, ya que su dificultad crece de manera exponencial.

Todos estos avances tecnológicos en busca de mayor capacidad computacional han sido encaminados en un mismo sentido: la miniaturización de los transistores y los circuitos. Actualmente, los procesadores más modernos que Intel está desarrollando son a 7nm (Sicard, E., 2017). Samsung, por su parte, presentó recientemente nodos de 5nm. Esta tendencia de reducir los tamaños de los circuitos y duplicar el número de transistores

cada dos años –Ley de Moore– lleva ocurriendo desde los años sesenta. Sin embargo, los analistas auguran que la tendencia desaparecerá pronto, pues la Ley de Moore está llegando a su final. La miniaturización está rozando los límites donde la física clásica entra en conflicto con la cuántica, lo que supone no poder reducir mucho más dichos circuitos. De hecho, empresas como Samsung y TSMC empiezan a ofrecer circuitos con medidas intermedias, es decir, procesadores, por ejemplo, de 6nm (Moore, S. K., 2019).

A medida que los dispositivos se reducen de tamaño, los efectos cuánticos comienzan a interferir en el funcionamiento de los dispositivos electrónicos, produciendo un fenómeno conocido como efecto túnel (ilustración 1). Cuando una partícula tiene que lidiar con barreras físicas, si esta no cuenta con la suficiente energía mecánica, entonces le será imposible atravesar dichas barreras. Este es el caso de los electrones de un circuito viajando por los buses. De esta manera, cuando los circuitos llegan a niveles extremadamente pequeños, la mecánica cuántica entra en juego violando los principios de la mecánica clásica. En este caso, la partícula cuántica es capaz de penetrar la barrera a pesar de contar con menor energía cinética. Esto se debe a su intrínseco comportamiento ondulatorio (Bonillo, V. M., 2013).

Ilustración 1. Diagrama del efecto túnel.



Cervera-Lierta, A., 2018. Fuente: https://medium.com/@quantum_wa/quantum-annealing-cdb129e96601

Es en este punto donde la mecánica cuántica entra en juego en la computación. Aunque estos avances han venido ocurriendo en los últimos años, el desarrollo de la computación cuántica data a partir de los años setenta. La primera aproximación a la teoría de la información cuántica fue realizada por Bell, basándose en la paradoja EPR –Einstein-Podolsky-Rosen– (Steane, A., 1998). Dicha paradoja presenta una nueva realidad de la mecánica cuántica donde las partículas de un sistema se encuentran entrelazadas y cuyos estados interaccionan sobre los estados del resto. A diferencia de la mecánica clásica, en la cuántica no se puede entender el conjunto sin la individualidad. Autores como Aharonov, Clauser, Holt y Horne contribuyeron a clarificar y extender dichos conceptos, así como llevar a cabo experimentos sobre las desigualdades de Bell (Aspect, A., Dalibard, J., & Roger, G., 1982). Más adelante, Feynman expuso sus correlaciones ERP-Bell. En ellas se explican ciertas propiedades de la mecánica cuántica (Feynman, R. P., 1982b). Además, fue el primer autor en considerar un sistema de simulación universal especialmente diseñado para ejecutar comportamientos de sistemas físicos (Feynman, R. P., 1985).

Así mismo, otros autores como Deutsch (1989), Shor (1994) o Grover (1997) comenzaron a elucubrar sobre una nueva serie de algoritmos: los algoritmos cuánticos. Al fin y al cabo, la mecánica cuántica existe desde inicios del siglo XX como un marco matemático, o un conjunto de reglas, para la construcción de teorías físicas. Para entonces, los primeros algoritmos que aparecieron fueron revolucionarios debido a que estos podían resolver algunos de los problemas que los algoritmos clásicos no podían realizar de forma eficiente.

No obstante, uno de los problemas de la computación cuántica, además de la compleja y abstracta base física y matemática, ha sido la falta de investigación y desarrollo en lo que corresponde al software para los sistemas informáticos cuánticos. Esta carencia viene encaminada, en parte, por la excesiva complejidad característica de la cuántica. La abstracta física y formulación de este campo no ha atraído a muchos ingenieros informáticos que pudiesen comenzar a desarrollar esta tecnología en un ámbito más práctico que el teórico.

Además, no queda claro que la computación cuántica pueda resolver aquellos problemas que la computación clásica no puede. Este suceso se conoce como *supremacía cuántica,* concepto apodado por Google (Preskill, J., 2012). Es recientemente cuando empresas y desarrolladores comienzan a elaborar ingeniería de software aplicada a la

computación cuántica, creando pseudocódigo, así como lenguajes de más alto nivel similar a los ya comunes entre los programadores clásicos, como es el caso de C y su análogo cuántico, *quC*.

1.2. Objetivos del proyecto

De este modo, el objetivo propuesto para este proyecto es el de recapitular y exponer el estado del arte de la computación cuántica con un trasfondo didáctico para la completa y amena comprensión de los conceptos más importantes al respecto y una desmitificación de muchas de las falsas ideas existentes alrededor de esta tecnología. Adicionalmente, este proyecto trata de relucir aquellas ventajas que la computación cuántica puede aportar en general a las personas, empresas, el mundo académico y gobiernos.

1.3. Marco regulador e impacto socioeconómico

La computación cuántica puede suponer un gran cambio de paradigmas para muchos de los sistemas actuales que dependen de dicha tecnología. La encriptación, la optimización de problemas o la simulación de sistemas cuánticos serán los principales afectados. No obstante, esta tecnología está en pleno desarrollo, por lo que actualmente no existe ningún tipo de regulación ni legislación que limite o controle los efectos que puede desembocar dicha tecnología. Asimismo, la investigación actual se está llevando a cabo por entidades tanto públicas como privadas, por lo que genera una dificultad añadida en cuanto a futuras estandarizaciones de la tecnología, así como de regulaciones.

Recientemente, el gobierno de Estados Unidos aprobó una legislación para incentivar la inversión y el desarrollo de los sistemas cuánticos (*H.R.6227 - National Quantum Initiative Act*)[1]. El objetivo es tener un sistema funcional en los próximos diez años. Esto, a su vez, supone el desarrollo de nuevas políticas respecto a la seguridad y privacidad. Así lo expone el documento de estrategia en ciberseguridad de Estados Unidos de 2018 (Trump, D., 2018, p. 8):

"Para protegerse contra la amenaza potencial de que los ordenadores cuánticos puedan romper la criptografía moderna de clave pública, el Departamento de Comercio, a través del Instituto Nacional de Estándares y Tecnología (NIST por sus siglas en

[1] Fuente: https://www.congress.gov/bill/115th-congress/house-bill/6227

inglés), continuará solicitando, evaluando y estandarizando algoritmos criptográficos de clave pública resistentes a los algoritmos cuánticos".

Cuando los ordenadores cuánticos sean suficientemente grandes y potentes, los sistemas RSA, basados en la clave pública y los números primos, se verán totalmente rotos. Esto se debe en parte al algoritmo de Shor (1994) y su capacidad de encontrar los factores de un número de forma eficiente. Es por ello por lo que es esencial la creación de nuevas herramientas y protocolos de criptografía que estén hechos a prueba de algoritmos cuánticos.

Actualmente existen dos versiones para achacar este problema: la *criptografía postcuántica* y la *criptografía cuántica*. La criptografía postcuántica está basada en problemas difíciles de computar incluso para ordenadores cuánticos. Estos problemas han tratado de resolverse durante años, pero siempre han fallado. El punto de interés surge en la falta de conocimiento sobre si estos problemas tienen una solución eficiente para ordenadores cuánticos, o incluso clásicos (De Wolf, R., 2017). El proyecto *FrodoKEM* es una colaboración entre universidades y empresas privadas como Microsoft y Google[2] para el desarrollo de sistemas criptográficos postcuánticos. Por su parte, los sistemas criptográficos cuánticos están basados en técnicas cuánticas y son sistemas mucho más avanzados y seguros. El sistema más conocido actualmente es la *distribución de clave pública* donde los sistemas cuánticos de comunicación no son de confianza (Bennett, C. H., & Brassard, G., 2014).

Por otra parte, la computación cuántica puede brindar unos efectos muy positivos en una gran diversidad de campos como la optimización de procesos logísticos o del tráfico de una ciudad, el incremento los ingresos de las empresas reduciendo costes en los procesos de producción, la creación de sistemas cuánticos aplicados a la química y biología, con el consiguiente desarrollo de nuevos métodos para el tratamiento de enfermedades. Por ejemplo, la mayoría de los sistemas químicos o físicos cuentan con grandes cantidades de elementos los cuales están interconectados generando una gran cantidad de combinaciones difíciles de simular. Los ordenadores cuánticos serán capaces de solventar estos problemas.

Por otra parte, la aplicación del algoritmo de búsqueda de Grover permitirá análisis de grandes datos a una velocidad nunca vista. De tal modo, aspectos como *machine learning* y reconocimiento de patrones, o los problemas de optimización como

[2] Fuente: https://www.microsoft.com/en-us/research/project/frodokem/.

el problema del viajante, verán grandes avances en este aspecto. Esto permitirá sistemas mucho más avanzados y rápidos con una consiguiente reducción de costes y recursos y, en definitiva, hará más eficiente los procesos de clasificación de datos, regresiones estadísticas, análisis, redes de neuronas, entre otros (Orus, R., Mugel, S., & Lizaso, E., 2019).

1.4. Motivación del trabajo y metodología

La motivación de este trabajo emana de tres aspectos importantes:

- Por una parte, la dificultad intrínseca del proyecto despertó el interés de afrontar un proyecto de tal envergadura y afrontarlo como un gran reto. El aprendizaje y entendimiento de una tecnología tan compleja e innovadora era una gran oportunidad de poner en común todos los conocimientos adquiridos durante los años de carrera.
- Además, la computación cuántica es una tecnología que aún no está muy establecida y, por ello, es poco conocida. Esto ha generado un gran revuelo de ideas falsas o erróneas sobre ella. La desmitificación de muchas de estas ideas que rondan a la cuántica, así como presentar una explicación amena sobre la física cuántica son otros dos puntos a favor de este trabajo.
- En último lugar, el gran potencial que la computación cuántica puede suponer para la ciencia, y la sociedad en general, es una gran baza para tener en cuenta. Por ello, este trabajo ha servido como primera aproximación a este campo con una perspectiva de seguir investigando y trabajando en ello en el futuro.

La metodología seguida en este proyecto ha sido simple y metódica, pero esencial para el correcto desarrollo del trabajo. Debido a la dificultad inicial con el tema, la primera etapa del proyecto fue la comprensión de los conceptos más básicos de la computación cuántica. Las fuentes consultadas fueron artículos, libros básicos de cuántica y tutoriales en la red. Tras ello, la siguiente etapa comenzó con la definición de la estructura y los objetivos principales del proyecto. Una de las fuentes principales seguidas fue el libro de Nielsen y Chuang (2001*): Quantum computing and quantum information*. Esta guía fue de gran ayuda para la comprensión y enfoque de los temas abarcados en este trabajo: las puertas lógicas y los sistemas cuánticos, las clases de complejidad y algoritmos cuánticos.

Una vez definido el esquema global, la metodología seguida ha sido la *técnica de la cebolla*. Esta técnica consiste, inicialmente, en la esquematización de aquellos temas

que se desea abarcar y, posteriormente, un desarrollo progresivo de los mismos, de tal forma que todos los apartados se encuentren en equilibrio en todo momento.

A medida que se profundiza en los diversos temas, la investigación también lo hace en cuanto a la bibliografía correspondiente. Esto supuso el salto constante de unas bibliografías a otras de forma iterativa. Uno de los métodos para mejorar la eficiencia de la búsqueda de bibliografía relevante es *Google* Scholar. Mediante este portal es fácil conocer el número de citaciones que un artículo o autor han obtenido. De esta forma se encontraron autores tanto modernos como más antiguos de gran importancia en la investigación de los distintos campos abarcados en este proyecto. Adicionalmente los tutores del proyecto, mediante sus comentarios y consejos, también facilitaron la búsqueda y realización de la investigación.

Tras el análisis y comprensión de los artículos, la siguiente labor consiste en la abstracción de todos los conceptos para una consiguiente simplificación y contextualización, además de la comparación de los conceptos, en caso de existir. Por último, aquellos temas relacionados con el software cuántico fueron consultados en las páginas oficiales de referencia, como es el caso de IBM, Google o D-Wave.

El trabajo global ha sido llevado a cabo en cuatro versiones. Estas versiones eran entregadas a los profesores y corregidas por ellos con su consiguiente devolución en la que se incluía *feedback* para la corrección de errores y/o meros consejos. La comunicación fue buena y constante, y los medios utilizados fueron el correo electrónico, conversaciones telefónicas y reuniones ocasionales en persona.

1.5. Estructura del documento

El documento está organizado de la siguiente manera. El capítulo dos presenta un marco teórico con los elementos fundamentales de la computación cuántica junto con los conceptos matemáticos y físicos en los que se basa. El capítulo tres expone las clases de complejidad existentes según los tipos de problemas, con especial hincapié en aquellos que la computación clásica no puede llegar a resolver. El cuarto capítulo detalla los tipos de algoritmos cuánticos que se han desarrollado, así como algunos ejemplos con más detalle. El capítulo cinco presenta el concepto de la supremacía cuántica, así como los distintos sistemas cuánticos que se están utilizando para llegar a la demostración. En último lugar, el sexto capítulo introduce la ingeniería del software en relación con la computación cuántica: simuladores y lenguajes de programación.

2. Computación cuántica

2.1. Postulados de la mecánica cuántica

La *computación cuántica* es una vertiente de la computación que hace uso de fenómenos de la mecánica cuántica tales como átomos, superposición, amplitudes de probabilidad o entrelazamiento cuántico. Dichas teorías comenzaron a tomar forma a principios del siglo XX con unas primeras aproximaciones de Planck, Bohr, Einstein o Broglie (Lomonaco, S. J., 2002). No obstante, no fue hasta la década de los años 20 cuando un grupo de físicos y matemáticos trabajaron de forma conjunta para unificar todas las diferentes visiones[3] y formalizar una serie de postulados que amoldaron la nueva teoría cuántica (Márquez, A. M., 2019):

Postulado I: los sistemas cuánticos están descritos por una función de onda –o vector estado– $\Psi(\mathrm{q},\mathrm{t})$ definida como finita y continua, con valores simples y derivadas continuas y de cuadrado integrable. La función está descrita por el espacio y por el tiempo, *q* y *t* respectivamente, y contiene toda la información disponible sobre el sistema.

Postulado II: Cada magnitud física observable (a, b, …) del sistema se asocia a un operador lineal y hermético $(\widehat{\mathrm{A}},\widehat{\mathrm{B}},\dots)$ definido en el espacio de las funciones. El término hermético hace referencia a valores propios reales, funciones ortogonales entre sí y funciones propias que forman un espacio completo.

Postulado III: Cualquier medida de un observable correspondiente a un operador A solo resultará en los autovalores a del operador correspondiente a dicho observable:

$$\widehat{\mathrm{A}}\,\Psi \rightarrow (a1, a2, a3, \dots) \rightarrow a_i$$

Postulado IV: Sea un sistema definido por una función de onda Ψ donde se realizan repetidas medidas de un observable *A* –el observable A genera el valor propio a– de un operador $\widehat{\mathrm{A}}$, entonces la medida de los valores obtenidos viene dada por:

[3] El primer conjunto de teorías cuánticas establecidas formalmente se llevó a cabo por Einstein, Schrödinger, Heisenberg, Born, Bohr, Von Neumann, de Broglie y Pauli, entre muchos otros.

$$\bar{a} = \langle \mathrm{a} \rangle = \frac{\langle \Psi | \hat{\mathrm{A}} | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$

2.2 Conceptos básicos de computación cuántica

Desde la invención de los primeros ordenadores en la década de 1940 (Gürer, D., 2002) la arquitectura utilizada en la computación ha permanecido casi invariable, basando sus principios en la máquina de Turing. Los computadores no pueden ser entendidos sin su parte lógica, tratándose esta del bit, la unidad mínima de información. Los bits pueden ser leídos o modificados por el hardware, cuya interacción es plasmada mediante las puertas lógicas. El bit representa uno de dos estados físico-electrónicos de un circuito: no hay presencia de voltaje en el circuito –representado con el 0–, o hay voltaje en el circuito –representado mediante el 1.

De forma análoga al bit clásico, en la computación cuántica existe un concepto similar apodado *qubit* –derivado de *quantum bit*. Un qubit es la unidad –objeto matemático– más básica de información cuántica que representa un sistema cuántico de dos estados cuyos valores pueden tomar el 0 o el 1. Además, existen un estado de *superposición* coherente que supone la combinación de 0 y 1, consecuencia de las leyes de la mecánica cuántica (Nielsen, M. A., & Chuang, I. L., 2001). Esto significa que un qubit puede ser 0, 1 o 0 y 1 a la vez. De tal forma, un sistema de dos bits, por ejemplo, puede representar cuatro estados: 00, 01, 10 y 11; pero solo podrá encontrarse en uno de ellos al mismo tiempo. Por su parte, un sistema de dos qubits también cuenta con el mismo número de estados: 00, 01, 10 y 11; pero la gran diferencia es que, debido a la superposición, el sistema se puede encontrar en los cuatro estados al mismo tiempo. Esto se podría ver como tener cuatro ordenadores clásicos trabajando conjuntamente.

El estado cuántico de un qubit puede describirse como un vector unitario combinación lineal de dos estados básicos ortonormales –vectores básicos. Estos estados básicos son conocidos como *estado fundamental* o *base* –$|0\rangle$– y *estado excitado* –$|1\rangle$– los cuales son analogías del 0 y 1 del bit clásico[4]. Los vectores base pueden ser expresados como vectores columnas ortonormales:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

[4] Dicha notación (| ⟩) se puede leer como *ket*, y fue introducida por el matemático Paul Dirac (Dirac, P. A. M., 1939).

De tal forma, un qubit puede representarse como la combinación lineal de los estados básicos |0⟩ y |1⟩ tal que:

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

donde α y β son conocidos como amplitudes de probabilidad y, generalmente, se tratan de números complejos que contienen información cuántica. Por tanto, $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ representa un estado del qubit siempre y cuando α y β sean números complejos y $|\alpha|^2 + |\beta|^2 = 1$. De esta forma, el estado de un qubit puede describirse mediante un vector columna bidimensional unitario, donde la magnitud al cuadrado de sus entradas debe sumar 1:

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

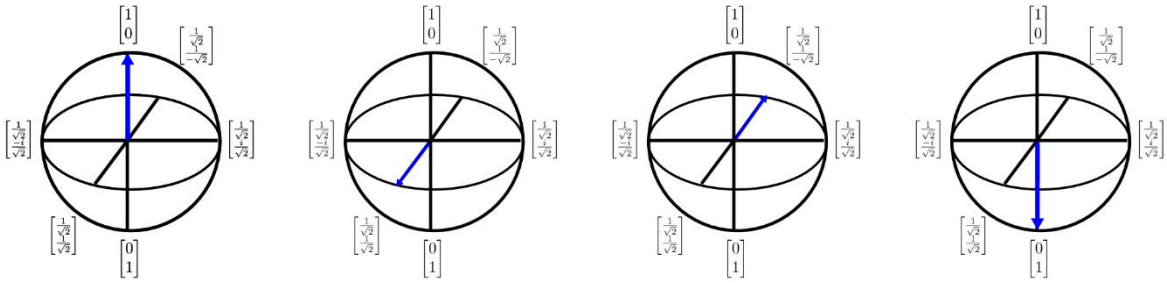
Llegados a este punto, se puede plantear el significado de los estados y lo que estos representan. Como se han comentado anteriormente un qubit es un elemento abstracto regido bajo las leyes de la cuántica. Los estados del qubit no son conocidos hasta que este es medido. Este término puede entenderse como un proceso de observación del qubit, el cual *colapsa* en uno de los dos estados base $\begin{bmatrix}1\\0\end{bmatrix}$ o $\begin{bmatrix}0\\1\end{bmatrix}$. En otras palabras, el qubit no es un valor discreto hasta que se mide. Cuando el vector estado $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ es medido, se obtiene uno de los valores clásicos: 0 o 1. A diferencia de la computación clásica, donde los estados son determinados y exactos, en la computación cuántica solo es posible hablar de los estados cuánticos en términos de probabilidad. Es decir, el estado no es otra cosa que la probabilidad con la que este mismo puede llegar a ser 0 o 1 cuando es observado. De tal forma, $|\alpha|^2$ es la probabilidad de que el resultado colapse a 0, mientras que $|\beta|^2$ es la probabilidad de que el resultado sea 1. Así, con la salida 0, el estado es $\begin{bmatrix}1\\0\end{bmatrix}$ y con salida 1, $\begin{bmatrix}0\\1\end{bmatrix}$ [5].

Como se ha comentado anteriormente, un qubit es una representación de un espacio vectorial bidimensional unitario cuyos valores α y β cuentan con dos grados de libertad cada uno. Existe una forma de representar los qubits en 3D: la esfera Bloch. La

[5] Los signos del vector estado son irrelevantes ya que las probabilidades de medir 0 y 1 dependen del cuadrado de los términos, por lo que la introducción de valores negativos no afectará a la probabilidad final. Así, negar un vector será irrelevante: $\alpha \to -a$ y $\beta \to -\beta$.

esfera de Bloch reduce los grados de libertad de cada valor complejo (α y β) a uno. De tal forma, los estados del qubit son repartidos a lo largo de la superficie como un vector tridimensional con dos grados de libertad, los cuales son reconocidos generalmente como ϕ y θ, donde θ comprende los valores entre 0 y π, y ϕ, los valores de 0 a 2π. La esfera Bloch se puede visualizar de la siguiente manera:

Ilustración 2. Representación de un qubit mediante la esfera Bloch.



Microsfot, 2017. Fuente: https://docs.microsoft.com/en-us/quantum/concepts/the-qubit?view=qsharp-preview

donde la superposición es comprendida como un continuo de estados entre $|0\rangle$ y $|1\rangle$ que no define ningún valor concreto hasta que el qubit colapsa. Una vez es observado, el estado adquiere un valor definido resultante en *0* o *1*, probabilísticamente, lo que se traduce como un único bit de información acerca del estado de qubit (Nielsen, M. A., & Chuang, I. L., 2001). Por ejemplo, el estado $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ tiene un cincuenta porciento de posibilidades de colapsar tanto a *0* como a *1*. Esto es así ya que $|1/\sqrt{2}|^2 = 0{,}5$.

Como cabe comprender, un qubit individual es una unidad de información muy pequeña. Así, como se ha visto previamente, el verdadero potencial de la computación cuántica aparece cuando varios qubits están entrelazados. El entrelazamiento significa que varias partículas funcionan como un conjunto inseparable donde los efectos de unas afectan a las otras, pues la dimensión del espacio vectorial crece exponencialmente con el número de qubits.

El número de qubits indica la cantidad de bits que pueden estar en superposición. Por ejemplo, en un sistema con dos qubits los estados-base son: $|00\rangle, |01\rangle, |10\rangle$ y $|11\rangle$.

$$|00\rangle \equiv \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \qquad |01\rangle \equiv \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix},$$

$$|10\rangle \equiv \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \qquad |11\rangle \equiv \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}.$$

donde $\otimes$ es el producto tensorial, una operación matricial con la que es posible formar un estado de dos qubits a partir de dos estados simples de un qubit.

Estos estados pueden formar un estado intermedio combinación lineal de los cuatro. El estado cuántico de *n* qubits se representa como un vector unitario de 2^n.

De nuevo, un sistema de dos qubits describe un vector:

$$\begin{bmatrix}\alpha_{00}\\\alpha_{01}\\\alpha_{10}\\\alpha_{11}\end{bmatrix},$$

si $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. Así mismo, los estados de los qubit, individualmente hablando, son descritos como $\begin{bmatrix}\alpha\\\beta\end{bmatrix}$ y $\begin{bmatrix}\gamma\\\delta\end{bmatrix}$ de tal modo que el estado global de los dos qubits es:

$$\begin{bmatrix}\alpha\gamma\\\alpha\delta\\\beta\gamma\\\beta\delta\end{bmatrix}.$$

Entonces ψ es la combinación lineal de los cuatro estados base $|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$, donde $|\alpha_{00}|^2$ es la probabilidad de que el qubit resulte en $|00\rangle$, $|\alpha_{01}|^2$ en $|01\rangle$, y así sucesivamente.

2.3 Sistemas físicos de la computación cuántica

Los qubits, al igual que los bits, son una representación abstracta de fenómenos físicos que tienen lugar en circuitos electrónicos. Los estados pueden ser manipulados cuánticamente aplicando diversas técnicas y estímulos superpuestos e interconectados a lo largo de varias etapas (Devoret, M. H., & Schoelkopf, R. J., 2013). (1) Para que dichos sistemas sean válidos, inicialmente los qubits deben estar totalmente controlados y estables el tiempo suficientemente para que puedan ser leídos, escritos y, por tanto, manipulados. (2) En segundo lugar, los sistemas deben ser capaces de ejecutar cualquier tipo de algoritmos cuánticos. (3) La tercera etapa consiste en corregir los errores mediante la lectura de los síndromes de error. En este punto, se introduce en el sistema un estado,

llamado síndrome, que informa del ruido o interferencias externas que afecten al conjunto del sistema, tema que será abordado posteriormente con más profundidad.

Los primeros sistemas físicos que consiguieron implementar las etapas descritas anteriormente fueron, por una parte, la *trampa de iones* (Cirac, J. I., & Zoller, P., 2000), (Jaksch, D., García-Ripoll, J. J., Cirac, J. I., & Zoller, P., 2016), y más recientemente, el *circuito superconductor* (Kim, T., Maunz, P., & Kim, J., 2011) (Reed, M. D., et al, 2012). No obstante, existen otros sistemas que han tenido también un éxito relevante y que permiten manipular los qubits. Algunos de estos ejemplos son las vacantes de Diamante, NRM o los sistemas ópticos.

La *trampa de iones* es el primer sistema que se desarrolló para la implementación de ordenadores cuánticos. Estos sistemas hacen uso de átomos de forma natural. Los átomos son capaces de almacenar información cuántica mediante sus espines –o niveles de energía nuclear. Pero a diferencia de un átomo normal, en la trampa de iones los átomos están cargados positivamente –átomo conocido como ion– de forma artificial, eliminando un electrón. Esta detracción es la causa de que los iones interactúen entre sí a través de los campos electromagnéticos generados por electrodos cercanos. Con esta técnica, se pueden llegar a capturar muchos iones, los cuales quedan atrapados, formando una especie de malla llamada cristal estacionario donde cada ion actúa de forma individual. Mediante láseres de control, los iones son empujados para entrelazar sus estados de giro a través de las vibraciones de los iones. En tal situación, los iones son considerados qubits individuales y completamente funcionales, pudiendo llevar a cabo operaciones computacionales. Pero cuando estas mallas cuentan con grandes números de qubits, su estabilidad y manipulación es muy compleja (Monroe, C. R., Schoelkopf, R. J., & Lukin, M. D., 2016).

Los *circuitos superconductores* se caracterizan por utilizar qubits controlables que pueden ser introducidos o eliminados selectivamente de circuitos de alta capacidad conductora. Aunque estos dispositivos contienen una gran cantidad de átomos, se caracterizan por comportarse como qubits simples y controlables con capacidad de entrelazarse y desacoplarse a otros qubits de forma selectiva y eficiente mediante la modificación de la dirección de la corriente del circuito que corresponde a los estados 0 (sentido horario) y 1 (sentido antihorario). Importantes entidades como Google e IBM están desarrollando sistemas superconductores capaces de manipular y entrelazar varios qubits con una técnica conocida como *electrodinámica cuántica de circuito –circuit*

quantum electrodynamic– (Wallraff, A., Schoelkopf, R. J., 2004). Una de las principales ventajas de este tipo de circuitos es su capacidad modular, es decir, es posible crear un gran circuito a través de la conexión de pequeños e independientes módulos mediante cables superconductores con una consiguiente reducción de las interferencias entre módulos. De tal forma, en caso de fallos o problemas puntuales de los componentes, podrían llegar a eliminarse u omitirse los módulos defectuosos, volviendo a establecerse las conexiones entre ellos para obtener la misma disposición, o para crear diferentes arquitecturas. De esta forma se evitan los grandes y complejos circuitos íntegros (Monroe, C. R., Schoelkopf, R. J., & Lukin, M. D., 2016).

Utilizar *qubits con espines de estado sólido* es otra técnica donde los qubits son capaces de almacenar información cuántica en los espines[6] de materiales de estado sólido, en lugar de gases o átomos discretos. Un candidato para realizar dicho almacenamiento cuántico de estado sólido es el *Centro de Defectos de Vacante de nitrógeno* –conocido como CDV– en el diamante. Esta técnica comienza llevando a cabo unos defectos artificiales en un sistema de átomos de carbono de diamante. De esta red se sustituye un átomo de carbono por uno de nitrógeno. Esta modificación genera un espacio vacío próximo llamado vecino vacío. La impureza del CDV genera una serie de impulsos electromagnéticos entre las partículas de tal forma que estas se relacionan manteniendo su individualidad y permitiendo la medición y control de alta fidelidad de los espines del qubit (Bernien, H., 2013). A pesar de ello, una de las grandes limitaciones de este tipo de sistemas es que una impureza de vacante de nitrógeno tiene solo unos pocos vecinos cercanos de carbono, lo que limita el número total de qubits por módulo a menos de una docena. No obstante, el estado sólido está presentándose como una seria alternativa a las trampas de iones debido a su requisitos ópticos menos exigentes y operaciones más rápidas. Así, según los últimos estudios, una pequeña placa de 4.5x4.5x0.5 mm dentro de un diamante basada en cinco qubits nucleares por CDV permitiría el uso de 5 millones de qubits (Stephen, C. J., 2018).

En los últimos tres años ha aparecido un nuevo concepto de ordenador cuántico basado en los fotones. Son los llamados *sistemas ópticos cuánticos*. Los fotones son las partículas cuánticas más elementales de los campos electromagnéticos y se caracterizan por su baja decoherencia y mínima pérdida de transmisión de información. Es por esto

[6] El espín es una propiedad intrínseca de las moléculas subatómicas, del mismo modo que la masa o la carga eléctrica. Esta característica tiene varias expresiones las cuales pueden ser traducidos como estados cuánticos. Fuente: https://astrojem.com/teorias/espin.html.

por lo que esta tecnología ha tomado gran protagonismo en la comunicación cuántica al poder codificar y procesar naturalmente sistemas cuánticos multidimensionales. Este proceso se lleva a cabo mediante una excitación coherente y controlable simultanea de muchos fotones los cuales se encuentran en varios modos simultáneos en una matriz de fotones (Dusanowski, Ł., Kwon, S. H., Schneider, C., & Höfling, S., 2019). Dichos fotones se entrelazan arbitrariamente permitiendo realiza mediciones multidimensionales de alta fidelidad. Esto ha sido plasmado en un chip fotónico cuántico integrado que es capaz de generar, manipular y medir los qubits entrelazados con una gran precisión, control y universalidad[7]. Además, el estado del sistema puede ser manipulado a alta velocidad, a diferencia de otros sistemas cuánticos (Wang, J., 2018). Esto abre la puerta a innumerables aplicaciones directas como el desarrollo de redes cuánticas o tecnologías multidimensionales avanzadas. Recientemente ha visto la luz un método de óptica cuántica el cual no hace uso directo de fotones, sino de átomos ultrafríos (Krinner, L., Stewart, M., Pazmiño, A., Kwon, J., & Schneble, D., 2018). Los átomos son controlados con rayos láser y gracias a esto el sistema muestran niveles de decoherencia muy bajos, ya que los átomos están muy bien aislados. En palabras de Ignacio Cirac, estos sistemas "*constituyen una plataforma casi perfecta para simular problemas cuánticos complejos, así como fenómenos de óptica cuántica*" (Cirac, J. I., & Tudela, A. G., 2019).

2.4 Computación reversible y puertas lógicas

La energía es un recurso fundamental en la computación, tanto en la clásica como en la cuántica, que está estrechamente relacionada con la reversibilidad. La reversibilidad es un concepto que puede ser entendido desde dos puntos de vista. Por una parte, la reversibilidad es la diferencia de energía que existe en un sistema tras haber pasado por un dispositivo electrónico. Estos dispositivos que pueden modificar, o no, las entradas, son conocidos como puertas lógicas. Si la diferencia de energía existente en un circuito es la misma después de aplicar una puerta lógica, esta se dice que es reversible, pues dada la salida de la operación, es posible inferir la entrada. Por el contrario, si la energía del sistema es menor después de aplicar una puerta lógica, entonces será imposible saber cuál fue la entrada a partir de la salida de la operación. Este concepto puede ser abordado en términos de información. Una puerta lógica irreversible elimina irreparablemente la

[7] Universalidad hace referencia a poder llevar a cabo operaciones universales en sistemas cuánticos para cualquier dimensión.

información de la entrada. De este modo, una puerta reversible será aquella que, durante de un proceso de computación, no eliminará ninguna información de la entrada (Nielsen, M. A., & Chuang, I. L., 2001).

La relación entre la energía y la irreversibilidad viene conectada con el principio termodinámico del procesamiento de la información de Landauer. Cuando un bit es eliminado, la entropía del sistema aumenta debido a la información de procesamiento de información o de su entorno (Bennett, C. H., 2003). Es decir, eliminar información de un sistema supone una disipación de energía.

Por tanto, si todas las puertas lógicas fuesen reversibles, ningún bit sería borrado durante el proceso de cálculo y el principio de Landauer implicaría existiría entropía en los sistemas de cómputo. Por tanto, sería lógico preguntarse si toda la computación podría llevarse a cabo sin eliminar información. Intuitivamente se puede pensar en las puertas lógicas irreversibles *AND* y *OR*.

Un requisito que deben cumplir todas las operaciones cuánticas es que deben ser reversibles, lo que está estrechamente relacionado con la computación universal reversible[8]. De esta forma, existen una serie de *puertas lógicas universales*[9] que se comportan igual que las puertas clásicas AND, NOT y NAND. Una de dichas puertas es la llamada *puerta Toffoli* o *puerta reversible CCNOT*, que será comentada posteriormente junto al resto de puertas universales.

Si los qubits son representados mediante vectores, las puertas lógicas son representadas mediante matrices. De esta forma, cualquier matriz definida se trata de una puerta lógica, es decir, las puertas cuánticas pueden involucrar uno, dos, tres o N qubits. Sin embargo, es muy difícil implementar físicamente puertas cuánticas de más de dos qubits. No obstante, cualquier operación puede ser aproximada con una precisión tan pequeña como se quiera mediante un conjunto de puertas cuánticas universal compuesto por puertas de 1 qubit y 2 qubits. Las puertas cuánticas de un solo qubit pueden entenderse en términos de la esfera Bloch. Las puertas cuánticas más simples –un qubit– e importantes se las conoce como *puertas Pauli*. Estas puertas son *X*, *Y* y *Z*.

[8] La computación universal está basada en el principio de que cualquier operación puede ser computada sin eliminarse ninguna información. No obstante, el uso de ciertas puertas irreversibles hace suponer que existe, al menos, una puerta universal que hace de las operaciones reversibles, operaciones universales (Nielsen, M. A., & Chuang, I. L., 2001).

[9] Un conjunto G de puertas es universal para un modelo computacional si cualquier función computable en este modelo puede ser calculada por un circuito que usa solo puertas en G (Arkhipov, A., 2009).

La *puerta X*, también llamada NOT, invierte el resultado de entrada, al igual que ocurre en una puerta NOT clásica. Esta puerta rota la esfera Bloch sobre el eje X en π radianes y viene definida por la matriz:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

La entrada de dicha puerta es un qubit representado en forma de matriz, por ejemplo, $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$, o $|0\rangle$. Dicho qubit es operado por la puerta X llevando a cabo una sencilla multiplicación de matrices lo que resulta en el valor opuesto:

$$X \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$$

La *puerta Y* actúa sobre un qubit rotando la esfera Bloch sobre el eje Y en π radianes. Esto significa que cuando la entrada es $|0\rangle$, la transforma en $i|1\rangle$, así como $|1\rangle$ en $-i|0\rangle$. Se representa con la matriz

$$Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.$$

Una de las más utilizadas es la *puerta Z*. Esta puerta rota la esfera Bloch sobre el eje Z en π radianes, lo que modifica únicamente $|1\rangle$, cambiando dicha amplitud por $-|1\rangle$. $|0\rangle$ queda invariable. La matriz se define:

$$Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$Z \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = -|1\rangle$$

Existen otras tres puertas de un solo qubit que cobran una gran importancia en la computación cuántica. Estas son *H*, *S* y *T*.

La *puerta H,* o *Hadamard*, es una de las puertas lógicas cuánticas de mayor importancia. Viene definida por la matriz $H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ y representa un giro de π sobre el eje $\frac{x+z}{\sqrt{2}}$. De tal manera, la puerta transforma el qubit de entrada con probabilidad uno en un qubit con probabilidad $\frac{1}{2}$ para cada valor. Así, si la entrada es $|0\rangle$:

$$H\begin{bmatrix}1\\0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}\right) = \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}$$

y si la entrada es $|1\rangle$:

$$H\begin{bmatrix}0\\1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\0\end{bmatrix} - \begin{bmatrix}0\\1\end{bmatrix}\right) = \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$$

Adicionalmente, las puertas T y S están estrechamente relacionadas ya que $S = T^2$. La puerta T se define mediante la matriz:

$$T = \begin{pmatrix}1 & 0\\0 & \exp\left(\frac{i\pi}{4}\right)\end{pmatrix}$$

tanto que la puerta S, que representa un giro de 90º sobre el eje Z, es definida por:

$$S = \begin{pmatrix}1 & 0\\0 & -i\end{pmatrix}.$$

Las puertas NOT y H conforman, junto con la puerta de fase compleja $-P(i)-$ y la puerta de rotación $-R_\theta-$, aquellos simples bloques con los que se pueden construir las puertas cuánticas universales.

$$P(i) = \begin{bmatrix}1 & 0\\0 & i\end{bmatrix}$$

$$R_\theta = \begin{bmatrix}\cos\theta & -sen\,\theta\\sen\,\theta & \cos\theta\end{bmatrix}$$

Este tipo de puertas son de especial importancia porque expresan el mínimo necesario para realizar cualquier cálculo cuántico en un computador cuántico, es decir, la esencia de la electrónica cuántica. De esta forma, es posible construir las puertas universales reversibles CNOT y CCNOT (Arkhipov, A., 2009).

La *puerta CNOT –Controlled NOT gate–* se caracteriza por tener dos entradas. La primera de ellas es el *qubit de control* que, dependiendo de su estado, tendrá un efecto determinado sobre la segunda entrada, el *qubit objetivo*. De tal forma, si el qubit de control es un cero, este deja indiferente al objetivo. En caso de ser un uno, lo invierte. Esta puerta se representa mediante la matriz:

$$U_{CN} = \begin{bmatrix}1 & 0 & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\\0 & 0 & 1 & 0\end{bmatrix}$$

Si el bit de control es $|0\rangle$, entonces el bit objetivo no varía:

$$|00\rangle \rightarrow |00\rangle;\ |01\rangle \rightarrow |01\rangle$$

Sin embargo, si el control es $|1\rangle$, el bit objetivo se transforma en el contrario:

$$|10\rangle \rightarrow |11\rangle;\ |11\rangle \rightarrow |10\rangle$$

La puerta *Toffoli*, también conocida como CCNOT –*Controlled-Controlled NOT gate*–, es una puerta de tres qubit de los cuales los dos primeros son qubits de control. Si se introduce dos unos, entonces el tercer qubit, el objetivo, será invertido. Debido a su universalidad, cualquier operador puede ser construido a partir de ella; sin embargo, no puede formar por si misma un circuito cuántico y debe implementarse junto a las puertas simples descritas anteriormente para que sea un sistema universal (Shi, Y., 2002). La puerta es definida con la matriz:

$$\text{CCNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Tabla 1.Tabla de verdad de CCNOT

a	**b**	**c**	**c⊗ab**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

3. Clases de complejidad. Dilema P vs NP

La *máquina de Turing* es un modelo matemático de computación desarrollado por Alan M. Turing (1937) para la ejecución de algoritmos. La máquina consiste en dos

elementos: por una parte, una cita infinita que transporta símbolos de un alfabeto finito, y, por otra parte, un controlador o una serie de reglas de un estado interno que permiten leer, escribir o borrar los símbolos. El estado inicial pertenece a un conjunto de estados finitos, por lo que el comportamiento que sigue la máquina es el mismo que el de un autómata, es decir, un modelo matemático formado por un conjunto de estados los cuales varían según la entrada. De esta forma, un programa es entendido como un conjunto de reglas que determinan al controlador y cuyo estado inicial determina el resultado de este.

Posteriormente, Turing presenta la *máquina de Turing universal*, una hipotética máquina que puede computar cualquier secuencia de símbolos que cualquier máquina de Turing puede computar, con una salida idéntica a la otra. Esto supuso el primer indicio de ordenadores programables que podían ejecutar programas distintos. Pero a su vez, supuso nuevos problemas, como el *problema de la parada*. Es decir, dada una máquina de Turing y un programa cualquiera junto con unas entradas, cuándo pararía de ejecutarse, o si, por el contrario, continuaría para siempre. Turing demostró que ningún algoritmo puede resolver el problema de la parada, es decir, es un problema indecidible. Tal caso es conocido por ser uno de los primeros problemas de decisión[10] y toma especial relevancia por el hecho de definir una clase de aplicaciones que ningún programa puede realizar.

De esta forma, las *tesis Church-Turing* recalca que "*una función es intuitivamente computable si y solo si es computable por una máquina de Turing, o de manera equivalente si está especificada por una función recursiva*" (Soare, R. I., 2009). Sin embargo, dicha afirmación no tiene en cuenta la eficiencia en la computación, pues un programa puede ser computable, pero el coste de cálculo es absurdo en términos de tiempo o espacio en memoria. Es decir, el problema es intratable.

3.1. Teoría de la complejidad computacional

La complejidad computacional de un algoritmo aumenta cuando el tamaño de este también lo hace en términos de variables o inputs. En este aspecto, existen diferentes niveles de complejidad según el algoritmo. La teoría de la complejidad clasifica los problemas en clases según la relación entre el tamaño del problema y el mínimo de recursos necesario para llegar a una solución. Estos recursos limitados y variables con los que un programa tiene que lidiar son el tiempo y el espacio en memoria. El tiempo se

[10] Un programa de decisión es aquel que resulta una de dos posibles salidas: aceptar o rechazar.

traduce como una el número de pasos computacionales, u operaciones, requeridas para solucionar el problema a partir del número de instancia. Asimismo, el tamaño hace referencia al número de bits en memoria o el número de celdas utilizadas en una máquina de Turing requeridas para la solución del algoritmo.

Es importante resaltar que la teoría de la complejidad se basa en la clasificación de los algoritmos según los niveles de complejidad –por lo que es posible diferenciar dos problemas por su complejidad resolutiva–, y no en buscar una resolución a problemas concretos.

La complejidad de un problema conduce a pensar que existen dos aproximaciones ante este: la complejidad de resolución y la complejidad de verificación de la solución. Así, un problema, generalmente, es más difícil de resolver que de chequear si una solución es correcta. Esta afirmación está íntimamente relacionada con el famoso problema de *P versus NP* y, concretamente, P diferente de NP ($P \neq NP$), donde P refiere a aquellos problemas que son resolubles eficientemente y NP, a aquellas tareas cuya solución es verificable eficientemente[11,12]. En este contexto el término *eficiente* significa que el número de operaciones crece de forma polinómica en relación con las entradas del problema. Por el contrario, un algoritmo ineficiente requiere un mayor tiempo que el polinómico, típicamente, exponencial (Cobham, A., 1964).

Dicho esto, podemos diferencia varios niveles jerárquicos de los problemas según su complejidad. Por un lado, los problemas considerados como *fáciles* son aquellos que utilizan o bien un número constante de operaciones, o bien o bien el número de operaciones crece a un ritmo logarítmico: $log_2(n)$. A continuación, se establecen aquellos algoritmos *tratables*, es decir, sus operaciones crecen con un coste polinómico. Estos problemas pueden ser lineales, cuadráticos, etc. Por último, en el escalafón superior se encuentran los problemas *intratables*, cuyas operaciones crecen a ritmo exponencial. Esta clase de problemas se encuentra en una frontera abstracta, pues puede darse el caso que un problema no cuente, hasta la actualidad, con ningún algoritmo que consiga

[11] Los algoritmos de clase P conciben su nombre del inglés *Polynomial-Time*, mientas que los de la clase NP son *Nondeterministic polynomial-time*.

[12] NP está relacionado con máquinas de Turing no deterministas, es decir, aquellas que tienen varios posibles movimientos en la cinta dada una configuración específica. A diferencia de una máquina determinista, donde existe un estado inicial seguido de uno o varios estados determinados única y exclusivamente por una la función de transición –lo que significa que solo hay un único estado sucesor–, en una máquina no determinista existe un estado inicial seguido de múltiples estados sucesores permitiendo a la máquina explorar todos ellos hasta que uno es aceptado. Aceptando el estado, toda la máquina es aceptada.

resolver dicho problema en tiempo polinomial, lo que no significa que nunca se vaya a descubrir uno, es decir, que siempre sea intratable.

De este modo, dicha frontera adquirió el nombre de NPC o NP-completo, albergando los problemas más complejos de NP, aquellos que su solución puede ser verificada, pero aún no es sabido si es posible hallar una solución eficientemente, debido a que, a medida que el problema crece y se añaden más variables, el tiempo de resolución del problema se incrementa exponencialmente. En otras palabras, un problema es NP-completo si cualquier problema en NP es reducible a él[13] (Goldreich, O., 2008). Si algún problema NP-completo es resuelto por un algoritmo en tiempo polinómico, entonces todos los problemas en NP también tendrán solución, lo que significaría que $P = NP$. Es decir, un problema que se encuentre en NP será NP-completo si cualquier problema en NP es reducible a ese. Por tanto, mostrando que un problema es NP-completo implica que ese problema no está en P, a menos que $P = NP$.

Adicionalmente, la aleatoriedad es un factor que suele estar estrechamente relacionado con los algoritmos. Los problemas BBP –*bounded-error probabilistic polynomial time*– son problemas que son resueltos por un algoritmo en tiempo polinómico aleatorio, con un error exponencialmente pequeño limitado a 1/3 para todas las entradas. Esta clase contiene a los problemas de P por lo que P ⊆ BPP. Aun así, existen problemas o entidades, como el conjunto de los números primos, que se encuentran en BPP (Solovay, R., & Strassen, V., 1977), pero que no es sabido si también se encuentran en P. La pseudoaleatoriedad para la generación de números en algoritmos aleatorios empieza a ser una razón de peso para pensar que $P = BPP$ (Cook, S., 2006).

Por último, existen dos conjuntos de problemas más generales en la jerarquía de las clases de complejidad: PSPACE –Polynomial Space– y EXP –Exponential. PSPACE engloba a aquellos problemas que una máquina determinista de Turing puede resolver en espacio polinómico, entendiendo espacio como el número de celdas en la cinta. Por otra parte, EXP alberga a aquellos problemas que pueden ser resueltos en una máquina determinista de Turing con límites de tiempo exponencial. Es decir, la máquina requiere $2^{p(n)}$ operaciones para *n* entradas y *p* polinómico. PSPACE puede compararse con EXP si es entendido en tiempo, de tal forma, si PSPACE requiere un número de espacios polinómico $p(n)$ y la máquina de Turing cuenta con k estados, el número total de

[13] Se dice que un problema es reducible a otro problema si es posible resolver eficientemente el primero cuando se le proporciona un algoritmo eficiente para resolver el segundo

configuraciones será $k\,p(n)2^{p(n)}$ que es como máximo, $2^{q(n)}$ para un algoritmo polinómico q, por lo que $PSPACE \subseteq EXP$ (Arora, S., & Barak, B., 2009).

Con todo lo anterior, es intuitivo llegar a la conclusión de que $P \subseteq NP$ así como $NPC \subseteq NP$ y $P \subseteq BPP$, con lo que una posible secuencia de inclusiones de clases de complejidad podría ser:

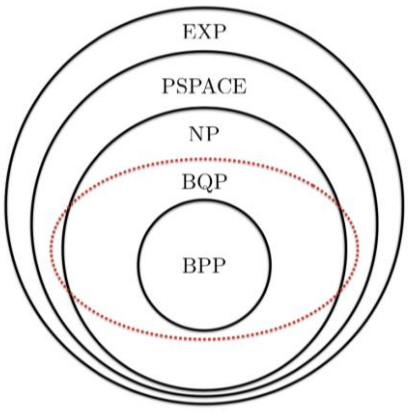
$$P \subseteq BPP, NP \subseteq PSPACE \subseteq EXP$$

Con los últimos avances en la computación cuántica la teoría sobre las clases de complejidad también ha sido expandida entorno a esta nueva cuestión (Bernstein, E., & Vazirani, U., 1997, p. 1414). La clase BQP –*bounded-error quantum polynomial time*– se atribuye a problemas de decisión resolubles en máquinas cuánticas de Turing (Deutsch, D., 1985, p. 97) con un error de probabilidad acotado menor que 1/3. Bernstein y Vazirani (1997) demuestran que $BPP \neq BQP$[14] y, por tanto:

$$PP \subseteq \text{BQP} \subseteq P^{\#P}$$

donde $P^{\#P}$ es el número de salidas correctas obtenidas de un algoritmo en P. Probar que $BPP \neq BQP$ demostraría que el poder de las computadoras cuánticas excede el de una máquina de Turing probabilística[15] (Sparrow, C., 2017). La ilustración 3 representa la jerarquía de las clases de complejidad computacional.

Ilustración 3. Jerarquía de las clases de complejidad computacional



Fuente: Sparrow, C., 2017.

[14] Bernstein y Vazirani desarollan un modelo artificial basado en un oráculo de caja negra, lo que no parece una demostración totalmente válida (Sparrow, C., 2017, p. 46).
[15] Una máquina de Turing probabilística se trata de una máquina de Turing no determinista donde cada estado sucesor es seleccionado de forma aleatoria de acuerdo con distribución probabilística.

Tabla 2. Definición de las clases de complejidad

Clase	Tipo	Definición
P	D	Problema determinista resoluble en tiempo polinómico en ordenador clásico
EQP	D	Problema determinista resoluble en tiempo polinómico en ordenador cuántico
BPP	D	Problema aleatorio resoluble en tiempo polinomial clásico con error acotado menor que 1/3
DQP	D	Problema aleatorio resoluble en tiempo polinomial cuántico con error acotado menor que 1/3
NP	D	Resultados verificables de un algoritmo en P en tiempo polinómico
PP	D	Problema aleatorio resoluble en tiempo polinomial clásico con error acotado menor que 1/2
#P	C	Cuenta el número de salidas correctas obtenidas de un algoritmo en P
GapP	Z	Diferencia entre el número de salidas correctas y rechazadas de un algoritmo en P
PSPACE	D	Memoria se incrementa de forma polinómica en ordenador clásico

Fuente: (Lund, A. P. et al, 2017).

3.2. Técnicas utilizadas para demostrar el problema P versus Np

Los problemas de clase NP y NP-completo han sido estudiados por una incontable cantidad de científicos usando técnicas muy variadas. Aun así, de momento, nadie ha sido capaz de hallar una solución válida (Fortnow, L., & Homer, S., 2003). Fortnow (2009) resume, de forma muy esquemática, las técnicas utilizadas y fallidas que han sido utilizadas para tratar de demostrar si $P \neq NP$.

La *búsqueda por fuerza bruta* es la técnica más simple que existe en la resolución de problemas y consiste en el sistemático análisis de todos los posibles candidatos mediante prueba y error para hallar una posible solución. Debido a las mejoras considerables en los computadores actuales, no es descabellado resolver problemas concretos pequeños mediante esta prueba. De esta forma, el problema del viajante,

comentado anteriormente, es posible llegar a una solución eficiente con más de 80.000 ciudades (Applegate, D. L., 2009). Los problemas de satisfacción de restricciones 3-SAT –explicados posteriormente– pueden encontrar una solución hasta las 100 variables. Pero debido a que los problemas NP crecen en tiempo exponencial, a medida que se incrementa en número de entradas, los tiempos de ejecución pasan a ser intratables.

La *diagonalización* es un método que comenzó a utilizarse en la computación en el siglo XIX. Esta técnica consiste en un lenguaje L en la clase NP diseñado para que un algoritmo polinómico no pueda calcular L correctamente. Este problema comenzó aplicándose al problema de la parada de Turing. Posteriormente, se utilizó para demostrar cotas inferiores exponenciales en problemas decidibles muy difíciles. Aunque esta técnica sea válida para dichos cálculos, el hecho de utilizar un conjunto de oráculos[16] relativos a $P = NP$ (Baker, T., Gill, J., & Solovay, R., 1975) sugiere que la diagonalización no es una técnica fiable para resolver el problema.

Otra técnica conocida es el uso de *tautologías, fórmulas booleanas y operadores lógicos*. Las tautologías en una formula booleana son fáciles de demostrar cuando estas son falsas, simplemente aplicando una asignación de variables con un resultado falso. No obstante, el proceso contrario es complejo mediante pruebas simples. Si se demuestra tal suceso, entonces se puede admitir que $P \neq NP$, pero tal cosa no ha ocurrido de momento. Los problemas de aproximación pueden servir para hallar resultados muy fiables en problemas de optimización NP-completo, usando, por ejemplo, la distancia euclídea. A pesar de esto, los problemas no dejan de ser NP-completo, por lo que no es válido para demostrar.

Otras técnicas utilizas son las *pruebas de complejidad, la complejidad parametrizada o el uso de heurísticas* y *casos promedio*. Aunque algunas de estas técnicas son prometedores en los ámbitos de la computación, como la complejidad de los casos promedio –los resultados promedios que obtiene un algoritmo cuando se introduce un set de entradas concreto–, el enfoque de los problemas mediante dichas técnicas para conjuntos NP-completos sigue siendo un problema sin solución.

[16] Un oráculo es una caja negra abstracta aplicada sobre una máquina de Turing usada en la resolución de problemas de decisión. Dicha caja puede solucionar problemas concretos en una simple operación.

3.3. Ejemplos de problemas según las clases de complejidad

La clase de complejidad P engloba muchos problemas clásicos resolubles en tiempo polinómico estudiados por la matemática desde tiempo inmemorial. Algunos de estos problemas son: la aritmética de enteros, álgebra lineal, flujo de red, programación lineal, así como muchos problemas de grafos: ruta más corta, árbol de expansión mínima, coincidencia bipartita, etc. (Cook, S., 2006).

En cambio, los problemas de tipo NP, son más complejos de encontrar y no fue hasta finales de siglo pasado cuando las tareas computacionales comenzaron a clasificarse por su dificultad. De tal forma, Johnson, D. S. (1982) aporta una gran lista de problemas NP-completos de los cuales algunos problemas naturales son de gran interés, entre los que se incluye la Satisfacción y 3-SAT, Clique o el gráfico hamiltoniano, entre otros.

Los problemas de satisfacción booleana –conocidos comúnmente como SAT– tratan de determinar, dada una expresión booleana, aquella asignación de valores que hacen que la expresión completa sea verdadera. 3-SAT es un caso concreto que requiere que cada clausula tenga exactamente tres literales. Este problema fue listado como NP-completo por S.Cook (1971).

A modo de ejemplo, si se tiene dos clausulas cualesquiera:

$$(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_6)$$

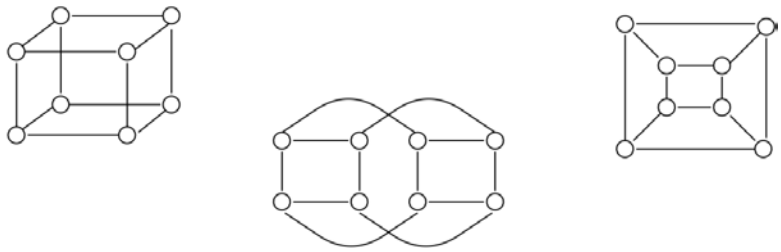
¿existe tal combinación de valores que hagan la expresión verdadera?

Otro ejemplo clásico de problema NP es el viajante de la mochila –t*ravelling salesman*. Este ejemplo se trata de un problema de minimización de costes, donde el objetivo es encontrar un camino hamiltoniano recorriendo un conjunto de nodos de tal forma que el coste total del recorrido sea el mínimo posible. El viajante parte desde un nodo inicial, debe pasar por cada nodo una única vez y volver al nodo inicial. Existen $(n-1)!$ diferentes caminos, por lo que a medida que el número de ciudades se incrementa, el tiempo de ejecución aumenta exponencialmente. De esta manera, un experimento llevado a cabo en los años 60 en un ordenador *IBM 7090* consiguió resolver el problema para 13 ciudades en 17 segundos. Cuando el número de ciudades creció hasta las 20, es decir, únicamente siete ciudades más, el tiempo aumento de forma exponencial hasta las 10 horas de ejecución (Held, M., & Karp, R. M., 1962). Así, existe un algoritmo que mejora estos resultados utilizando una técnica llamada retroceso: *algoritmo de Little* (Little, J. D.,1963). No obstante, como regla general, los autores establecieron que el

tiempo de este problema crece en un factor de 10 cada vez que se añaden 10 ciudades nuevas.

Por último, existen ciertos problemas de NP de los que no se puede decir nada sobre la clase donde están contenidos, ya sea en P o NP-completo. Este caso lo recoge el *problema del grafo isomorfo* (Fortin, S., 1996): dados dos grafos no dirigidos, determinar si son isomorfos (la ilustración 4 recoge algunos ejemplos). Se piensa que si $P \neq NP$, entonces existe una clase intermedia donde los grafos isomorfos pueden encontrar cabida. Una clase intermedia significa que existe un algoritmo capaz de resolver esta clase de problemas en tiempo moderadamente exponencial, es decir, que siguen una relación: $n^k < t < a^k$, donde *t* es el tiempo, *n* es el tamaño del problema y *a*, *k* son números reales arbitrarios, lo que significa que la función exponencial crece a un ritmo menor a *n*.

Ilustración 4. Grafos isomorfos.



Fuente: Fortin, S., 1996.

4. Algoritmos cuánticos

Un algoritmo cuántico es un algoritmo que es ejecutado en un computador cuántico y que presenta ventajas computacionales respecto a los algoritmos clásicos. Estas ventajas suelen ser en términos de velocidad y eficiencia en relación con aquellos problemas que para un ordenador clásico no son tratables. De tal modo, aunque los algoritmos clásicos y cuánticos sean semejantes, es decir, un algoritmo clásico podría ser ejecutado en un computador cuántico, la verdadera potencia de los algoritmos cuánticos es explotada cuando utilizan técnicas cuánticas como el entrelazamiento o la superposición. Asimismo, los algoritmos cuánticos presentan diferencias que los clasifican dependiendo de una serie de técnicas en las que están basados. Las técnicas más importantes son la transformada de Fourier, la amplificación de amplitud, la estimación de fase, las caminatas y la simulación cuánticas.

4.1. Técnicas en la creación de algoritmos cuánticos

Existen una serie de algoritmos cuánticos que están basados en la *transformada cuántica de Fourier*, una analogía cuántica de la transformada clásica de Fourier. La transformada de Fourier es una función matemática que descompone una serie temporal compleja en el total de ondas simples sinusoidales que componen dicha serie. Por su parte, la transformada cuántica se aplica al vector de amplitudes de un estado cuántico.

Un ordenador clásico puede realizar una transformada de Fourier utilizando $O(n2^n)$ puertas lógicas de Hadamard[17], o lo que es lo mismo, $N \log(N)$, ya que $N = 2^n$, donde *N* es la longitud del vector de números complejos que recibe la transformada como entrada y *n* es un entero que indica el número de qubits de un ordenador cuántico. Un ordenador cuántico puede resolver una transformada en $\log^2(N) = n^2$, lo que se traduce como un ahorro exponencial.

No obstante, el algoritmo tiene en cuenta cierta información de la que no es posible saber nada acerca de ella. Esto se debe a que la computación cuántica trabaja en un continuo entre las amplitudes $|0\rangle$ y $|1\rangle$ donde existen infinitos valores que no son conocidos y no pueden medirse. Aun así, cuando la salida es medida, esta colapsa en uno de los estados $|0\rangle$ o $|1\rangle$ haciendo conocido el resultado de la transformada. Así, la ventaja que aporta la computación cuántica para la resolución de la transformada es que para *n* qubits, los cálculos son realizados en 2^n amplitudes. Esto permite resolver eficientemente muchos problemas que no son solucionables en un ordenador corriente.

El algoritmo de Shor es un ejemplo de algoritmo que hace uso de la transformada de Fourier. El algoritmo de Shor busca factores de un número *n* tal que ese valor se encuentre entre 1 y *n* y, además, divida *n*. Estas operaciones se realizan en tiempo polinomial, exactamente a un ritmo de tiempo $O(log\ n^3)$ y $O(\log n)$ de espacio, donde n es el número que se desea factorizar. Esto supone una ventaja muy grande en comparación con los algoritmos clásicos ya que el mejor clásico podía resolver dicho algoritmo en $O(2^{(\log N)^{\frac{1}{3}}})$ (Sparrow, C., 2017).

Realmente el algoritmo está dividido en dos partes. Por un lado, una parte clásica que descompone los valores y por otra, un algoritmo cuántico para ordenar dichos valores mediante la búsqueda del periodo de una función. En este punto se reciben un par de registros de entrada a los cuales se les aplican una función cuántica y transformada de

[17] Una puerta Hadamard es un caso concreto de la transformada de Fourier

Fourier. El resultado de la transformada es un estado que puede ser medido y cuyo registro de salida será el resultado general (Bonillo, V. M., 2013).

La *amplificación de amplitud* es una generalización del algoritmo de búsqueda de Grover. Este problema de búsqueda sin información realiza alteraciones en un espacio de Hilbert modificando la amplitud del valor buscado por otra con un valor absoluto mayor, y sustituyendo el resto de las amplitudes por otras con amplitudes en valor absoluto menores, lo que incrementa la amplitud de ecuación de onda de la partícula. Es decir, para encontrar un valor *k*, se ubica un vector base |k⟩ en el espacio de Hilbert con una amplitud α_k y se sustituye por α'_k tal que $|\alpha'_k| > |\alpha_k|$. El algoritmo devuelve como resultado el valor *k* con una probabilidad $|\alpha_k|^2$ alta.

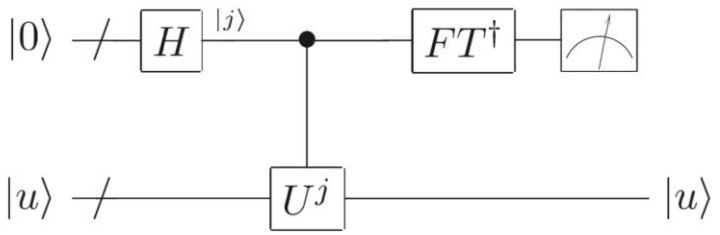
El algoritmo de búsqueda de Grover (Grover, L. K., 1996) es un caso particular de la amplificación de amplitud. Se trata de uno de los primeros algoritmos cuánticos que destacan los beneficios que aporta la mecánica cuántica. Este algoritmo tiene el objetivo de localizar un elemento concreto de una lista de elemento. Si se cuenta con una base de datos con N registros, este algoritmo es capaz de encontrar el resultado deseado en $\sqrt{N}$, a diferencia de un algoritmo clásico que necesita, en el peor de los casos, *N* operaciones.

El algoritmo de Grover es un algoritmo muy versátil ya que es aplicado para resolver otros tipos de problemas. Por ejemplo, el algoritmo puede ser aplicado en un problema de satisfacibilidad como es 3-SAT. En este caso, el algoritmo necesita exactamente en $O(1{,}41\ldots^n\ poly(n))$ pasos, donde *poly* hace referencia a un incremento polinómico. Sin embargo, el mejor algoritmo clásico para este problema es capaz de encontrar un resultado en $O(1{,}328\ldots^n)$ operaciones (Rolf, D., 2003). Esto demuestra que el algoritmo de Grover no aporta siempre una mejora de eficiencia ya que en este caso no funciona tan bien aplicándolo en problemas de satisfacibilidad. De tal modo, si en este algoritmo es aplicada una amplificación de la amplitud invocando un algoritmo A $O(\frac{1}{\sqrt{\epsilon}})$ veces, entonces esto supone la obtención de un resultado en $O\left(\sqrt{1{,}329\ldots^n}\right) = O(1{,}153\ldots^n)$ pasos, lo que implica una mejora cuadrática en el tiempo (Ambainis, A., 2004).

La *estimación de fase* es un procedimiento general de la transformada de Fourier usado en muchos algoritmos más. Existen un valor unitario U con vector propio |u⟩, y un valor propio $e^{2\pi i\varphi}$. El objetivo principal de la estimación de fase es hallar el valor φ haciendo uso de unas cajas negras u oráculos que preparan el estado |u⟩ de tal forma que sea resoluble. Esto significa que el algoritmo cuántico no es un algoritmo universal, pues

dichos oráculos son introducidos externamente. Así, puede entenderse la estimación de fase como un módulo que permite el cálculo de ciertos problemas concretos. La ilustración 5 muestra un esquema del procedimiento de la estimación de fase general mediante puertas cuánticas que utiliza dos registros. Por una parte, un registro con *t* qubits inicializados a $|0\rangle$ –el número de qubits depende de varios ajustes elegibles relacionados con el número de dígitos de φ– y por otra parte un registro $|u\rangle$ que es el vector propio que indica el número de qubits necesarios para almacenar el vector per se. Adicionalmente, '/' hace referencia a un conjunto de cables. En la estimación de fase, primero se aplica una transformada de Hadamard al primer registro y después una puerta de control al segundo registro. En la segunda etapa, se aplica la transformada inversa de Fourier al primer registro y, por último, se realiza una lectura del estado del primer registro que proporciona el resultado aproximado de φ.

Ilustración 5. Esquema de puertas lógicas para la estimación de fase

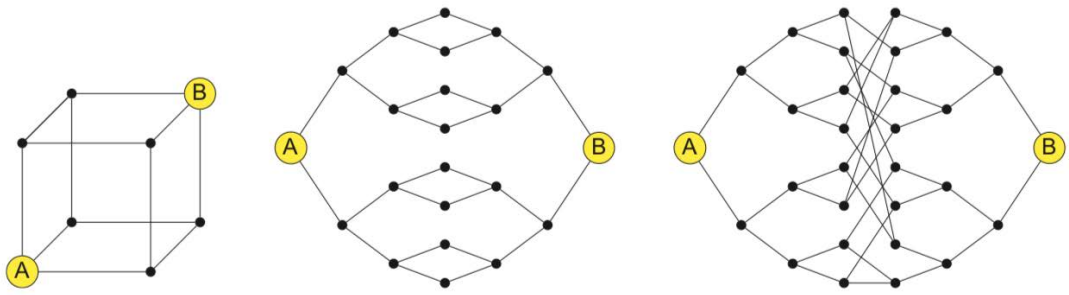


Fuente: (Nielsen, M. A., & Chuang, I. L., 2001).

La *caminata cuántica* es otra técnica derivada de las caminatas aleatorias y, en concreto, de la caminata del borracho. Esta técnica es aplicada en los problemas de búsqueda y de satisfacción de restricciones, como 3-SAT, los cuales no dejan de ser representaciones matemáticas de procesos físicos (Andraca, S. E. V., 2008). Este tipo de problemas están basados en una partícula localizada en una estructura gráfica la cual se mueve de forma aleatoria. Cuando se habla de caminata cuántica hace referencia a una partícula cuántica que se mueve en un grafo bajo los efectos de la mecánica cuántica. Estos efectos, a diferencia de la caminata clásica, permiten a la partícula encontrar un vértice objetivo más rápido a partir del vértice desde el que parte, o propagarse de forma más rápida por el conjunto de vértices del grafo. Una caminata clásica y una cuántica difieren en los tiempos necesarios para que una partícula encuentre un vértice. Un algoritmo clásico necesita del orden de $N^{\frac{1}{6}}$ operaciones, siendo N el número de vértices.

Esto que significa que cuando N es pequeño el algoritmo puede hacer búsquedas rápidas. No obstante, a medida que el número de vértices crece, el algoritmo no podrá resolver el grafo eficientemente, ya que la parte central aleatoria será demasiado grande (ejemplos de grafos en la ilustración 6). Por su parte, el algoritmo cuántico necesita $O(\log N)$ pasos (Montanaro, A. 2016). Tal es la diferencia que un grafo de 10 millones de vectores será resuelto por un algoritmo clásico en más de 14 operaciones, mientras que el algoritmo cuántico solo requerirá 7 operaciones, es decir, el doble de recursos.

Ilustración 6. Ejemplos donde de grafos donde un paseo aleatorio clásico requiere exponencialmente más tiempo que un paseo cuántico para llegar de A a B



Fuente: Montarano, A., 2016.

El algoritmo de Grover también puede aplicarse en estos problemas. El punto de interés surge en el número de consultas definidas por el problema. Por ejemplo, si hay dos consultas, el número total de pasos será dos veces la búsqueda del algoritmo de Grover $\left(\sqrt{N} \cdot \sqrt{N}\right) = N$. En un algoritmo clásico, esto requiere $O(N)$ consultas (Benioff, P., 2000). Si el problema cuenta con tres dimensiones, entonces se requieren $O\left(\sqrt{N}\sqrt[3]{N}\right) = O\left(N^{\frac{5}{6}}\right)$ que es mejor que la búsqueda tradicional, pero, aun así, sigue obteniendo peores resultados que la búsqueda de Grover simple. Así, las caminatas cuánticas arrojan mejores resultados haciendo búsquedas en $O\left(N/\sqrt{M}\right)$ operaciones siendo $M = N^{2/3}$ (Ambainis, A., 2004).

Los Algoritmos de *simulación cuántica* son usado para resolver sistemas de mecánica cuántica, por ejemplo, relacionados con la química cuántica, superconductividad, materiales… (Montanaro, A., 2016). Estos sistemas hacen uso de números complejos que suelen crecer de manera exponencial con el tamaño del sistema. Un sistema que contenga n distintos componentes ocupa en memoria c^n bits (siendo c una constante). Sin embargo, un ordenador cuántico solo necesitará kn qubits para llevar a cabo la simulación, donde k es una constante. Esto significa que cualquier sistema

clásico solo podría abordar este tipo de simulaciones en términos exponenciales de crecimiento a medida que los sistemas crecen en tamaño (Feynman, R. P., 1982a).

De tal modo, los sistemas cuánticos simulados han permitido avanzar en una gran diversidad de campos, como por ejemplo los algoritmos basados en el enfoque básico Trotter-Suzuki utilizado en la química cuántica (Hastings, M. B. et al., 2014). Asimismo, las simulaciones químicas cuánticas también se han aplicado a sistemas cuánticos basados en las trampas de iones con sus consiguientes algoritmos (Hempel, C. et al., 2018). Georgescu y sus compañeros (2014), listan una gran cantidad de proyectos donde la simulación cuántica ha mejorado potencialmente las aproximaciones mediante algoritmos clásicos.

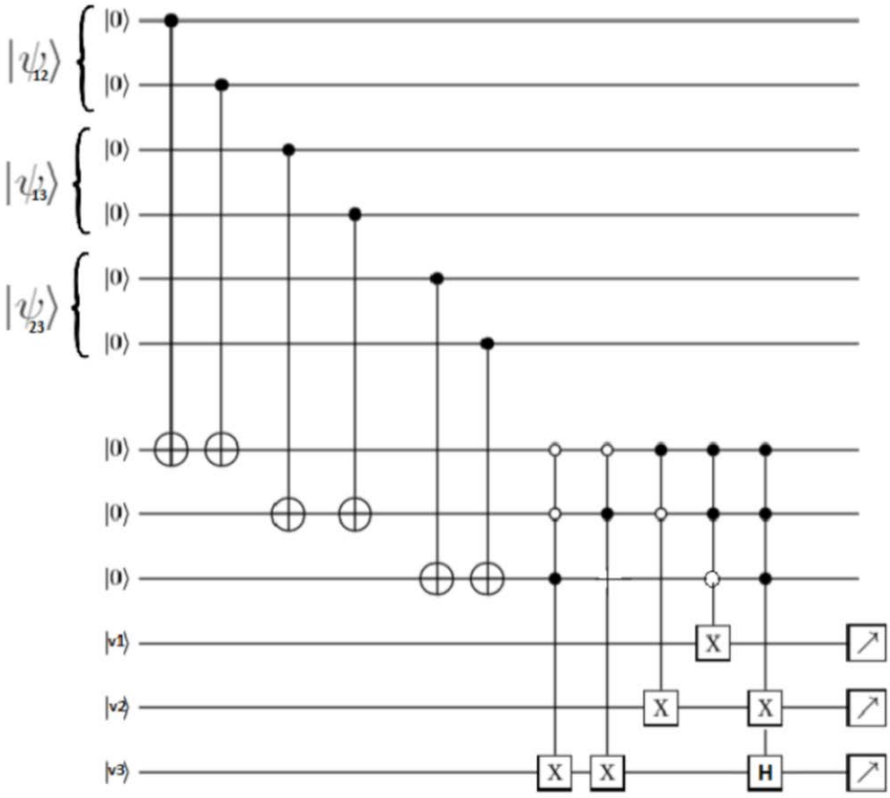
4.2. Clases de algoritmos

Los *algoritmos de búsqueda* tratan de hallar un elemento concreto en un espacio de búsqueda sin saber previamente nada acerca de la información que contiene. Así en un ordenador clásico, la búsqueda se realizaría en N operaciones, esto es, el peor de los casos, donde el objetivo fuese el último elemento del grupo. En un ordenador cuántico la búsqueda daría con un resultado en $\sqrt{N}$ operaciones. Los algoritmos de búsqueda heurística tienen muchas más aplicaciones que aquellos que usan la transformada de Fourier.

La *teoría de grafos* juega un papel importante en los problemas que complejidad. Muchos de estos problemas se consideran NP-completos. De ahí se desprende que no existen algoritmos que puedan resolver dichos problemas en tiempo polinómico a no ser que se demuestre que P=NP. Hoy en día este tipo de algoritmos se resuelven mediante definición de restricciones. Por ejemplo, restringiendo cada vértice a dos colores, el problema se resuelve en tiempo polinomial (2-SAT). Un ejemplo de este tipo de algoritmos es el *problema 3-coloring*: dado un grafo, el objetivo del problema es asignar *k* colores a los vértices de tal forma que la coloración sea apropiada, es decir, que los vértices adyacentes sean de colores diferentes (Mouatadid, L. 2014). Para los problemas de coloración, todavía no se han encontrado algoritmos cuánticos que resuelvan dicha tarea. No obstante, sí existen algoritmos que permiten comprobar si un grafo con *k* vértices está correctamente resuelto para 2 y 3 colores (Saha, M. et al., 2019). La ilustración 7 muestra un circuito cuántico para el problema de los 3 colores, donde cada

qubit representa un vértice distinto (1, 2 y 3): $|\psi_{12}\rangle$, ψ_{13} y ψ_{23}. El objetivo será colorear los vértices llamados v_1, v_2 y v_3 mediante puertas lógicas. Para ello se aplica la puerta lógica de identidad, representada por el $\oplus$ cuya matriz es aquella que deja invariable a la entrada $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$, la puerta NOT, o X y la puerta de Hadamard, o H.

Ilustración 7. Circuito cuántico para 3 colores.



Fuente: Saha, M. et al., 2019

Por otra parte, los algoritmos para encontrar *vectores y valores propios* son sumamente importantes en el campo de la física y química cuántica, ya que es necesario conocer los valores y vectores propios de un operador hamiltoniano. Hasta ahora estos problemas eran computados en simuladores cuánticos ejecutados en ordenadores comunes. De tal modo, estos algoritmos obtienen resultados exponenciales, generando además resultados poco deseados. Sin embargo, existe un algoritmo cuántico que resuelve este problema en tiempo polinómico cuántico (Abrams, D. S., & Lloyd, S., 1999).

Como fue comentado anteriormente, los algoritmos de *satisfacción de restricciones* se encuentran dentro del grupo de los problemas de optimización combinatoria y la mayoría de ellos se encuentra en la clase de complejidad NP-difícil. Es por esto por lo que los algoritmos clásicos resuelven dichos problemas en tiempo

exponencial. Uno de los algoritmos más famosos es el algoritmo de retroceso o *backtracking* que, a pesar de su tiempo súper polinómico, es más rápido que la resolución por la fuerza bruta. Aunque generalmente los algoritmos cuánticos pueden resolver problemas complejos en tiempo polinómico –mientras que un algoritmo clásico lo hace en tiempo exponencial– hay que aclarar que esto no sucede siempre ya que depende del tipo de algoritmo que se está tratando. Entrando en más detalle, los problemas 3-SAT acaparan mucho interés.

4.3. Machine learning e inteligencia artificial

Una de las posibilidades de los algoritmos cuánticos con las que se especula es la aplicación de la computación cuántica para la resolución de problemas relacionados con la inteligencia artificial y el *machine learning*. Existen algunos intentos al respecto, como por ejemplo D-Wave, que con su sistema adiabático de 2000 qubits están tratando de resolver problemas. Mediante el aprendizaje automático los algoritmos aprenden automáticamente a reconocer ciertos patrones recurrentes a partir de grandes cantidades de datos. Como es lógico pensar, existe una gran cantidad de combinaciones posibles a la hora de analizar y computar los datos, lo que requiere una gran cantidad de poder computacional. Debido a la cantidad de datos involucrados en este proceso y al inmenso número de combinaciones potenciales de elementos de datos, este es un problema de optimización muy costoso desde el punto de vista computacional.

D-wave, empresa canadiense de computación cuántica de la que se hablará posteriormente, cuenta con algunos proyectos donde se está aplicando el aprendizaje automático. QxBranch es un proyecto que aplica esta tecnología para predicciones en finanzas, aeroespacial, defensa y otras tecnologías[18]. Los *Alamos National Lab* está realizando análisis de grandes cantidades de imágenes mediante factorización de matrices de forma cuántica[19]. El método usado en este proceso es la factorización de matrices no negativas (NMF por sus siglas en inglés), que restringe las matrices de tal modo que deben ser siempre positivas. Esto es interesante porque cuando una gran base de fotos es analizada, muchas de estas fotos pueden no contener las imágenes deseadas, lo que

[18] Proyecto QxBranch. Fuente: https://www.dwavesys.com/sites/default/files/2018-04-04%20-%20Max%20Henderson%20-%20Quantum%20Machine%20Learning%20for%20Election%20Modeling.pdf

[19] Proyecto Los Alamos National Lab. Fuente: https://www.dwavesys.com/sites/default/files/ML%20with%20Facial%20Recognition.pdf

normalmente se tacha como un componente negativo. Dicho esto, si se utilizase otra técnica llamada PCA (Principle Components Analysis) se observaría que este método no elimina aquellas fotos no deseadas, siendo estos los cuadrados rojos (ilustración 8). Sin embargo, si se aplica NMF, entonces el aprendizaje no contará con esas imágenes, mientras que todas las fotos restantes proporcionan cierto valor al análisis. La factorización de matrices sigue una formula simple: $A = B \cdot C$. La ilustración 9 representa esta factorización mediante cuadros que representan las matrices. El primer cuadro (B) alberga algunas características de ciertas imágenes con las se puede llegar a construir una imagen real. Las filas contienen píxeles de una imagen. El segundo cuadro (C) contiene imágenes y características presentes en una imagen. En último lugar, el cuadro de la derecha (A) es la solución final compuesto por columnas que contienen una versión vectorizada de una imagen y, cada fila, un píxel de esa imagen.

Ilustración 8. Método PCA aplicado a imágenes de caras.

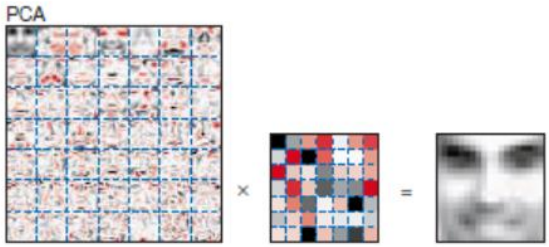
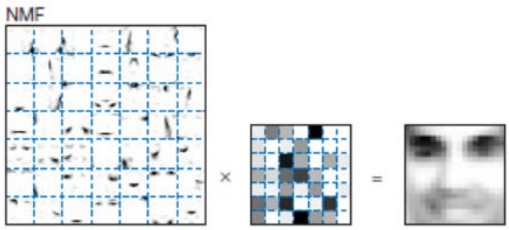


Ilustración 9. Método NMF aplicado a imágenes de caras.



Fuente ilustración 8 y 9: proyecto los Alamos National Lab.

Aunque la computación cuántica parece aportar muchas ventajas a los algoritmos de aprendizaje automático, como se ha visto anteriormente, todavía existen ciertas razones por las que no permiten considerar a esta tecnología como una alternativa a la computación clásica (Biamonte, J., Wittek, P. et al. 2017). Para Peter Wittek (2017) existen aún muchas barreras hardware y software que deben ser superadas para poder

hablar de una verdadera supremacía cuántica. Los cuatro principales problemas a los que se debe prestar especial atención para resolver son:

En primer lugar, la lectura de los *inputs* de los problemas es especialmente compleja si se compara con el procesamiento de los datos, y esta complejidad puede suponer un coste alto para los algoritmos cuánticos, los cuales pueden perder parte de su ventaja de procesamiento rápido. En segundo lugar, las *salidas* del problema no son completamente entendidas, pues el entrelazamiento de qubits computa infinitas soluciones hasta que los estados colapsan hasta un estado final concreto. En tercer lugar, se piensa que la computación cuántica ofrece especial ventaja ante problemas de gran tamaño, pero debido a las limitaciones hardware, esto todavía no se ha demostrado. En último lugar, es necesario mejores sistemas de evaluación de algoritmos para apreciar las ventajas de los algoritmos cuánticos frente a los clásicos.

5. En busca de la supremacía cuántica

> *"Usando mecánica cuántica en un ordenador se puede llegar a computar más eficientemente que en un ordenador clásico"*
> – Shor, P. W., 1994

Como se ha comentado anteriormente, existen distintas clases de problemas de acuerdo con su complejidad de resolución. Los problemas de tipo P son aquellos que pueden ser resueltos de forma eficiente, es decir, en tiempo polinómico. Por ejemplo, determinar si un número es primo entra dentro de la clase de problemas de tipo P (Agrawal, M., Kayal, N., & Saxena, N., 2004). Por otro lado, ciertos problemas, a medida que crecen, su dificultad de resolución lo hace de forma no polinomial –generalmente exponencial. Estos problemas son los de clase NP o su subclase NP-completo –problemas incluso más complejos. El problema del viajante o la factorización de enteros son ejemplos que entran dentro de esta clase.

Con el desarrollo de la mecánica cuántica y de los primeros algoritmos cuánticos, los problemas NP parecen atisbar algo de luz en cuanto a una resolución tratable y eficiente. Las computadoras cuánticas, cuyos principios están basados en las propiedades de la mecánica cuántica, prometen un procesamiento más rápido y eficiente –con menor esfuerzo computacional– que los sistemas clásicos (Preskill, J., 2012). Dicha declaración

es conocida como *supremacía cuántica* y su demostración es uno de los retos más importantes a los que se enfrentan los científicos hoy en día (Harrow, A. W., & Montanaro, A., 2017). No obstante, parece que este camino aún queda lejos, pues el poder que aporta la mecánica cuántica no es completamente entendido y, hasta la fecha, no ha sido posible demostrar completamente la supremacía cuántica.

Actualmente, existen dos ramas de investigación para llegar a la demostración de la supremacía. Por un lado, la demostración se puede llevar a cabo mediante experimentos que muestren que un ordenador cuántico es capaz de resolver algoritmos que los ordenadores clásicos no pueden realizar de forma eficiente. Por otra parte, demostrar que un ordenador clásico no es capaz de realizar un cálculo de forma eficiente[20]. De este modo, queda claro que la demostración de la supremacía cuántica es una tarea importante para la comunidad científica en cuanto a la resolución de problemas complejos, ya que dará resultados a muchos problemas que hoy en día no son abarcables.

La primera rama de investigación está encabezada por los *problemas de muestreo cuántico*, una técnica innovadora que tiene grandes papeletas de demostrar la supremacía del algoritmo cuántico (Lund, A. P., 2017). Existen dos clases de problemas de muestreo cuántico que son sumamente difíciles de resolver por los ordenadores clásicos: el muestreo de bosones y el muestreo instantáneo de tiempo polinómico cuántico.

5.1. Problemas de muestreo para la demostración de la supremacía cuántica

El muestreo de bosones –*BosonSampling*– es un protocolo cuántico propuesto por Aaronson y Arkhipov (Aaronson, S., & Arkhipov, A., 2013) capaz de resolver de forma eficiente problemas de muestreo mediante redes ópticas lineales. En otras palabras, dichos sistemas cuentan con partículas cuánticas de luz –fotones– controladas mediante chips ópticos con el fin de resolver cuánticamente problemas de muestreo. Uno de los objetivos principales es la demostración de la supremacía cuántica. Hasta ahora, esta clase de problemas parecían ser intratables por los ordenadores clásicos cuando los problemas crecían mínimamente[21]. En tal caso, los tiempos de ejecución crecían de forma exponencial a medida que los problemas crecían en tamaño. Pero en el muestreo de bosones, un estado de muchos fotones es capaz de dar lugar a una distribución de

[20] En muchas ocasiones se piensa que los algoritmos clásicos no pueden resolver eficientemente ciertos problemas, pero realmente puede ocurrir que, simplemente, no se haya descubierto ningún algoritmo clásico mejor para dicho problema, lo que no quiere decir que no pueda existir ninguno.
[21] Un ejemplo de problema de tamaño medio suele contar con 50 bosones y 2500 caminos.

probabilidad muestreada eficientemente –a diferencia de lo que ocurre con un algoritmo clásico. Si se consigue verificar que un computador cuántico puede resolver este problema, entonces la supremacía cuántica podría ser demostrada (Carolan, J. et al, 2015).

Sin embargo, actualmente los procesadores ópticos lineales universales (LPU) encuentran problemas con la selección y manipulación arbitraria de los fotones, o qubits. Las futuras investigaciones parecen estar dirigidas hacia la combinación de nuevos componentes con circuitos más grandes de bajas pérdidas (Elshaari, A. W. et al, 2017). Asimismo, algunos enfoques ya proporcionan filtrados y enrutamientos eficientes de los fotones a niveles individuales, permitiendo complejas estructuras y experimentos con varias partículas, lo que abre la puerta a un mayor grado de control del crecimiento de los sistemas cuánticos

El muestreo instantáneo de tiempo polinómico cuántico –*Instantaneous Quantum Polynomial-time*– o IQP por sus siglas en inglés, es otra técnica de muestreo cuántico basada en la computación cuántica no universal mediante puertas de conmutación (Shepherd, D., & Bremner, M. J., 2009). En dicho sistema, las puertas cuánticas pueden utilizarse simultáneamente. La definición formal en (Bremner, M. J., Jozsa, R., & Shepherd, D. J., 2010, p. 463) lo describe como *un circuito con n líneas de qubits donde cada puerta es diagonal en la base X* $\{|0\rangle \pm |1\rangle\}$*, la entrada es un conjunto de qubits inicializados a* $|0\rangle$ *y la salida es una medición final en relación con el conjunto de líneas*.

Debido a la dificultad de simular el muestreo en sistemas clásicos y a los errores multiplicativos –aquel en el que la variable dependiente es un producto de la variable independiente– este sistema también es considerado como una promesa para la demostración de la supremacía cuántica. Además, cualquier arquitectura cuántica puede implementar el muestreo IQP y las técnicas de corrección de ruidos pueden funcionar en cualquier sistema –tema abarcado más adelante. Por todo ello, todos los sistemas que cuentan con puertas lógicas universales elegidas aleatoriamente tienen grandes posibilidades de alcanzar la supremacía cuántica. Los últimos análisis hablan de, aproximadamente, 50 qubits superconductores, lo que equivale a 2^{50} posibles estados (Lund, A. P., 2017).

5.2. Simulación cuántica. Sistemas universales vs sistemas adiabáticos.

En los años 80 Richard Feynman sugirió que las simulaciones de sistemas cuánticos debían ser ejecutadas por ordenadores con un comportamiento cuántico para

hallar resultados realistas. Los primeros esbozos de la computación cuántica contemplaron dos entornos donde simular los sistemas cuánticos: computadores clásicos y máquinas cuánticas. Desde un inicio, se descartó la simulación de sistemas cuánticos en ordenadores clásicos por el simple motivo de que los ordenadores clásicos no son capaces de simular los sistemas cuánticos de forma eficiente debido al problema de variable oculta, lo que supone no poder representar los estados cuánticos completamente. Como se comentó anteriormente, hasta que un qubit no colapsa, este contiene infinita información del sistema. De tal manera, generar las probabilidades y los resultados con la probabilidad cuántica correcta es imposible por un ordenador clásico, así como almacenar todos los valores. De tal modo, dicho fenómeno debía ser imitado y, consecuentemente, no resultaba totalmente fiable (Feynman, R. P., 1982a). Sin embargo, hasta que el famoso problema $P\ versus\ NP$ no sea demostrado, no será posible afirmar que la mecánica cuántica no puede ser simulada clásicamente (Harrow, A. W., & Montanaro, A., 2017).

QtVM es un proyecto que simula sistemas universales cuánticos en ordenadores clásicos. Como es lógico pensar, dicho sistema cuenta con problemas de crecimiento exponencial en memoria a medida que se añaden más qubits en la simulación, ya que, al existir el entrelazamiento, los estados crecen a un ritmo de 2^n, siendo n el número de qubits. Aun así, la plataforma parece ser un buen sistema de pruebas para testear y entender mejor los algoritmos cuánticos, así como para demostrar la supremacía cuántica (Hong, Q. T. et al, 2018).

Las principales investigaciones sobre la simulación cuántica han sido encaminadas hacia la segunda vertiente: simuladores universales cuánticos en computadores cuánticos, cuyas investigaciones comenzaron a finales de los años ochenta (Deutsch, D., 1989). Hoy en día existen dos principales arquitecturas cuánticas: la computación cuántica universal, basada en el control de estados cuánticos mediante el uso puertas lógicas y el entrelazamiento de qubits; y la computación cuántica adiabática, basado en las mismas ideas, pero cuyos qubits no necesitan encontrarse en superposición en todo momento, siempre y cuando existan conjuntos grandes de qubits entrelazados.

5.2.1. Computación cuántica universal

La *computación universal* se implementa como una secuencia de transformaciones matriciales unitarias discretas que forman un circuito cuántico. Este

circuito trabaja con una gran cantidad de niveles de energía y a medida que crece en número de qubits, el sistema se hace más inestable a causa de la *decoherencia*. Este fenómeno no es otra cosa que la corrupción de la información que aparece cuando algunos qubits de la computación se entrelazan con el entorno, colapsando en estados alterados (Shor, P. W., 1995). El desarrollo de estos ordenadores es muy complejo y, hasta ahora, solo existen unos pocos casos funcionales: es el caso de IBM y su *IBM Q 50 prototype* con 50 qubits (Linke, N. M. et al, 2017), Google y su ordenador de 72 qubits (Kelly, J. A, 2018) o Rigetti, con 19 qubits, pero con planes de implementar un ordenador cuántico de 128 qubits para finales de 2019[22]. Por otra parte, IonQ ha optado por desarrollar un sistema basado en las trampas de iones. Hasta la fecha han conseguido implementar un sistema cuántico de 11 qubits para ejecutar algoritmos complejos en puertas lógicas de dos qubits. No obstante, para puertas lógicas simples de un qubit han llegado a mantener en cadena 79 qubits. Su sistema les ha valido para obtener a finales de 2018 el reconocimiento del mejor ordenador cuántico hasta la fecha en términos de potencia y rendimiento[23].

Mientras tanto, Microsoft utiliza la tecnología de qubits topológicos para su propio ordenador cuántico[24]. Los qubits topológicos son qubits protegidos del ruido exterior mediante una duplicidad de sus valores repartidos en dos puntos distintos. Esto se consigue mediante el fraccionamiento electrónico, es decir, la división de un electrón en dos cuya información cuántica es distribuida en ambas partes. De esta forma, si una de las dos mitades se ve afectada por alteraciones, la segunda mitad puede respaldar a la primera. Además, los qubits topológicos han sido diseñados con una protección adicional ante interferencias mediante una técnica llamada *degeneración del estado fundamental*, con la que se consigue que un qubit cuente con dos estados fundamentales discriminados mediante otra técnica que logra diferenciar ambos estados, llamada *trenzado*. Esta diferenciación de estados es sensible a los cambios del ruido ambiental por lo que en cualquier momento es posible saber si los qubits sufren interferencias respecto a su estado inicial.

[22] Rigetti, C. (2018). *The Rigetti 128-qubit chip and what it means for quantum.* Fuente: https://medium.com/rigetti/the-rigetti-128-qubit-chip-and-what-it-means-for-quantum-df757d1b71ea.
[23] Collage Park. (2018). *IonQ harness single-atom qubits to build the wolrd's most powerful quantum computer*. Fuente: https://ionq.com/news/december-11-2018.
[24] Microsoft Quantum Team. (2018). *Developing a topological qubit*. Fuente: https://cloudblogs.microsoft.com/quantum/2018/09/06/developing-a-topological-qubit/

Aunque existan ciertos métodos para reducir la decoherencia, a medida que un ordenador cuántico cuenta con más qubits, el riesgo de una alta decoherencia aumenta. Asimismo, es necesario contar con sistemas cuánticos controlados y suficientemente grandes a fin de que los algoritmos cuánticos obtengan resultados completamente eficientes (Berthiaume, A., Deutsch, D., & Jozsa, R. 1994), (Shor, P. W., 1995). Como se comentó anteriormente, en 2017-2018 se pronosticó que la supremacía podía alcanzarse con los 50 qubits. No obstante, Google, con su ordenador de 72 qubits, aún no ha podido demostrar la supremacía cuántica. Dicho esto, los ingenieros de dicha empresa predicen que esto puede ocurrir a finales de 2019. Según las declaraciones de Hartmut Neven, director del equipo cuántico de Google, es posible que pronto se llegue a un punto en el que el ordenador cuántico sea suficientemente potente para la demostración de la supremacía. Esta afirmación está basada en dos factores:

Por un lado, la superposición de los qubits aporta una ventaja natural exponencial al representar y tratar la información de cada qubit. Un procesador con n qubits cuenta con 2^n estados –puede realizar la misma cantidad de trabajo que 2^n bits clásicos– por lo que es mucho más eficiente que un ordenador clásico. De este modo, un ordenador clásico de 4 bits es equivalente a uno cuántico de 2 qubits, 65.536 bits a 16 qubits, y así sucesivamente.

Por otra parte, la velocidad con la que los chips cuánticos están siendo mejorados crece también a un ritmo exponencial. Esta cifra está en torno a un crecimiento exponencial doble respecto a la tasa de crecimiento del poder computacional de los ordenadores clásicos. Por ello, ya que la Ley de Moore en la computación cuántica parece ser una función doble, la ley de Neven o de Moore para ordenadores cuánticos puede quedar representada como un crecimiento 2^{2^n}. La tabla 3 muestra como hipotéticamente un ordenador cuántico es mucho más eficiente tratando datos en términos de bits clásicos.

Según el director del grupo de computación cuántica de Google, la supremacía será alcanzada a finales de 2019, hablando de meses y no años[25]. No obstante, desde IBM son algo más precavidos y piensan que la supremacía no podrá ser alcanzada hasta dentro de tres años[26]. Para ello, primero se debe hacer frente a algunos de los problemas que generan errores en los sistemas cuánticos: la decoherencia, los errores en los circuitos

[25] Harnett, K. (2019). *A New Law to Describe Quantum Computing's Rise?* Fuente: https://www.quantamagazine.org/does-nevens-law-describe-quantum-computings-rise-20190618/.
[26] Loeffler, J. (2019). *IBM Expects Commercialization of Quantum Computers in 3 to 5 Years.* Fuente: https://interestingengineering.com/ibm-expects-commercialization-of-quantum-computers-in-3-to-5-years.

cuánticos ocurridos por el entrelazamiento de los bits y el ruido cuántico externo procedente de otras fuentes de energía. Es por esto por lo que es necesario un sistema de calibración y corrección de errores de tal modo que la computación cuántica a gran escala sea útil.

Tabla 3. Poder computacional de un ordenador clásico, un ordenador cuántico, y un ordenador cuántico según la ley de Neven

n	2^n	2^{2^n}
1	2	4
2	4	16
3	8	256
4	16	65.536
5	32	4.294.967.296
6	65	18.446.744.073.709.551.616
7	128	3,402823669209384634633746074317e38
8	256	1,157920892373161954235709850086 9e77
9	512	1,340780792994259709957402499820 6e154
10	1.024	1,797693134862315907729305190789e308

Hasta la fecha, los métodos no eran lo suficientemente eficientes y no podían lidiar con los errores en sistemas con más de 10 qubits (Harper, R., Flammia, S. T., & Wallman, J. J., 2019a). Algunas investigaciones han sido enfocadas hacia la calibración óptima de puertas lógicas de dos qubits (Patterson, A. D. et al., 2019). Pero muchos de estos métodos no han sido totalmente útiles para los sistemas a gran escala (múltiples qubits) hasta que Harper, R. y sus compañeros (2019a) expusieron un protocolo que demuestra la funcionalidad un sistema de corrección de errores preciso para un sistema con hasta 100 qubits en línea. Este tamaño es muy superior al de cualquier otro método actual (> 10 qubits) y puede llegar a suponer el primer eslabón en la creación de controles cuánticos para obtener puertas lógicas óptimas, circuitos con reconocimiento de ruido o la generación de mapas de ruido para la corrección de errores.

Actualmente el error que las puertas cuánticas suelen generar está entorno al 0,5-1% (Auger, J. M., 2018). Esto supone que, de media, se produce un error por cada 200 operaciones cuánticas, lo que supone cinco veces más fallos que los que tiene un circuito

clásico[27]. Para obtener cálculos cuánticos fiables es necesario reducir las tasas de error hasta el 0,1%. Sin embargo, esta reducción de la tasa de errores puede llevar un tiempo. Harper y Flammia (2019b) muestran en su artículo sobre las puertas lógicas en el sistema de *IBM Quantum Experience* que, incluso utilizando un mapa de ruido de alta fidelidad, el ruido correlacionado puede afectar negativamente a los resultados de la computación. De esta forma, los autores advierten de que todo sistema de antierrores eficiente debe definirse de acuerdo con tres características. Primero, el sistema debe tratarse de un espacio de código en el que pueden aplicarse puertas cuánticas lógicas sobre la información cuántica codificada. En segundo lugar, las puertas lógicas deben tener cierta tolerancia a fallos. Por último, el sistema es capaz de detectar si se ha producido un cierto número de errores.

Como se ha visto, la supremacía cuántica está muy relacionada con la corrección de errores. Mientras que estos errores no sean controlados, será imposible demostrar que los ordenadores cuánticos son superiores a los ordenadores clásicos. Existen autores más pesimistas que piensan que la corrección de errores no es posible de controlar y, por tanto, creen que la supremacía cuántica nunca se dará (Kalai, G., 2019). Por el contrario, muchas investigaciones (comentadas previamente) avanzan, más o menos rápido, hacia sistemas estables. El resultado queda como una pregunta abierta a futuras investigaciones y mejor entendimiento de la mecánica cuántica.

5.2.2. Sistemas adiabáticos

Como una alternativa a la computación universal aparece la computación *cuántica adiabática*, en la que los qubits del sistema no se encuentran en perfecta superposición, sino que ciertos qubits se entrelazan de forma aleatoria. El término *adiabático* está estrechamente unido al término cuántico *cuasiestático*, relacionado con la velocidad a la que un proceso cuántico se lleva a cabo, el cual se realiza suficientemente lento a fin de que prevalezca un equilibrio en el sistema. De tal forma, este modelo de computación cuántica basa sus cálculos en un *hamiltoniano* inicial cuyo estado fundamental es fácil de preparar con el objetivo de alcanzar un hamiltoniano final cuyo estado fundamental

[27] Hartnett, K. (2019). *Quantum Supremacy Is Coming: Here's What You Should Know*. Fuente: https://www.quantamagazine.org/quantum-supremacy-is-coming-heres-what-you-should-know-20190718/

codifica la solución al problema computacional (Albash, T., & Lidar, D. A., 2018). El hamiltoniano tiene la forma:

$$H_v = -\frac{v^2}{2}\frac{\partial^2}{\partial x^2} + V(x)$$

donde H_v representa la función de costes a minimizar (Albeverio, S., 1977).

Dicho de otra forma, los sistemas adiabáticos aprovechan la evolución natural de los estados –los cuales pueden evolucionar continuamente en el tiempo– partiendo desde un estado deslocalizado hasta llegar a una configuración del sistema final, la cual será el objetivo buscado. Este sistema, a diferencia de un ordenador cuántico universal –el cual es inestable y puede estar expuesto a ruidos–, realiza dicha transición con una velocidad suficientemente lenta para llegar a una solución óptima, lo que garantiza una robustez inherente frente a errores. Esta ventaja asegura que los algoritmos cuánticos no presentan errores relacionados con el ruido (Childs, A. M., Farhi, E., & Preskill, J., 2001). Adicionalmente, los sistemas adiabáticos permiten controlar mediante la simulación a los hamiltonianos. Si la variación de estos es progresivamente lenta, el sistema es capaz de rastrear el estado fundamental instantáneo, lo que incrementa la calidad de las soluciones de los algoritmos (Albash, T., & Lidar, D. A., 2018). Esto se relaciona estrechamente con la complejidad computacional y los algoritmos heurísticos. De tal manera, una de las primeras aplicaciones directas que se da a los sistemas adiabáticos es la resolución de problemas clásicos de optimización combinatoria y satisfacibilidad, debido a su gran poder computacional en tiempo polinómico, con resultados parecidos a los de la computación cuántica universal (Aharonov, D. et al., 2008).

Muchos problemas computacionales son fácilmente transformables en problemas de asignación variable donde el objetivo es el de buscar una función de energía que minimice dicho problema. De esta forma el problema es fácilmente abordable como ocurre, por ejemplo, con los problemas 3-SAT (Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M., 2000). A este método de optimización se lo conoce como *temple cuántico* (Apolloni et al., 1988).

El temple cuántico –en inglés, *quantum annealing*– es un algoritmo de búsqueda heurística aplicada a problemas de optimización con múltiples variables independientes. El objetivo del temple cuántico es encontrar el máximo o mínimo de una función de costes, el cual está basado en el temple simulado que, análogamente, se trata de un algoritmo de optimización de búsquedas basado en sistemas termodinámicos

(Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi, 1983). La gran diferencia reside en que el sistema cuántico realiza *tunneling* en las barreras energéticas, al contrario del temple simulado, el cual solo permite un salto probabilístico sobre dichas barreras (Albash, T., & Lidar, D. A., 2018)

Dicho esto, recientemente se han realizado experimentos donde se ha demostrado que la ejecución de algoritmos cuánticos, frente a los clásicos, aportaba un incremento considerable de velocidad en la resolución. Estos incrementos, como es lógico pensar, varían de acuerdo con el tipo de problema. Es por esto por lo que existe una clasificación de algoritmos cuánticos de acorde con el incremento de velocidad que la computación cuántica aporta. Dicho valor se obtiene de la ratio entre el tiempo empleado en resolver un problema clásico de tamaño N –denotado como $C(N)$– y el tiempo empleado en una máquina cuántica – $Q(N)$ (Rønnow, T. F., 2014):

$$S(N) = \frac{C(N)}{Q(N)}$$

donde $S(N)$ es el incremento de velocidad. Existen cinco tipos de incrementos de la velocidad:

Tabla 4. Clasificación de algoritmos por su velocidad

Velocidad cuántica demostrable	Existen pruebas que demuestran que los algoritmos clásicos no pueden mejorar los resultados de un algoritmo cuántico. El mejor ejemplo de este tipo es el algoritmo de búsqueda de Grover (Grover, L. K., 1997) el cual ofrece un incremento de velocidad cuadrático demostrable (Bennett et al., 1997).
Velocidad cuántica fuerte	Los algoritmos cuánticos mejoran la velocidad del mejor de los algoritmos clásicos, aunque este algoritmo puede estar, o no, descubierto hasta la fecha. Desafortunadamente, el rendimiento del mejor algoritmo clásico posible es desconocido para muchos problemas interesantes. Este es el caso de la factorización, donde todos los algoritmos clásicos tienen tiempos súper-polinomiales a diferencia del algoritmo de Shor, el cual conlleva un tiempo polinomial. No obstante, aún no se sabe si algún algoritmo clásico puede reducir los tiempos exponenciales
Velocidad cuántica	Los algoritmos cuánticos mejoran la velocidad del mejor de los algoritmos clásicos descubiertos.

Velocidad cuántica potencial	Los algoritmos cuánticos mejoran la velocidad de un conjunto de algoritmos clásicos a falta de consenso en cuál de estos es el mejor algoritmo clásico.
Velocidad cuántica limitada	Los algoritmos cuánticos están diseñados para hacer uso de los efectos cuánticos, pero no se sabe si estos efectos proporcionan una ventaja sobre los algoritmos clásicos.

Fuente: Rønnow, T. F., 2014).

5.3. D-Wave como alternativa a la computación cuántica universal

D-Wave es una compañía de origen canadiense especializada en computación cuántica, pero a diferencia de sus más próximos competidores – Google, IBM y Rigetti– D-wave opta por una estrategia distinta: la computación adiabática. Mientras que las grandes empresas están destinando sus recursos al modelo de puertas lógicas, computación cuántica universal, D-wave ha decidido ir por una vía nueva de investigación llamada temple cuántico, algo más parecido a un simulador cuántico que a un ordenador cuántico propiamente.

Una de las grandes diferencias respecto con un ordenador cuántico universal, en el cual todos los bits deben estar perfectamente entrelazados entre sí, es que los qubits forman subconjuntos conectados entre sí, pero sin existir un entrelazamiento total. Además, dichos computadores cuentan con sistemas con tiempos de coherencia menores que los ordenadores universales, por lo que existe menor probabilidad de inducir a ruido y errores. Para ello, la QPU –*quantum processing unit*– debe trabajar a temperaturas cercanas al cero absoluto, inferiores a 15 milikelvin, lo que significa cerca de 180 veces más frío que el espacio interestelar. Asimismo, debe existir una protección especial contra el ruido externo, como son las interferencias electromagnéticas.

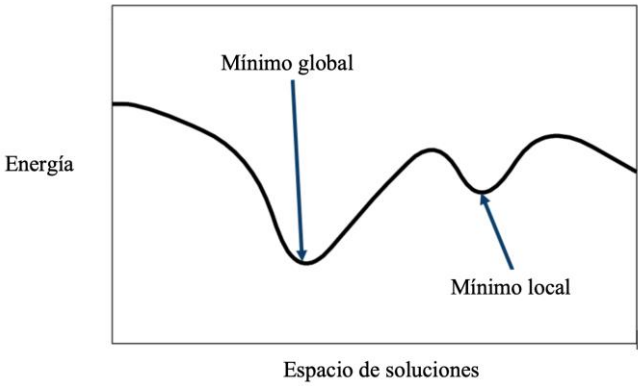
De este modo, los ordenadores adiabáticos de D-wave tienen unos objetivos muy claros y específicos: resolver problemas de optimización, muestreo y aprendizaje automático[28]. Los problemas de optimización consisten en hallar el mínimo coste para un problema que está limitado por una serie de restricciones impuestas. Estos problemas, cuando cuentan únicamente con unas pocas variables, son fácilmente resolubles por

[28] D-Wave. (2018). D-Wave 2000Q Technology Overview. Fuente: https://www.dwavesys.com/sites/default/files/D-Wave%202000Q%20Tech%20Collateral_1029F.pdf

ordenadores clásicos, pero a medida que el número de opciones aumenta, la dificultad crece exponencialmente. Con 270 opciones existen más combinaciones que átomos en el universo.

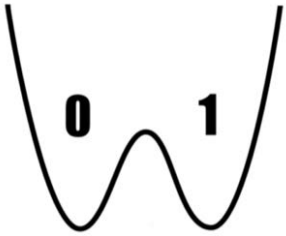
De acuerdo con el tipo de problemas qué estos computadores cuánticos resuelven y la forma en la que son abordados, todos los problemas deben ser transformados en una función de energía. De tal modo, todos los valores se encuentran repartidos a lo largo de la función en la que los puntos máximos corresponden a aquellas soluciones menos probables que violan la mayoría de los criterios, mientras que las cotas inferiores corresponden a soluciones en las que se satisfacen la mayoría o la totalidad de las restricciones. El objetivo final es encontrar el punto donde se minimice completamente la función y, por tanto, el consumo de energía. Esto se debe a que la física tiende a buscar el estado más eficiente en cuanto gasto de energía. Esto corresponde al algoritmo de temple cuántico que es el encargado de encontrar aquellos estados de baja energía (ver ilustración 10).

Ilustración 10. Función de energía



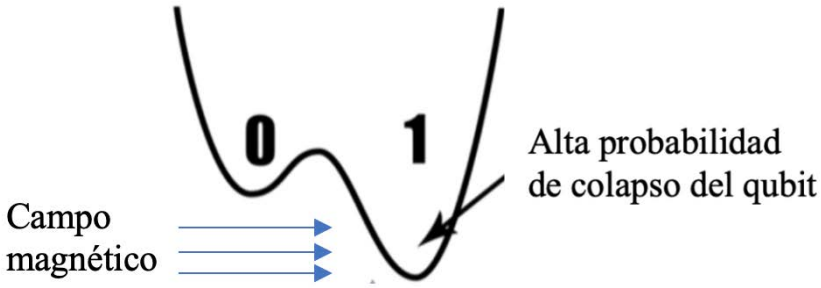
La función de energía tiene una estrecha relación con los conceptos de computación cuántica, pues cada punto de la función corresponde con un estado de un qubit. La ilustración 11 representa una función de energía conocida como *potencial del doble pozo* donde el punto bajo del valle izquierdo corresponde al estado 0, y el punto bajo del valle derecho corresponde al estado 1. El qubit contempla ambos estados, los cuales están entrelazados, y no será hasta el final del algoritmo de temple cuántico cuando este colapsa el estado del qubit a uno de los dos, concretamente a aquel con el menor gasto de energía.

Ilustración 11. Estados de un qubit representados en función de energía



La ilustración 11 muestra un caso en el que la probabilidad de colapso es la misma para ambos estados, es decir, cada estado (0 o 1) tienen un 50% de posibilidades. Dicho esto, el temple cuántico permite aplicar campos magnéticos externos para modificar dichas probabilidades (ilustración 12). Dicho esto, el campo magnético aumenta la probabilidad de que el qubit colapse hacia el mínimo global, es decir, el punto óptimo.

Ilustración 12. Aplicación de campo magnético en el qubit y colapso de este al mínimo inferior.

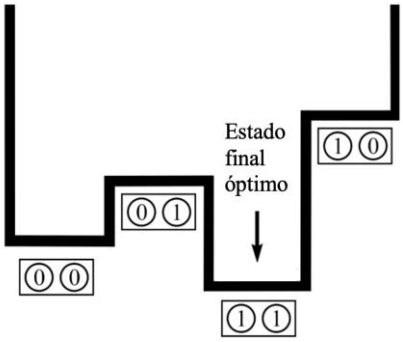


Las ilustraciones previas (11 y 12) estaban argumentadas con un solo qubit. No obstante, generalmente estos sistemas cuentan con muchos qubits que trabajan de forma conjunta, es decir, entrelazados. Los qubits se encuentran entrelazados por unos artefactos conocidos como acopladores[29]. Los qubits entrelazados funcionan como un todo con muchos estados interrelacionados, de tal forma que todos son dependientes de todos. De esta forma, un sistema que tuviese dos qubits estaría comprendido por cuatro estados –(0,0), (0,1), (1,1) y (1,0)– donde la energía de cada estado depende y repercute en el resto. Cuando el temple cuántico es aplicado, el estado final del qubit colapsará a aquel punto

[29] Los acopladores –*couplers*- son los artefactos que se encargan de enlazar a los qubits de tal forma que los qubits puedan influenciarse entre ellos. Los acopladores pueden hacer que los qubits enlazados tiendan al mismo valor o a valores opuestos.

donde exista un mínimo global, como ocurre en la ilustración 13, que muestra el estado final tras el colpaso: (1,1).

Ilustración 13. Función de energía con resultado óptimo



Existe una técnica muy extendida para hallar los mínimos de una función objetivo conocida como QUBO –*quadratic unconstrained binary optimization*. QUBO es un problema de satisfacción con dos posibles resultados: VERDADERO y FALSO, que se corresponden con 1 y 0. El objetivo de este problema es minimizar la función objetivo:

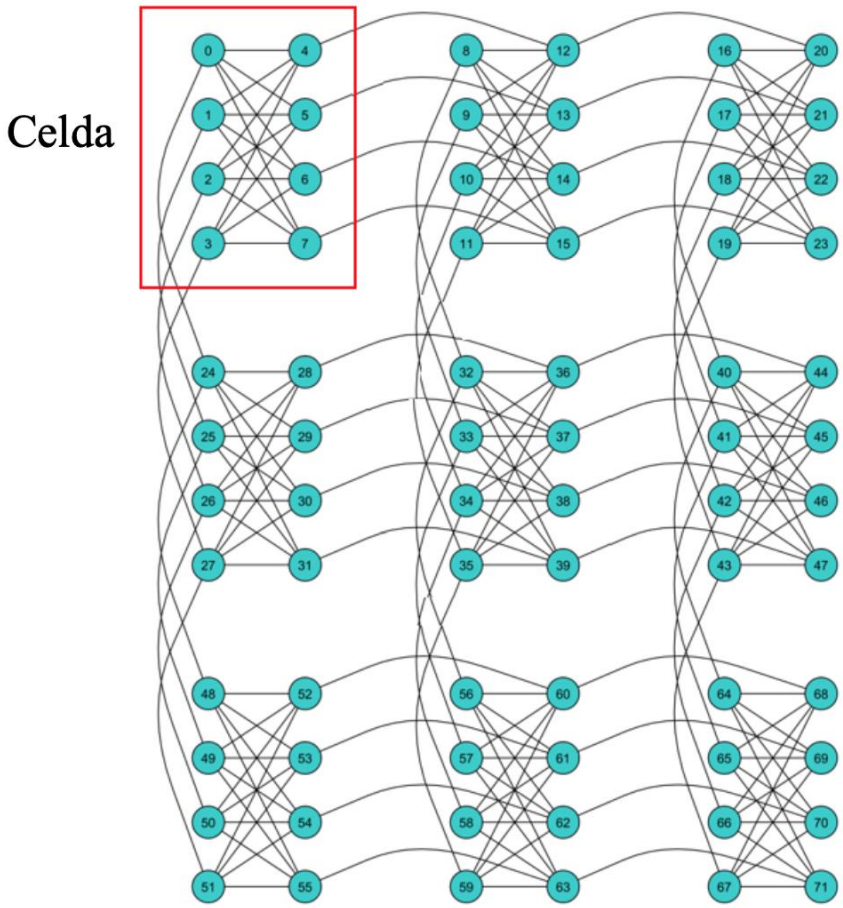
$$\min x \in \{0,1\}^n \, x^T Q x$$

donde *x* es un vector de variables binarias y Q es una matriz de tamaño *N x N* superior triangular de pesos reales.

Como fue comentado anteriormente, los sistemas cuánticos de D-Wave no solo cuentan con uno o dos qubits, sino con una gran cantidad de ellos entrelazados. No obstante, debido a que se trata de un sistema adiabático, los qubits no se encuentran entrelazados perfectamente entre todos. Lo que realmente se da es una agrupación de subconjuntos de bits entrelazados los unos con los otros mediante los acopladores. La forma en la que se conectan los qubits en la arquitectura de D-wave se conoce como *Chimera* (ilustración 14).

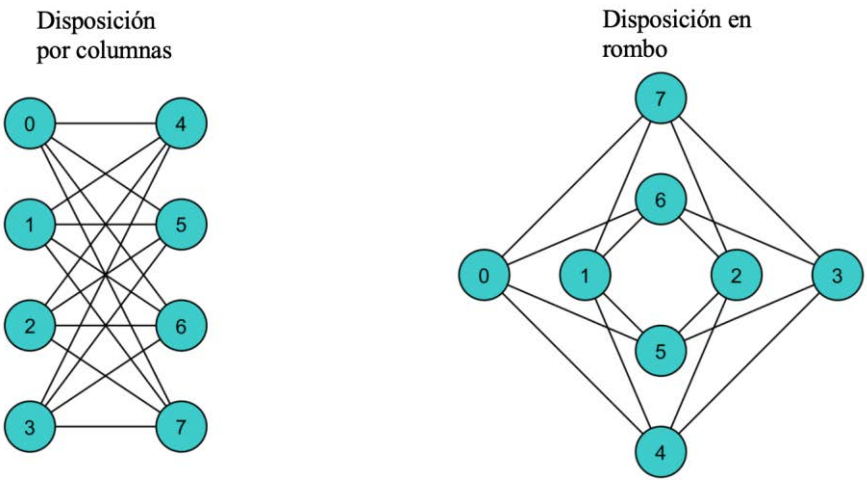
La red de qubits está compuesta principalmente por celdas de qubits conectados entre sí mediante acopladores definiéndose como una matriz de N x N celdas, cuya denotación se define CN. En el caso de la ilustración 9, el grado es C3 compuesto de 3 x 3 celdas. Así mismo las celdas tiene dos formas de componerse: en formato columna o en formato rombo (ilustración 15).

Ilustración 14. Arquitectura quimera



Fuente: D-Wave. https://docs.dwavesys.com/docs/latest/c_gs_4.html

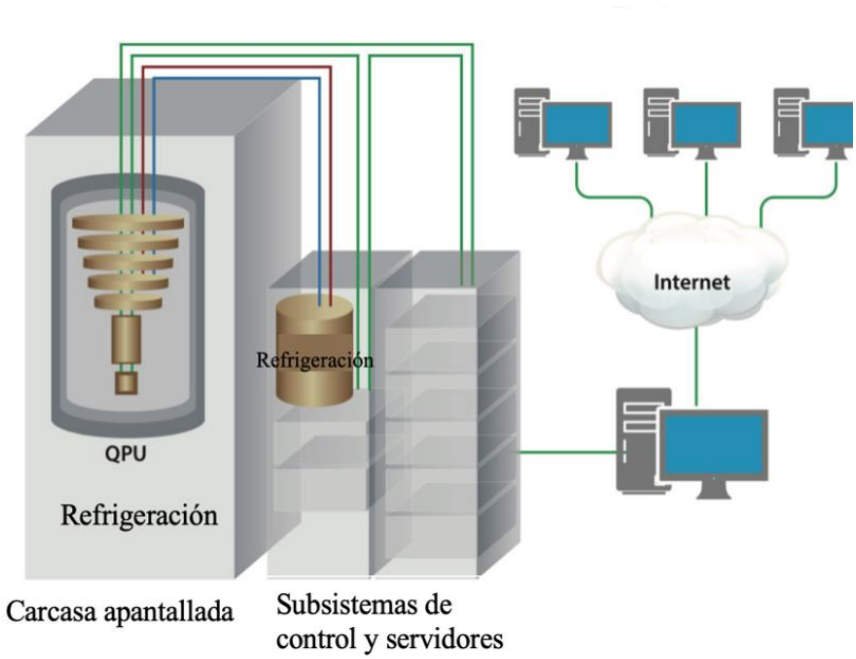
Ilustración 15. Conectividad de las celdas



Fuente: D-Wave. https://docs.dwavesys.com/docs/latest/c_gs_4.html

El ordenador más actual de D-wave, llamado *D-Wave 2000Q* (ilustración 16), cuenta con una tipología[30] de 6 qubits, 2048 qubits y 6016 acopladores. A principios de 2019 D-Wave anunció para mediados de 2019 un nuevo computador adiabático de 5.000 qubits basado en una tipología llamada *Pegasus*, con una tipología de 15 qubits[31]. Este ordenador cuenta con una refrigeración especial para mantener unas temperaturas extremadamente bajas, así como un blindaje denso para evitar interferencias externas. El procesador cuántico se conecta a su vez con unos sistemas de control que se aseguran de mantener las condiciones perfectas, junto a los servidores del sistema que realizan la conexión con el exterior o con centros de datos comunes. Además, usuarios comunes tienen acceso a través de la nube y de la plataforma online de D-Wave.

Ilustración 16. Entorno del ordenador 2000Q



Fuente: D-Wave, https://www.dwavesys.com/sites/default/files/D-Wave%202000Q%20Tech%20Collateral_0117F.pdf.

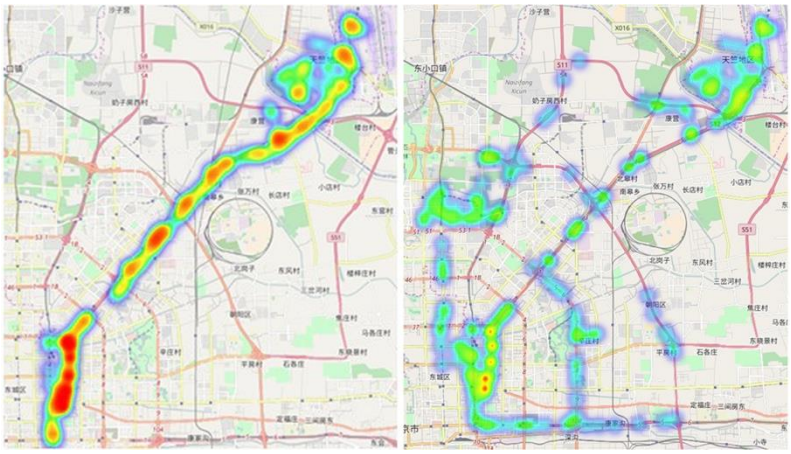
A pesar del avance que se está dando la tecnología cuántica de D-Wave no ha demostrado todavía ser más eficiente y rápida que los algoritmos clásicos. Algunos estudios demuestran que los tiempos de ejecución de los algoritmos de D-wave crecen de

[30] La tipología es una característica que representa la cantidad de qubits con los que un qubit está entrelazado.
[31] D-wave. (2019). D-Wave Previews Next-Generation Quantum Computing Platform. Fuente: https://www.dwavesys.com/press-releases/d-wave-previews-next-generation-quantum-computing-platform

forma exponencial a diferencia de un algoritmo clásico como el peso mínimo con perfecto emparejamiento – *minimum-weight perfect-matching*– el cual escala a ritmo polinómico con la introducción de nuevas variables. Si para sistemas pequeños el chip D-Wave 2000Q es rápido, no ocurre lo mismo cuando el número de variables lógicas en el chip se duplica, pues un algoritmo clásico como el de peso mínimo puede llegar a ser tres veces más rápido (Mandra, S., Katzgraber, H. G., & Thomas, C., 2017).

En los últimos dos años han aparecido algunos casos prácticos donde la tecnología de D-wave se está utilizando para resolver problemas de optimización reales. Volkswagen es una de las empresas que ha confiado en esta tecnología para solucionar un problema que le afecta directamente: la predicción y optimización del tráfico (Neukart, F. et al., 2017). El estudio recogió datos de muchos taxis de la ciudad de Pekín y dividió la ciudad en segmentos. Mediante el análisis del comportamiento de los taxis fueron capaces de optimizar el flujo de tráfico. De los 50 experimentos que se realizaron, el sistema descongestionó el tráfico de todos ellos (ver ejemplo en ilustración 17).

Ilustración 17. Ejemplo de un área congestionada (izquierda) y un área descongestionada después de aplicar la optimización (derecha)



Fuente: D-Wave. https://www.dwavesys.com/sites/default/files/VW.pdf

Es por esto por lo que los resultados aportados por el procesador de D-Wave fueron aceptables y adecuados para este tipo de problemas de optimización. Sin embargo, el estudio fue realizado en un entorno muy controlado y con un número de variables muy limitado: el experimento fue realizado con un número limitado de coches, sin comunicación con la infraestructura, sin otros participantes en el tráfico, y sin otros objetivos de optimización, excepto la minimización de la congestión de las carreteras. En futuras investigaciones se desea añadir

dichas variables con esperanzas de que el nuevo computador con la tipología *Pegasus* y 5000 qubits pueda resolver problemas mucho más complejos.

6. Ingeniería del software en la computación cuántica

La ingeniería del software "*es la aplicación sistemática de conocimientos, métodos y experiencias científicas y tecnológicas al diseño, implementación, pruebas y documentación de software*" (ISO, I., 2010).

La computación cuántica es una de las ramas de la tecnología de la computación que ha comenzado a desarrollarse en la práctica más recientemente. El primer ordenador cuántico en ver la luz fue desarrollado por IBM en el 2000. Desde entonces, numerosas empresas se han sumado al desarrollo de computadores y sistemas cuánticos. Es ahora cuando estos dispositivos comienzan a estar disponibles para el uso y desarrollo de software para ingenieros y desarrolladores. IBM es un ejemplo de ello.

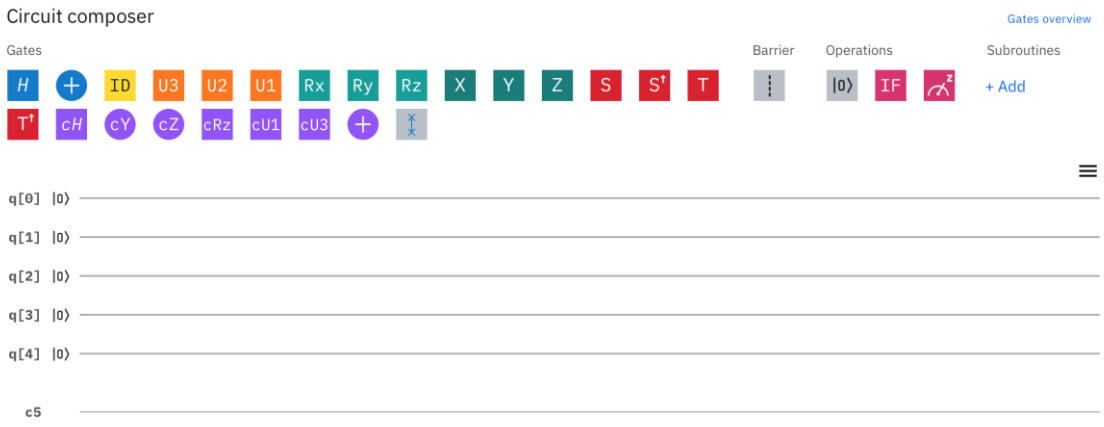
6.1. Simuladores y compiladores cuánticos

IBM cuenta con la plataforma online *Q Experience*, desde donde el público general tiene acceso a una gran cantidad de recursos, así como al ordenador cuántico de IBM a través de la red. Este servicio permite la ejecución de algoritmos cuánticos en las máquinas, la realización de experimentos y el análisis de los resultados, todo ello desde un entorno visual simple e intuitivo. El compositor de circuitos es la herramienta principal que ofrece realizar simulaciones en un ordenador cuántico de cinco qubits, así como el uso de una gran variedad de puertas lógicas (ilustración 18).

Generalmente, los simuladores de circuitos cuánticos suelen ser muy flexibles a la hora de restringir el número de puertas cuánticas que pueden utilizarse, lo que permite recrear una gran cantidad de complejos circuitos, sobre todo, gracias al uso de puertas lógicas de dos qubits. No obstante, todos los computadores cuánticos reales cuentan con una característica muy restrictiva: la topología cuántica. Por tanto, cuando se desea ejecutar algún algoritmo en un ordenador cuántico, es necesario adaptar el circuito a las

características del equipo donde será ejecutado. Esto es algo que solo unos pocos simuladores permiten hacer (Fingerhuth, M., Babej, T., & Wittek, P., 2018).

Ilustración 18. Compositor de circuitos de la plataforma IBM Q Experience



Fuente: IBM Q Experience

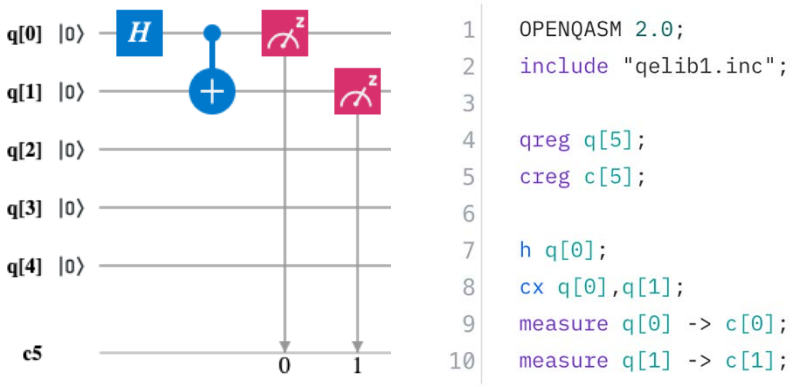
Uno de los programas cuánticos más sencillos que se pueden programar en *Q Experience* es el entrelazamiento de dos qubits –ilustración 19–. La ilustración 18 muestra el entorno de composición de circuitos. Se dispone de cinco líneas horizontales –q[0], q[1], q[2]…– donde cada una se corresponde con un qubit independiente. Cada qubit está inicializado a $|0\rangle$.

El primer paso del algoritmo es el diseño del circuito. Para ello, sencillamente se arrastran las puertas lógicas –las cajas de colores de la ilustración 18– a cada una de las líneas deseadas. Inicialmente, el primer paso es añadir una puerta Hadamard en el q[0]. De esta forma, el estado de q[0] pasará de $|0\rangle$ a $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Como q[1] todavía no ha sido modificado, su estado sigue siendo $|0\rangle$, por lo cual, el estado combinado actual es $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$ obtenido del producto tensorial $\frac{|0\rangle+|1\rangle}{\sqrt{2}} \otimes |0\rangle$.

El siguiente paso es aplicar una puerta *controlled-NOT* (CX) desde q[0] a q[1]. Como breve recordatorio, la puerta CX invierte los valores del qubit objetivo–es decir, 1 se convierte en 0 y viceversa– siempre y cuando el qubit de control –el primer bit en este caso q[0]– sea 1. De tal modo, el resultado global sin aplicar la puerta CX contiene en la segunda parte un bit de control 1 seguido por un 0: $\frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$. Tras aplicar la puerta CX $|10\rangle$ pasará a ser $|11\rangle$, resultando en el estado global $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, que es

el estado de dos qubits entrelazados. Este circuito queda reflejado en la ilustración 19, donde las dos cajas rosas son las puertas de *medición-Z*,

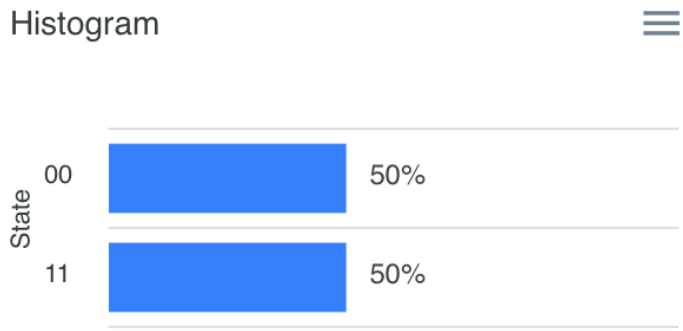
Ilustración 19. Circuito cuántico para entrelazar dos qubits (visual y código)



Fuente: IBM Q Experience

Donde *qreg* es el número de qubits, *creg* es el registro de resultados, *h* es la puerta Hadamard y *cx* es la puerta controlled-NOT. Tras ejecutar el programa, la medición (*measure*) muestra que existen un 50% de posibilidades de que el estado resulte en $|00\rangle$ o $|11\rangle$, observable en la ilustración 20.

Ilustración 20. Probabilidades de medición del entrelazamiento de dos qubits



Fuente: IBM Q Experience

A pesar de que muchos proyectos sean de código cerrado, cada vez son más las versiones que aparecen de código abierto. Este es el caso del entorno *ScaffCC*[32], un

[32] Fuente: https://scaffcc.llvm.org.cn

compilador y planificador que utiliza un propio lenguaje cuántico: *Scaffold*. Con ello, los usuarios pueden programar y analizar tanto circuitos como programas cuánticos. Otros compiladores están basados en lenguajes clásicos de programación. *PyZX* es un compilador que implementa *cálculos ZX*[33] mediante *Python* para la creación, visualización y optimización de grandes circuitos cuánticos.

Además, existe una gran cantidad de *frameworks* de desarrollo de código abierto que proporcionan un enfoque cuántico completo más allá que los compiladores o simuladores cuánticos. Uno de ellos, creado por IBM, es la librería *full-stack Qiskit*[34]. Esta librería permite crear, ejecutar y analizar algoritmos cuánticos de una forma similar al sistema de puertas de *Q Experience*. Este proyecto está dividido a su vez en subproyectos. *Forest* y *Terra* son dos librerías que permiten al usuario definir, compilar y simular circuitos cuánticos. Por otra parte, *Aqua* es una colección de algoritmos cuánticos implementados con Terra. Posteriormente aparece *Aer*, una librería que proporciona un marco de trabajo de simulador de alto rendimiento para Qiskit. Adicionalmente, cuenta con una serie de herramientas para la reducción de ruido y tratamiento de errores durante la simulación.

Google también cuenta con su propia plataforma de desarrollo *Cirq*[35], una librería en Python para escribir, manipular y optimizar circuitos cuánticos tanto en ordenadores como simuladores cuánticos.

Por último, existen ciertos compiladores para sistemas de temple cuántico como el de D-Wave. *Qbsolv*[36] es un solucionador de problemas que permite tanto crear como analizar los resultados para problemas de tipo QUBO. Este entorno cuenta con una interfaz en Python y otra de línea de comandos. Asimismo, existen varias APIs como *dwave-system*[37] y *Dimod*[38] permiten resolver problemas QUBO y BQP –*binary quadratic model*. Terceras compañías también cuentan con herramientas para la ejecución de algoritmos de tipo QUBO. Otro ejemplo de tecnología relacionada con los problemas QUBO y el temple cuántico de D-Wave es D-WIG[39] –*D-Wave Instance Generator*–, que permite crear programas cuánticos a partir de clases de problemas parametrizados.

[33] Los cálculos ZX son un tipo de diagramas tensoriales generados a partir de combinaciones de mapas lineales para la representación de puertas cuánticas lógicas y entendimiento de sus estructuras.
[34] Fuente: https://qiskit.org
[35] Fuente: https://github.com/quantumlib/Cirq
[36] Fuente: https://docs.ocean.dwavesys.com/projects/qbsolv/en/latest/#
[37] Fuente: https://docs.ocean.dwavesys.com/projects/system/en/latest/intro.html
[38] Fuente: https://docs.ocean.dwavesys.com/projects/dimod/en/latest/
[39] Fuente: https://github.com/lanl-ansi/dwig

6.2. Lenguajes de programación cuántica

Hoy en día es posible la programación de circuitos cuánticos con lenguajes de programación tradicionales como *Python* o *JavaScript*. No obstante, está comenzado a desarrollarse nuevos lenguajes intermedios diseñados específicamente para el desarrollo de programas cuánticos. Uno de los lenjuages más usados actualmente es OpenQASM –*Open Quantum Assembly Language*–, el cual es básicamente un lenguaje ensamblador de instrucciones para experimentar con circuitos cuánticos (Cross, A. W. et al., 2017). El entorno permite desarrollar los circuitos mediante diferentes herramientas como el compositor, una herramienta de software de visual, o manuscritamente.

Existen otros lenguajes de más alto nivel como *quC*, *CQL*, *Q* y *Q#* (Prieto Idarraga, B. N. 2018). Uno de los primeros lenguajes en aparecer fue QCL –*Quantum Computation Language*–[40]. Se trata de un lenguaje de alto nivel muy emparentado con *C*, puesto que la sintaxis y las estructuras están basadas en los mismos principios[41]. A modo de ejemplo se detalla el algoritmo de Grover en QCL (Mutiara, A. B., & Refianti, R., 2010). Este algoritmo reduce el mínimo número de operaciones de búsqueda de un objeto concreto en una base de datos de N elementos. Mientras que un algoritmo clásico necesita $\frac{N}{2}$ comparaciones de media, el algoritmo de Grover consigue encontrar un elemento específico en $O(\sqrt{N})$ pasos. Los procesos para ejecutar el algoritmo de Grover en QCL se detallan a continuación (ilustración 22).

La primera tarea del algoritmo consiste en conocer el valor que se desea buscar en la lista. Por ello, el programa pide al usuario introducir el número entero a buscar, el cual será almacenado en la variable `findvalue1`. A continuación, se inicializan las dos variables más importantes del programa. El número de qubits necesarios para la resolución del algoritmo –`sumqubit` –, así como el número de iteraciones que se llevará a cabo –`iteracion`. Para calcular dichos valores se utilizan las siguientes fórmulas:

$$sumqubit = \lfloor (\log_2 \text{findvalue1}) + 1 \rfloor$$

$$iteracion = \left\lfloor \frac{\pi}{8} \cdot \sqrt{2^{sumqubit}} \right\rfloor$$

Adicionalmente, se inicializan los registros para los qubits –`qureg q[sumqubit]`, así como aquellas variables para almacenar los resultados –`resultadoMedicion`.

[40] Fuente: http://tph.tuwien.ac.at/~oemer/qcl.html
[41] Mirar estructuras y tipos de datos. Fuente: http://tph.tuwien.ac.at/~oemer/doc/quprog/node12.html

Ilustración 21. Algoritmo de Grover en QCL (divido en dos pasos).

Paso 1: entrada e inicialización

```
input "Introduzca el valor enetero que desee buscar:", findvalue1;

procedure grover(int n) {
   int sumqubit = floor(log(findvalue1,2)) + 1;  //número de qubits
   int iteracion = ceil (pi/8*sqrt 2^sumqubit)); //numero de iters.

   int resultadoMedicion;
   int i;

   qureg q[sumqubit];
   qureg f[1];

   print "Número de qubits usados:",sumqubit;
   print "Número de iteraciones necesarias:", iteracion;
```

El paso 2 consiste en el bucle principal del programa, donde se lleva a cabo la lógica del algoritmo. En este punto, el primer paso es la inicialización de todos los qubits a $|0\rangle$ –mediante *reset*– y aplicar la puerta Hadamard –*H(q)*– a todos los qubits registrados para obtener una correcta superposición de los estados. A continuación, se realiza el bucle tantas veces como iteraciones se necesiten:

- La función *query* rota el estado en una fase de π radianes, y voltea la función *f* si el qubit analizado coincide con el valor buscado *findvalue1*.
- La función *CPhase* aplica un proceso de fase entre π y la función *f*.
- Se deshace la función *query*.
- Se aplica la función *diffuse*. Esta función amplifica aquel estado que contiene un escalar negativo, convirtiéndolo en un escalar positivo mayor que el anterior escalar. De esta forma, la probabilidad de dicho estado aumenta. En este proceso se aplica una puerta Hadamard sobre el qubit, luego lo invierte, aplica de nuevo el proceso de fase y después deshace, por una parte, la inversión y, por otra, la trasformada de Hadamard.
- Pasadas unas cuantas iteraciones, en último lugar, se lleva a cabo la medición de los qubit. El resultado será aquel estado con una probabilidad superior a ½, pues anteriormente, tras la aplicación de la transformada de Hadamard y con el entrelazamiento, el estado tenía un 50% de posibilidades de colapsar tanto

a $|00\rangle$ como a $|11\rangle$: $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. El resultado se comprueba para confirmar que coincide con el esperado. En caso contrario se repite el proceso.

Paso 2: bucle principal

```
  {
    reset;
    H(q);

    for i= 1 to iteracion {          // Bucle principal
      query(q,f,findvalue1);         // calcular C(q)
      CPhase(pi,f);
      !query(q,f,findvalue1);        // Deshacer C(q)
      diffuse(q);                    // Operador de difusión
    }

    measure q, resultadoMedicion;    // Medición
    if resultadoMedicion == findvalue1{
      print "Medido!", resultadoMedicion;
    }
  } until resultadoMedicion == findvalue1;
}
reset;
```

La función *diffuse* se muestra a continuación:

```
operator diffuse(qureg q) {
  H(q);                // Transformada de Hadamard
  Not(q);              // Invertir q
  CPhase(pi,q);        // Rotar si q == findvalue1
  !Not(q);             // deshacer inversión
  !H(q);               // deshacer la transformada de Hadamard
}
```

Existe otro tipo de lenguajes basados en notación matricial de operaciones, lo que permite una mayor precisión y cohesión con la computación cuántica ya que los principios matemáticos de la cuántica son matrices y vectores. Así uno de estos lenguajes es *QPMC* (Anticoli, L., Piazza, C., Taglialegne, L., & Zuliani, P., 2017). A continuación, se muestra un mero ejemplo de QPMC con el algoritmo de Deutsch (ilustración 23) para poder visualizar su sintaxis, pero no será detallado.

Ilustración 22. Algoritmo de Deutsch en QPMC.

```
Qmc

const matrix A1 = [(1 / 2),(1 / 2),(1 / 2),(1 / 2);(1 / 2),(-1 /
2),(1 / 2),(-1 / 2);(1 / 2),(1 / 2),(-1 / 2),(-1 / 2);(1 / 2),(-1 /
2),(-1 / 2),(1 / 2)];

const matrix A2 = [1,0,0,0;0,1,0,0;0,0,0,1;0,0,1,0];

const matrix A3 = [(sqrt(2) / 2),0,(sqrt(2) / 2),0;0,(sqrt(2) /
2),0,(sqrt(2) / 2);(sqrt(2) / 2),0,((-1 * sqrt(2)) / 2),0;0,(sqrt(2)
/ 2),0,((-1 * sqrt(2)) / 2)];

const matrix A4 = [1,0,0,0;0,1,0,0;0,0,0,0;0,0,0,0];

const matrix A5 = [0,0,0,0;0,0,0,0;0,0,1,0;0,0,0,1];

module test

s: [0..5] init 0;

[] (s =0) -> <<A1>> : (s' = 1);
[] (s =1) -> <<A2>> : (s' = 2);
[] (s = 2) -> <<A3>> : (s' = 3);
[] (s = 3) -> <<A4>> : (s' = 4) + <<A5>> : (s' = 5);
[] (s = 4) -> (s' = 4);
[] (s = 5) -> (s' = 5);

endmodule
```

Uno de los grandes problemas a los que se enfrenta la ingeniería del software cuántico es que los algoritmos cuánticos suelen diseñarse en un nivel análogo al funcionamiento de las puertas lógicas (Francik J., 2002), lo que dificulta el desarrollo de nuevos algoritmos por ingenieros informáticos. Con el creciente desarrollo de un nivel intermedio, desde el punto de vista del software, como está ocurriendo con entornos tales como *IBM Q Experience*, la tendencia puede cambiar hacia sistemas más familiares como los clásicos entornos de programación.

7. Conclusiones y discusión

En este trabajo se ha realizado una revisión bibliográfica del estado del arte de la computación cuántica con el objetivo final de conocer y comprender el estado actual del desarrollo la tecnología cuántica, los logros que se han alcanzo hasta ahora –junto la capacidad computacional actual–, así como las perspectivas en los próximos años.

En primer lugar, se ha explicado los principios básicos sobre los que se sustenta la computación cuántica, resaltando las cualidades cuánticas del entrelazamiento y la superposición. Estas características, que permiten el manejo y procesamiento de grandes cantidades de información de forma mucho más eficiente que los computadores clásicos, son las razones por las que la computación cuántica puede convertirse en una tecnología muy potente.

En segundo lugar, se introdujeron las clases de complejidad. Los problemas son clasificados según su complejidad computacional, resaltando principalmente los problemas tratables y los no tratables. Los primeros son resolubles en tiempo eficiente, es decir, constante, logarítmico o polinómico. Por el contrario, los segundos no encuentran solución eficiente pues su complejidad aumenta a un ritmo exponencial. De tal modo, estos problemas son los que más importancia tienen actualmente, puesto que, si un día llegan a ser resolubles, entonces muchos de los retos actuales en diversas áreas podrían llegar a ser resueltos.

En tercer lugar, se presentan algunos de los algoritmos cuánticos más conocidos. Estos algoritmos están basados en diferentes principios, como la transformada de Fourier o la estimación de fase. Ciertamente cabe destacar que los algoritmos cuánticos suelen estar centrados es un abanico de problemas, de momento, no muy amplio. Los principales algoritmos son diseñados para problemas optimización, problemas de satisfacción de variables, problemas de grafos, problemas de búsqueda, y problemas de simulación cuántica, como la búsqueda de autovalores y autovectores.

Posteriormente, se ha presentado el concepto de la supremacía cuántica, es decir, la demostración de que los ordenadores cuánticos son mejores resolviendo problemas que el mejor de los ordenadores clásicos. Pero como se ha visto, esta demostración aún no ha ocurrido. Mientras que, por una parte, ciertos sectores de la investigación se mantienen positivos para llegar este punto pronto, es el caso de Google, existen otras voces que piensan que esto nunca acabará por ocurrir.

Adicionalmente, se han detallado los dos principales sistemas cuánticos con los que se trata de llegar a la supremacía cuántica. Por un lado, los sistemas cuánticos universales, basados en circuitos cuánticos y puertas lógicas. Este modelo es el más prometedor, no obstante, se enfrenta a ciertas limitaciones tecnológicas puesto que los sistemas cuánticos, para poder demostrar la supremacía, deben contar con al menos 50 qubits. El problema surge por la extrema estabilidad que debe existir en estos sistemas

bajo unas condiciones extremas. Para una perfecta computación, todos los qubits deben estar entrelazados correctamente, pero existen ciertas interferencias y factores externos que pueden generar errores en el cómputo y, por tanto, en el resultado. A pesar de ello, Google ha afirmado en rotundas ocasiones que para finales de 2019 serán capaces de llegar a tal situación. Actualmente ya cuentan con un sistema de 72 qubits, pero esto no ha supuesto la demostración de la supremacía cuántica. Por otra parte, IBM es algo más recelosa y no prevé que esto ocurra hasta los próximos tres años.

Como alternativa a la computación cuántica universal existe otra tecnología liderada por D-Wave conocida como computación cuántica adiabática. La computación adiabática está abordando algunos de los problemas comentados previamente de forma diferente, ya que los qubits no se encuentran en perfecta superposición, sino que ciertos qubits se entrelazan de forma aleatoria, lo que supone depender, en menor medida, de los errores comunes de la computación universal.

Además, estos sistemas hacen uso de algoritmos de temple cuántico, los cuales se tratan de algoritmos de búsqueda heurística aplicada a problemas de optimización con múltiples variables independientes. Esto aporta una gran ventaja debido a que muchos problemas computacionales son fácilmente transformables en problemas de asignación variable. Así, el objetivo final del temple cuántico es encontrar el máximo o mínimo de una función. Actualmente los computadores cuánticos más avanzados de D-Wave han conseguido contar con más de 2.000 qubits, pero esto no ha sido suficiente para demostrar la supremacía cuántica, ya que no se beneficia de forma real y completa de las técnicas cuánticas per se. No así, sus algoritmos son capaces de resolver grandes problemas de optimización como la gestión del tráfico de una ciudad o la gestión de los procesos logísticos de un centro distribuidor. En dichos casos los resultados parecen aportar ciertas ventajas que pueden ser aprovechadas para enfrentarse a muchos de los problemas del mundo.

En último lugar, se ha destinado tiempo a comentar el estado de la ingeniería del software cuántico. En la actualidad, existe una oferta interesante de simuladores y librerías cuánticas para el diseño, ejecución y análisis de circuitos y algoritmos cuánticos. Por su parte, los lenguajes de programación cuántica también están ganando algo de presencia entre los desarrolladores – QCL o OpenQASM. Aunque también existen varias alternativas con lenguajes más comunes como Python.

Dicho esto, queda claro que el desarrollo de sistemas y software cuántico acaba de comenzar y, a pesar de la existencia de múltiples plataformas para el desarrollo de algoritmos cuánticos, es indispensable que se produzca un progreso en cuanto al enfoque de trabajo de dichas plataformas para que sean menos abstractas –en términos de conceptos físico-cuánticos y electrónicos como las puertas lógicas– a favor de plataformas más familiares a las herramientas clásicas de un ingeniero de software con el objetivo de desarrollar verdaderas aplicaciones y algoritmos cuánticos que resuelvan los problemas actuales y del futuro.

8. Bibliografía

Aaronson, S., & Arkhipov, A. (2013). Bosonsampling is far from uniform. arXiv preprint arXiv:1309.7460.

Abrams, D. S., & Lloyd, S. (1999). Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. Physical Review Letters, 83(24), 5162.

Agrawal, M., Kayal, N., & Saxena, N. (2004). PRIMES is in P. Annals of mathematics, 781-793.

Aharonov, D., Van Dam, W., Kempe, J., Landau, Z., Lloyd, S., & Regev, O. (2008). Adiabatic quantum computation is equivalent to standard quantum computation. SIAM review, 50(4), 755-787.

Albash, T., & Lidar, D. A. (2018). Adiabatic quantum computation. Reviews of Modern Physics, 90(1), 015002.

Albeverio, S., Ho/egh-Krohn, R., & Streit, L. (1977). Energy forms, Hamiltonians, and distorted Brownian paths. Journal of Mathematical Physics, 18(5), 907-917.

Ambainis, A. (2004). Quantum search algorithms. ACM SIGACT News, 35(2), 22-35.

Andraca, S. E. V. (2008). Caminatas cuánticas: definiciones y algoritmos.

Anticoli, L., Piazza, C., Taglialegne, L., & Zuliani, P. (2017). Verifying Quantum Programs: From Quipper to QPMC. arXiv preprint arXiv:1708.06312.

Apolloni, B., N. Cesa-Bianchi, and D. de Falco (1988). in Proceedings of the Ascona/Locarno Conference, p. 97.

Applegate, D. L., Bixby, R. E., Chvátal, V., Cook, W., Espinoza, D. G., Goycoolea, M., & Helsgaun, K. (2009). Certification of an optimal TSP tour through 85,900 cities. Operations Research Letters, 37(1), 11-15.

Arkhipov, A. (2009). Universal quantum gates. Scott Aaronson's course projects at MIT.

Arora, S., & Barak, B. (2009). Computational complexity: a modern approach. Cambridge University Press.

Aspect, A., Dalibard, J., & Roger, G. (1982). Experimental test of Bell's inequalities using time-varying analyzers. Physical review letters, 49(25), 1804.

Auger, J. M. (2018). Fault-tolerant Quantum Computation with Realistic Error Models (Doctoral dissertation, UCL (University College London)).

Baker, T., Gill, J., & Solovay, R. (1975). Relativizations of the P=?NP question. SIAM Journal on computing, 4(4), 431-442.

Beigel, R., & Eppstein, D. (2005). 3-coloring in time O (1.3289 n). Journal of Algorithms, 54(2), 168-204.

Bennett, C. H., & Brassard, G. (2014). Quantum cryptography: public key distribution and coin tossing. Theor. Comput. Sci., 560(12), 7-11.

Benioff, P. (2000). Space searches with a quantum robot. arXiv preprint quant-ph/0003006.

Bennett, C. H. (2003). Notes on Landauer's principle, reversible computation, and Maxwell's Demon. Studies In History and Philosophy of Science Part B: Studies In History and Philosophy of Modern Physics, 34(3), 501-510.

Bennett, C. H., Bernstein, E., Brassard, G., & Vazirani, U. (1997). Strengths and weaknesses of quantum computing. SIAM journal on Computing, 26(5), 1510-1523.

Bernien, H., Hensen, B., Pfaff, W., Koolstra, G., Blok, M. S., Robledo, L., ... & Hanson, R. (2013). Heralded entanglement between solid-state qubits separated by three metres. Nature, 497(7447), 86.

Bernstein, E., & Vazirani, U. (1997). Quantum complexity theory. SIAM Journal on computing, 26(5), 1411-1473.

Berthiaume, A., Deutsch, D., & Jozsa, R. (1994). Proceedings of the Workshop on Physics and Computation—PhysComp'94.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. Nature, 549(7671), 195.

Bonillo, V. M. (2013). Principios fundamentales de computación cuántica. Universidad de La Coruña.

Bremner, M. J., Jozsa, R., & Shepherd, D. J. (2010). Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 467(2126), 459-472.

Carolan, J., Harrold, C., Sparrow, C., Martín-López, E., Russell, N. J., Silverstone, J. W., ... & Marshall, G. D. (2015). Universal linear optics. Science, 349(6249), 711-716.

Childs, A. M., Farhi, E., & Preskill, J. (2001). Robustness of adiabatic quantum computation. Physical Review A, 65(1), 012322.

Cirac, J. I., & Tudela, A. G. (2019). Óptica cuántica sin fotones. Investigación y ciencia, (510), 14-16.

Cirac, J. I., & Zoller, P. (2000). A scalable quantum computer with ions in an array of microtraps. Nature, 404(6778), 579.

Cobham, A. (1964). The intrinsic computational difficulty of functions Proc. Int. Cong. for Logic, Methodology and Philosophy of Science II (North Holland).
Computational complexity: a conceptual perspective. ACM Sigact News, 39(3), 35-39, 2-3.

Cook, S. (2006). The P versus NP problem. The millennium prize problems, 87-104.

Cook, S. A. (1971, May). The complexity of theorem-proving procedures. In Proceedings of the third annual ACM symposium on Theory of computing (pp. 151-158). ACM.

Cook, S. A. (1973). A hierarchy for nondeterministic time complexity. Journal of Computer and System Sciences, 7(4), 343-353.

Courtland, R. (2017). Google aims for quantum computing supremacy [News]. IEEE Spectrum, 54(6), 9-10.

Cross, A. W., Bishop, L. S., Smolin, J. A., & Gambetta, J. M. (2017). Open quantum assembly language. arXiv preprint arXiv:1707.03429.

Das, A., & Chakrabarti, B. K. (2008). Colloquium: Quantum annealing and analog quantum computation. Reviews of Modern Physics, 80(3), 1061.

Deutsch, D. (1985). Quantum theory, the Church–Turing principle and the universal quantum computer. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400(1818), 97-117.

Deutsch, D. (1989). Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 425 (1868), 73.

Devoret, M. H., & Schoelkopf, R. J. (2013). Superconducting circuits for quantum information: an outlook. Science, 339(6124), 1169-1174.

De Wolf, R. (2017). The potential impact of quantum computers on society. Ethics and Information Technology, 19(4), 271-276.

Dirac, P. A. M. (1939, July). A new notation for quantum mechanics. In Mathematical Proceedings of the Cambridge Philosophical Society (Vol. 35, No. 3, pp. 416-418). Cambridge University Press.

Dusanowski, Ł., Kwon, S. H., Schneider, C., & Höfling, S. (2019). Near-unity indistinguishability single photon source for large-scale integrated quantum optics. Physical review letters, 122(17), 173602.

Elshaari, A. W., Zadeh, I. E., Fognini, A., Reimer, M. E., Dalacu, D., Poole, P. J., ... & Jöns, K. D. (2017). On-chip single photon filtering and multiplexing in hybrid quantum photonic circuits. Nature communications, 8(1), 379.

Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000). Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106.

Feynman, R. P. (1982a), International Journal of Theoretical Physics 21 (6), 467.

Feynman, R. P. (1982b). Simulating physics with computers. International journal of theoretical physics, 21(6), 467-488.

Feynman, R. P. (1985). Quantum mechanical computers. Optics news, 11(2), 11-20.

Fingerhuth, M., Babej, T., & Wittek, P. (2018). Open source software in quantum computing. PloS one, 13(12), e0208561.

Fortin, S. (1996). The graph isomorphism problem.

Fortnow, L. (2009). The status of the P versus NP problem. Communications of the ACM, 52(9), 78-86.

Fortnow, L., & Homer, S. (2003). A short history of computational complexity. Boston University Computer Science Department.

Francik, J. (2002). Quantum software. Studia informatica, 23(2A), 48.

Georgescu, I. M., Ashhab, S., & Nori, F. (2014). Quantum simulation. Reviews of Modern Physics, 86(1), 153.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. arXiv preprint quant-ph/9605043.

Grover, L. K. (1997). Quantum mechanics helps in searching for a needle in a haystack. Physical review letters, 79(2), 325.

Grover, L. K. (1997). Quantum mechanics helps in searching for a needle in a haystack. Physical review letters, 79(2), 325.

Gürer, D. (2002). Pioneering women in computer science. ACM SIGCSE Bulletin, 34(2), 175-180.

Harper, R., & Flammia, S. T. (2019b). Fault-tolerant logical gates in the ibm quantum experience. Physical review letters, 122(8), 080504.

Harper, R., Flammia, S. T., & Wallman, J. J. (2019a). Efficient learning of quantum noise. arXiv preprint arXiv:1907.13022.

Harrow, A. W., & Montanaro, A. (2017). Quantum computational supremacy. Nature, 549(7671), 203.

Hastings, M. B., Wecker, D., Bauer, B., & Troyer, M. (2014). Improving quantum algorithms for quantum chemistry. arXiv preprint arXiv:1403.1539.

Held, M., & Karp, R. M. (1962). A dynamic programming approach to sequencing problems. Journal of the Society for Industrial and Applied mathematics, 10(1), 196-210.

Hempel, C., Maier, C., Romero, J., McClean, J., Monz, T., Shen, H., ... & Aspuru-Guzik, A. (2018). Quantum chemistry calculations on a trapped-ion quantum simulator. Physical Review X, 8(3), 031022.

Hong, Q. T., Ge, Z. Y., Wang, W., Lang, H. F., Wang, Z. A., Peng, Y., ... & Fan, H. (2018). A Universal Quantum Computing Virtual Machine. arXiv preprint arXiv:1806.06511.

ISO, I. (2010). IEC/IEEE 24765: 2010 systems and software engineering-vocabulary. Technical report, Institute of Electrical and Electronics Engineers, Inc.

Jaksch, D., García-Ripoll, J. J., Cirac, J. I., & Zoller, P. (2016). Quantum Computing with Cold Ions and Atoms: Theory. Quantum Information: From Foundations to Quantum Technology Applications, 483-517.

Johnson, D. S. (1982). The NP-completeness column: An ongoing gulde. Journal of Algorithms, 3(4), 381-395

Kalai, G. (2019). The Argument against Quantum Computers. arXiv preprint arXiv:1908.02499.

Kelly, J. A preview of Bristlecone, Google's new quantum processor (2018). URL https://ai. googleblog. com/2018/03/a-preview-ofbristlecone-googles-new. html.

Kim, T., Maunz, P., & Kim, J. (2011). Efficient collection of single photons emitted from a trapped ion into a single-mode fiber for scalable quantum-information processing. Physical Review A, 84(6), 063423.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983), Science 220 (4598), 671.

Krinner, L., Stewart, M., Pazmiño, A., Kwon, J., & Schneble, D. (2018). Spontaneous emission of matter waves from a tunable open quantum system. Nature, 559(7715), 589.

Linke, N. M., Maslov, D., Roetteler, M., Debnath, S., Figgatt, C., Landsman, K. A., ... & Monroe, C. (2017). Experimental comparison of two quantum computing architectures. Proceedings of the National Academy of Sciences, 114(13), 3305-3310.

Little, J. D., Murty, K. G., Sweeney, D. W., & Karel, C. (1963). An algorithm for the traveling salesman problem. Operations research, 11(6), 972-989.

Lomonaco, S. J. (2002). A Rosetta stone for quantum mechanics with an introduction to quantum computation. In Proceedings of Symposia in Applied Mathematics (Vol. 58, pp. 3-66).

Lund, A. P., Bremner, M. J., & Ralph, T. C. (2017). Quantum sampling problems, BosonSampling and quantum supremacy. npj Quantum Information, 3(1), 15.

Mandra, S., Katzgraber, H. G., & Thomas, C. (2017). The pitfalls of planar spin-glass benchmarks: raising the bar for quantum annealers (again). Quantum Science and Technology, 2(3), 038501.

Márquez, A. M. (2019). Postulados de la mecánica cuántica.

Monroe, C. R., Schoelkopf, R. J., & Lukin, M. D. (2016). Quantum connections. Scientific American, 314(5), 50-57.

Montanaro, A. (2016). Quantum algorithms: an overview. npj Quantum Information, 2, 15023.

Moore, S. K. (2019). Another step toward the end of Moore's law: Samsung and TSMC move to 5-nanometer manufacturing-[News]. IEEE Spectrum, 56(6), 9-10.

Mouatadid, L. (2014). Introduction to Complexity Theory: 3-Colouring is NP-complete.

Mutiara, A. B., & Refianti, R. (2010). Simulation of Grover's Algorithm Quantum Search in a Classical Computer. International Journal of Computer Science and Information Security, 1.

Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., & Parney, B. (2017). Traffic flow optimization using a quantum annealer. Frontiers in ICT, 4, 29.

Nielsen, M. A., & Chuang, I. L. (2001). Quantum computation and quantum information. Phys. Today, 54, 60-2.

Orus, R., Mugel, S., & Lizaso, E. (2019). Quantum computing for finance: overview and prospects. Reviews in Physics, 100028.

O'Regan, G. (2008). A brief history of computing. Springer Science & Business Media.

Patterson, A. D., Rahamim, J., Tsunoda, T., Spring, P., Jebari, S., Ratter, K., ... & Leek, P. J. (2019). Calibration of the cross-resonance two-qubit gate between directly-coupled transmons. arXiv preprint arXiv:1905.05670.

Preskill, J. (2012). Quantum computing and the entanglement frontier. arXiv preprint arXiv:1203.5813.

Prieto Idarraga, B. N. (2018). Lenguajes de Programación Cuánticos (LPC).

Reed, M. D., DiCarlo, L., Nigg, S. E., Sun, L., Frunzio, L., Girvin, S. M., & Schoelkopf, R. J. (2012). Realization of three-qubit quantum error correction with superconducting circuits. Nature, 482(7385), 382.

Rolf, D. (2003). Improving Randomized Local Search by Initializing Strings of 3-Clauses.

Rønnow, T. F., Wang, Z., Job, J., Boixo, S., Isakov, S. V., Wecker, D., ... & Troyer, M. (2014). Defining and detecting quantum speedup. Science, 345(6195), 420-424.

Saha, Muskan & Behera, Bikash & Panigrahi, Prasanta. (2019). Quantum Algorithms for Colouring of Graphs. 10.13140/RG.2.2.19222.19523.

Shepherd, D., & Bremner, M. J. (2009). Temporally unstructured quantum computation. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 465(2105), 1413-1439.

Shi, Y. (2002). Both Toffoli and controlled-NOT need little help to do universal quantum computation. arXiv preprint quant-ph/0205115.

Shor, P. W. (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science (pp. 124-134). Ieee.

Shor, P. W. (1995). Scheme for reducing decoherence in quantum computer memory. Physical review A, 52(4), R2493.

Sicard, E. (2017). Introducing 7-nm FinFET technology in Microwind

Soare, R. I. (2009). Turing oracle machines, online computing, and three displacements in computability theory. Annals of Pure and Applied Logic, 160(3), 368-399.

Solovay, R., & Strassen, V. (1977). A fast Monte-Carlo test for primality. SIAM journal on Computing, 6(1), 84-85.

Sparrow, C. (2017). Quantum interference in universal linear optical devices for quantum computation and simulation.

Steane, A. (1998). Quantum computing. Reports on Progress in Physics, 61(2), 117.

Stephen, C. J., Green, B. L., Lekhai, Y. N. D., Weng, L., Hill, P., Johnson, S., ... & Newton, M. E. (2018). Three-dimensional solid-state qubit arrays with long-lived spin coherence. arXiv preprint arXiv:1807.03643.

Trump, D. (2018). National Cyber Strategy of the United States of America. Washington, DC.

Turing, A. M. (1937). On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London mathematical society, 2(1), 230-265.

Wallraff, A., Schuster, D. I., Blais, A., Frunzio, L., Huang, R. S., Majer, J., ... & Schoelkopf, R. J. (2004). Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics. Nature, 431(7005), 162.

Wang, J., Paesani, S., Ding, Y., Santagati, R., Skrzypczyk, P., Salavrakos, A., ... & Bonneau, D. (2018). Multidimensional quantum entanglement with large-scale integrated optics. Science, 360(6386), 285-291.