



Proyecto pedagógico de

Computación Cuántica

Juan Pablo Salas

Contenido

I

Computación clásica

1	Bits.....	7
1.1	Bitjack.....	9
2	Compuertas lógicas	11
2.1	Compuertas lógicas con circuitos	15
3	Algoritmos	17
3.1	Pensamiento algorítmico	18
4	Limitaciones	19

II

Computación cuántica

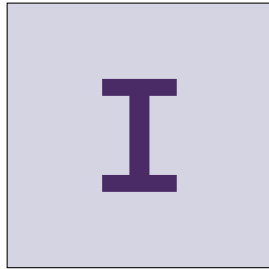
5	Qubits.....	23
5.1	Qubit con luz láser	25
6	Compuertas cuánticas.....	27
7	Entrelazamiento.....	31
7.1	Teleportación cuántica	34

III

Realizaciones físicas

8	Luz	41
9	Iones atrapados	43
10	Superconductores	45

11	Búsqueda	51
12	Criptografía cuántica	53



Computación clásica



1	Bits.....	7
1.1	Bitjack	
2	Compuertas lógicas.....	11
2.1	Compuertas lógicas con circuitos	
3	Algoritmos.....	17
3.1	Pensamiento algorítmico	
4	Limitaciones	19

Bits

Definición

Un **bit** es la unidad más básica de información de un computador. Representa un valor de verdad (verdadero/falso) comúnmente con **0** y **1**.

La información en un computador (imágenes, texto, archivos, etc.) se almacena utilizando una cadena de bits. Los bits se pueden representar de distintas maneras como **verdadero/falso**, **sí/no**, **on/off**, **+/-** y **0/1**. Esta última es la más común porque nos permite manipular las cadenas de bits utilizando el *sistema binario*. Se puede pensar en un bit como un interruptor de luz que puede estar o prendido o apagado.

¿Para qué sirven?

El uso principal de los bits en los sistemas computacionales es de almacenar información en la memoria del sistema. Esto se hace en la **memoria principal** del sistema, una colección de circuitos que pueden almacenar el valor de un bit cada uno. Esta memoria normalmente está organizada en celdas de 8 bits cada una, equivalente a 1 byte ¡Un computador puede tener hasta el orden de 10^9 celdas de memoria! [8]

Por lo general, los computadores tienen un número total de celdas que corresponde a una potencia de 2. Por esta razón, se estableció el prefijo *kilo* para el número 1024, equivalente a 2^{10} .



¿Cuántos **bits** hay en una memoria de 2 *gigabytes*?

Sistema binario

El sistema binario es un sistema numérico que permite representar cualquier número utilizando solamente **0** y **1**.

En nuestro sistema decimal, la representación del número 1325 es una suma de cada dígito multiplicado por la potencia de 10, asociada a la posición del dígito

$$1325 = 1 \times 10^3 + 3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$

En el sistema binario, tenemos una suma de cada dígito multiplicado por la potencia de 2, asociada a la posición del dígito

$$\begin{aligned} 10100101101 &= 1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 \\ &\quad + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 1024 + 0 + 256 + 0 + 0 + 32 + 0 + 8 + 4 + 0 + 1 = 1325 \end{aligned}$$

Para convertir un número en representación decimal a representación binaria basta con dividirlo entre 2 y al final anotar los residuos de la división en orden inverso. Para 23, por ejemplo

$$23/2 = 11(\text{residuo } 1)$$

$$11/2 = 5(\text{residuo } 1)$$

$$5/2 = 2(\text{residuo } 1)$$

$$2/2 = 1(\text{residuo } 0)$$

$$1/2 = 0(\text{residuo } 1)$$

El número 23 en sistema binario es 10111.

1.1 Bitjack

Objetivos de aprendizaje

- Familiarizarse con el sistema binario.
- Emplear los algoritmos de conversión de sistema decimal a sistema binario en un juego de estrategia.
- Comparar valores de distintas magnitudes en su representación binaria.

Materiales

- Tablero con cinco casillas (uno por jugador/equipo)
- 1 dado
- 52 cartas del naipes de **Bitjack**

Objetivo del juego

Formar el número **21** en representación binaria o acercarse lo más posible a este número sin pasarlo. Si no se logra formar este número, también se consideran ganadores los que completan el número **0** o el **31**.

¿Cómo jugar?

1. Un jugador llamado **repartidor** tendrá inicialmente las 52 cartas de la baraja.
2. Para comenzar, cada jugador tira el dado que determina la posición en la que la primera carta irá en su tablero. El repartidor entrega una carta al azar y la ubica en la posición del dado boca arriba.
3. En cada turno, el jugador debe tirar el dado y recibir una carta del repartidor que se ubicará en la posición del dado. Estas cartas son puestas boca abajo, sólo para que el dueño del tablero las vea.
4. Si al caer el dado marca una posición que ya tiene carta, esta se reemplazará por una carta al azar.
5. Una vez un jugador completa el número **21**, o cree estar lo más cerca posible a este puede decidir plantarse y el repartidor dará cartas por una ronda más.
6. El jugador que llegue al número **21** en binario o se acerque lo máximo a este sin pasarse, será el ganador.

Preguntas orientadoras

Antes de comenzar a jugar, responde estas preguntas.

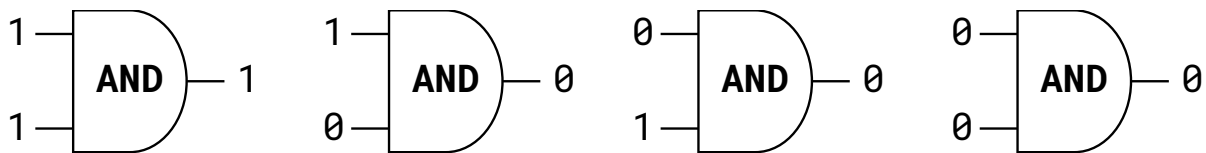
1. ¿Qué número representa un tablero con todos **0**? ¿Y todos **1**?
2. ¿Cuál es el número más grande que se puede obtener en un tablero de 2 casillas en representación binaria?
3. ¿Cuál es el número más grande que se puede obtener en un tablero de 4 casillas en representación binaria?
4. ¿Cuál es el número más grande que se puede obtener en un tablero de 5 casillas en representación binaria?
5. ¿Cuál es el número más grande que se puede obtener en un tablero de n casillas en representación binaria?
6. ¿Cómo se representa el número **21** en el sistema binario?
7. Cuando se pone un 0 al final de un número en el sistema decimal, se multiplica por 10 (por ejemplo 9 se vuelve $90 = 9 \times 10$). ¿Qué pasa cuando se pone un 0 al final de un número en su representación binaria?

Compuertas lógicas

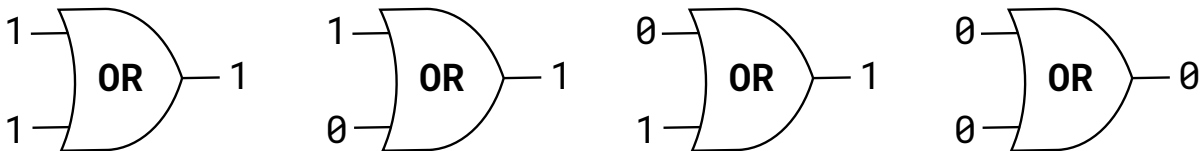
Después de entender que los bits son la unidad más básica de información de un computador, necesitamos explorar cómo estos se pueden manipular y combinar para alterar su estado y así transmitir información. Así pues, se utilizan las **operaciones booleanas** para modificar los valores de los bits. Estas operaciones pueden ser **unitarias**, es decir que reciben un único bit, o **binarias**, cuya entrada es de dos bits.

Las principales operaciones booleanas son **AND**, **OR**, **XNOT** y **NOT** cuya definición se muestra a continuación. Recordemos del capítulo anterior que el valor 1 representa el valor **verdadero** y el valor 0 representa el valor **falso**. [8]

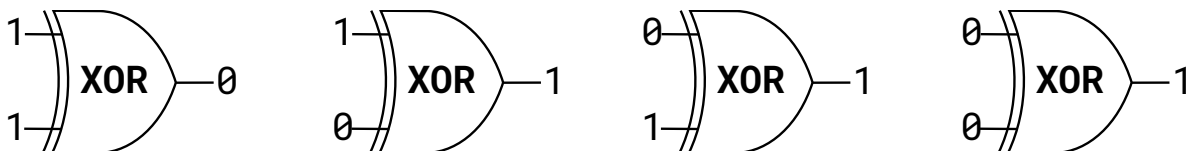
- **AND** Su resultado es verdadero si el primer bit **y** el segundo son verdaderos, de lo contrario es falso.



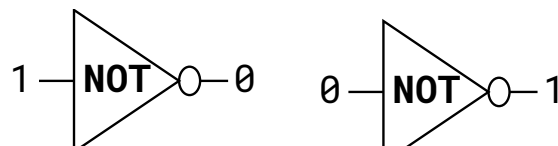
- **OR** Su resultado es verdadero sólo si el primer bit **o** el segundo son verdaderos.



- **XOR** Su resultado es verdadero si el primer bit **o** el segundo bit son verdaderos, de forma **exclusiva**.



- **NOT** Su resultado es el valor de verdad inverso al valor de verdad de entrada.



A partir de estas operaciones se pueden construir algunas operaciones más complicadas. Estas operaciones son la base de las compuertas lógicas.

Definición

Una **compuerta lógica** es un dispositivo produce el resultado de una operación booleana. Estos dispositivos pueden ser electrónicos u ópticos. En los computadores de hoy en día se utilizan valores de voltaje para representar los **0** y **1** y así manipularlos.

Propiedades

A continuación podemos ver algunas propiedades que se cumplen cuando **A, B, C** son bits 0 o 1.

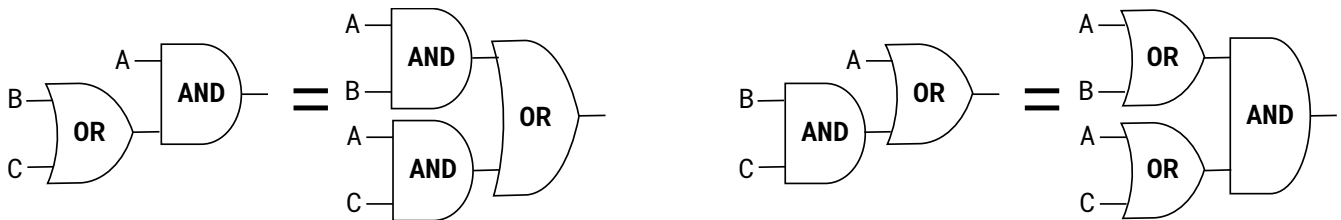


Figura 2.1: Estas dos propiedades también se pueden escribir como $A \text{ AND } (B \text{ OR } C) = (A \text{ AND } B) \text{ OR } (A \text{ AND } C)$. La de la derecha se puede escribir como: $A \text{ OR } (B \text{ AND } C) = (A \text{ OR } B) \text{ AND } (A \text{ OR } C)$.

Las siguientes nos ayudan a negar un **AND** o un **OR**.

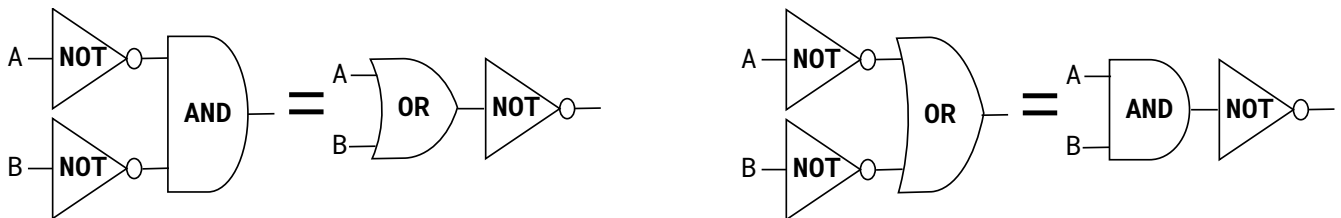
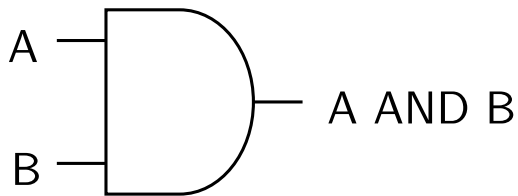


Figura 2.2: Estas dos propiedades también se pueden escribir como $(\text{NOT } A) \text{ AND } (\text{NOT } B) = \text{NOT } (A \text{ OR } B)$. La de la derecha se puede escribir como $(\text{NOT } A) \text{ OR } (\text{NOT } B) = \text{NOT } (A \text{ AND } B)$.

Compuerta AND



A	B	A AND B
1	1	1
1	0	0
0	1	0
0	0	0

Figura 2.3: Representación gráfica de la compuerta lógica AND.

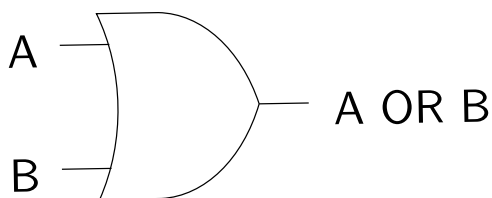
Tabla 2.1: Tabla de verdad para la compuerta lógica AND.

Cuando se trabaja en base binaria, esta operación puede entenderse como la multiplicación de los dos bits de entrada.

$$A \text{ AND } B = A \cdot B$$

En varios libros también se utiliza el símbolo \wedge para denotar la compuerta **AND**.

Compuerta OR



A	B	A OR B
1	1	1
1	0	1
0	1	1
0	0	0

Figura 2.4: Representación gráfica de la compuerta lógica OR.

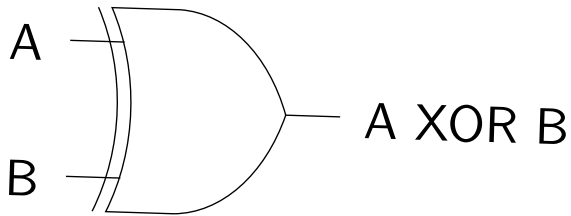
Tabla 2.2: Tabla de verdad para la compuerta lógica OR.

Cuando se trabaja en base binaria, esta operación puede entenderse como la adición de los dos bits de entrada.

$$A \text{ OR } B = A + B$$

En varios libros también se utiliza el símbolo \vee para denotar la compuerta **OR**.

Compuerta XOR



A	B	A XOR B
1	1	0
1	0	1
0	1	1
0	0	0

Figura 2.5: Representación gráfica de la compuerta lógica XOR.

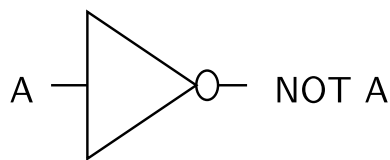
Tabla 2.3: Tabla de verdad para la compuerta lógica XOR.

Cuando se trabaja en base binaria, esta operación puede entenderse como la adición módulo 2 de los dos bits de entrada.

$$A \text{ XOR } B = A \oplus B$$

En varios libros también se utiliza el símbolo \neq para denotar la compuerta XOR.

Compuerta NOT



A	NOT A
1	0
0	1

Figura 2.6: Representación gráfica de la compuerta lógica NOT.

Tabla 2.4: Tabla de verdad para la compuerta lógica NOT.

En varios libros también se utiliza el símbolo \neg , \sim , \overline{A} para denotar la compuerta NOT.

2.1 Compuertas lógicas con circuitos

Objetivos de aprendizaje

- Descubrir el funcionamiento de las cuatro operaciones booleanas principales.
- Simular los conceptos de compuertas lógicas y operaciones booleanas en dispositivos reales.
- Proponer equivalencias lógicas a distintas compuertas y operaciones booleanas a partir de operaciones más sencillas (**AND** y **NOT**).
- Diseñar circuitos lógicos para funciones booleanas.

Materiales

- 4 simuladores de compuertas lógicas

Instrucciones

Se sugiere que esta actividad se realice en grupos pequeños de 4-5 estudiantes.

Los cuatro simuladores de esta actividad corresponden a las cuatro compuertas lógicas desarrolladas en esta unidad. Estos están contru-
idos a partir de interruptores (**ON/OFF**), los cuales corresponden a los
valores **1** y **0** de un bit. El resultado de estas operaciones se muestra
con unos bombillos LEDs donde **prendido** representa el valor **1** y **apagado**
representa **0**.

1. Con los diagramas de compuertas lógicas de los simuladores cubiertos, utiliza las tablas de verdad y prueba las diferentes opciones de los interruptores para clasificarlos.
2. Cada integrante del grupo debe escoger una de las propiedades de los operadores booleanos y con ayuda de los interruptores construir su tabla de verdad para verificarla. Después de que las tengan, las deben compartir con su grupo.
3. Utilizando estos simuladores, responde las preguntas orientadores a continuación.

Preguntas orientadoras

1. Construye la tabla de verdad de las cuatro compuertas lógicas utilizando los simuladores.
2. Construye la tabla de verdad de las siguientes operaciones. ¿Qué

puedes concluir al respecto?

- a. $A \text{ OR } 0$
 - b. $A \text{ AND } 1$
 - c. $A \text{ AND } A$
 - d. $A \text{ OR } (A \text{ AND } B)$
3. Construye las tablas de verdad de las siguientes operaciones de tres bits A,B y C.
- a. $A \text{ AND } (B \text{ OR } C)$
 - b. $\text{NOT } (A \text{ AND } B \text{ OR } C)$
 - c. $A \text{ XOR } (B \text{ XOR } C)$
 - d. $A \text{ OR } (\text{NOT } (A) \text{ AND } B) \text{ OR } (\text{NOT } (A) \text{ AND } \text{NOT } (B))$
 - e. $A \text{ OR } (\text{NOT } (A) \text{ AND } B)$
4. Usando una tabla de verdad y los simuladores de compuertas lógicas, verifica que las propiedades mostradas en el capítulo se cumplen.
5. Cuando una operación cualquiera tiene tres bits, ¿cuántos posibles resultados tiene su tabla de verdad? ¿Y cuando tiene n bits?
6. Utilizando las propiedades de las operaciones booleanas, simplifica las siguientes expresiones para utilizar el menor número de compuertas lógicas. Dibuja el resultado utilizando las representaciones gráficas de las compuertas lógicas.
- a. $(A \text{ AND } B) \text{ OR } (B \text{ AND } (B \text{ OR } \text{NOT } (C))) \text{ OR } (\text{NOT } (B) \text{ AND } C)$
 - b. $A \text{ AND } (B \text{ OR } A \text{ AND } B) \text{ OR } (A \text{ AND } C)$
 - c. $(A \text{ OR } B) \text{ AND } (\text{NOT } (A) \text{ OR } \text{NOT } (B))$
 - d. $(C \text{ AND } A) \text{ OR } (C \text{ AND } \text{NOT } (A)) \text{ OR } (C \text{ AND } B)$
7. ¿Es posible reescribir el siguiente circuito utilizando solamente compuertas **AND** y **NOT**?

$$A \text{ AND } B \text{ OR } A \text{ AND } (B \text{ OR } C) \text{ OR } B \text{ AND } (B \text{ OR } C)$$

Algoritmos

Las compuertas lógicas estudiadas anteriormente son la base para la construcción de los computadores actuales. Sin embargo, para que un computador pueda completar una tarea, es necesario que tenga un **algoritmo** que le diga precisamente qué hacer.

Definición

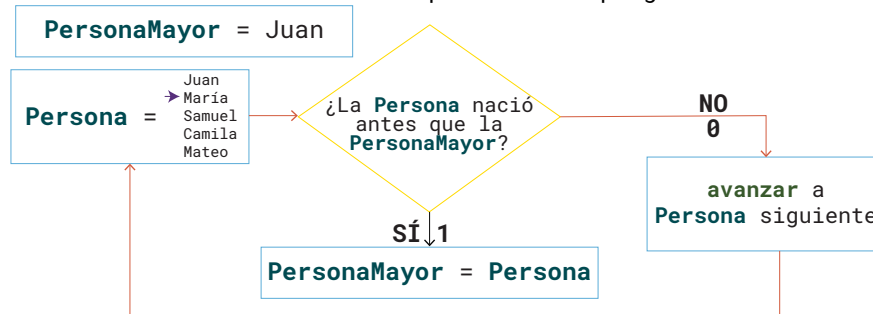
Un **algoritmo** es un conjunto *ordenado* de pasos que definen un proceso con un objetivo. [8]

Los algoritmos están presentes en nuestra vida de tantas maneras que a veces no nos damos ni cuenta. Acciones cotidianas como amarrarnos los zapatos o hacer una torta son uno de los cuantos ejemplos de algoritmos. Sin embargo, para que un computador pueda entender un determinado algoritmo, es necesario que este escrito en un cierto *lenguaje de programación*.

En ciencias de la computación, los algoritmos son las herramientas que nos permiten resolver problemas por lo que su desarrollo requiere de gran astucia y creatividad. Algunos de estos problemas a los que se enfrenta un computador pueden ser: ordenar un grupo de objetos, buscar un elemento en una lista, encontrar la raíz cuadrada de un número o incluso factorizar un número. A continuación veamos un ejemplo de un algoritmo para encontrar la persona más vieja de un grupo de gente.

Problema: Dado un grupo de personas que se saben su fecha de nacimiento, encuentra la persona más vieja del grupo.

Para resolver este problema, empezamos escogiendo una **PersonaMayor** cualquiera, como por ejemplo Juan. Ahora, debemos ir a cada **Persona** a preguntar si nacieron antes que quien decimos es la **PersonaMayor**. Si sí nació primero, hemos encontrado a una nueva **PersonaMayor** igual a la **Persona** actual. Esto se debe repetir hasta preguntarle a todas las personas.



Casi todos los algoritmos, como el que acabamos de mostrar, están

basados en preguntas condicionales, cuya respuesta es **SÍ** o **NO**. Un computador, procesa estos valores de verdad como **0** y **1** por lo que un algoritmo se puede pensar como una serie de instrucciones que leen y modifican los bits que estudiamos previamente.

3.1 Pensamiento algorítmico

Objetivos de aprendizaje

- Emplear un algoritmo de búsqueda de máximo en la vida real.
- Practicar el uso de condicionales y ciclos en contexto.
- Construir algoritmos para resolver problemas en la vida real.

Instrucciones

Se sugiere que esta actividad se realice en grupos pequeños de 4-5 estudiantes.

1. Ejecuten el algoritmo mostrado en la sección anterior para encontrar la persona mayor en su grupo.
2. Escriban los siguientes algoritmos como un diagrama (igual que el ejemplo de la persona más vieja).
 - a. Dependiendo de si hoy lloverá, sacar un paraguas.
 - b. Si me alcanza la plata, me compro un chocorramo. Si no me alcanza, me compro un menta.
 - c. Cuando llego a una fiesta, saludar a todas las personas que no he saludado.
3. Por grupos escojan uno de los siguientes problemas y escriban un algoritmo para solucionarlo. Piensen en cómo un computador pensaría para resolverlo de manera más eficiente.
 - a. Dado un número entero, descomponer en sus factores primos.
 - b. Ordenar un grupo de personas por sus edades.
 - c. Dadas dos palabras, determinar si son un anagrama.
 - d. Entre un número de personas, buscar a la que su grupo sanguíneo es 0-.

Limitaciones

Los científicos computacionales no sólo se encargan de pensar en algoritmos para resolver problemas utilizando computadores, sino que también se deben asegurar que estos algoritmos sean lo *más eficiente posible*. Esto, porque la idea es utilizar el menor tiempo y la menor cantidad de espacio posible.

En la sección anterior mostramos un algoritmo para encontrar la mayor persona de cierto grupo de personas el cual al ejecutarse para un grupo pequeño, no tarda más de un segundo en un computador. Ahora bien, ¿qué pasa si tenemos que encontrar la mayor persona dentro de 10,000 personas? ¿o de todo un país? Quizás ya no sea tan viable usar este algoritmo que se puede demorar hasta horas buscando la persona, sino que se debe buscar otro algoritmo que ataque el mismo problema de manera más eficiente.

Sin embargo, por más de que se busquen los algoritmos más eficientes, hay ciertos problemas que sencillamente son muy difíciles, o hasta imposibles de resolver utilizando un computador clásico. Ciertos de estos algoritmos se pueden resolver utilizando propiedades de la mecánica cuántica y su eficiencia aumenta significativamente. Estos algoritmos permiten la creación de nuevas tecnologías y tienen un gran número de aplicaciones, como las mostradas a continuación en la figura 4.1.

Si bien la computación cuántica resuelve un gran número de problemas y tiene importantes aplicaciones en distintas áreas de conocimiento, todavía está a varios años de ser completamente desarrollada a tal punto de reemplazar nuestras laptops por computadores cuánticos. Además, no significa que la computación cuántica reemplace por completo a la computación normal puesto que hay ciertos problemas para los cuales no existe ventaja de la primera. Por ejemplo, esta nueva ola de computación no significa que las películas en Netflix carguen instantáneamente pero como se puede ver en la siguiente página, sus aplicaciones y consecuencias si representan un avance importantísimo desde la ciencia y la tecnología [7].



(a) **Ciberseguridad.** Los sistemas actuales de seguridad informática, utilizan formas de encriptación que un computador normal tardaría un tiempo lo suficientemente grande para que sea insostenible descifrar. Sin embargo, un computador cuántico podría descifrar esta información en un tiempo significativamente menor, lo cual representa una preocupación a largo plazo [7]. Por esto uno de los retos más grandes en la computación es desarrollar criptografía que pueda ser indescifrable hasta para los computadores cuánticos[23] [4].



(b) **Creación de fármacos.** La creación de nuevos medicamentos (y materiales) es un problema de alta complejidad puesto que es necesario entender todas las posibles combinaciones en que los átomos se pueden enlazar. Los químicos buscan entonces simular computacionalmente estas reacciones químicas. Con un computador clásico, esta tarea se vuelve extremadamente demorada e insostenibles, sin embargo, como lo propuso Richard Feynman los computadores cuánticos podrían simular procesos cuánticos como reacciones a nivel molecular [12] y así llevar al descubrimiento de materiales y fármacos útiles en medicina y otras áreas [7] [26].

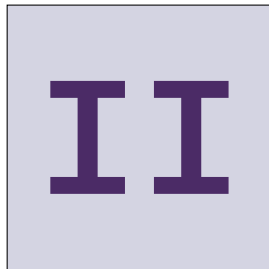


(c) **Finanzas.** En el sector de los bancos y finanzas también hay un fuerte interés por la computación cuántica. Esto es porque al calcular el puntaje crediticio (es decir, qué tan confiable es que un banco le preste dinero a una persona), entran muchos factores lo cual aumenta la complejidad del problema. Esto conlleva a que los bancos pongan ciertas restricciones que pueden resultar en errores: aceptar clientes que no pagarán sus deudas o rechazar clientes que pueden generar ingresos [7]. Este problema similar al de la simulación de la decisión de invertir en un portafolio de acciones requieren de niveles de computación más avanzada, para evaluar todos los posibles escenarios y escoger un óptimo [28] [22].



(d) **Inteligencia artificial.** Una de las aplicaciones de la inteligencia artificial es la de optimizar los procesos de manufactura de varios productos. Esto es porque utilizando análisis estadístico, se puede encontrar por qué ocurren posibles fallas en los procesos y así minimizarlos. Cuando los procesos tienen un gran número de fallas, es fácil hacer este análisis usando programación clásica pero cuando tienen un pequeño número de fallas, se requieren de computadores cuánticos. Este pequeño número de fallas es igual de importante puesto que para ciertos productos, pueden ser enormemente costosos [7]. Estos algoritmos utilizan métodos de frontera conocidos como aprendizaje automático cuántico [6].

Figura 4.1: Algunas aplicaciones para las cuales la computación clásica se ve limitada y se requiere el uso de la computación cuántica.



Computación cuántica



5 Qubits 23

5.1 Qubit con luz láser

6 Compuertas cuánticas..... 27

7 Entrelazamiento..... 31

7.1 Teleportación cuántica

Qubits

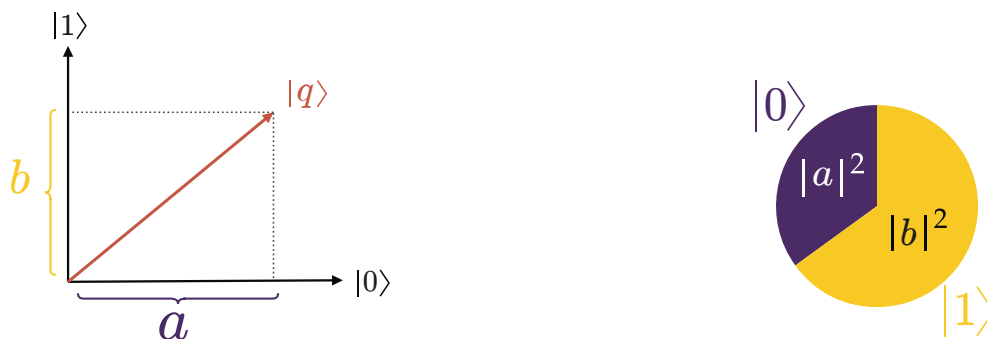
Ha llegado el momento de adentrarnos al mundo de la computación cuántica, teniendo en cuenta nuestras bases en computación clásica. La primera gran diferencia que encontramos es que la unidad básica de información de un computador cuántico se conoce como qubit (por *quantum bit* en inglés).

Definición

Un **qubit** es la unidad más básica de información cuántica. A diferencia de un bit, puede estar en estados diferentes a $|0\rangle$ y $|1\rangle$ pues puede estar en una combinación de estos. Esto se conoce como **superposición** y su estado se escribe

$$|q\rangle = a|0\rangle + b|1\rangle$$

No se deben dejar asustar por los símbolos $| \rangle$. Estos se conocen como *notación de Dirac* y es la forma en la que se escriben los estados de mecánica cuántica. En este caso, nuestro qubit está representado por el símbolo $|q\rangle$. Otra forma de entender esta combinación lineal es por medio de **vectores**, el cual es una cantidad que tiene magnitud y dirección. Esto lo podemos ver en la figura 5.2 donde la flecha roja se obtiene al moverse a unidades a lo largo del eje horizontal $|0\rangle$ y luego subir b unidades a lo largo del eje vertical $|1\rangle$.



(a) Los qubits también se pueden representar como vectores en un sistema de coordenadas $|0\rangle$ y $|1\rangle$.

(b) Los números $|a|^2$ y $|b|^2$ representan la probabilidad de que el qubit esté en estado $|0\rangle$ o en estado $|1\rangle$.

Figura 5.2: Representaciones de un qubit.

A diferencia de un computador clásico, no podemos saber si nuestro qubit está en $|0\rangle$ o en $|1\rangle$, sino que más bien podemos saber qué tan probable es que sea $|0\rangle$ o que sea $|1\rangle$. Para entender esto un poco mejor, podemos pensar en una marca de chocolates que produce chocolates sorpresa. Es

decir, cuando abrimos el empaque de nuestro chocolate, este puede ser o bien negro o blanco. La probabilidad de que sea negro, lo cual lo representaremos mediante $|0\rangle$ o blanco ($|1\rangle$) depende de estos números a y b como se muestra en la figura 5.2.

Los números a y b pueden también ser números imaginarios, pero ahora sólo pensaremos en números reales. Recuerda que un número imaginario es un múltiplo de la unidad imaginaria $i = \sqrt{-1}$.

La probabilidad de que esté en estado $|0\rangle$ está dada por la magnitud de este número al cuadrado, $|a|^2$ y la probabilidad de que esté en el otro estado será $|b|^2$. Recordemos que las probabilidades son proporciones que deben sumar todas a 100% por lo que tenemos la propiedad,

$$|a|^2 + |b|^2 = 100\% = 1$$

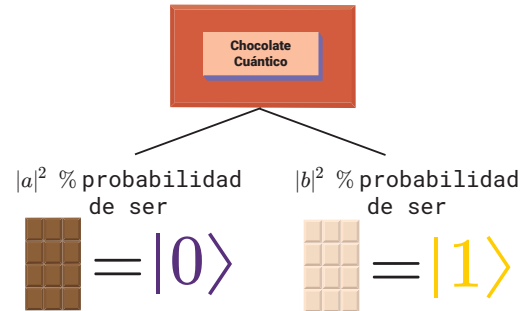


Figura 5.3: Nuestro chocolate tiene cierta probabilidad de

En la computación cuántica es muy común que se manipule la información alterando el estado de un qubit, es decir, cambiando las probabilidades y por ende los pedazos del gráfico de torta que se muestra en 5.2.

P Si $a=1$ y $b=0$, ¿cuál es la probabilidad de que el qubit esté en el estado $|0\rangle$? ¿y en el estado $|1\rangle$? ¿cómo se asemeja esto a un bit clásico?

Los qubits están hechos de algún sistema cuántico que tenga dos estados, uno de ellos se le pone el nombre de $|0\rangle$ y al otro $|1\rangle$. Cuando medimos el estado de nuestro qubit obtendremos o bien el estado $|0\rangle$ o $|1\rangle$, ¿qué pasó entonces con los números a y b ?

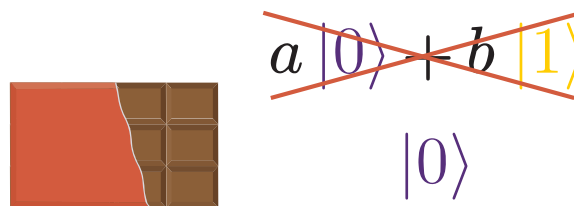


Figura 5.4: Cuando destapamos el chocolate, se altera el estado del qubit pues ya sabemos con certeza su color.

Resulta que una de las propiedades de la mecánica cuántica es que el acto de *medir*, cambia totalmente nuestro sistema. Para comprender esto, veamos la figura 5.4. Cuando destapamos nuestro chocolate, ya no tenemos incertidumbre de su color sino que sabemos con certeza que es o bien negro (como en la figura) o bien blanco. Esto significa que nuestro qubit se *colapsa* y perdemos la información de los números a y b . Gran parte de la magia de la computación cuántica ocurre sin destapar los chocolates.

5.1 Qubit con luz láser

Objetivos de aprendizaje

- Construir un montaje óptico
- Identificar distintos elementos ópticos y su uso.
- Probar los fotones como representación de un qubit.

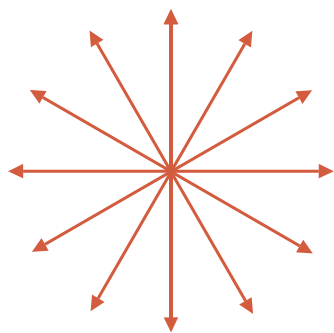
Materiales

- Láser
- Polarizador
- Divisor de haz polarizador

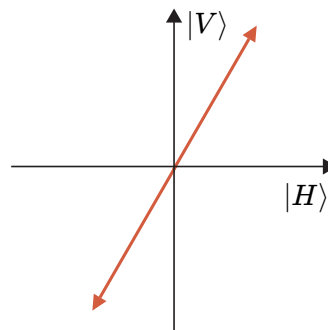
⚠ ¡Cuidado! Nunca pongas tus ojos a la misma altura del láser. Hacerlo podría tener consecuencias negativas para tu visión. Asegúrate de no tener anillos o elementos reflectivos al acercarte al montaje óptico. [31].

Descripción

Un qubit puede ser cualquier sistema cuántico que tenga dos estados. La partícula cuántica asociada con la luz se conoce como el **fotón** y podemos pensar que un rayo de luz láser está compuesto de millones de fotones. Una de las características de la luz es su polarización.



(a) La luz natural está compuesta de fotones vibrando en todas las direcciones.



(b) La luz polarizada en una dirección se puede descomponer en polarización horizontal $|H\rangle$ y vertical $|V\rangle$.

La luz se puede describir como una onda que está vibrando en todas las direcciones. Esta luz se puede **polarizar** para que vibre solamente en una dirección. Esta dirección se puede descomponer por componentes, los cuales llamamos *polarización horizontal* o $|H\rangle$ y *polarización vertical* o $|V\rangle$. Podemos representar estos estados como los estados de un qubit, $|H\rangle = |0\rangle$ y $|V\rangle = |1\rangle$.

Instrucciones

1. Antes de preparar nuestro montaje, debemos entender para qué se utiliza cada elemento. Relaciona cada elemento con su descripción.

Elemento

Láser ○

Descripción
Bloquea el paso de la luz en todas las direcciones excepto una.

Polarizador ○

Separa la luz en dos polarizaciones perpendiculares.

Divisor de haz polarizante ○

Fuente que produce un haz de luz.

2. El montaje para este experimento se encuentra a continuación. Ubica los nombres de los cuatro elementos.

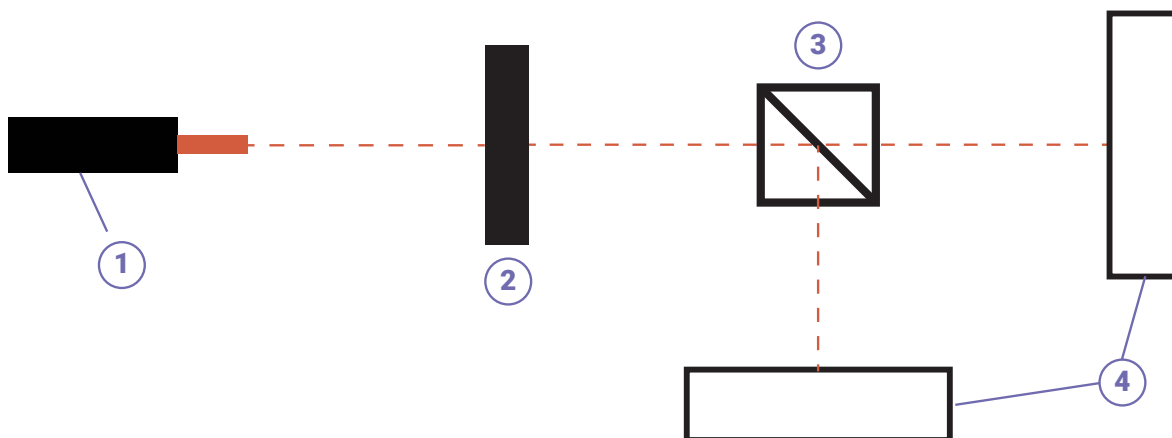


Figura 5.6: Montaje óptico.

3. Construye el montaje a partir de los bloques de construcción.

Preguntas orientadoras

1. Utiliza otro polarizador para bloquear la luz que llega al divisor de haz. ¿Cómo se puede usar esto para saber en qué dirección oscila la luz?
2. ¿Cómo sabemos en cual pantalla llega el haz horizontal, $|H\rangle$ y el vertical $|V\rangle$? ¿Esta elección es arbitraria?
3. Dibuja la vibración de la luz después de pasar por los puntos 1, 2, 4 del montaje en la figura 5.6. Puedes ayudarte de la figura 5.5a.

Compuertas cuánticas

Ahora explicaremos otras formas en las que los qubits se pueden representar que nos servirán para otras aplicaciones. Ya hemos visto en la figura 5.2 una forma de entender un estado como dos vectores, o un estado como las probabilidades de estar en $|0\rangle$ o $|1\rangle$. Ahora, también podemos representar estos qubits como una matriz. Una **matriz** es un arreglo rectangular de números que se escriben dentro de corchetes [2]. Por ejemplo, un qubit será una matriz de 2×1 puesto que tiene 2 filas y 1 columna, así

$$|q\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$

Esto significa que el número de arriba en la matriz representa el coeficiente de $|0\rangle$ y el número de la segunda fila representa el coeficiente de $|1\rangle$. Otra forma de representar los qubits es usando la *esfera de Bloch*. Esta es una esfera tridimensional (similar al globo terráqueo) donde el polo norte es el estado $|0\rangle$ y el polo sur es el estado $|1\rangle$. De esta manera, podemos dibujar un qubit $|q\rangle$ como se muestra en la figura 6.1.

Así como hicimos para los bits, estamos interesados en manipular y combinar los qubits para alterar su estado y llevar a cabo algoritmos. Esto lo haremos mediante **compuertas cuánticas** que cambiarán los estados de los qubits. Antes, una compuerta de un bit (como la **NOT**) recibía una sola entrada y tenía una sola salida. En este caso, una compuerta de un sólo qubit recibirá dos entradas (dadas por sus coeficientes de superposición de $|0\rangle$ y $|1\rangle$) y tendrá dos salidas. Esto muestra también que en un sólo qubit, podemos almacenar dos números de información mientras que en

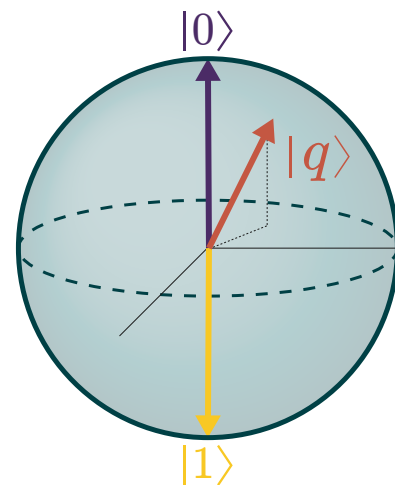
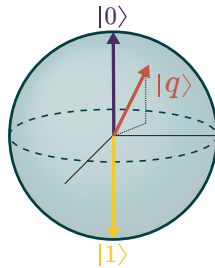


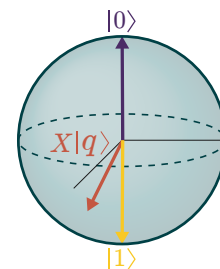
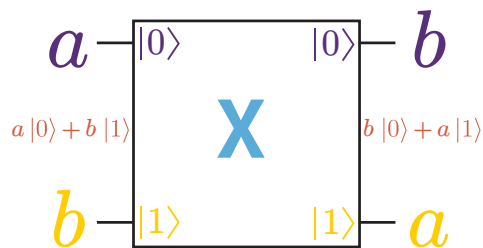
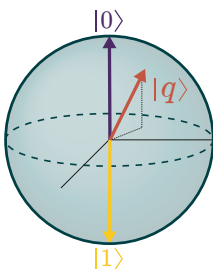
Figura 6.1: Representación de esfera de Bloch para un qubit

$$|q\rangle = \begin{bmatrix} a \\ b \end{bmatrix}$$



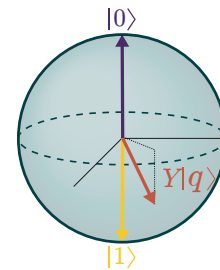
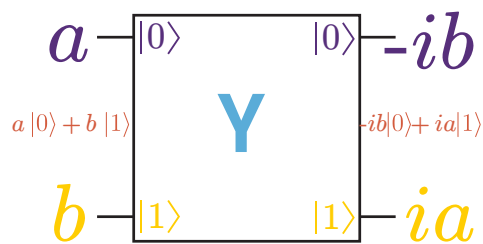
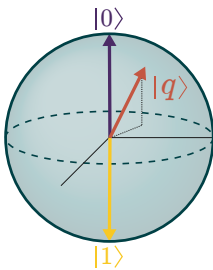
$$|q\rangle = a |0\rangle + b |1\rangle$$

Compuerta X



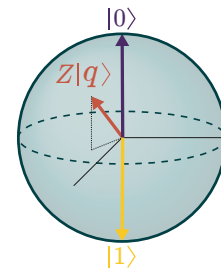
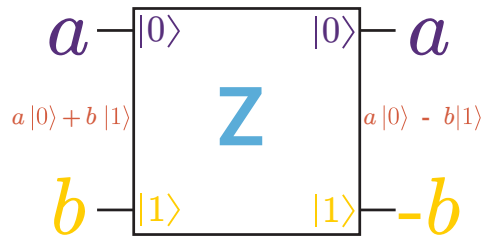
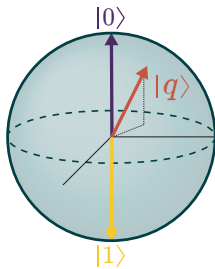
El efecto de esta compuerta sobre el qubit es $X |q\rangle = b |0\rangle + a |1\rangle$.

Compuerta Y



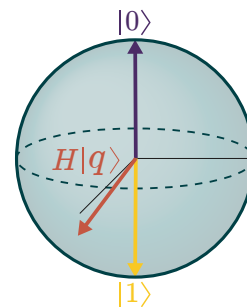
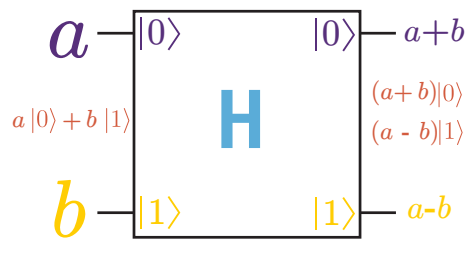
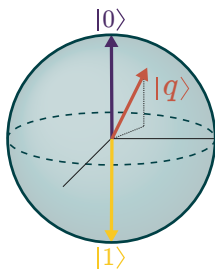
El efecto de esta compuerta sobre el qubit es $Y |q\rangle = -ib |0\rangle + ia |1\rangle$.

Compuerta Z



El efecto de esta compuerta sobre el qubit es $Z|q\rangle = a|0\rangle - b|1\rangle$.

Compuerta H



El efecto de esta compuerta sobre el qubit es $H|q\rangle = (a+b)|0\rangle + (a-b)|1\rangle$.

En general, existe un número de compuertas cuánticas, cada una de ellas que representan una rotación en la esfera de Bloch. Sin embargo, estas cuatro son las más usadas para un único qubit. Cuando aumenta el número de qubits que usamos, aumentan también las posibilidades puesto que nuestras bases ahora serán cuatro: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Nota que la base $|01\rangle$ indica el estado en el que el primer qubit está en estado $|0\rangle$ y el segundo en estado $|1\rangle$. También se puede tener una superposición de estos estados

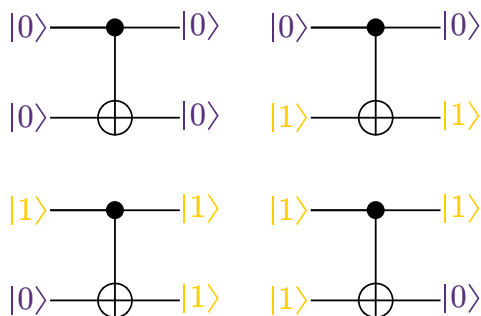
$$|q\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$



¿Cuántos estados base se tienen con un sistema de 3 qubits? ¿Y de n qubits?

También tenemos compuertas de dos qubits cuyo resultado son dos qubits también. La más importante se conoce como **CNOT**.

Compuerta CNOT



Esta compuerta realiza la siguiente operación: si el primer qubit está en el estado 1, cambia el segundo qubit. Por esta razón, al primer qubit lo llamamos **control** y al segundo **objetivo**.

Preguntas orientadoras

1. ¿Cómo describirías geoméricamente lo que le pasa al qubit con la compuerta X en la esfera de Bloch?
2. ¿Cuál es la probabilidad del qubit de estar en $|0\rangle$ o $|1\rangle$ después de las compuertas X y Y ? ¿Estas compuertas son iguales?
3. Calcula las probabilidades de obtener un vector $|0\rangle$ y $|1\rangle$ después de la compuerta de Hadamard si a es un número real y b un número complejo.
4. ¿Qué ocurre cuando aplicamos la compuerta Hadamard al qubit base $|0\rangle$? ¿Y al qubit base $|1\rangle$?
5. ¿Qué ocurre cuando aplicamos la compuerta **CNOT** al qubit $|q\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$?

Entrelazamiento

Llegó el momento de empezar a ver la magia de la computación cuántica y un claro ejemplo de su ventaja sobre su análogo clásico. Para esto, debemos adentrarnos al mundo del entrelazamiento cuántico.

¿Qué es?

Definición

El **entrelazamiento cuántico** es el fenómeno físico que ocurre cuando múltiples qubits están correlacionados. Es decir, la medición de uno de los qubits nos otorga información sobre el otro, sin importar que tan lejos estén el uno del otro.

La ventaja del entrelazamiento es que al tener dos chocolates, con sólo destapar el primero y ver su color, ya sabremos el sabor color del segundo. Veamos un ejemplo de un estado entrelazado para dos chocolates. En la figura 7.1, vemos que solamente existen dos estados para nuestro chocolate. Uno de estos es tal que nuestro primer chocolate es oscuro y el segundo blanco y el otro es tal que nuestro primer chocolate es blanco y el segundo oscuro. La ventaja es que con solamente medir uno de estos chocolates, ya sabremos de que color es el otro chocolate.

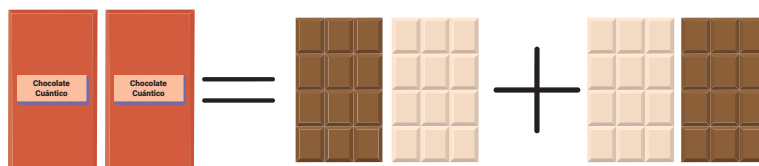



Figura 7.1: Un ejemplo de estado entrelazado

-  Si nuestros chocolates están en el estado de la figura 7.1 y destapamos nuestro primer chocolate en la figura 7.2, ¿cuál es el color del segundo chocolate? Podemos saber esto aún sin destapar el chocolate.

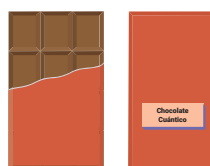


Figura 7.2: Con solo destapar un chocolate, ya podemos saber el color de su par entrelazado.

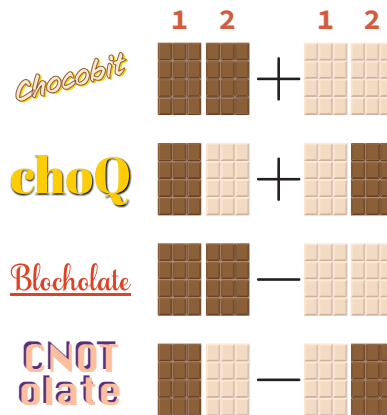


Figura 7.3: Cuatro maneras en las que nuestros chocolates pueden estar entrelazados. En cada una de estas, con sólo destapar nuestro primer chocolate, ya sabremos el sabor del segundo chocolate. Estas cuatro maneras corresponden a cuatro marcas de chocolate distintas.

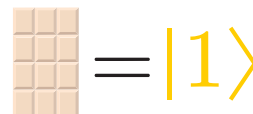
Existen cuatro formas en las que nuestro chocolate está entrelazado, los cuales se muestran en la figura 7.3. Cada marca de chocolate tiene distintos combos. Por ejemplo, si sabemos que dos chocolates son de la marca **Blocholate**, sabemos que ambos son oscuros o ambos son blancos. Nota que para cada uno de estos, con sólo saber de qué color es el primer chocolate, ya sabemos el color del segundo puesto que no hay más combinaciones posibles en este estado.

¿Cómo se hace?

Ahora veamos que debemos hacer para crear un estado entrelazado. Antes de eso debemos recordar un par de aspectos de nuestros qubits-chocolates. A continuación podemos ver la representación de qubits de nuestros chocolates.



(a) Nuestro chocolate oscuro será $|0\rangle$.



(b) Nuestro chocolate blanco será $|1\rangle$.

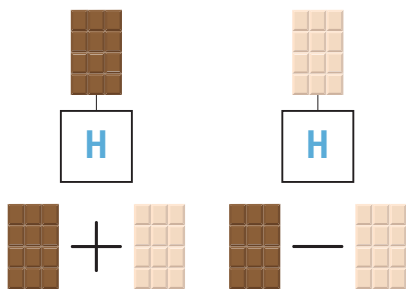


Figura 7.5: Resultado de pasar un chocolate negro o un chocolate blanco por una compuerta de Hadamard.

Recordemos que pasa cuando pasamos estos chocolates por nuestra compuerta de Hadamard como vemos en la figura 7.5. Cuando pasamos nuestro chocolate por la compuerta Hadamard, ya no podemos decir que es negro o blanco, sino que tiene cierta posibilidad de ser negro o blanco. Esto es lo que hemos visto antes como una *superposición de estados*. Ahora tenemos todas las herramientas para crear nuestro par entrelazado. Empezamos con un chocolate negro y uno blanco, el primero de esos lo pasamos por la compuerta de

Hadamard.

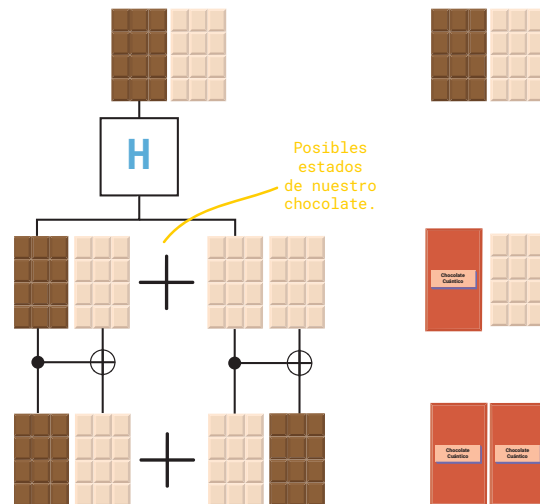


Figura 7.6: Creando chocolates entrelazados.



¿A qué marca corresponde el chocolate creado en la figura 7.6?

Empecemos con un chocolate negro y un chocolate blanco. Cuando pasamos nuestro chocolate negro por la compuerta Hadamard, creamos una superposición. Esto significa que ya no sabemos en que estado está este chocolate, por lo que lo podemos pensar como un chocolate empacado. Ahora, pasamos estos dos chocolates por una compuerta CNOT. Solamente si el primer chocolate es blanco, el segundo chocolate cambiará su estado. Pero como no sabemos con certeza si nuestro primer chocolate es negro o blanco, ahora tenemos incertidumbre sobre ambos entonces tenemos los chocolates empacados. Lo importante de este nuevo estado es que sólo existen dos posibilidades, que el primero sea negro y el segundo sea blanco, o que el primero sea blanco y el segundo sea negro. De esta manera, con sólo destapar uno de los dos, ya sabremos el estado del otro y los conocemos como estados *entrelazados*.

7.1 Teleportación cuántica

Objetivos de aprendizaje

- Utilizar el concepto de entrelazamiento cuántico en la teleportación cuántica mediante un ejemplo didáctico
- Formular un cambio de base para medir una propiedad indirecta de un sistema cuántico.
- Investigar un proceso de teleportación cuántica.

Materiales

- Chocolates

Descripción

Varias películas de ciencia ficción nos hablan de la teleportación, ¿cuántas veces no hemos soñado con llegar de un lado a otro instantáneamente? Pues la teleportación cuántica ya sucede actualmente. No de la forma en la que uno pueda mandar un paquete de un lado de la ciudad a otro, sino de la forma en la que uno puede teleportar información. Es decir, todavía no podemos irnos de vacaciones a Cartagena inmediatamente, pero si podemos mandar información de Armenia a Bogotá casi que inmediatamente.

Problema: El estado cuántico de un chocolate está determinado por sus coeficientes a, b . ¿Cómo hacemos para teleportar nuestro chocolate $|q\rangle = a|0\rangle + b|1\rangle$ desde Armenia hasta Bogotá de la forma más eficiente posible?

La solución a este problema está en la teleportación cuántica. En esta actividad mostraremos cómo es posible realizar esto utilizando las propiedades de la mecánica cuántica. El estado de nuestro chocolate está dado por $|q\rangle = a|0\rangle + b|1\rangle$ como se muestra en la figura 7.7.

Para lograr esto, necesitaremos un par entrelazado, el cual fabricaremos usando el procedimiento de la sección anterior.



Figura 7.7: Estado del chocolate que vamos a teleportar.

1. Describe los pasos para crear un estado entrelazado.
2. Si el primer chocolate es blanco en nuestro estado entrelazado, ¿cómo podemos determinar el color del segundo chocolate?

Una vez tenemos nuestro par entrelazado, dejaremos uno de los chocolates en Armenia y llevaremos el otro a Bogotá. Como se muestra en la figura 7.8.

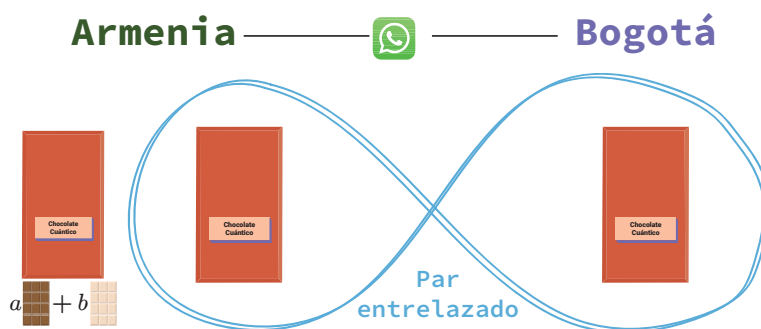


Figura 7.8: Diagrama de la teleportación. Queremos teleportar nuestro chocolate empacado en Armenia hasa Bogotá, usando nuestros pares entrelazados.

Esto significa que en total tenemos un chocolate $|q\rangle$ y un par entrelazado. Es importante aclarar en este punto que no conocemos el color preciso del chocolate que queremos teleportar ni del par entrelazado, pues hacer esto implicaría desempacarlos. De igual manera, nota que los primeros dos chocolates están en Armenia y el último está en Bogotá.


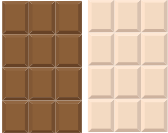

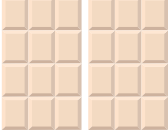
$$\begin{array}{c}
 \begin{array}{|c|} \hline \text{Chocolate Cuántico} \\ \hline \end{array} \left(\begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array} + \begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array} \right) \\
 \left(\begin{array}{|c|c|} \hline a & b \\ \hline \end{array} \right) \left(\begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array} + \begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array} \right)
 \end{array}$$

Si hacemos esta multiplicación, vemos que nuestro estado global es

$$a \begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array} + a \begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array} + b \begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array} + b \begin{array}{|c|c|} \hline \text{Armenia} & \text{Bogotá} \\ \hline \end{array}$$

Todavía no podemos ver de que color son los chocolates en Armenia, pero si podemos medir otra característica de estos, como por ejemplo su marca. Para esto, debemos primero escribir cada combinación de chocolates en Armenia como las marcas correspondientes. Observemos que la primera posibilidad de tener dos chocolates oscuros en Armenia es equivalente a sumar las marcas **Chocobit** y **Blocholate** de nuestra figura 7.3. Siguiendo está idea, completa la tabla siguiente:

1. ¿Cómo podemos escribir cada estado del chocolate y su empaque en términos de estas marcas? Completa la tabla:

	$\text{chocobit} + \text{Blocholate}$	
		

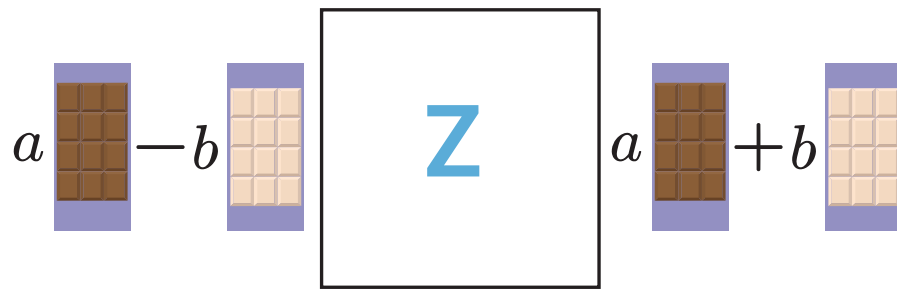
Cuando reescribimos nuestro estado global utilizando estas marcas tendremos:

$$\begin{aligned}
 & a \text{ chocobit} + \text{Blocholate} \\
 & + a \text{ choQ} + \text{CNOTolate} \\
 & + b \text{ choQ} - \text{CNOTolate} \\
 & + b \text{ chocobit} - \text{Blocholate}
 \end{aligned}$$

¿Qué sucede entonces si medimos la marca de los chocolates en Armenia como **CNOTolate**? Todas las demás marcas se anulan y terminamos con el estado

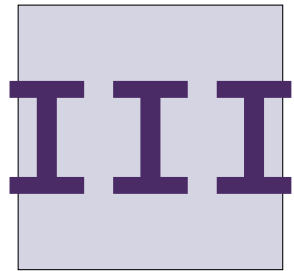
$$a \text{ CNOTolate} - b \text{ CNOTolate}$$

¿Cómo sabemos en Bogotá que la marca es **CNOTolate**? La única manera es que haya una comunicación no cuántica desde Armenia, por ejemplo via WhatsApp [24]. Esto es un ejemplo de como la computación cuántica no reemplaza totalmente la computación clásica, sino que se requieren ambas para lograr ciertos algoritmos. Una vez en Bogotá se sabe que el chocolate es de esta marca, ya podemos recrear nuestro chocolate inicial utilizando alguna compuerta cuántica, en este caso la compuerta Z puesto que:



1. Si le decimos al sujeto en Bogotá que nuestro chocolate es de marca ~~CNOT~~ olate, ¿cuál es el estado en el que está el sistema? ¿Qué compuerta(s) se debe(n) aplicar para volver al qubit original?
2. Si le decimos al sujeto en Bogotá que nuestro chocolate es de marca Blocholate, ¿cuál es el estado en el que está el sistema? ¿Qué compuerta(s) se debe(n) aplicar para volver al qubit original?

Otro punto importante es el momento en el que *medimos* la marca del chocolate, puesto que ahí alteramos el sistema, destruzándolo. No obstante, la medición indirecta nos permitió destruir el estado sin perder la información que queríamos transmitir originalmente, $|q\rangle = a|O\rangle + b|B\rangle$. Esta destrucción es importante puesto que nos permitió *teleportar* este estado, sin *clonarlo*.



Realizaciones físicas



8	Luz	41
9	Iones atrapados.....	43
10	Superconductores	45

Luz

A continuación nos dedicaremos a explorar algunas de las propiedades necesarias para la creación de un computador cuántico. Como hemos visto anteriormente, un qubit es un sistema cuántico de dos niveles. Sin embargo, este qubit puede ser alterado por medio de compuertas cuánticas, entonces físicamente necesitaremos algo que pueda alterar el estado de nuestro qubit. Por esta razón, necesitaremos poder preparar nuestros qubits en un estado inicial fijo, y no sólo que tengan un estado aleatorio debido a su naturaleza. Por último, también debemos poder medir el estado de este qubit si así lo deseamos [24]. Estas cuatro condiciones son cruciales para la búsqueda de un computador cuántico. No solamente tenemos que poder construir un qubit, sino también, el que sea el más fácil de manipular.

Uno de los posibles candidatos es la utilización de la óptica cuántica para la realización de un computador cuántico. Antes de entender cómo funciona, debemos preguntarnos algo más sencillo: ¿qué es la luz? En el siglo XX, se descubrió que la luz es una onda electromagnética. Ahora bien, la visión contemporánea explica que la luz también se puede entender como una partícula sin masa llamadas **fotones** [15].

Por ejemplo, la luz que viene del sol llega a nosotros como estas pequeñas partículas sin masa. Sin embargo, para varias aplicaciones, podemos tratar la luz sencillamente como una onda electromagnética [15]. Lo interesante de estas pequeñas partículas es que exhiben propiedades cuánticas que se pueden aprovechar para la computación cuántica.

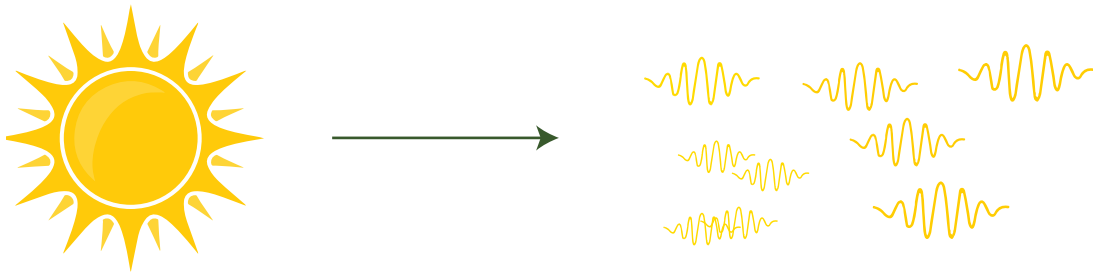


Figura 8.1: La luz en la naturaleza está compuesta de distintas partículas llamadas *fotones*.

Como hemos visto anteriormente, la luz puede tener una polarización dependiendo en qué dirección vibre. La polarización horizontal se refiere a $|H\rangle$ y se puede asociar con el elemento de la base $|0\rangle$ mientras que la polarización vertical se refiere a $|V\rangle$ y asociar con $|1\rangle$.

La polarización es una de las propiedades del fotón que podemos tratar como base cuántica. Sin embargo, también podemos utilizar otras de sus propiedades como característica cuántica. En un experimento, se tenían dos caminos diferentes por los que un sólo fotón podía pasar, el camino

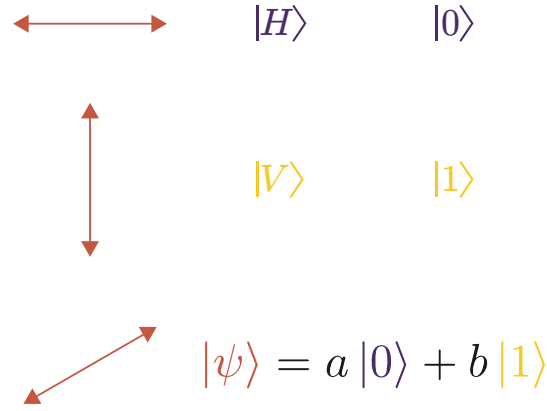


Figura 8.2: Los diferentes estados de polarización de la luz y de los fotones.

A y el camino B [19]. Esto implica que el camino por el cuál viaja el fotón también es una característica cuántica que se puede utilizar como qubit. En ese caso se tiene,

$$|q\rangle = a|A\rangle + b|B\rangle$$

De esta manera, se tienen dos posibilidades para utilizar como qubit con un sólo elemento: el fotón. La ventaja de la polarización es la utilización de distintos elementos que pueden cambiar fácilmente esta polarización, que pueden servir como compuertas cuánticas. Por ejemplo, en nuestro montaje experimental visto anteriormente el divisor de haz polarizante (PBS) sirve para dividir la polarización de la luz en dos, lo cual se puede tratar como una compuerta de Hadamard.

Ya hay experimentos que utilizan otros tipos de componentes ópticos como retardadores de onda para simular lógica cuántica [10]. Inclusive, se han preparado estados entrelazados de fotones que utilizan la polarización de los fotones y su posición espacial, es decir, en qué camino está [19]. Este estado enredado será

$$|q\rangle = \frac{1}{\sqrt{2}}(|A, H\rangle + |B, V\rangle)$$

Esto implica que sólo los fotones polarizados horizontalmente pasarán por el camino A y sólo los fotones polarizados verticalmente pasarán por el camino B . Si bien los fotones pueden parecer una buena alternativa para la computación cuántica, también tienen una desventaja: la dificultad de obtener un sólo fotón, en vez de una gran cantidad de estos, como lo que sale por un láser. [24].

Iones atrapados

Otra implementación física de la computación cuántica se basa en utilizar iones atrapados en un punto del espacio. Antes de entender su funcionamiento, debemos primero recordar el modelo del átomo. Como ya sabemos, un átomo cuenta con un núcleo que tiene neutrones y protones. El núcleo se encuentra rodeado de una nube de electrones. De forma similar a nuestro qubit, no podemos saber en dónde está este electrón, pero tenemos una probabilidad de que esté en esta nube.

Lo importante es que estos electrones están en diferentes niveles de energía alrededor del átomo. Como podemos ver en la figura 9.1, los electrones que están en estas nubes de probabilidad alrededor del núcleo están en distintos niveles de energía. El estado más pequeño de energía se conoce como el estado fundamental, y cualquier estado de energía por encima de este se conoce como un estado *excitado*.

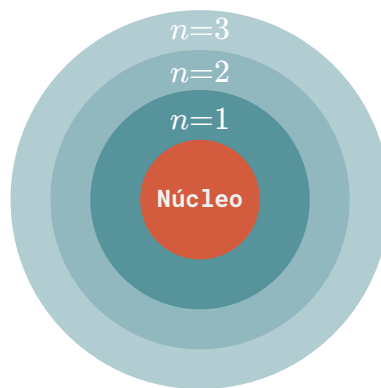


Figura 9.1: Niveles de energía de los electrones alrededor de un núcleo atómico. Estos electrones tienen una probabilidad de estar en estos niveles.

Estos electrones pueden saltar de un estado a otro cuando les damos energía en forma de luz [16]. En realidad podemos tener muchos estados excitados de energía pero también podemos limitarnos únicamente a los dos primeros estados: el estado **fundamental** y el primer estado **excitado**. De esta manera, tendremos nuestro sistema con dos propiedades. Usaremos el $|0\rangle$ para el estado fundamental y el $|1\rangle$ para el estado excitado, como se muestra en la figura 9.2.

Para esto tenemos un problema fundamental y es el hecho de que esta diferencia de energía es demasiado pequeña por lo que es difícil de observar y controlar. Esto es por que nuestros átomos están en movimiento constante y su energía de movimiento (**energía cinética**) es mucho mayor. ¿Qué pasaría entonces si logramos reducir esta energía de movimiento a su mínimo? De esta manera, la única energía que observaríamos y controlaríamos es la de su estado interno electrónico.

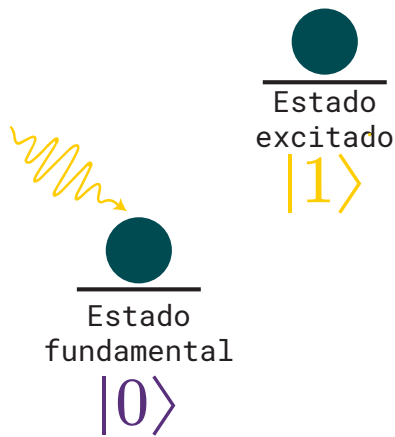


Figura 9.2: Cuando un electrón recibe luz, puede subir de energía a su estado excitado. El estado fundamental y el primer estado excitado pueden representar nuestro $|0\rangle$ y $|1\rangle$.

Esta idea de atrapar los átomos para que su velocidad sea cero, no es tarea fácil. En 1997, el premio Nobel fue otorgado a tres científicos que desarrollaron un método para atrapar átomos utilizando luz láser. La idea consiste en disparar a los átomos con luz hasta que eventualmente su velocidad es lo más cercana a cero posible [30]. Después de lograr que los átomos estén atrapados debemos lograr controlarlos. Para esto, se pueden utilizar láser o microondas para modificar los estados de estos átomos, haciendo que cambien de su estado fundamental a su estado excitado y viceversa [9].

Superconductores

Otra alternativa para la creación de un computador cuántico es el fenómeno de **superconductividad**. Para entender la superconductividad, debemos entender antes el flujo de electricidad. La corriente en un material conductor ocurre porque los electrones, pequeñas partículas que tienen carga negativa, viajan a cierta velocidad a través del material. Este flujo, sin embargo, a veces está restringido por el material.

Imaginemos que queremos viajar de un punto a otro en carro. Si la carretera no tiene huecos y está ordenada, este viaje se puede hacer de la manera más eficiente. Sin embargo, cuando hay huecos, el viaje es más demorado así vaya el mismo número de carros. Esto mismo pasa con los electrones dentro de un material pues cuando viajan a través de este encuentran *resistencia*.



(a) A temperatura ambiente, el material ejerce resistencia sobre el paso de electrones y hace que la conducción de electrones no fluya totalmente libre. (b) A temperaturas muy bajas, del orden de los 200°C , los electrones fluyen con resistencia cero y se da lugar a la superconductividad.

En 1911, un físico llamado Heike Onnes descubrió que a muy bajas temperaturas, esta resistencia desaparece. Casi como cuando en un día frío, la gente prefiere no salir de sus hogares y no hay trancón, en este caso los electrones pueden viajar libremente por el material. Esto da lugar a lo que conocemos como superconductividad [25]. La superconductividad es otro fenómeno físico de gran interés puesto que tiene varias aplicaciones, más allá de la computación cuántica como la creación de imanes de potencias muy grandes.

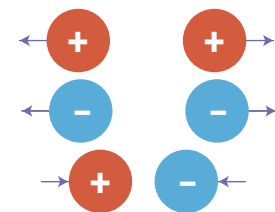
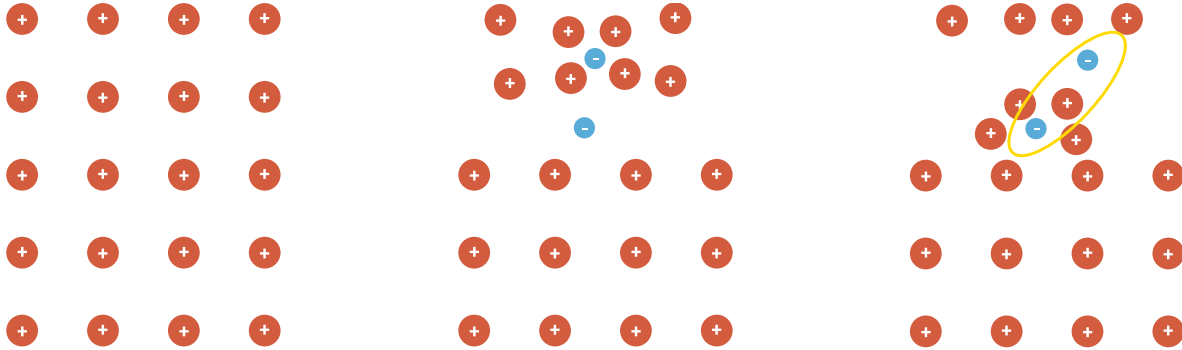


Figura 10.2: Representación de esfera de Bloch para un qubit

El funcionamiento de la superconductividad es el siguiente. Sabemos ya, del refrán popular¹ que las cargas iguales se repelen y las cargas opuestas se atraen como se puede ver en la figura 10.2. Podemos pensar entonces en un material como un arreglo de cargas positivas. Cuando

¹Los opuestos se atraen.

llega un electrón que quiere pasar por este material, todas estas cargas positivas se van a acercar a él, impidiendo su paso. Esto hace que las cargas negativas se desordenen de su arreglo de cuadrícula. Sin embargo, cuando llega otro electrón, las cargas van a querer acercarse a este. Esta interacción, hace que al final de cuentas sea más fácil para los pares de electrones moverse a través de este material. Estos pares de electrones se conocen como **pares de Cooper** y son el fenómeno que explica la superconductividad. [18].



(a) Un material superconductor se puede entender como un arreglo cuadrangular de átomos positivos.

(b) Cuando llega el primer electrón, los átomos positivos se acercan a él, deformando la cuadrícula. Esta deformación hace que sea más fácil para el segundo electrón atravesar el material.

(c) Cuando llega el segundo electrón, los átomos se acercan a este y le dan vía libre al primer electrón. Estos dos electrones se conocen como **pares de Cooper**.

Ahora, ¿cómo podemos unificar este cuento de la superconductividad con la computación cuántica? Estos pares de Cooper son en sí un sistema cuántico, entonces sólo nos hace falta definir los estados base. Para esto, los científicos desarrollaron el concepto de **isla superconductora**, que no es más que una región del espacio que puede o no tener un par de Cooper. Si en esta isla, hay un par de Cooper entonces se le asocia el estado $|1\rangle$ y si no, se le asocia el estado $|0\rangle$, como se muestra en la figura 10.4 [1]. A partir del par de Cooper, se desarrollaron los **transmones**, que son qubits superconductores que generan menos errores como que un qubit $|0\rangle$ se lea como uno $|1\rangle$ o viceversa. [20]. Uno de los retos más grandes en la construcción de estos computadores es la posibilidad de que en un algoritmo, un qubit cambie de estado, de $|0\rangle$ a $|1\rangle$ o viceversa. Esto puede producir que el algoritmo falle en su totalidad, por lo que asegurar está fidelidad de los qubits es de suma importancia.

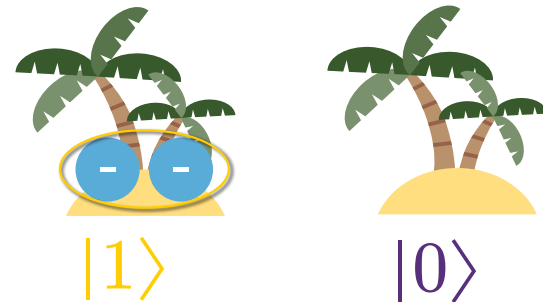


Figura 10.4: La base de los qubits es la presencia o ausencia de un par de Cooper en una isla superconductora.

Desde la década de 1980, los científicos han tratado de encontrar sistemas cuánticos con los que se pueda construir un computador cuántico a gran escala. Estas tres realizaciones físicas están entre los candidatos, pero siguen teniendo algunos problemas debido a su escalabilidad y la generación de ruido. El desarrollo de los computadores cuánticos se encuentra con el gran problema de que todos los fenómenos que permiten el aprovechamiento de características cuánticas como el entrelazamiento o la incertidumbre de medición, son precisamente muy difíciles de controlar. Los computadores cuánticos desarrollados actualmente se conocen como tecnología ruidosa de escala intermedia (o NISQ en inglés [29]) puesto que todavía son muy susceptibles a ruido aleatorio que impide la ejecución de ciertos algoritmos cuánticos correctamente. Estos computadores ya tienen entre 50–100 qubits y el sistema cuántico de preferencia son los superconductores, como veremos a continuación en los computadores cuánticos construidos por Google y IBM, dos compañías que están en la frontera de la innovación en este área [14].

Sycamore de Google

Desde 2016, Google ha desarrollado procesadores cuánticos que pretenden ejecutar algoritmos para resolver problemas en química y optimización. Estos hacen parte de la era NISQ (Noisy Intermediate-Scale Quantum). Su más reciente avance se conoce como el procesador **Sycamore**, un procesador cuántico que tiene 53 qubits. Los qubits utilizados son transmones, construidos a partir de circuitos superconductores y están dispuestos en un arreglo rectangular.

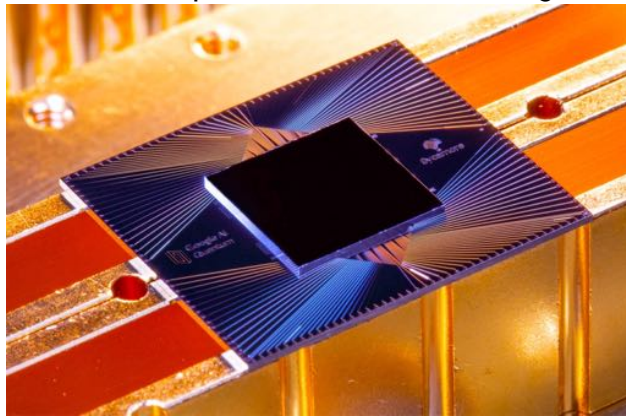


Figura 10.5: Procesador Sycamore de Google. Imagen tomada de [21]

Los circuitos diseñados se utilizaron para entrelazar un conjunto de qubits y posteriormente aplicar compuertas de un qubit y dos qubits. Lo anterior fue uno de los retos más grandes de los científicos en Google cuyo resultado final fue un circuito de 53 qubits, 1113 compuertas cuánticas de un qubit, 430 compuertas cuánticas de dos qubits con un error del 0.2%. Los investigadores estiman que un circuito de 50×10^{12} horas (alrededor de 400,000 veces la edad del universo [5]) tardaría 600 segundos [3].

Eagle de IBM

Otra compañía que se encuentra a la frontera de la investigación en computación cuántica es IBM. A la par de la creación e investigación en computadores cuánticos, esta compañía le apuesta a la simulación con el fin de que cualquier persona pueda ejecutar algoritmos cuánticos desde un computador clásico. Claramente, no con la misma eficiencia, pero en aras de avanzar la investigación de la programación cuántica.

IBM tiene unas metas considerablemente ambiciosas, pues pretenden para el 2025 construir un computador cuántico de más de 4000 qubits llamado *Kookaburra* [17]. Su más reciente avance fue en el 2021 y se trata de la construcción un procesador cuántico llamado *Eagle* el cual tiene una capacidad de 127 qubits. Al igual que el computador de Google, este utiliza transmones superconductores como qubits [11].

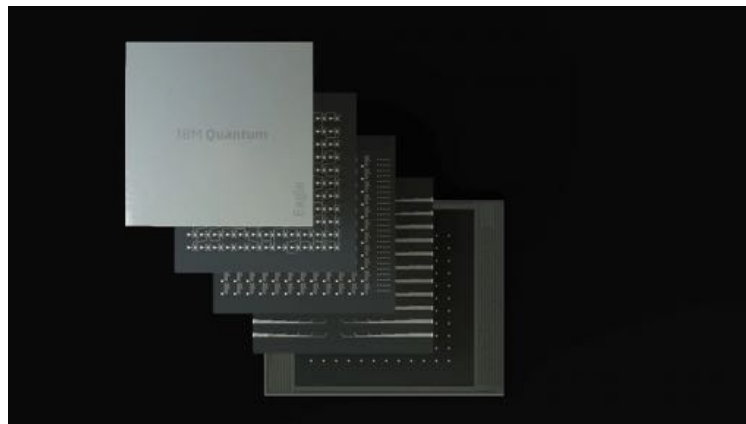


Figura 10.6: IBM Eagle. Imagen tomada de [11]

IV

Problemas cuánticos



11	Búsqueda	51
12	Criptografía cuántica	53

Búsqueda

El problema de buscar un elemento nos compete en muchas situaciones de la vida. Desde buscar las llaves de un carro hasta buscar un restaurante para ir que cumpla ciertas condiciones, todo el tiempo estamos realizando algoritmos de búsqueda. Por eso, la importancia de desarrollar un algoritmo de búsqueda eficiente que responda a esta problemática.

Consideremos por ejemplo el problema de buscar un restaurante adecuado dado ciertas restricciones:

- Debe tener opción vegetariana
- Debe estar cerca
- Debe tener opción de domicilio
- Debe estar abierto

Usando un computador clásico

Para resolver este problema usando un computador clásico, debemos tener una lista de restaurantes sobre la cuál vamos a buscar. Supongamos que tenemos 10 restaurantes, deberíamos preguntarle a cada restaurante si cuenta con estas características. Si por casualidad, el restaurante que cumple con estas características es el primero de la lista, no tendremos mayor problema. ¿Pero qué tal si el restaurante está de últimas y debemos preguntarle a 9 restaurantes antes? ¿Qué pasará si en cambio tenemos 10,000 restaurantes? A medida que el número de datos aumenta, la ejecución de este algoritmo se vuelve más insostenible para un computador clásico, por lo que aparece el algoritmo cuántico.

Usando un computador cuántico

Para un computador cuántico, podemos utilizar un algoritmo conocido como el **algoritmo de Grover**. En resumen, este algoritmo crea una superposición de todos los estados posibles. Para dos qubits, esta superposición sería

$$|s\rangle = |00\rangle + |01\rangle + |10\rangle + |11\rangle$$

Si nuestro elemento buscado está codificado por $|10\rangle$, denominamos un estado **correcto** y un estado **incorrecto** como

$$\begin{aligned} |c\rangle &= |10\rangle \\ |i\rangle &= |00\rangle + |01\rangle + |11\rangle \end{aligned}$$

Ahora ponemos nuestro vector de todos los estados de superposición en un plano formado por el qubit correcto y el qubit incorrecto.

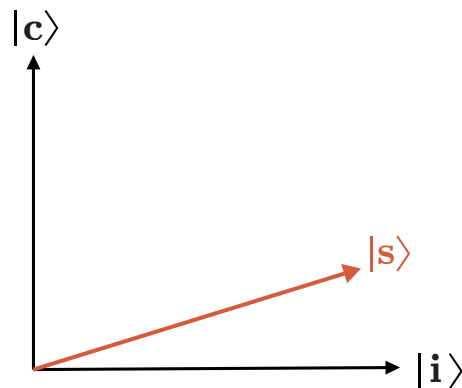
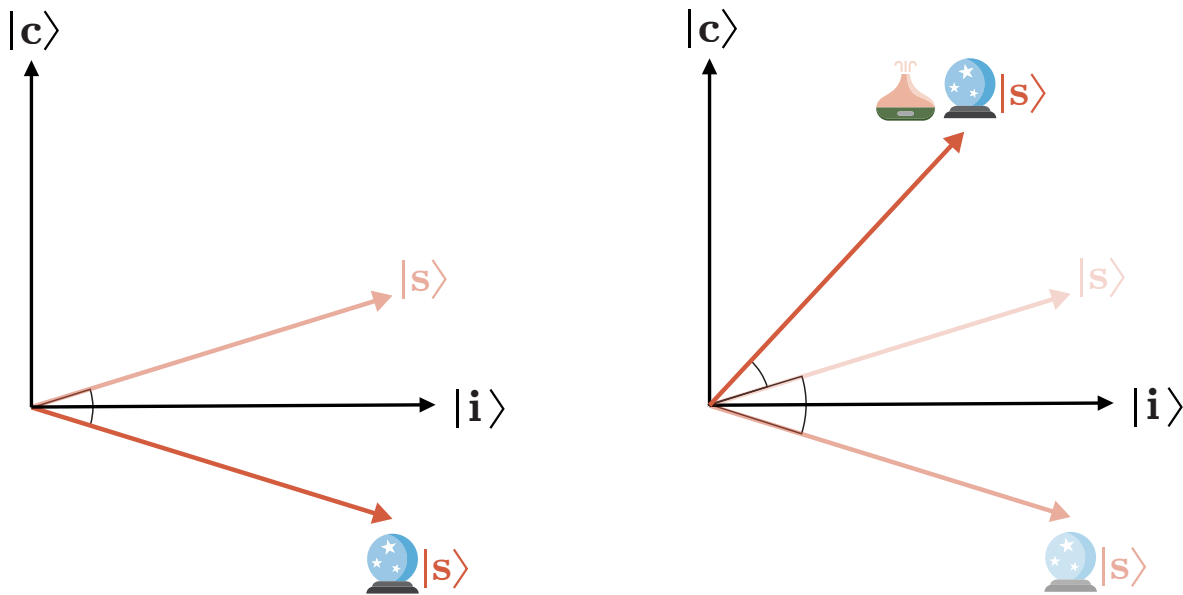


Figura 11.1: Estado de superposición $|s\rangle$ en plano de estado correcto $|c\rangle$ y estado incorrecto $|i\rangle$.

Ahora, usando dos compuertas cuánticas llamadas **oráculo** y **difusor**, cambiamos el estado de nuestro estado inicial para que se vaya acercando a nuestro estado correcto. Esto lo hacemos repetidas veces hasta que nuestros qubits se encuentren en el estado correcto, y así podemos saber cuál es.



(a) Cuando aplicamos la compuerta del oráculo, nuestro estado se refleja en el eje del estado incorrecto.

(b) Cuando aplicamos la compuerta del difusor, nuestro estado se refleja en el estado de superposición inicial, acercándonos al estado correcto.

La ventaja computacional es que si usando un computador clásico tenemos que buscar entre 100 elementos, usando el algoritmo de Grover sólo tenemos que buscar 10 veces. En general, para N elementos, debemos ejecutar el algoritmo \sqrt{N} veces [13].

Criptografía cuántica

El problema de la factorización

Quizás el problema de factorización puede parecer no tan importante. Después de todo, desde primaria hemos estudiado cómo factorizar un número como

$$24 = 2 \times 2 \times 2 \times 3$$

Un problema más grande puede ser el de factorizar un número que es un producto de dos números primos, como

$$33 = 3 \times 11$$



¿Cómo encontramos los factores de 221?

Por lo general, para un número que es el producto de dos primos, nuestra única opción es verificar si todos los números primos antes de ese número son factores. Por ejemplo, para 221 tendríamos que ver si 1 es factor, si 2 es factor, si 3 es factor, si 5 es factor, si 7 es factor y así sucesivamente. Este problema se vuelve exponencialmente grande cuando hablamos de números más grandes, por ejemplo para un número como 246,331 tendríamos que ver todos los números primos antes de ese. Sin embargo, producir un número así es bastante fácil, sólo es necesario multiplicar dos números primos cualesquiera como 787 y 313. Así mismo, es fácil verificar si este número es un producto de estos dos. En resumen tenemos

- Es fácil encontrar y/o verificar que un número es el producto de dos primos.
- Es exponencialmente difícil encontrar los factores primos de un número grande.

Si bien esto puede todavía parecer no tener aplicación en la vida real, este problema tan aparentemente sencillo como encontrar dos factores primos de un número puede volver vulnerable la seguridad informática de varios sistemas de internet, como los sistemas bancarios o las cuentas privadas.

Criptografía

Hoy en día vivimos en la era de la información y podemos consultarla con tan sólo un par de clics. Así mismo podemos compartir información con casi cualquier persona de manera muy sencilla. Ahora bien, seguramente no queremos que esta información esté disponible públicamente, sino que sólo le llegue a un receptor específico. La criptografía es precisamente la ciencia que estudia como mantener la información segura entre dos partes cuando hay riesgo de que exista un espía.

En particular, uno de los tipos de criptografía se conoce como la **criptografía de llave pública**. Supongamos que Bernardo le quiere mandar un mensaje a Alicia sin que nadie más lo lea. Alicia tiene dos llaves, una conocida como la llave privada y otra como la llave pública, las cuales están asociadas. Como Bernardo conoce la llave pública, la utiliza para encriptar el mensaje secreto para Alicia. Resulta que la única manera de desencriptar este mensaje es utilizando la *llave privada* que solamente posee Alicia. De esta manera, Alicia es la única persona que puede desencriptar el mensaje secreto.

La mayoría de sistemas de ciberseguridad funcionan con un protocolo de encriptación conocido como encriptación RSA. Su funcionamiento se basa en el hecho de que por más de que se tenga un número que es el producto de dos factores primos, es por lo general casi imposible encontrar estos dos factores primos. El primer número funcionaría como la *llave pública* y sus dos factores serían la *llave privada*. Si un espía quisiera desencriptar el mensaje secreto de Bernardo, debería encontrar los factores primos del número de la llave pública, lo cual como hemos visto previamente le tomaría a un computador normal miles de años [27].

En 1995, el matemático Peter Shor desarrolló un algoritmo cuántico que puede factorizar dos números cualesquiera en un menor tiempo, de tal manera que sea viable ejecutarlo en un computador cuántico. Si bien este algoritmo ya ha sido extensivamente estudiado, todavía no se ha podido llevar a cabo en un computador cuántico real si bien este algoritmo, conocido como *algoritmo de Shor*.

Si se logra llevar a cabo este algoritmo, muchos aspectos de la seguridad informática se verían comprometidos. Por ejemplo, los sistemas financieros de los bancos, los secretos militares, la información personal, entre otros, se volverían de acceso público. Por lo que las cuentas se podrían *hackear* muy fácil [27]. Esto se lograría utilizando el *algoritmo de Shor*, el cual usa computadores cuánticos para factorizar dos números primos.

Referencias

- [1] Amira Abbas et al. *Learn Quantum Computation Using Qiskit*. 2020. URL: <http://community.qiskit.org/textbook> (cited on page 46).
- [2] A.R. Angel, D.C. Runde, and V.C. Olguín. *Álgebra intermedia*. Ciudad de México, 2019. ISBN: 9786073248556. URL: <https://books.google.com.co/books?id=RXgJzwEACAAJ> (cited on page 27).
- [3] Frank Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (Oct. 2019), pages 505–510. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1666-5. URL: <https://doi.org/10.1038/s41586-019-1666-5> (cited on page 48).
- [4] J Barande. *Innovation and Research Symposium Cisco and Ecole Polytechnique 9-10 April 2018 Artificial Intelligence and Cybersecurity*. École Polytechnique, Apr. 2018. URL: <https://flickr.com/photos/117994717@N06/40631791164> (cited on page 20).
- [5] M. Bersanelli. "The Planck mission: From observations to cosmological parameters". In: *Fourteenth Marcel Grossmann Meeting - MG14*. Edited by M. Bianchi, R. T. Jansen, and R. Ruffini. 2018, pages 243–257. DOI: 10.1142/9789813226609_0015 (cited on page 48).
- [6] Jacob Biamonte et al. "Quantum machine learning". In: *Nature* 549.7671 (Sept. 2017), pages 195–202. DOI: 10.1038/nature23474. URL: <https://doi.org/10.1038/nature23474> (cited on page 20).
- [7] Francesco Bova, Avi Goldfarb, and Roger G. Melko. "Commercial applications of quantum computing". In: *EPJ Quantum Technology* 8.1 (Jan. 2021). DOI: 10.1140/epjqt/s40507-021-00091-1. URL: <https://doi.org/10.1140/epjqt/s40507-021-00091-1> (cited on pages 19, 20).
- [8] G. Brookshear and D. Brylow. *Computer Science: An Overview PDF eBook, Global Edition*. Pearson Education, 2015. ISBN: 9781292061801. URL: <https://books.google.com.co/books?id=jF1TDwAAQBAJ> (cited on pages 7, 11, 17).
- [9] Kenneth R. Brown et al. "Materials challenges for trapped-ion quantum computers". In: *Nature Reviews Materials* 6.10 (Mar. 2021), pages 892–905. DOI: 10.1038/s41578-021-00292-1. URL: <https://doi.org/10.1038/s41578-021-00292-1> (cited on page 44).
- [10] N. J. Cerf, C. Adami, and P. G. Kwiat. "Optical simulation of quantum logic". In: *Physical Review A* 57.3 (Mar. 1998), R1477–R1480. DOI: 10.1103/physreva.57.r1477. URL: <https://doi.org/10.1103/physreva.57.r1477> (cited on page 42).
- [11] Jerry Chow, Oliver Dial, and Jay Gambetta. *IBM Quantum breaks the 100 qubit processor barrier*. Nov. 2021. URL: <https://research.ibm.com/blog/127-qubit-quantum-processor-eagle?lnk=ushpv18nf2> (cited on page 48).
- [12] Richard P Feynman. "Simulating physics with computers". In: *International Journal of Theoretical Physics* 21.6 (1982), pages 467–488 (cited on page 20).
- [13] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. DOI: 10.48550/ARXIV.QUANT-PH/9605043. URL: <https://arxiv.org/abs/quant-ph/9605043> (cited on page 52).
- [14] Vikas Hassija et al. "Present landscape of quantum computing". In: *IET Quantum Communication* 1.2 (Dec. 2020), pages 42–48. DOI: 10.1049/iet-qtc.2020.0027. URL: <https://doi.org/10.1049/iet-qtc.2020.0027> (cited on page 47).
- [15] E. Hecht. *Optics, eBook, Global Edition*. Pearson Education, 2016. ISBN: 9781292096964. URL: <https://books.google.com.co/books?id=kv4yDQAAQBAJ> (cited on page 41).
- [16] Paul G Hewitt. *Conceptual physics*. Pearson Education, 2015 (cited on page 43).
- [17] IBM. *Our new 2022 Development Roadmap*. URL: <https://www.ibm.com/quantum-computing/roadmap> (cited on page 48).
- [18] Alan M. Kadin. "Spatial Structure of the Cooper Pair". In: *Journal of Superconductivity and Novel Magnetism* 20.4 (Mar. 2007), pages 285–292. DOI: 10.1007/s10948-006-0198-z. URL: <https://doi.org/10.1007/s10948-006-0198-z> (cited on page 46).

- [19] Yoon-Ho Kim. "Single-photon two-qubit entangled states: Preparation and measurement". In: *Physical Review A* 67.4 (Apr. 2003). DOI: 10.1103/physreva.67.040301. URL: <https://doi.org/10.1103/2Fphysreva.67.040301> (cited on page 42).
- [20] Jens Koch et al. "Charge-insensitive qubit design derived from the Cooper pair box". In: *Physical Review A* 76.4 (Oct. 2007). DOI: 10.1103/physreva.76.042319. URL: <https://doi.org/10.1103/2Fphysreva.76.042319> (cited on page 46).
- [21] Erik Lucero. *Photograph of the Sycamore processor*. Google. URL: https://1.bp.blogspot.com/-4pbQ6nBDyxY/XbC8MHKgtCI/AAAAAAAAAE10/wu0JGYKYZ-wyCUIQRTvYt2PGzCPKmhSrAClCBGAsYHQ/s1600/Google_Quantum_Nature_cover_art_Sycamore_device_small.png (cited on page 47).
- [22] Money Photo 3. CafeCredit.com, Sept. 2016. URL: <https://www.flickr.com/photos/cafecredit/29229864513/in/photolist-LwWGng-2jTLJ4p-23wd5fu-2jTLCP1-23wd5cU-yocWw-2jTGSsx-hweGeZ-9tZK5X-CzobYs-dCNVNL-EtfssF-DvwcT3-c19J85-c19JFf-c19JEC-c19JaW-c19Jao-ZULtKi-HjtVHk-St8PG1-TKGQqp-TKGQqZ-bsQzcH-5k4jRx-4dTkQ5-Kteq7f-23wgNHj-2frvtV-uAM6rG-qLdd5K-bgfJs6-gsjtfe-DunPKX-ErLmnB-ErLmFT-fZcYwr-dQAEAv-ErLmKk-snq5xC-28L6DZK-iYdhq-PL31F3-LwWF6i-r9zp1S-rqzfeq-m3kmC-s683JV-5tXw2w-4K5haA> (cited on page 20).
- [23] Dustin Moody et al. *Status report on the second round of the NIST post-quantum cryptography standardization process*. Technical report. July 2020. DOI: 10.6028/nist.ir.8309. URL: <https://doi.org/10.6028/nist.ir.8309> (cited on page 20).
- [24] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. American Association of Physics Teachers, 2010 (cited on pages 36, 41, 42).
- [25] H. Kamerlingh Onnes. "Further experiments with liquid helium. C. On the change of electric resistance of pure metals at very low temperatures etc. IV. The resistance of pure mercury at helium temperatures". In: *Through Measurement to Knowledge*. Springer Netherlands, 1991, pages 261-263. DOI: 10.1007/978-94-009-2079-8_15. URL: https://doi.org/10.1007/978-94-009-2079-8_15 (cited on page 45).
- [26] *Pastillas y fármacos*. PxHere, Nov. 2017. URL: <https://pxhere.com/es/photo/678861> (cited on page 20).
- [27] Anastasia Perry et al. *Quantum Computing as a High School Module*. 2019. DOI: 10.48550/ARXIV.1905.00282. URL: <https://arxiv.org/abs/1905.00282> (cited on page 54).
- [28] Marcos Lopez de Prado. "Generalized Optimal Trading Trajectories: A Financial Quantum Computing Application". In: *SSRN Electronic Journal* (2015). DOI: 10.2139/ssrn.2575184. URL: <https://doi.org/10.2139/ssrn.2575184> (cited on page 20).
- [29] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (Aug. 2018), page 79. DOI: 10.22331/q-2018-08-06-79. URL: <https://doi.org/10.22331/q-2018-08-06-79> (cited on page 47).
- [30] *The nobel prize in physics 1997*. URL: <https://www.nobelprize.org/prizes/physics/1997/summary/> (cited on page 44).
- [31] Frank Träger, editor. *Springer Handbook of Lasers and Optics*. Springer New York, 2007. DOI: 10.1007/978-0-387-30420-5. URL: <https://doi.org/10.1007/978-0-387-30420-5> (cited on page 25).