Escuela Politécnica Superior

Aplicación didáctica sobre criptografía y computación cuántica

Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Oscar Agudo Ruano

Tutor/es:

José Vicente Aguirre Pastor

Antonio Zamora Gómez

Universitat d'Alacant
Universidad de Alicante

Puedo afirmar sin miedo a equivocarme que

nadie entiende la mecánica cuántica.

Richard Feynman

Justificación y Objetivos

La física cuántica es algo fascinante y a la vez difuso. Sus aplicaciones en la computación y criptografía es lo que me ha llevado a realizar este trabajo. Los computadores cuánticos son todavía algo novedoso y que está en una fase muy temprana de su desarrollo, sería equivalente a los computadores clásicos de los años 70. Muy lejos aún de su versión definitiva.

El objetivo de este trabajo es mostrar a los estudiantes de ingeniería informática que existe esta tecnología, que está en desarrollo y que es el futuro de nuestro campo. Todo ello sin tener amplios conocimientos en física cuántica, de manera que quede lo más claro posible, abstrayendo al estudiante de fórmulas matemáticas, demostraciones de teoremas, etc. Ya que todo el contenido de este trabajo surge de lo que está siendo aplicado hoy en día en la computación y criptografía cuántica.

Índice

Índice de figuras

Índice de tablas

1. Introducción

Fue a principios del siglo XX que el físico Max Planck publicó su trabajo '*La teoría de la ley de distribución de energías del espectro normal'* [1] el cual revolucionaría la física de una manera que nadie se podría imaginar. En este trabajo, propuso la idea de que la radiación no es emitida de forma continua sino en cuantos de energía discreta, los fotones.

Planck no consiguió explicar por qué la energía se presentaba en paquetes, algo que sí consiguió Albert Einstein con su estudio sobre el efecto fotoeléctrico.

A pesar de esto, la física cuántica no fue desarrollada hasta años más tarde con físicos como Erwin Schrödinger, Werner Heisenberg, Paul Dirac y de Broglie, llegando a conclusiones sorprendentes sobre la materia, de las que para este trabajo nos interesan dos de ellas: la superposición de estados y el entrelazamiento cuántico [2].

"La superposición de estados es un principio fundamental de la mecánica cuántica que sostiene que un sistema físico tal como un electrón, existe en parte en todos sus teóricamente posibles estados de forma simultánea, pero, cuando se mide, da un resultado que corresponde a una y solo una de las posibles configuraciones" (Dirac 1958) [3]. Para entender este principio, Erwin Schrödinger ideó un experimento mental en 1935, el famoso gato de Schrödinger.

En este experimento se introducía a un hipotético gato en una caja sellada con un vial de gas venenoso y con un dispositivo de activación. Según los cálculos, en el momento de sellar la caja, el gato tenía las mismas probabilidades de seguir vivo que de estar muerto, esto es debido a que la física cuántica demuestra que las partículas pueden asumir diferentes estados al mismo tiempo. De esta manera, Schrödinger demostró que efectivamente el gato estaba vivo y muerto al mismo tiempo mientras la caja permaneciera sellada. Las partículas que constituían al gato asumirían uno de los dos estados, de forma aleatoria, y solo cuando el observador abriera la caja.

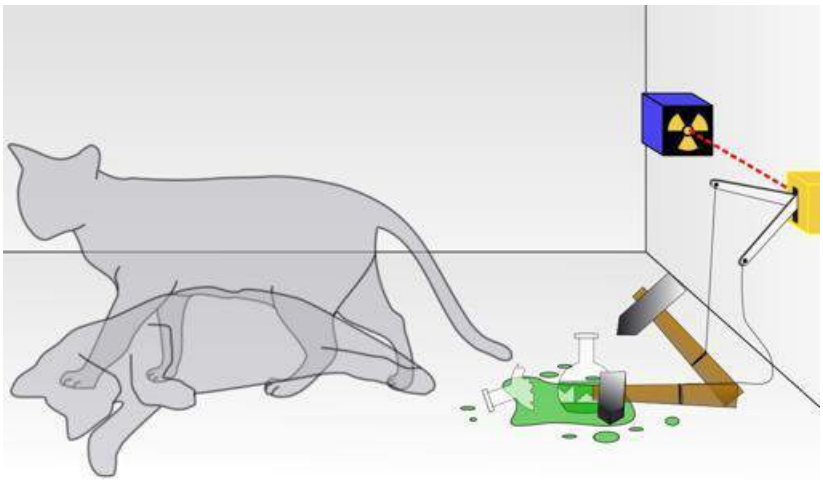

Figura 1: Representación del gato de Schrödinger

Por otro lado, el entrelazamiento cuántico fue postulado en 1935 por Albert Einstein junto a Boris Podolsky y Natham Rosen en un artículo conocido como EPR. En dicho artículo los físicos mostraron que, según la teoría cuántica, dos sistemas pueden estar conectados de manera extremadamente ligada, tal que una partícula puede influenciar el estado de otra instantáneamente, aunque estén separadas a años luz de distancia.

Pensemos en dos sistemas que guardan entre sí alguna relación, como por ejemplo haber colisionado tiempo atrás y más tarde han sido separados el uno del otro, el artículo EPR nos enseña que la mecánica cuántica predice que la medición de uno de los sistemas cambia el estado cuántico del otro en tiempo 0, es decir, inmediatamente y sin importar la distancia que hay entre ellas. Einstein, que en su teoría de la relatividad afirmaba que nada puede viajar más rápido que la velocidad de la luz llamó a este fenómeno "acción fantasmal a distancia" [4].

Para entender por qué nos interesan estas propiedades debemos pensar en cómo funciona un ordenador clásico. En éste, la unidad básica de información es el bit, el cual puede tener dos estados posibles, 0 o 1, con los que podemos realizar diferentes operaciones lógicas con una combinación de las llamadas puertas lógicas. De esta manera un ordenador clásico puede operar con n estados distintos, donde n es el número de bits, mientras que un computador cuántico opera con 2^n estados a la vez.

"Cuando analizamos las unidades fundamentales de la realidad, las que componen todo a nuestro alrededor, ya no debemos pensar en estas unidades como fragmentos de energía o materia, sino que deberíamos pensar en ellas como unidades de información, ya que según la mecánica cuántica no se puede decir que algo exista o no a no ser que se haya realizado

una medición, por tanto, las unidades de información son lo que crean la realidad, no las unidades de materia o energía" (Vedral, 2010) [5].

Pero no fue hasta 1981 que Paul Benioff [6] propuso la idea de una computadora cuántica, más tarde apoyado por Richard Feyman que planteó que una computadora que se basase en las leyes de la mecánica cuántica, podría hacer cálculos complejos de manera más rápida.

Un computador cuántico, parte de la base de que un sistema pueda estar en varios estados a la vez, haciéndolo más eficiente que un computador clásico. Un computador actual trabaja con bits, que pueden tener máximo 2 estados, 0 o 1, mientras que los ordenadores cuánticos trabajan con qubits, los qubits pueden tener una superposición que les permite adoptar 2 valores simultáneamente, esto permite realizar múltiples tareas a la vez, como descifrar de una manera extremadamente veloz los factores de un número muy grande, búsqueda en miles de datos o simular sistemas tan complejos como es el clima.

Es gracias a estas propiedades lo que hace que la computación cuántica sea a la vez peligrosa si no lo tratamos adecuadamente y adoptamos unos nuevos estándares de seguridad y un gran avance en la historia de la humanidad, ya que con la potencia de estos computadores cuánticos seremos capaces de realizar grandes avances en medicina, economía, prevención de catástrofes climáticas, etc.

Si bien es cierto que la idea surgió en la década de los 80 y se empezó a desarrollar en los 90, y hemos hecho un gran avance hasta la fecha, aún nos queda un largo camino hasta alcanzar la supremacía cuántica. La supremacía cuántica se alcanzará cuando un computador cuántico pueda resolver un problema que un supercomputador actual no pueda resolver o tarde demasiado tiempo en resolverlo.

Por el gran potencial de los computadores cuánticos es por lo que se han desarrollado algoritmos cuánticos que veremos más adelante, algoritmos que teóricamente resuelven un problema en horas e incluso minutos y del que un supercomputador actual requeriría cientos o miles de años. Algoritmos que ponen en peligro la seguridad actual de la criptografía, con lo que cualquier contraseña y cualquier medio criptográfico quedaría obsoleto. Es por ello que se han desarrollado nuevos protocolos de seguridad para que cuando llegue el día de que la computación cuántica alcance esa supremacía, estemos preparados y seguros para una nueva era en la historia de la computación.

A lo largo de este trabajo conoceremos mejor cómo es y cómo funciona un computador cuántico, en qué se diferencia de uno clásico, una serie de algoritmos cuánticos a tener en cuenta y los lenguajes de programación más importantes actualmente para desarrollar esos algoritmos. Por parte de la criptografía, veremos también en qué se diferencia la criptografía cuántica de la actual, los sistemas de seguridad cuánticos, algunos de los protocolos más importantes y algunos de los principales ataques a sistemas cuánticos. Por último, tendremos una visión general sobre las puertas cuánticas que permiten a un computador cuántico realizar operaciones, el hardware que requiere un ordenador cuántico y los sistemas operativos desarrollados para los computadores cuánticos.

Toda esta información ha sido plasmada en una aplicación web didáctica, donde se mostrará al usuario en un entorno amigable de la manera más clara posible y donde el usuario podrá finalmente poner a prueba los conocimientos adquiridos en la materia.

2. Estado del arte

2.1. Computación

La computación es la ciencia que trata el estudio de las computadoras, incluyendo su diseño, las operaciones que puede realizar y su uso en el procesamiento de los datos. También se puede añadir, que es una tecnología que permite estudiar el tratamiento de la información a través de máquinas automatizadas.

Desde las primeras calculadoras mecánicas diseñadas en el siglo XVII hasta los años 40, se han inventado infinidad de máquinas y computadoras mecánicas, analógicas o electrónicas que han intentado mejorar y acelerar la precisión de los cálculos [7].

En 1840 el matemático Charles Babbage diseñó el motor analítico, el cual se trataba de una máquina de calcular programable mediante tarjetas perforadas. Esta máquina estaba diseñada para trabajar en base 10 y permitía que sus cálculos realizaran saltos condicionales y bucles. Más tarde, Ada Lovelace, hija del poeta lord Byron, siguió con el trabajo de Babbage y en 1843 publicó el trabajo "*Sketch of the analytical engine invented by Charles Babbage*" [8] que describe el motor analítico y construye un ejemplo completo de cómo hacer que la máquina produzca la secuencia de los números de Bernoulli. Considerando a Ada Lovelace como la primera programadora de la historia [9].

Más tarde, en 1936 Alan Turing formalizó la idea abstracta de computador, haciendo uso de un modelo de procesamiento muy sencillo: una máquina abstracta con un escáner que lee y escribe ceros y unos de una cinta infinita (lo que hoy llamamos memoria) y se mueve y los escribe en función de una tabla definida en la máquina (programa).

En 1945 John Von Neumann, el cual trabajaba en la construcción del ENIAC, propuso su famosa arquitectura en la que se proponen las dos ideas claves de los ordenadores de propósito general: un programa que es almacenado en memoria y una serie de instrucciones de procesamiento que incluye un direccionamiento indirecto [10].

Fue en 1998 cuando nació la primera máquina de 2 qubits, esta fue presentada en la Universidad de Berkeley, California. Un año más tarde, IBM diseñó la primera máquina de 3 qubits que fue capaz de ejecutar por primera vez el algoritmo de búsqueda de Grover.

El objetivo de IBM para 2021 es presentar una máquina de 127 qubits, para 2022 una de 443 y para 2023 pretenden desarrollar una computadora cuántica de nada menos que 1121 qubits. Este objetivo es muy ambicioso, teniendo en cuenta que el ordenador cuántico más potente en la actualidad (2020) tiene 65 qubits [11].

2.1.1. Computación clásica vs cuántica

Las diferencias entre un ordenador clásico y un ordenador cuántico son más que evidentes a simple vista, pero más allá del hardware y software utilizados que veremos más adelante, nos centraremos en los procesos internos que hacen que un computador pueda realizar operaciones lógicas.

Como ya sabemos, la unidad mínima de información que utilizan los ordenadores clásicos es el bit. Un bit es un sistema binario que puede adoptar uno de dos valores, 0 o 1. Esto se traduce en el paso de electricidad por un transistor. Si el voltaje es alto se representa en 1, mientras que si es más bajo se representa en 0. De esta manera, haciendo uso del álgebra de Boole mediante puertas lógicas, un ordenador clásico es capaz de realizar una gran cantidad de operaciones lógicas que nos permiten manejar los ordenadores tal como los conocemos.

En cambio, en la computación cuántica no hacemos uso del bit tradicional como unidad mínima de información, sino que nos encontramos con el bit cuántico o qubit. A diferencia del bit binario en el que solo podemos tener uno de los estados, el qubit puede adoptar uno de esos valores o una superposición de ambos valores a la vez, de esta manera tenemos que un qubit puede ser 0, 1 o 0 y 1 a la vez, superponerse y entrelazarse según las leyes de la física cuántica. Esto hace que los qubits a diferencia de los bits, puedan tomar varios valores a la vez y por tanto desarrollar cálculos que no puede realizar un ordenador convencional [12].

Si bien, como hemos visto antes los bits emplean las puertas lógicas, éstas no nos sirven para un sistema no binario, por ello los computadores cuánticos emplean puertas cuánticas que también veremos más adelante.

Para hacernos una idea de lo que es capaz un ordenador cuántico, mientras que un ordenador convencional de n bits opera con n bits a la vez, un ordenador cuántico de n qubits opera con

2^n. Es decir, mientras que el bit crece con la proporción lineal, el qubit crece con la proporción 2^n.

Por ejemplo, si tenemos un sistema de 20 bits, se entregan 20 resultados, pero con 20 qubits tendremos 1.048.576 resultados (2^{20}). Con estas propiedades, se estima que un computador cuántico de 600 qubits podría llegar a realizar cálculos que contengan la información de todos los átomos del universo en cuestión de segundos [13].

Otra de las principales diferencias que tiene el qubit respecto al bit es que cada uno de sus procesos es independiente. Es decir, que mientras que en la computación clásica la resolución de un problema es lineal, en la computación cuántica se puede resolver más de una operación al mismo tiempo.

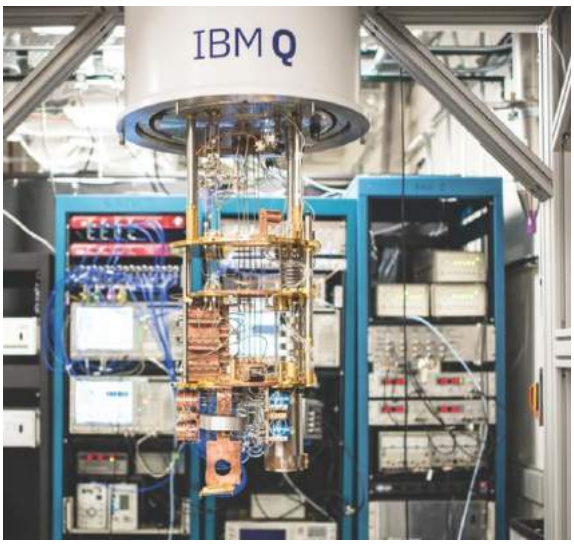

Figura 2: IBM Q System One

2.1.2. Qubits

Ya hemos visto que la unidad mínima de información en los computadores cuánticos es el bit cuántico o qubit, lo equivalente al bit en los computadores convencionales. Pero si bien sus propiedades los hacen muy superiores al bit, la creación y medición de esos qubits es lo que hacen que este sea un sistema mucho más complejo.

Grandes compañías como IBM o Google utilizan circuitos superconductores enfriados casi al cero absoluto, otras compañías como IonQ, gestionan los qubits atrapando átomos individuales en campos electromagnéticos mediante chips de silicio en cámaras de ultra vacío. En los dos casos, el objetivo es aislar los qubits en un estado cuántico controlado para su medición.

Aunque existen varios esquemas con los que podemos describir los estados de un sistema cuántico, el más conveniente para este trabajo es la notación de Dirac, que se ha convertido en un estándar en la física moderna, en esta notación un estado es representado por un vector ket indicado por $|\psi\rangle$.

Los dos estados básicos de un qubit son $|0\rangle$ y $|1\rangle$, pero también puede encontrarse en una superposición de ambos $|\psi\rangle = \alpha|0\rangle + \beta\,|1\rangle$, donde α y β son amplitudes de probabilidad complejas.

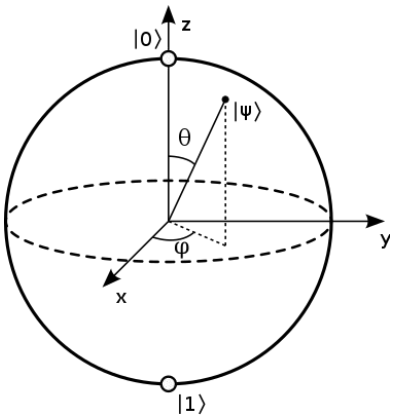

Figura 3: Representación tridimensional de un qubit en función de la esfera de Bloch

Un qubit se puede representar en una esfera de Bloch para visualizar su comportamiento de manera más clara. En la figura 1 podemos observar la línea representada con $|\psi\rangle$ que será la que indique el estado del qubit, cuanto más arriba se encuentre, más relevante será el estado $|0\rangle$ y cuanto más abajo esté, más relevante será el estado $|1\rangle$. Pero eso no es todo, ya que las combinaciones también pueden ser negativas e incluso números complejos, lo que aumenta infinitamente las posibles configuraciones de un qubit. Aunque todo esto parezca complejo, se sigue manteniendo lo anterior, cuanto más arriba apunte la fecha, más relevante será el estado $|0\rangle$ y cuanto más abajo apunte la fecha, más relevante será el estado $|1\rangle$.

A pesar de que un qubit puede estar en infinitas superposiciones distintas, observarlas es complicado, ya que en el mundo cuántico medir es una operación que perturba el estado inicial y obliga al qubit a apuntar a uno de los dos estados básicos.

Si medimos una serie de qubits de la misma manera, observaremos que algunas veces nos dan el resultado $|0\rangle$ y otras veces el resultado $|1\rangle$, esto es debido a que cuanto más relevante sea uno de los estados en superposición, más probable es que se acabe observando.

La probabilidad exacta de medir cada estado es el módulo cuadrado del coeficiente que le acompaña, donde $|\alpha_0|^2 + |\alpha_1|^2 = 1$, es decir, si tenemos la superposición de estados con este resultado:

$0.81|0\rangle + 0.58|1\rangle$, la probabilidad de medir $|0\rangle$ será de $|0.81|^2$ y la probabilidad de medir $|1\rangle$ será de $|0.57|^2$.

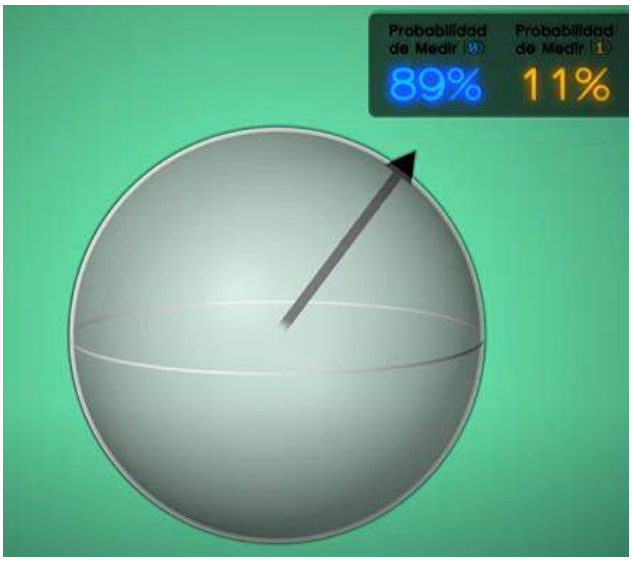

Figura 4: Representación de probabilidades de colapso del estado de un qubit a un estado básico

De esta manera tenemos que, si la flecha apunta totalmente para arriba, con total seguridad cuando midamos nuestro qubit obtendremos el valor $|0\rangle$ y cuando la flecha esté totalmente para abajo, obtendremos el valor $|1\rangle$. Pero en cambio, si la flecha apunta a cualquier otra dirección, el resultado será totalmente aleatorio. Esto es debido a que el estado natural de qubit no está determinado ni en $|0\rangle$ ni en $|1\rangle$, sino que es una propiedad que está indefinida, pero cuando medimos el qubit le obligamos a que nos dé un valor y el qubit se proyecta en el eje y se pierde la superposición.

Esto ocurre cuando medimos en el eje Z, pero sin embargo cuando medimos en el eje Y de nuestra figura 1 ocurre lo mismo, pero en vez del estado $|0\rangle$ y $|1\rangle$ obtenemos los estados $|\leftarrow\rangle$ o $|\rightarrow\rangle$. En términos de $|0\rangle$ y $|1\rangle$, la derecha corresponde a sumar equitativamente $|0\rangle$ y $|1\rangle$ y la izquierda corresponde a restar equitativamente $|0\rangle$ y $|1\rangle$.

Imaginemos que queremos medir un qubit, primero lo medimos con respecto al eje Y y obtenemos el estado $|\rightarrow\rangle$, pero como este estado es una mezcla perfecta de $|0\rangle$ y $|1\rangle$, tenemos la misma probabilidad de que se proyecte en uno u otro.

De esta manera si medimos un qubit primero en el eje Y y luego en el eje Z no obtenemos el mismo resultado que si medimos primero el eje Z y luego el eje Y, aquí nos encontramos con el principio de incertidumbre de Heisenberg [14].

Fue Benjamin Schumacher quien descubrió en 1995 una forma de interpretar los estados cuánticos como información. Se le ocurrió una forma de comprimir la información en un estado y almacenar la información en un número menor de estados. Esto se conoce hoy en día como la compresión Schumacher, que sería el equivalente al teorema de codificación sin ruido de Shannon de los computadores clásicos [15].

2.1.3. Algoritmos

Ya hemos visto las sorprendentes cualidades de los qubits, ahora veremos cómo se han aplicado esas propiedades para crear una serie de algoritmos que pueden resolver problemas que los supercomputadores clásicos no son capaces de resolver.

Para este trabajo analizaremos 3 de los algoritmos más importantes creados hasta la fecha: El algoritmo de Shor para resolver el problema de la factorización de números primos, el algoritmo de Grover para la búsqueda en una secuencia no ordenada de datos y el algoritmo Deutsch-Jozsa para determinar si una función es constante o balanceada.

2.1.3.1. Algoritmo de Shor

Peter Shor propuso en 1994 un algoritmo cuántico para resolver de modo eficiente tanto el problema de la factorización de un número entero como el problema del logaritmo discreto.

Este algoritmo implica que la criptografía de clave pública podría romperse fácilmente, dada una computadora cuántica lo suficientemente grande.

Un mensaje cifrado con RSA puede ser descifrado descomponiendo en factores la llave pública N, que es el producto de dos números primos grandes. Los algoritmos clásicos conocidos no pueden hacer esta descomposición en tiempo O $((\log N)^k$ para ningún k, por lo que son poco prácticos a medida que se aumenta N. El algoritmo de Shor puede romper RSA en tiempo polinómico.

Como todos los algoritmos cuánticos, el algoritmo de Shor es probabilístico. Puede dar la respuesta correcta con alta probabilidad y la probabilidad de fallo se reduce repitiendo el algoritmo.

Dado un número entero N, tratamos de encontrar otro número entero p entre 1 y N que divide N.

El algoritmo de Shor está compuesto por dos partes.

- La primera convertimos el problema de encontrar un factor propio de un número en el problema de encontrar el periodo de una función, esto se puede implementar clásicamente y sin necesidad de la computación cuántica.
- La segunda parte trata de encontrar el período, usando la QFT (transformada de Fourier cuántica), y este paso es el responsable de la aceleración cuántica.

1. Elegir aleatoriamente a entre 1 y N-1
2. Si mcd(a,N)$\neq$1, devolver mcd(a,N)
3. Determinar t, tal que $a^t \equiv 1 \bmod N$
4. Si t es impar devolver fallo
5. Si mcd($a^{t/2}$+1,N) $\neq N$, devolver mcd($a^{t/2}$+1,N)
6. Devolver fallo

Figura 5: Algoritmo para encontrar un factor propio

Si el algoritmo no falla, devuelve un factor propio de N.

El paso 3 es el más costoso y para reducir su coste se puede usar la transformada cuántica de Fourier.

1. Elegir aleatoriamente a entre 1 y N-1
2. Si mcd(a,N)≠1, **devolver** mcd(a,N).
3. Determinar el período T de la función $f(k)=a^k \bmod N$:
   (a) Inicializar el (n,m)-qubit: $|0\rangle \otimes |0\rangle$
   (b) Aplicar la QFT, F_n al primer registro.
   (c) Aplicar el operador U_f asociado a la función f.
   (d) Aplicar nuevamente F_n al primer registro.
   (e) Obtener la medida k y calcular la fracción continua de k/Q
   (f) Tomar como posibles valores de T los denominadores de las convergentes de la fracción continua.
4. Para cada T, hacer:
   (a) Si T es impar **devolver** fallo.
   (b) Si T es par y mcd($a^{T/2}+1$,N) ≠N, **devolver** mcd($a^{T/2}+1$,N)
   (c) En otro caso **devolver** fallo

Figura 6: Algoritmo de Shor usando QFT

Complejidad del algoritmo de Shor:

1. Aritmética básica → $O(\log^3(N))$
2. Elección aleatoria de un número entre 1 y N-1 → $O(\log(N))$
3. Transformada cuántica de Fourier $O(\log^2(N))$
4. Exponenciación modular cuántica U_f → $O(\log^3(N))$
5. Medida cuántica del primer registro → $O(\log(N))$

El algoritmo de búsqueda de períodos de Shor se basa en gran medida en la capacidad que tiene una computadora cuántica para estar en multitud de estados simultáneamente. Este comportamiento es llamado por los físicos como superposición de estados. Para calcular el período de una función f, debemos evaluar la función en todos sus puntos simultáneamente.

La física cuántica no nos permite acceder a toda la información directamente. Ya que una medición arrojará solo uno de todos los valores posibles, destruyendo con esta medición todos los demás. De manera que tenemos que transformar muy cuidadosamente la superposición a otro estado que nos devolverá la respuesta correcta con una alta probabilidad. Esto se logra mediante la transformada cuántica de Fourier [16] [17].

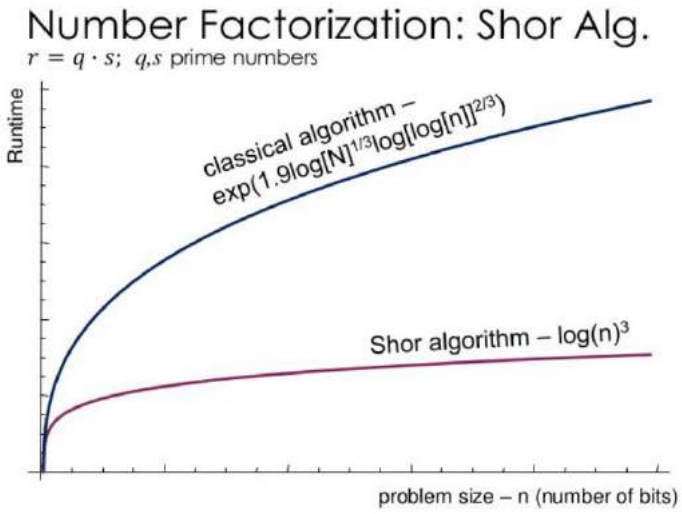

Figura 7: Comparación entre el tiempo que le lleva factorizar un número al mejor algoritmo clásico, comparado con el algoritmo cuántico de Shor

2.1.3.2. Algoritmo de Grover

Este algoritmo fue desarrollado por Lov Grover en 1996 para resolver el problema de las búsquedas en una secuencia no ordenada de datos, como pueden ser las bases de datos.

Clásicamente, la búsqueda en una base de datos sin clasificar requiere una búsqueda lineal O (N) en el tiempo. El algoritmo de Grover lo resuelve en O (N/2), lo que lo convierte en el algoritmo cuántico más rápido para buscar en una base de datos sin clasificar. A diferencia de otros algoritmos cuánticos como el algoritmo de Shor que daba una aceleración exponencial con respecto a los algoritmos clásicos, el algoritmo de Grover proporciona una aceleración cuadrática. Sin embargo, incluso una aceleración cuadrática es considerable cuando N es muy grande.

Como hemos visto anteriormente, la gran mayoría de los algoritmos cuánticos son probabilísticos, en el sentido de que da la respuesta correcta con alta probabilidad y esa probabilidad aumenta conforme se repita el algoritmo.

El algoritmo de Grover también se puede utilizar para estimar la media y la mediana de un conjunto de números.

Consideremos una base de datos clásica como una agenda telefónica con D nombres, cada uno de ellos va seguido de un número de teléfono. En el mejor de los casos para encontrar

un nombre dado un número de teléfono nos valdrá con mirar una sola entrada, pero en el peor de los casos tendremos que revisar todas las entradas para encontrar el número que queremos. De media este número de revisiones necesarias sería de $D/2$.

Desde un punto de vista computacional, tenemos una función lógica de N variables booleanas T o F, y la función solamente devolverá T al encontrar el número que buscamos probando cada una de las D posibilidades. Usando el algoritmo de Grover tendremos una reducción de $D/2$ a $(\pi/4)\sqrt{D}$. De esta manera tenemos que la base de datos o función lógica debe ponerse en la forma de oráculo cuántico. Un oráculo clásico devuelve una respuesta de 1 bit (sí o no) en función de si el resultado es el esperado o no, en cambio, un oráculo cuántico debería aceptar una serie de superposiciones cuánticas de preguntas y devolver la correspondiente superposición cuántica de respuestas [18].

Usando la superposición, se nos permite consultarle al oráculo todas las posibles preguntas a la vez. Pero recordemos que medir los qubits hace que el estado cuántico colapse en uno de los estados básicos, haciendo de esta manera imposible determinar cuál era el estado de superposición anterior a la medida. Sin embargo, Grover mostró una manera de obtener la información deseada del oráculo con muchos menos intentos que los necesarios por un algoritmo clásico.

El oráculo es quien contiene la información de cuál es la respuesta correcta y funciona de la siguiente manera:

- Si al oráculo se le da cualquiera de las D – 1 preguntas incorrectas devuelve la misma amplitud de probabilidad. Por ejemplo, si la respuesta correcta es 111, entonces 110 no lo será y por tanto $\Omega|100\rangle = |100\rangle$.

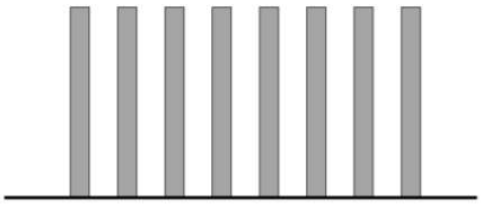

Figura 8: Inicialización del algoritmo de Grover con un estado de superposición uniforme

- Pero si en cambio se le hace la pregunta adecuada, el oráculo devolverá la amplitud de probabilidad cambiada de signo. Por ejemplo, $\Omega|100\rangle = -|100\rangle$.

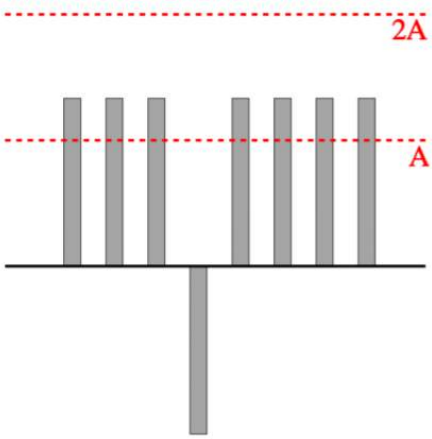

Figura 9: Cambio de signo de la solución tras aplicar el oráculo y cálculo de la media

- Luego, aplicamos el operador de difusión de Grover (inversión del promedio) y obtendremos la nueva amplitud de probabilidad, siendo con cada repetición más cercana al 1, mientras que cualquiera de las incorrectas cada vez será menor. Hay que tener en cuenta que, si se realizan demasiadas iteraciones, la probabilidad de éxito disminuye [19] [20].

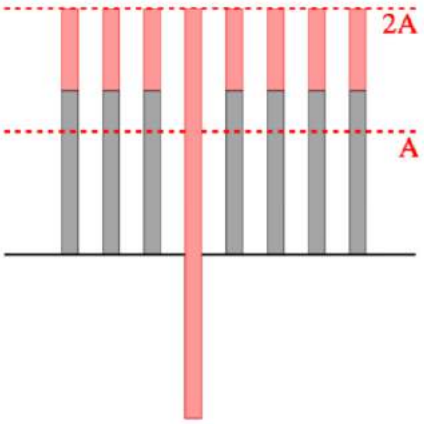

Figura 10: Proceso de inversión sobre la media y amplificación de la amplitud

A continuación, veremos una gráfica de un ejemplo para buscar el número decimal 6 en una base de datos de tamaño 2^3 en el que se han realizado 2 iteraciones.

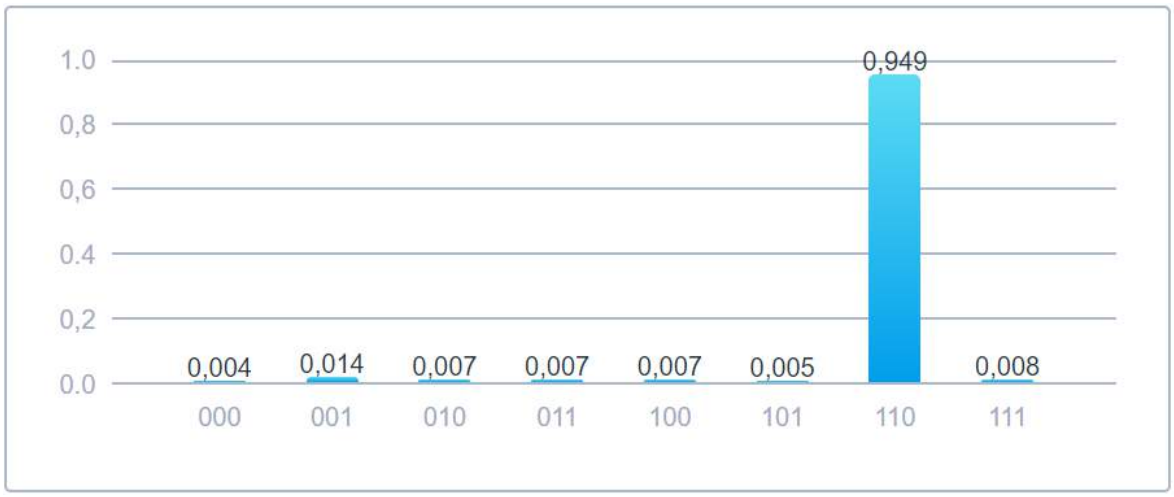

Figura 11: Gráfica de probabilidades aplicando el algoritmo de Grover

Como vemos en la imagen anterior, con solo dos iteraciones obtenemos un 95% de probabilidades de éxito [21].

2.1.3.3. Algoritmo Deutsch-Jozsa

El algoritmo de Deutsch-Jozsa fue propuesto por David Deutsch y Richard Jozsa en 1992, mejorado más tarde por Richard Cleve, Arthur Ekert, Chiara Macchiavello y Michele Mosca en 1998. Fue uno de los primeros ejemplos de un algoritmo cuántico y ha sido demostrado que es exponencialmente más rápido que cualquier algoritmo clásico aprovechando los principios de superposición cuántica y entrelazamiento.

Su función es determinar si una función de tipo caja negra $f\,(x)\colon \{0,1\}^n \to \{0,1\}$ es "constante" o "balanceada". Esto quiere decir, dada una función que para una entrada de n bits da un solo bit de salida, determinar si la salida es independiente de la entrada o si para la mitad de las estradas es 0 y para la otra mitad es 1. Es decir, la función es constante si $f\,(x) = 0$ *o* $f\,(x) = 1$ para todos los valores de X. Una función está equilibrada si $f\,(x) = 0$ para la mitad de los posibles valores de entrada X y $f\,(x) = 1$ para la otra mitad.

Este algoritmo hace uso de un oráculo tal que $Uf\ |x\rangle = (-1)f(x)\,|x\rangle$ y sigue los siguientes pasos:

1. Inicializar los n qubits en el estado cero $|0, \ldots, 0\rangle$.
2. Aplicar la puerta Hadamard H a cada qubit.

3. Aplicar el circuito Oráculo U_f.
4. Repetir el paso 2.
5. Medir cada qubit

Este algoritmo es un ejemplo simple de un algoritmo cuántico que se puede utilizar para acelerar una búsqueda. Puede determinar si una función tiene una determinada propiedad (estar equilibrada) al requerir que la función solo necesita llamarse una vez con un algoritmo cuántico en lugar de dos veces con un algoritmo clásico [22].

2.1.4. Programación

Un lenguaje de programación es un conjunto estructurado de reglas, notaciones, símbolos y caracteres que permiten a un programador poder expresar el procesamiento de datos y sus resultados por medio del uso de computadores. Cada lenguaje posee una sintaxis propia, lo cual los hace muy diferentes entre sí. Podemos encontrar varios tipos de paradigmas de programación, como los imperativos, declarativos u orientado a objetos. Existen una gran variedad de lenguajes de programación entre los que se encuentran Java, Python, PHP, etc.

Hay lenguajes de programación de alto nivel, que son aquellos que se pueden leer y escribir de forma más sencilla ya que se asemejan al lenguaje humano y los lenguajes de programación de bajo nivel que dependen del tipo de máquina y no se puede migrar o utilizar en otras máquinas, aquí nos encontramos con el lenguaje ensamblador el cual es un derivado del lenguaje máquina que usa operaciones fundamentales para el funcionamiento de la máquina.

Todo lo anterior es aplicable a la computación clásica, pero aquí nos interesa la computación cuántica y cómo comunicarnos y enviar órdenes a las máquinas cuánticas. IBM presentó en 2017 el primer computador cuántico, que puede ser usado por medio de la nube, y nos permite crear circuitos cuánticos a través de un modelo interactivo de pinchar y arrastrar compuertas cuánticas, a través de su lenguaje ensamblador QASM y para los usuarios más avanzados, haciendo uso de una API desarrollada en Python [23].

Pero estos no son los únicos lenguajes de programación existentes, ya que en los últimos años se han desarrollado otros lenguajes de programación entre los que destacaremos Qiskit, Silq y Q#.

2.1.4.1. Qiskit

Qiskit es un SDK de código abierto creado por IBM bajo licencia apache para trabajar con computadoras cuánticas a nivel de pulsos, circuitos y módulos de aplicación. Usa el lenguaje Python, aunque también hay disponibles versiones para Swift y JavaScript.

Qiskit acelera el desarrollo de aplicaciones cuánticas al proporcionar el conjunto completo de herramientas necesarias para interactuar con simuladores y sistemas cuánticos.

Qiskit está compuesto de cuatro elementos fundamentales:

- Qiskit Terra: Es la base sobre la que se asienta el resto del software. Proporciona una base para la composición de programas cuánticos a nivel de circuitos y pulsos, para optimizarlos en base a las restricciones de un dispositivo en particular y para administrar la ejecución de lotes de experimentos en dispositivos de acceso remoto.
- Qiskit Aqua: Está enfocado para expertos en dominios de química, optimización o inteligencia artificial, donde van a poder explorar los beneficios de usar computadores cuánticos como aceleradores para tareas computacionales específicas.
- Qiskit Ignis: Dedicado a combatir el ruido y los errores. Mejora la caracterización de errores, de las compuertas y la computación en presencia de ruido.
- Qiskit Aer: Se puede usar para verificar que los computadores cuánticos actuales y futuros funcionen correctamente. Para ello realiza una simulación con los efectos del ruido realista en el cálculo [24].

Figura 12: Logo de Qiskit

2.1.4.2. Q#

Q# es un lenguaje de programación de alto nivel de código abierto orientado a la computación cuántica que ofrece un enfoque intuitivo y moderno para el desarrollo de programas cuánticos creado por Microsoft en 2017 como parte del Quantum Development Kit.

Q# incluye bibliotecas Q#, simuladores cuánticos, extensiones para otros entornos de programación y documentación API. Además de la biblioteca Standard Q#, QDK incluye bibliotecas de química, aprendizaje automático y numérico.

Los programas Q# pueden ejecutarse como una aplicación de consola, a través de Jupyter Notebooks, o usar un programa de host Python o .NET [25].

Figura 13: Logo de Q#

2.1.4.3. Silq

Silq es un lenguaje de programación de alto nivel para computación cuántica con un fuerte sistema de tipo estático, desarrollado en ETH Zürich.

Silq no está diseñado en torno al hardware, sino pensando en la forma de pensar de los programadores para resolver un problema. Este es el primer lenguaje de programación que identifica y borra automáticamente valores que ya no son necesarios, su método de cálculo automático usa solo comandos de programación que están libres de cualquier operación cuántica especial [26].

Figura 14: Logo de Silq

2.2. Criptografía

La criptografía es la ciencia y arte de escribir mensajes en forma cifrada o en código secreto. Es parte de un campo de estudio que trata las comunicaciones secretas, usadas, entre otras finalidades, para autentificar la identidad del usuario, autentificar y proteger el sigilo de comunicaciones personales y de transacciones comerciales y bancarias y para proteger la integridad de transferencias electrónicas de fondos [27].

La criptografía tiene su origen en la escritura. Proteger lo escrito mediante su conversión a información cifrada. Desde tiempos inmemorables se han utilizado códigos secretos para lograr que un mensaje resultara incomprensible para las personas no autorizadas a leerlo. A lo largo de la historia, siempre ha existido la necesidad de transmitir secretamente información de una persona a otra. Desde las tumbas del antiguo Egipto, pasando por los primeros sistemas criptográficos que conocemos como la scitala espartana, hasta llegar a la máquina Enigma utilizada por el ejército alemán en la segunda guerra mundial. Esta fue la que impulsó a la computación y llevó a Alan Turing a desarrollar su máquina de Turing para descifrar esta máquina Enigma [28].

Figura 15: Scitala Espartana

Con la aparición de los ordenadores los métodos de cifrado anteriores resultaron muy vulnerables por la capacidad de cálculo de estos. En 1949, el matemático estadounidense Claude Shannon publicó su artículo "*Communication Theory of Secrecy Systems*" [29] basado en su obra sobre la Teoría de la Información, sentando las bases para el tratamiento matemático de la criptología. Con lo que se empezó a desarrollar la criptografía computacional que conocemos hoy en día.

2.2.1. Criptografía clásica vs cuántica

La codificación y decodificación de un mensaje dentro de la criptografía clásica, se realiza por medio de claves que pueden ser de conocimiento privado, en la que el propietario de la clave tiene que guardar el secreto, o de conocimiento público, donde cualquiera puede acceder a estas claves. Esta criptografía se clasifica según el tipo de clave que usa: privada o pública. También se las conoce con el nombre de criptografía simétrica y asimétrica respectivamente.

La seguridad de la transmisión de un mensaje se mide según la distribución de la clave, el almacenamiento de la clave, la codificación del mensaje y la resistencia a ataques de terceros.

Por un lado, tenemos la criptografía de clave privada o simétrica, donde se usa una misma clave tanto para cifrar como para descifrar el mensaje. Para que exista una comunicación segura entre dos usuarios ambos deben compartir la misma clave, siendo este el punto débil del sistema ya que resulta más fácil interceptar una clave que se ha transmitido sin seguridad.

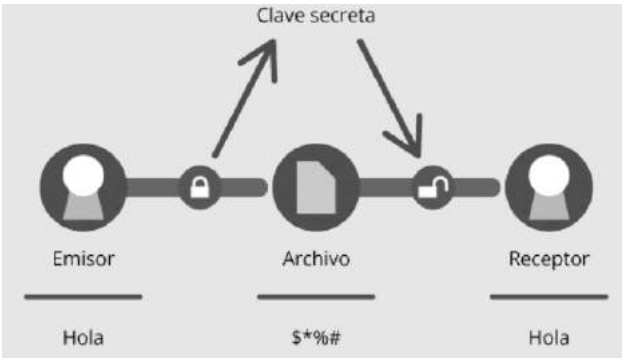

Figura 16: Comunicación entre emisor y receptor mediante clave simétrica

Teóricamente debería resultar más sencillo conocer la clave interceptándola que probándola una a una mediante un ataque por fuerza bruta, teniendo en cuenta que la seguridad de un mensaje cifrado debe recaer sobre la clave y nunca sobre el algoritmo. Ya que debería ser una tarea titánica descifrar la clave.

Este sistema se considera seguro cuando una vez que se cifra el mensaje, no se pueda obtener la clave de cifrado ni tampoco el texto en claro por ningún método y cuando, conocidos el texto en claro y el mensaje cifrado, resulte más costoso obtener la clave para acceder al texto en claro que el posible valor que pueda tener la información que se consiga interceptar.

Algunos ejemplos de algoritmos de clave simétrica son DES o AES entre otros, estos se caracterizan en que son muy rápidos.

Por otro lado, nos encontramos con la criptografía asimétrica o de clave pública, que se basa en el uso de dos claves: la pública, que deberá publicarse junto con la identidad del usuario y la privada que ha de mantenerse en secreto en todo momento. Así, cuando se quiera enviar un mensaje seguro a un usuario se hará uso de la clave pública, que se utilizará para cifrar el mensaje a enviar.

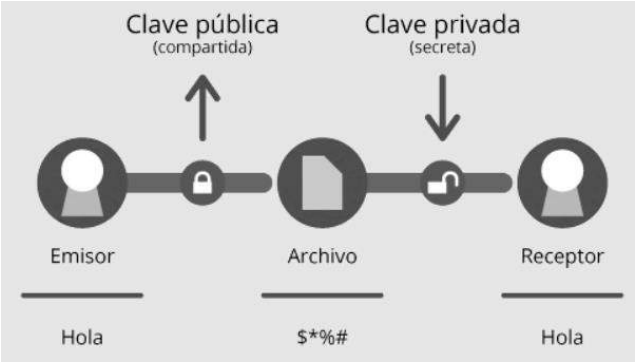

Figura 17: Comunicación entre emisor y receptor mediante clave asimétrica

Los algoritmos de clave asimétrica están basados en funciones matemáticas fáciles de resolver en un sentido, pero muy complicadas de realizar en sentido inverso, salvo que se conozca alguna trampa. Ambas claves están relacionadas matemáticamente, pero esta relación debe ser lo suficientemente compleja como para que resulte muy difícil obtener una a partir de la otra.

La ventaja de los criptosistemas asimétricos o de clave pública es que la clave pública y el algoritmo de cifrado pueden ser de dominio público y no es necesario poner en peligro la clave privada enviándola por canales inseguros. La desventaja con respecto a los criptosistemas de clave privada es que son mucho más lentos ya que deben realizar cálculos mucho más complejos.

El algoritmo de clave pública más utilizado es RSA que es en el que radica la mayor parte de la seguridad en internet. La seguridad de este algoritmo se basa en el problema de la factorización de números enteros. Se cree que RSA será seguro mientras no se conozcan formas rápidas de descomponer un número grande en producto de primos. Aunque como ya hemos visto anteriormente, el algoritmo de Shor puede romper este sistema criptográfico.

Actualmente el método común que se emplea en la criptografía es el uso de la criptografía híbrida, el cual es un criptosistema que usa la clave pública del receptor para cifrar una clave simétrica que se usará en el proceso de comunicación cifrada. De esta manera tenemos la seguridad de la transmisión de claves de la criptografía de clave pública y la rapidez de la criptografía de clave privada, aprovechando así sus ventajas y paliando sus inconvenientes.

Pero como ya sabemos, los qubits no se comportan igual que los bits tradicionales y son sus propiedades extraordinarias las que los convierten en un arma casi infalible contra el espionaje. La criptografía cuántica hace uso de dos canales de comunicación entre el emisor y el receptor. Un canal cuántico, el cual tiene un único sentido y que generalmente es de fibra óptica y un canal convencional, público y de dos vías.

La primera vez que apareció la idea de la criptografía cuántica fue a manos de Stephen Weisner en 1970, quien propuso dos modalidades de comunicación que no eran posibles por medio de la física clásica: un canal de multiplexación cuántica y un billete infalsificable [30].

La criptografía cuántica se ha enfocado en tratar uno de los problemas a los que se enfrenta la seguridad de los sistemas de comunicación, el problema de la distribución de claves, naciendo así la distribución de clave cuántica o QKD.

Si en la criptografía tradicional nos encontrábamos con el peligro de que un tercero interceptara la clave para poder descifrar nuestro mensaje sin darnos cuenta, esto no es tan sencillo en la criptografía cuántica, ya que, aprovechándose de una de sus propiedades más destacables, la tentativa de supervisión dentro del canal cuántico provoca necesariamente cambios detectables en la señal. Esto quiere decir, que como ya hemos visto anteriormente en el mundo cuántico medir perturba el resultado, con lo que enseguida nos daríamos cuenta de que estamos siendo espiados. A esto se le conoce como principio de incertidumbre de Heisenberg, el cual a su vez soporta al teorema de no clonación que veremos más adelante [31].

2.2.2. Seguridad

La seguridad de la criptografía cuántica descansa en las bases de la mecánica cuántica, donde tenemos como piedra angular el principio de incertidumbre de Heisenberg. Este principio a grandes rasgos nos enseña que no pueden conocerse simultáneamente con exactitud dos

variables complementarias, como pueden ser la posición y la velocidad de una partícula. Si se mide una propiedad, necesariamente se altera la complementaria, perdiéndose cualquier noción de su valor exacto. Cuanto más precisa sea la medición de una de las propiedades, mayor será la incertidumbre de la otra propiedad.

Generalmente las partículas utilizadas en la criptografía cuántica son los fotones que son la unidad de energía más pequeña en una onda de luz y que no se puede dividir.

La polarización de un fotón tiene la cualidad cuántica de vibrar en dos direcciones al mismo tiempo, pero al medirlo el fotón sufre una alteración y se ve obligado a elegir una de esas dos direcciones.

Supongamos que tenemos un fotón que puede estar polarizado en una de cuatro direcciones distintas: vertical (↕), horizontal (↔), diagonal a la izquierda (⤡) o diagonal a la derecha (⤢). Estas cuatro polarizaciones forman dos bases ortogonales, + y X respectivamente, que corresponderían al eje Z y al eje Y que vimos en los qubits.

El principio de incertidumbre de Heisenberg nos impide que podamos conocer en cuál de las cuatro posibles polarizaciones se encuentra nuestro fotón. Para ello debemos emplear un filtro que esté orientado en una de las cuatro direcciones. Supongamos entonces que tenemos un filtro orientado horizontalmente (↔) y un fotón polarizado en la misma dirección, este fotón pasará por el filtro sin problema y tendremos un 100% de probabilidades de medir el fotón correctamente, pero imaginemos ahora que nuestro fotón está polarizado en la dirección ⤡, este fotón no pasará el filtro y su polarización será modificada de manera aleatoria [32].

Para usar estas propiedades en la criptografía, debemos acordar representar con un 1 o un 0 según la polarización de los fotones que se envían. Por ejemplo, en la base + podríamos acordar que ↔ representaría al 1 y ↕ al 0. De igual manera, en la base X, ⤡ podría representar al 1 y ⤢ al 0. Este acuerdo lo deben llevar a cabo entre el emisor y el receptor.

Visto lo anterior podríamos pensar que la solución más fácil para burlar el principio de incertidumbre de Heisenberg sería capturar todos esos fotones o qubits, crear una copia de ellos para analizarlos más tarde y devolver los originales al receptor, de manera que el intruso pasaría desapercibido. Esto no es posible gracias a otra de las propiedades fundamentales de la mecánica cuántica en la que se apoya la seguridad en la criptografía cuántica, el teorema de no clonación cuántica.

Este teorema fue desarrollado por los físicos Wootters, Zurek y Dieks en 1982 y declara que es imposible crear una copia idéntica de un estado cuántico desconocido arbitrario, con lo que podemos proteger nuestros qubits.

2.2.3. Protocolos

Desde la propuesta del primer protocolo de distribución de claves en 1984 hasta hoy, han aparecido diferentes posibilidades para el intercambio seguro de claves basadas en los principios de la física cuántica. Por un lado, encontramos las propuestas de protocolos basados en la transmisión de un único qubit como BB84 o B92. Y, por otro lado, los protocolos basados en pares entrelazados, también conocidos como pares EPR, donde se encuentra el protocolo E92.

A partir de este momento trabajaremos con una arquitectura básica para el estudio de una comunicación en la que intervienen un emisor y un receptor, comúnmente conocidos como Alicia y Bob, por otro lado, tendremos un espía conocido como Eva. También tenemos dos canales de comunicación, uno cuántico de un único sentido para enviar fotones y que generalmente es de fibra óptica y un canal convencional, público y de dos vías y que está autenticado para garantizar la seguridad de los protocolos QKD.

El trabajo de Eva será obtener la máxima información posible acerca de la clave intercambiada por Alicia y Bob, para lo cual, dispondrá de acceso a los canales cuántico y convencional bajo los siguientes supuestos:

1. El acceso al canal cuántico será total, teniendo la capacidad de hacer cualquier cosa que no esté prohibida por las reglas de la mecánica cuántica, y por lo tanto respetando el teorema de no clonación, por lo que no podrá copiar los qubits, y el principio de incertidumbre de Heisenberg, con lo que al medir un qubit provocará una perturbación en dicho qubit.
2. Eva podrá leer toda la información del canal convencional, pero nunca podrá modificarla, ya que el canal se encuentra autenticado.

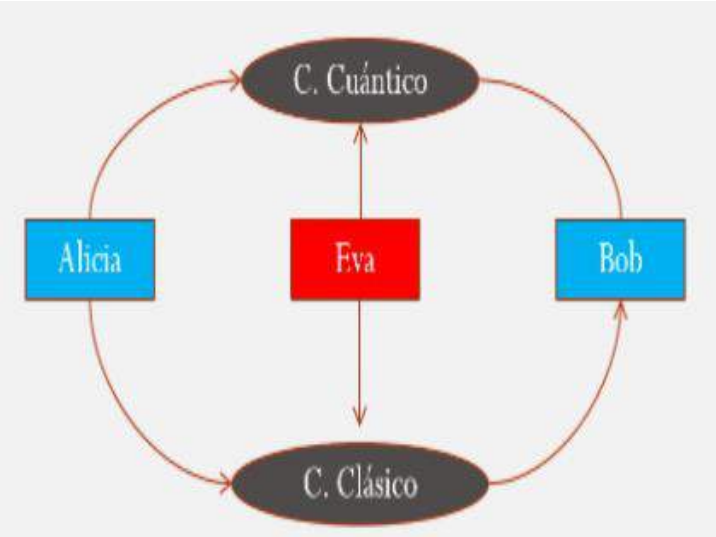

Figura 18: Escenario de un sistema QKD

2.2.3.1. Protocolo BB84

El protocolo BB84 es el primer protocolo de distribución cuántica de claves. Fue propuesto por Bennet y Brassard en 1984.

La primera parte del protocolo BB84 se desarrolla por medio de un canal de comunicación cuántico, a través del cual, Alicia y Bob, intercambiaran una serie de qubits implementados en fotones y codificados conforme a los cuatro estados de polarización que hemos visto anteriormente.

En primer lugar, Alicia genera una serie muy larga de bits elegidos aleatoriamente que los escribirá en una cadena de qubits, para ello Alicia tiene que hacer una elección, codificarlos en el eje Y o en el eje Z. Si elige el eje Y, Alicia traduce los bits 0 preparando estados llamados $|\rightarrow\rangle$, mientras que traduce los bits 1 preparando estados $|\leftarrow\rangle$. Por otro lado, si elige el eje Z, traduce los valores 0 y 1 preparando respectivamente los estados llamados $|0\rangle$ y $|1\rangle$. Alicia, haciendo estas selecciones está generando una cadena de qubits formada por cuatro posibles estados: $|0\rangle$, $|1\rangle$, $|\rightarrow\rangle$ o $|\leftarrow\rangle$.

Para cada qubit, Alicia apunta su estado especificando el eje en el que lo ha preparado y el valor del bit que representa. Cuando la cadena es lo suficientemente grande, se la envía a Bob.

Bob atrapa los qubits que le ha mandado Alicia y elige aleatoriamente un eje para medirlo empleando los filtros. Si Bob mide en el mismo eje que ha escogido Alicia, tiene un 100% de probabilidades de medir exactamente el valor escogido por Alicia. Pero, por el contrario, si se equivoca de eje al medir, el estado preparado por Alicia se proyectará en ese eje destruyéndolo y el resultado será completamente aleatorio.

Bob mide toda la cadena alternando aleatoriamente los filtros para medir los qubits de Alice y apunta el eje en el que ha decidido medir cada qubit y el resultado que ha obtenido sin conocer cómo había preparado Alicia los qubits. Una vez Bob termina de medir los qubits, hace público el resultado de los ejes, pero no el valor que ha medido. Alicia al verlo, hace público el eje de todos los qubits que había preparado.

Ahora Alicia y Bob comparten las dos listas de ejes y eliminan los casos en los que no han coincidido, esto es debido a que, en estos casos, los resultados de Bob son completamente aleatorios, por lo tanto, puede haber acertado o no. Pero, sin embargo, cuando Alicia y Bob coinciden con el eje, siempre coinciden con el resultado, enviándose así un bit clásico. Bob tiene una probabilidad del 50% de acertar el eje de Alicia, con lo que la cadena de qubits quedará reducida aproximadamente a la mitad.

Una vez purgada la lista, Alicia le pide a Bob que haga público el valor de una cierta cantidad de qubits. Cuando Bob los hace públicos, Alicia los compara con los suyos. Lo normal es que Alicia observe exactamente lo mismo en las dos tablas, ya que, si Bob ha acertado el eje, el resultado debe ser el mismo. Pero podría pasar que uno de los resultados de Bob no coincida con los de Alicia. Esto podría ser debido a un fallo de conexión, en el aparato de medida de Bob o un fallo en la preparación de los qubits, pero también podría ser una prueba de que alguien los está espiando [33].

Imaginemos que Eva, intentando capturar la clave ha estado interceptando los qubits que Alicia le enviaba a Bob para medirlos ella primero y luego devolvérselos a Bob sin que nadie se diera cuenta. Pero debido al principio de incertidumbre de Heisenberg, al medir el qubit ha perturbado el resultado del qubit. Pongamos un ejemplo:

Alicia prepara un qubit en el estado |0⟩ del eje Z y se lo envía a Bob, pero Eva lo intercepta antes de que le llegue. Eva, al igual que Bob, desconoce el estado inicial del qubit, y debido al teorema de no clonación, no puede crear una copia del qubit para medirla más tarde, por lo que debe decidir si medirlo en el eje Y o en el eje Z. Si Eva elige el eje Z, habrá acertado y podrá medirlo correctamente sin perturbarlo, devolviéndoselo a Bob sin que nadie se percate. Pero, sin embargo, si Eva decide medirlo en el eje Y, se habrá equivocado y el estado |0⟩ se proyectará aleatoriamente en el estado |→⟩ o |←⟩, supongamos que sale |→⟩. Eva habrá destruido el estado inicial en el que Alicia había preparado el qubit sin saberlo, ya que no tiene manera de saber si ha medido correctamente o no el qubit. Eva le envía el qubit a Bob, ahora es el turno de Bob para elegir en qué eje mide el qubit. Si Bob escoge el eje Y, se habrá equivocado con respecto a Alicia, por lo que Eva pasaría inadvertida ya que cuando Alicia y Bob comparen los resultados, descartaran ese qubit al no haber coincidido en el eje. Pero si Bob decide medir el qubit en el eje Z, cuando comparen los ejes lo darán por bueno y pasará el filtro. Bob ha medido el qubit en el mismo eje que lo preparó Alice, pero recordemos que Eva lo ha perturbado proyectándolo en el eje Y. Por lo tanto, al medirlo en el eje Z, hará que el qubit colapse en dos posibles estados, el |0⟩ o el |1⟩, si el resultado es el |0⟩ Eva habrá tenido suerte y ni Alicia ni Bob se darán cuenta de que están siendo espiados. Si, por el contrario, el resultado es |1⟩, Alicia se dará cuenta de que había preparado el qubit en el estado |0⟩ y a pesar de medirlo correctamente a Bob le ha dado como resultado el estado |1⟩.

De esta manera Alicia se da cuenta de que el canal de comunicación ha sido comprometido y descarta la clave que había propuesto. Alicia y Bob deberán empezar de nuevo.

La probabilidad de que Alicia se percate de que está siendo espiada es de un 25% si solo se revisa un qubit, si revisan 10, la probabilidad sube a un 95% y con 50, Alicia tiene una seguridad del 99,9999%. Una vez garantizado que la comunicación es segura, Alicia y Bob utilizan el resto de resultados que no han hecho públicos como una clave [34].

Esto convierte al protocolo BB84 en un método infalible teóricamente hablando, ya que, esto ocurriría en un escenario ideal sin las perturbaciones y desajustes que pueden ocasionar los aparatos de medición, los generadores de fotones, interferencias en el canal de comunicación, etc.

Bits aleatorios de Alicia	0	1	1	0	1	0	0	1

Ejes de Alicia	Z	Z	Y	Z	Y	Y	Y	Z
Fotones enviados por Alicia	$\|0\rangle$	$\|1\rangle$	$\|\leftarrow\rangle$	$\|0\rangle$	$\|\leftarrow\rangle$	$\|\rightarrow\rangle$	$\|\rightarrow\rangle$	$\|1\rangle$
Ejes de Eva	Z	Y	Z	Z	Y	Z	Y	Z
Polarizaciones que mide Eva	$\|0\rangle$	$\|\rightarrow\rangle$	$\|1\rangle$	$\|0\rangle$	$\|\leftarrow\rangle$	$\|1\rangle$	$\|\rightarrow\rangle$	$\|1\rangle$
Ejes de Bob	Z	Y	Y	Y	Z	Y	Z	Z
Polarizaciones que mide Bob	$\|0\rangle$	$\|\rightarrow\rangle$	$\|\rightarrow\rangle$	$\|\leftarrow\rangle$	$\|1\rangle$	$\|\rightarrow\rangle$	$\|0\rangle$	$\|1\rangle$
Clave secreta compartida	0		0			0		1
Errores en la clave	✓		✗			✓		✓

Tabla 1: Ejemplo aplicación protocolo BB84

2.2.3.2. Protocolo E91

El protocolo E91 fue propuesto por Artur Ekert en 1991 y basa su seguridad en el uso de pares EPR. Un par EPR es un estado cuántico de 2 qubits entrelazados. El esquema de comunicación es bastante parecido al del protocolo BB84, a diferencia de que en este escenario necesitaremos además una fuente que produzca una serie de fotones entrelazados. Esta fuente puede estar en manos de Alicia, de Bob o de un tercero de confianza, lo importante es que de cada par de fotones entrelazados que se produzcan, un fotón debe llegar a Alicia y el otro a Bob.

Al igual que ocurría en el protocolo BB84, Alicia y Bob eligen de manera independiente y aleatoria el eje en el que medirán cada qubit. Alicia y Bob se intercambian un listado con los ejes que han utilizado para medir sus qubits, de esa manera sabrán que bits son los que corresponden a la clave generada.

Si un intruso intentase medir de alguna forma alguno de los fotones entrelazados, no podrá predecir los ejes de Alicia y Bob por lo que tendría que usar un eje aleatorio, esto alteraría el resultado de Alicia y Bob, ya que al igual que ocurría en el protocolo BB84, se realiza una

verificación de los qubits intercambiados con el fin de detectar intrusos, interferencias o errores de transmisión [30].

La ventaja que ofrece este protocolo es que la clave se genera aleatoriamente de forma natural y hace imposible conocer de antemano que polarización tendrá cada fotón.

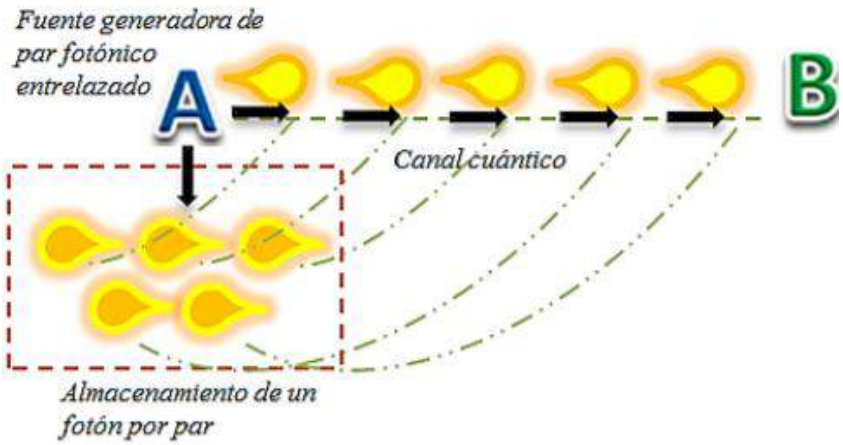

Figura 19: Generación de par fotónico entrelazado.

2.2.4. Ataques

Desde tiempos inmemoriales usamos la criptografía para cifrar nuestra información y protegerla de los intrusos, pero los intrusos no han desistido de la tarea de romper nuestro cifrado para obtener nuestra información. A lo largo de la historia se han desarrollado numerosos sistemas criptográficos, y con ellos nuevos ataques para romper esos cifrados. Alan Turing dio un paso más allá y desarrolló lo que hoy conocemos como el predecesor de la computación, la máquina de Turing, para romper el sistema criptográfico que utilizaba el ejército alemán durante la segunda guerra mundial, la máquina enigma.

A partir de entonces y hasta nuestros días se fue desarrollando la criptografía computacional y con ello diversos ataques, como la fuerza bruta o man-in-the-middle entre otros.

Ahora es el turno de la criptografía cuántica, para la que, como hemos visto antes, se han desarrollado nuevos protocolos para realizar una distribución de clave segura.

Hoy en día, tenemos suficientes conocimientos sobre los ataques más poderosos que Eva podría realizar contra el canal cuántico, asumiendo que Eva no tiene absolutamente ningún límite tecnológico, es decir, puede hacer todo lo que la física cuántica le permita.

Pero, claramente, los ataques de Eva no están limitados solo al canal cuántico. Eva podría por ejemplo atacar a los aparatos de medición de Alicia y Bob o podría explotar las debilidades en la implementación real de los QKD.

La física cuántica no ayuda a proteger los aparatos de Alicia y Bob, de hecho, tan pronto como la información pase a estar codificada en un sistema de física clásico, es vulnerable a la copia y difusión. Por lo tanto, Alicia y Bob deben proteger sus aparatos por medios clásicos.

Las implementaciones reales de los usos abstractos de QKD usando la tecnología actual se alejan algo del esquema ideal. Por tanto, es de vital importancia para QKD analizar correctamente las consecuencias de estas vulnerabilidades [35].

Por tanto, es importante construir un catálogo de ataques conocidos contra sistemas QKD prácticos. Además, es imperativo estudiar las defensas específicas contra los ataques propuestos.

2.2.4.1. Man-in-the-middle

Está de sobra probado que los protocolos QKD son seguros frente a una escucha pasiva de Eva. Pero no existe ningún protocolo QKD que pueda resistir el clásico ataque de intromisión. Este es un ataque activo, en el que Eva suplanta a Bob cuando habla con Alicia, mientras que suplanta a Alicia cuando habla con Bob. Esto lo consigue entrometiéndose en el canal convencional y desarrollando dos sesiones simultáneas de comunicación con Alicia y Bob [36] [37].

Este tipo de ataque solo se puede evitar mediante la autenticación previa de Alicia y Bob en el canal convencional.

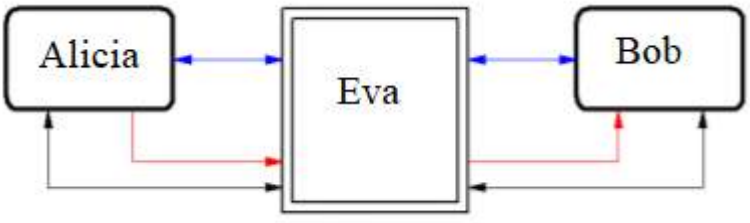

Figura 20: Representación del ataque de intromisión

2.2.4.2. Caballo de Troya

El canal cuántico en sí es potencialmente una puerta para un intruso que le da acceso a los aparatos de Alicia y Bob. Incluso si la puerta está correctamente diseñada, Eva podría utilizarlo al mismo tiempo que los usuarios legítimos.

Este ataque consiste en enviar potentes pulsos de luz por el canal cuántico que permiten a Eva sondear cómo está configurado el emisor de fotones de Alicia, de esta manera, Eva podría conocer con anterioridad los ejes que ha elegido Alicia para sus qubits y, por consiguiente, medir todos los qubits sin provocar ningún fallo.

Para limitar esta posibilidad, el sistema debe diseñarse de tal manera que:

1. Solo puede entrar luz con una longitud de onda determinada, es decir, usando filtros.
2. El canal cuántico debe estar abierto solo durante periodos breves, es decir, los componentes ópticos de codificación deben estar activos el menor tiempo posible y solo cuando emitan o reciban qubits.
3. El canal cuántico es de solo una dirección, de manera que Eva no pueda enviar los pulsos de luz al emisor de fotones de Alicia [35].

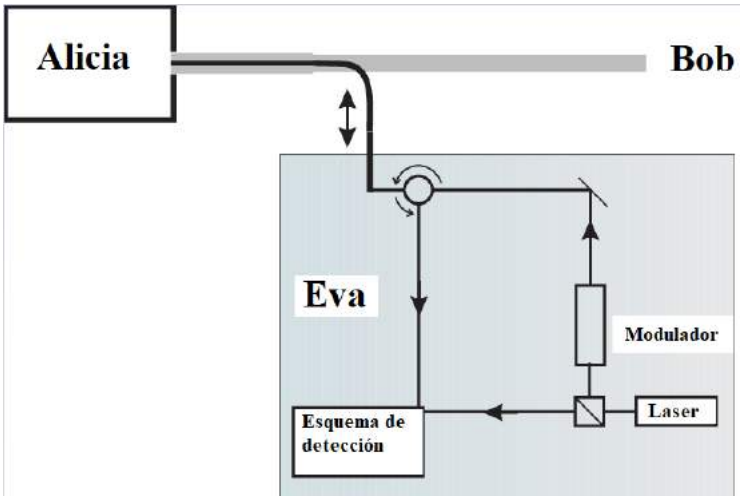

Figura 21: Principio del ataque del caballo de Troya

2.2.4.3. Remapeo de fases

Las dificultades prácticas asociadas con las inestabilidades de fase y polarización en la fibra de larga distancia, han llevado al desarrollo de dos estructuras QKD bidireccionales: la estructura “plug & play” y la estructura Sagnac QKD. En ambos casos, uno de los usuarios envía fuertes pulsos de láser al otro usuario. Alicia codifica su información sobre la fase de pulso fuerte, la atenúa a un nivel de fotón único y luego la envía a Bob. Debido a que Alicia permite que las señales entren y salgan de su dispositivo, esto abre una puerta trasera para que Eva lance varios ataques del tipo de caballo de Troya que hemos visto anteriormente.

El ataque de remapeo de fases es un tipo específico del ataque de caballo de Troya. Este ataque consiste en que Eva, mediante el envío de fuertes pulsos engañe al láser de Alicia y le haga creer que necesita ser recalibrado, modificando los qubits a otros que a Eva le convienen más. En ciertos casos, Eva puede quedar por debajo del umbral de errores que Alice capta de sus propios aparatos y pasar desapercibida [38].

Al tratarse de un ataque de tipo caballo de Troya, podemos evitarlo de la misma manera que hemos visto anteriormente.

2.2.4.4. Escisión de fotones

Este ataque aprovecha las imperfecciones actuales de los aparatos generadores de pulsos, ya que en algunas realizaciones prácticas ocurre que en vez de codificar cada qubit en un solo fotón, se codifica en varios fotones para que resistan la disipación del canal o porque el generador de pulsos tiene algún desajuste y ha enviado sin querer más de uno.

Como todos los fotones del pulso representan el mismo qubit, Eva tan solo tiene que dividir los fotones adicionales, coger uno de los fotones de Alicia y dejar pasar el resto a Bob. Eva almacena estos fotones adicionales en una memoria cuántica hasta que Alicia y Bob publiquen sus ejes, Eva podrá medir sus fotones en el eje correcto y obtener información sobre la clave sin introducir errores que puedan ser detectados y sin que nadie sospeche [33].

Hay varias soluciones a este ataque. Por un lado, podemos empezar a contar los fotones que envía Alicia para ver si concuerdan con los que recibe Bob, otra solución sería implementar un protocolo de estado de señuelo, en el que Alicia envía aleatoriamente algunos de sus pulsos a modo de señuelo. Eva al capturarlos, no tiene forma de saber qué pulsos son señuelos y cuáles son los que contienen la clave [39].

2.3. Otros elementos cuánticos

Visto lo anterior y llegados a este punto podríamos pensar ¿Cómo funciona un computador cuántico internamente?, ¿Cómo se hace un computador cuántico? y, ¿Cómo podemos interactuar con un computador cuántico?

Esta sección se ha desarrollado para resolver esas preguntas y ampliar un poco más nuestros conocimientos acerca de la computación cuántica, de manera que podamos entender plenamente esta nueva tecnología.

2.3.1. Puertas cuánticas

Para manejar la información almacenada en un conjunto de bits en la computación clásica, utilizamos las llamadas puertas lógicas, que se basan en el Álgebra Booleana. La puerta lógica más simple y de carácter unitario es la puerta NOT, también tenemos las puertas binarias AND y OR. El resto de puertas lógicas se pueden expresar en función de estas tres puertas.

Las computadoras cuánticas están formadas por puertas cuánticas, que realizan cambios en las probabilidades de obtener un estado base. Estas puertas cuánticas son reversibles, a diferencia de la mayoría de las puertas lógicas clásicas. La aplicación de puertas cuánticas no colapsa los estados y permite realizar distintas operaciones [40].

Las puertas lógicas cuánticas son representadas mediante una matriz unitaria, es decir, una matriz cuya inversa es igual a su conjugada traspuesta.

Las puertas cuánticas operan sobre k qubits. Esto significa que, como matrices, las puertas cuánticas se pueden describir mediante matrices de 2^k x 2^k.

2.3.1.1. Puerta Hadamard

La puerta cuántica Hadamard H es usada por los algoritmos para generar superposiciones a partir de los estados de la base computacional. Es una operación de un solo qubit que mapea el estado base $|0\rangle$ a $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$ y $|1\rangle$ a $\frac{|0\rangle - |1\rangle}{\sqrt{2}}$, creando así una superposición de los dos estados base.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

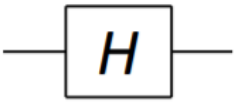

Figura 22: Puerta Hadamard

La puerta Hadamard es equivalente a hacer una rotación de π radianes sobre el eje XZ en la esfera de Bloch [41].

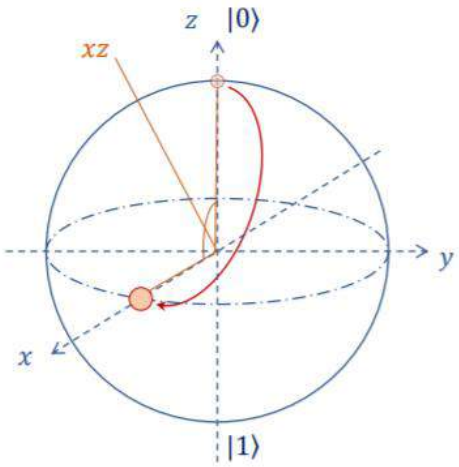

Figura 23: Representación de la puerta Hadamard en la esfera de Bloch

2.3.1.2. Puerta Z

La puerta Z mapea el estado $|1\rangle$ al - $|1\rangle$, dejando $|0\rangle$ inalterado. Equivale a una rotación de π radianes alrededor del eje Z [42].

$$Z = V^2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Figura 24: Puerta Z

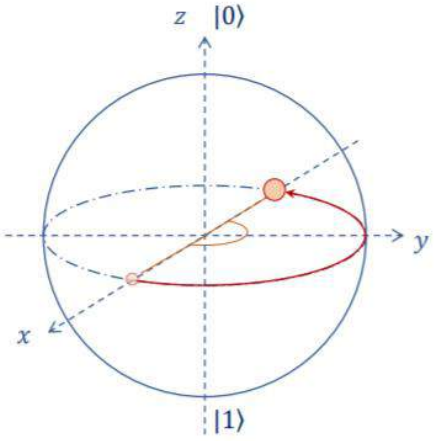

Figura 25: Representación de la puerta Z en la esfera de Bloch

2.3.1.3. Puerta X

La puerta X es el equivalente cuántico a la puerta NOT. La puerta X representa una negación. Mapea el estado base |0⟩ al |1⟩, y el estado |1⟩ al |0⟩. Equivale a una rotación de π radianes alrededor del eje Y [42].

$$X = HZH = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

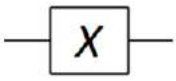

Figura 26: Puerta X

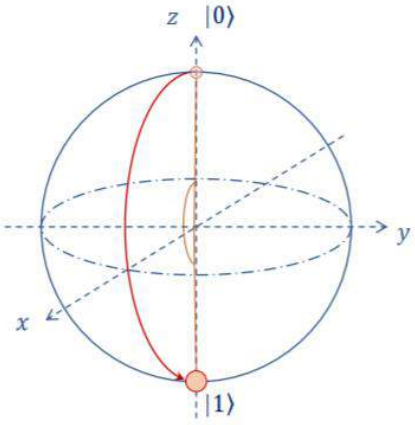

Figura 27: Representación de la puerta X en la esfera de Bloch

2.3.2. Hardware

No hay una única manera de construir computadoras cuánticas, pero sí que hay una serie de reglas que se deben cumplir para que sean consideradas como tales. Son 5 puntos y se los conoce como la lista Di Vincenzo [43]:

1. El sistema ha de poder inicializarse. Esto significa llegar a un estado de partida conocido y controlado.
2. Ha de ser posible hacer manipulaciones a los qubits de forma controlada con un conjunto de operaciones que forme un conjunto universal de puertas lógicas.
3. El sistema ha de mantener su coherencia cuántica a lo largo del experimento. Como ya hemos visto, el poder de la computación cuántica se basa en el paralelismo cuántico, es decir, poner a los qubits en un estado de superposición. Mantener estos estados no es fácil y existe lo que se denomina tiempo de coherencia, que es el tiempo disponible para ejecutar el algoritmo antes de que se produzca la decoherencia.
4. Ha de poder leerse el estado final del sistema tras el cálculo.
5. El sistema ha de ser escalable. Es decir, debe de haber una forma definida de aumentar el número de qubits para tratar con problemas con mayor coste computacional.

A continuación, vamos a ver dos formas con la que son posibles construir un ordenador cuántico.

2.3.2.1. Trampas de Iones

Los átomos son partículas con carga eléctrica neutra, esto es debido a que tienen la misma cantidad de electrones que de protones. Pero los átomos pueden ganar o perder electrones, con lo que la carga del átomo deja de ser neutra. A estos se les llaman iones, hay dos tipos de iones: iones cationes, cuando la carga resultante es positiva y los iones aniones, cuando la carga resultante es negativa.

Las trampas de iones son dispositivos electrodinámicos que usan un campo eléctrico o magnético para capturar iones en una región de un sistema o tubo al vacío.

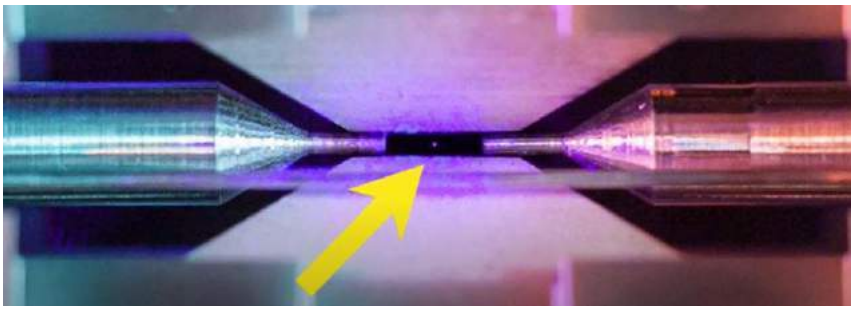

Figura 28: Átomo atrapado en una trampa de iones

Una vez atrapados los iones, hay que enfriarlos para poder manipularlos. Pero no se usa ningún tipo de refrigerador para ello, ya que estamos hablando de la temperatura como una medida relacionada con la energía cinética promedio de las partículas. Al tratarse de una sola partícula, nos basta con dejar la partícula muy quieta. Esto se logra empujándola de un lado y el otro con un láser [44].

Lo que resta para fabricar un ordenador cuántico basado en trampa de iones, es fabricar todos los dispositivos necesarios para ir cumpliendo con las 5 reglas de Di Vincenzo.

1. Para inicializar los qubits, se emplea un proceso llamado bombeo óptico.
2. Para poder manipular los qubits hay dos maneras dependiendo del tipo de qubits que se trate. Por un lado, tenemos las transiciones de dipolo magnéticas y la otra manera se la llama transiciones de cuadripolo eléctricas.
3. Para poder realizar la medición se utiliza un láser que excita a los iones que produce que liberen un fotón cuando se desintegran, estos fotones son capturados por un tubo fotomultiplicador.
4. La escalabilidad es un problema ya que se puede almacenar solo un número finito de qubits en cada trampa de iones mientras conservan sus capacidades computacionales. Por lo que sería necesario interconectar estas trampas de iones de forma que se pueda compartir la información cuántica.
5. La decoherencia se produce cuando los qubits interactúan de forma no deseada con el entorno externo, por lo que su implementación práctica es muy difícil.

2.3.2.2. Circuitos Superconductores

Las computadoras cuánticas basadas en circuitos superconductores son las que más están siendo investigadas por compañías como Google, IBM, Intel y Microsoft.

A diferencia de otras computadoras cuánticas que basan su funcionamiento en partículas individuales, estas computadoras basan su funcionamiento en circuitos superconductores formados por muchos átomos. Estos circuitos son relativamente fáciles de construir y manejar debido a las semejanzas de las técnicas usadas para fabricar los chips convencionales de silicio.

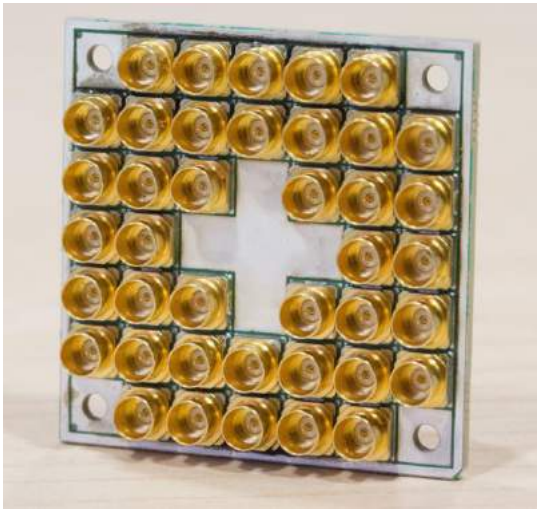

Figura 29: Chip superconductor de 17 qubits fabricado por Intel

Estos circuitos se basan en la unión de dos materiales superconductores donde se genera una súper corriente de pared Cooper. Existen 3 arquetipos de qubits superconductores, que son: los qubits de fase, los qubits de carga y los qubits de flujo.

El principal problema es que para que estos materiales tengan propiedades superconductoras es que necesitan alcanzar temperaturas muy bajas, de aproximadamente 15 mK, lo que sería aproximadamente -273º C, muy cerca del 0 absoluto [45].

Siguiendo las reglas de Di Vincenzo:

1. Inicializar los qubits es bastante sencillo. Simplemente hay que esperar a que los qubits se relajen a su estado fundamental de energía, aunque también existen técnicas de sintonización que permiten inicializarlos de manera más rápida.

2. La manipulación de los qubits superconductores permiten rotaciones arbitrarias en la esfera de Bloch mediante la utilización de señales de microondas pulsadas, que se inducen en una antena o línea de transmisión acoplada al qubit.
3. La medición se suele hacer mediante el acoplamiento de un resonador de microondas, donde la frecuencia de resonancia es desplazada por el estado del qubit.
4. La escalabilidad de esta tecnología es relativamente sencilla. Esto es debido a que los qubits superconductores se fabrican en chips utilizando técnicas similares a las que se utilizan en la fabricación de los chips de silicio convencionales.
5. La decoherencia de los qubits superconductores se ve afectada por múltiples factores. Principalmente debido a las imperfecciones en el sustrato del chip.

2.3.3. Sistemas operativos

Las primeras versiones de ordenadores clásicos no tenían sistemas operativos. En la década de los 60 los ordenadores usaban procesamientos por lotes y fue durante estos años cuando comenzaron a desarrollarse los sistemas operativos.

En la computación cuántica nos encontramos en ese punto, por lo que, por el desarrollo y evolución de esta tecnología, empieza a ser necesario un entorno amigable con el que poder interactuar con la computadora cuántica.

Actualmente no existen sistemas operativos cuánticos en sí, pero empezamos a tener una aproximación de ellos.

2.3.3.1. t|ket>

Fue desarrollado por Cambridge Quantum Computing (CQC), una empresa líder mundial en software de computación cuántica.

t|ket> es una pila de software cuántico independiente de la arquitectura y el mejor compilador de su clase. t|ket> traduce algoritmos independientes de la máquina en circuitos ejecutables, optimizando el diseño físico de qubit y reduciendo el número de operaciones necesarias. El

motor de compilación optimiza las aplicaciones de alto rendimiento, con una interfaz coherente en enrutamiento, optimizaciones y otras transformaciones de circuitos.

t|ket> tiene aplicaciones en química cuántica, aprendizaje automático y en ciberseguridad cuántica [46].

2.3.3.2. LIQUi |>

Microsoft no se podía quedar atrás en la carrera por el desarrollo de software cuántico, con lo que desarrolló LIQUi |>.

Es una arquitectura de software y una suite de herramientas para la computación cuántica. Incluye un lenguaje de programación, algoritmos de optimización y programación y simuladores cuánticos.

LIQUi |> se puede utilizar para traducir un algoritmo cuántico escrito en forma de un programa de alto nivel en las instrucciones de máquina de bajo nivel para un dispositivo cuántico.

Algunos de los algoritmos específicos que se pueden simular con LIQUi |> son:

- Teletransportación cuántica simple.
- Algoritmo de factorización de Shor.
- Química cuántica: calcular la energía del estado fundamental de una molécula.
- Corrección de error cuántico.
- Memoria asociativa cuántica.
- Álgebra lineal cuántica.

Podría parecer que LIQUi |> es un lenguaje de programación entre sentencias normales y datos cuánticos, pero al permitir una comunicación e interacción entre el núcleo cuántico y las instrucciones que procesa, podría definirse en cierta manera que es como un sistema operativo [47].

3. Metodología y herramientas

Para la realización del proyecto se ha hecho uso de metodologías ágiles, que pese a estar orientadas a equipos pequeños, se han utilizado de manera reducida ya que es relativamente difícil aplicar todos los pasos en un proyecto donde únicamente interviene un desarrollador.

Se ha utilizado la metodología de tipo SCRUM mediante el empleo de incrementos entregables. El resto de las características de una metodología SCRUM se simplifican al aplicarse sobre un solo desarrollador.

El desarrollo de la aplicación web ha sido dividido en diferentes *sprints* donde se han ido obteniendo una versión entregable de la aplicación web. Sobre esta versión se han ido añadiendo más funcionalidades en los siguientes *sprints*, llegando así a la versión final de esta aplicación web.

También, se han mantenido reuniones telemáticas por Google Meet cada 15 días con el tutor para el seguimiento del desarrollo tanto de la aplicación web como de la memoria de este trabajo, donde se han ido entregando diferentes versiones, siguiendo de esta manera con la metodología SCRUM.

Por otra parte, cuanto a la metodología empleada para la investigación se ha seguido una metodología siguiendo una serie de procedimientos de colección de datos cualitativos [48]. Estos procedimientos han sido: documentos cualitativos (estudios, trabajos, publicaciones) y materiales digitales (páginas web). Esta metodología de investigación es de carácter crítico-interpretativa revisando los estados producidos por otras personas en su representación bibliográfica conformada por tres fases. La primera se trata de la planificación y diseño, la segunda fase trata la gestión y análisis y, por último, la tercera fase consta de la formalización y elaboración.

La fase de planificación y diseño es donde se condicionan los requisitos y exigencias para realizar el proceso de investigación, se establece el tema a tratar y se realiza una primera exploración documental para seleccionar una serie de fuentes de información.

La fase de gestión y análisis es donde una vez obtenidas una serie de fuentes bibliográficas se realiza una selección de todos los documentos que nos interesen para la elaboración de nuestro trabajo.

La fase de formalización y elaboración es donde se realiza la escritura del trabajo final teniendo ya unas fuentes contrastadas y se formaliza mediante una publicación del trabajo en cualquier media. Es en esta fase donde se ha implementado todo el contenido del estado del arte y donde se ha desarrollado la aplicación web.

Por otro lado, las herramientas utilizadas para la realización de este trabajo han sido:

JIRA

Jira es una herramienta para la administración de tareas de un proyecto, en esta herramienta puedes planificar *sprints* asignando diferentes tareas, llamadas *issues* o incidencias y seleccionando el responsable de cada una de estas tareas. Se establece un nivel de dificultad a cada *sprint* y con ello un tiempo para la realización del *sprint*.

Figura 30: Ejemplo sprint computación desarrollado en la aplicación web didáctica

Los *sprints* han sido planificados y clasificados según su dificultad y tiempo de desarrollo en el que se han establecido entre 1 y 2 semanas por *sprint*. Cada *sprint* contenía una serie de tareas o incidencias que se han ido desarrollando de modo incremental en cuanto a dificultad y dependientes de otras.

Como podemos ver en la imagen anterior, Jira incorpora un sistema de clasificación de las incidencias en el que podemos marcar las incidencias como pendientes, en curso o ya realizadas.

Figura 31: Logo Jira

GitHub

GitHub es un sistema de control de versiones en la nube que permite a los desarrolladores almacenar el código y llevar un registro y control de cualquier cambio sobre este código.

Se ha utilizado esta herramienta por la familiaridad obtenida durante la realización del grado ya que ha sido utilizado en varias asignaturas.

Otro de los motivos de utilizar GitHub, además de por el sistema de control de versiones, es como sistema de seguridad ante cualquier fallo o problema que hubiera podido ocurrir en el ordenador en el que ha sido desarrollado el código, de manera que se habrían minimizado las pérdidas ante cualquier inconveniente.

Figura 32: Logo GitHub

Mockingbird

Se ha utilizado la herramienta en línea Mockingbird para el desarrollo de los Mockups de la aplicación web. Esta herramienta ha sido seleccionada por su sistema de "*drag* and *drop*" de elementos de la interfaz de usuarios en la página y por su gran variedad de elementos para representar una página web.

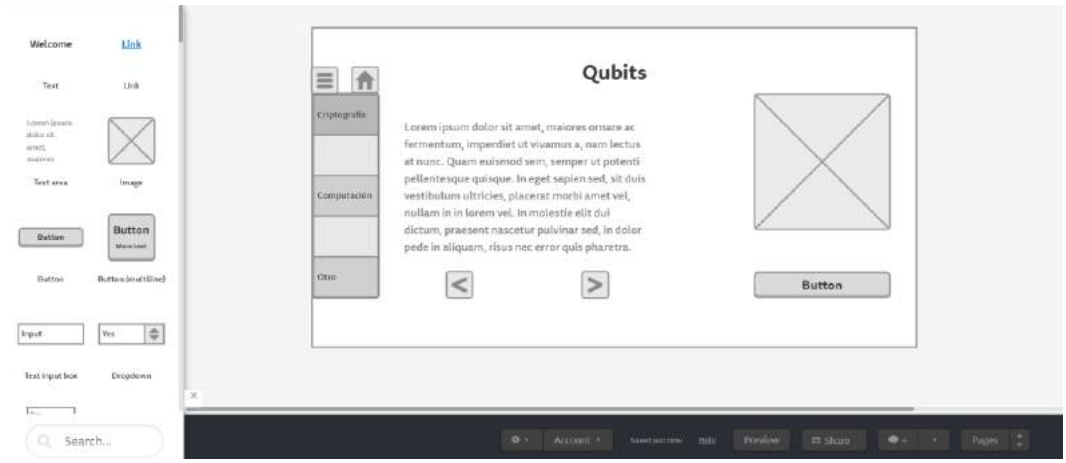

Figura 33: Ejemplo creación de mockup usando la herramienta mockingbird

IntelliJ

Se ha utilizado IntelliJ como entorno de desarrollo debido a que ofrece una experiencia inteligente y ultrarrápida aportando sugerencias relevantes en cada contexto: finalización del código instantánea e inteligente, análisis del código al momento y herramientas de refactorización fiables. IntelliJ es compatible con Maven, Gradle, Ant y otras herramientas de compilación, por lo que ofrece mucha flexibilidad en este aspecto. Además, se puede integrar el sistema de control de versiones, haciendo su uso mucho más cómodo.

Figura 34: Logo IntelliJ

Angular

Para la implementación del código se ha utilizado Angular, que es un framework para aplicaciones web desarrollado en TypeScript, de código abierto y mantenido por Google que se utiliza para crear aplicaciones webs.

Angular adopta el estándar de los componentes web. Estos componentes son un conjunto de APIs que permiten crear nuevas etiquetas HTML personalizadas, reutilizables y auto-contenidas, que luego pueden ser utilizadas en otras páginas y aplicaciones web.

El lenguaje de programación en el que se ha realizado este trabajo es TypeScript para las funcionalidades de la página debido a su baja curva de aprendizaje por sus semejanzas con lenguajes como Java o JavaScript. Por otro lado, se ha utilizado HTML5 y CSS3 para el diseño de las páginas de la aplicación web.

Figura 35: Logo Angular

Node .js

Node .js es un entorno de ejecución multiplataforma que nos permite ejecutar en el servidor, de manera asíncrona, con una arquitectura orientada a eventos nuestra aplicación web. Esta librería se ejecuta sobre JavaScript y ha sido creada por Google.

Se ha hecho uso de esta herramienta debido a su facilidad de uso y porque nos permite visualizar los cambios de código al instante, de esta manera se han podido desarrollar las diferentes páginas de una manera más veloz ya que cada vez que se realiza un cambio se compilan únicamente esos cambios y no todo el proyecto.

Figura 36: Logo node .js

4. Implementación

El primer paso para poder implementar esta aplicación web didáctica ha sido la generación de los Mockups (ver anexos). En estos Mockups se ha establecido una primera fase de distribución de los contenidos expuestos en el estado del arte.

Por otra parte, se ha realizado un proceso de investigación en el que se han ido adquiriendo los conocimientos sobre computación y criptografía cuántica que han sido plasmados en el mencionado estado del arte de este trabajo.

El siguiente paso fue desarrollar la aplicación web siguiendo el patrón generado en los Mockups y empleando las tecnologías mencionadas con anterioridad. Una vez creadas las pantallas se han ido completando con los conocimientos obtenidos en el proceso de investigación, de manera que se pueda visualizar de una manera clara y amigable para el usuario. Estos conocimientos se han ido plasmando en la aplicación web de una manera ordenada y categorizada, dividido en 3 secciones: computación cuántica, criptografía cuántica y otros elementos cuánticos. En cada sección, el usuario encontrará los apartados divididos de la misma manera que en el estado del arte.

La información contenida en la aplicación web es una versión reducida y resumida del contenido de este trabajo, apoyándose en diferentes videos, gráficas e imágenes que ayudan a adquirir conocimientos y ampliar la información. Algunos de estos apartados contienen una serie de ejercicios interactivos, donde el usuario podrá poner en práctica lo aprendido y evaluar así los conocimientos adquiridos.

Todo el proceso de investigación se ha plasmado en una aplicación web interactiva para facilitar la profundización en el tema a los alumnos o usuarios interesados en esta materia.

Se trata, por lo tanto, de una aplicación web didáctica orientada a ampliar el contenido de la asignatura “Estrategias de Seguridad”, cursada durante el cuarto curso del grado en Ingeniería Informática en la especialidad de Tecnologías de la Información de la Universidad de Alicante.

En las siguientes imágenes podemos ver una demo de la aplicación web didáctica desarrollada:

En la figura 37 podemos ver la página de inicio de nuestra web. En esta página nos encontramos con un menú lateral desplegable donde podremos acceder directamente a la sección deseada. En la barra de arriba también tendremos la posibilidad de acceder de forma más rápida a la sección que deseemos consultar.

Por último, destacar que el contenido de esta página se basa en un carrusel de imágenes en la que podemos desplazarnos y acceder a la sección que vayamos a ver.

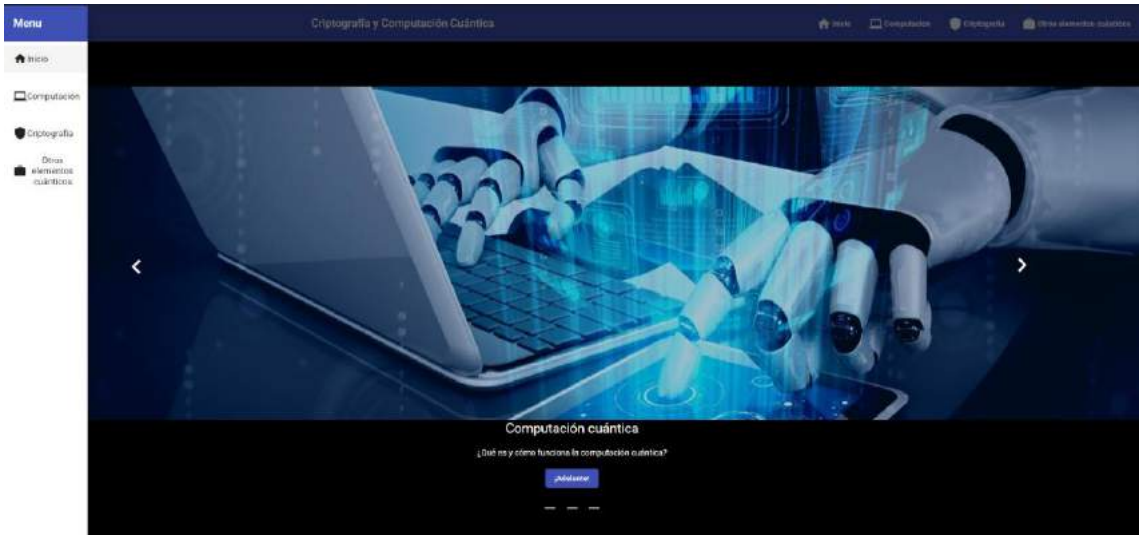

Figura 37: APLICACIÓN WEB: Pantalla de Inicio

La figura 38 se corresponde con la pantalla de computación cuántica, donde podemos ver las diferentes partes de las que está compuesta, pudiendo acceder a su contenido, y, en el caso de que lo tenga, su apartado de evaluación. En esta pantalla tenemos una primera sección donde encontraremos las diferencias entre la computación clásica y la computación cuántica y en la siguiente sección podremos conocer qué es un qubit y cómo se comporta. En cuanto a la parte de abajo nos encontramos con una sección donde se expondrán una serie de algoritmos cuánticos y finalmente un apartado de programación cuántica, donde conoceremos diferentes lenguajes de programación para la computación cuántica.

Figura 38: APLICACIÓN WEB: Página de Computación Cuántica

En la figura 39 podemos ver cómo queda el apartado de algoritmos cuánticos. Vemos que este apartado se apoya en un video extraído de YouTube para ayudar al usuario de una mejor comprensión con ejemplos visuales.

Figura 39: APLICACIÓN WEB: Página de Algoritmos cuánticos

Por otro lado, nos encontramos la pantalla de criptografía cuántica en la figura 40. Esta sección tiene en su parte superior los apartados de diferencias entre la criptografía clásica y la criptografía cuántica y un apartado de seguridad en la criptografía cuántica. En su parte inferior nos encontramos con un apartado sobre los ataques más destacables de la criptografía cuántica, así como una serie de protocolos cuánticos.

Figura 40: APLICACIÓN WEB: Página de Criptografía Cuántica

En la figura 41 podemos ver el resultado de la pantalla de protocolos cuánticos. Como vemos, en esta pantalla nos encontramos una tabla de ejemplo para ayudar a comprender mejor cómo funciona el protocolo BB84.

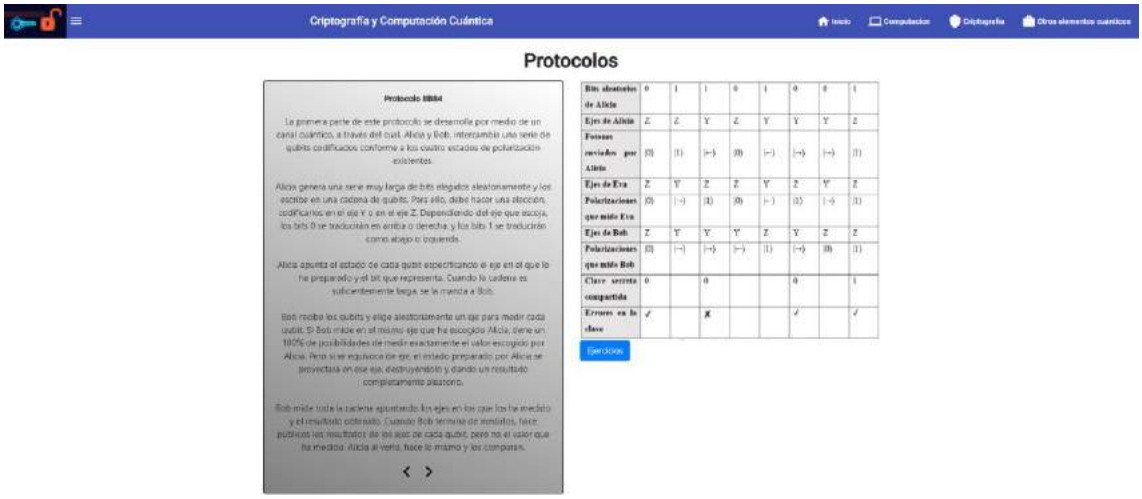

Figura 41: APLICACIÓN WEB: Página de Protocolos cuánticos

Por ultimo nos encontramos en la figura 42 la última sección de esta aplicación web. Esta sección sirve como ampliación del contenido anterior y podemos encontrar un apartado sobre puertas cuánticas, sistemas operativos cuánticos, el hardware utilizado en la computación cuántica y, por último, encontraremos un apartado de evaluación donde se han

agrupado todos los ejercicios propuestos a lo largo de toda la página web y un apartado donde podremos consultar diferentes fuentes a modo de ampliación de conocimientos.

Figura 42: APLICACIÓN WEB: Página de Otros Elementos Cuánticos

En la figura 43 podemos ver un ejemplo de un ejercicio de tipo test en el que nos dan 3 opciones, si el usuario responde correctamente se mostrará esta respuesta en color verde.

Figura 43: APLICACIÓN WEB: Página de ejercicios (acierto)

En la figura 44 podemos ver el caso contrario al anterior, si el usuario falla la respuesta, esta se mostrará en color rojo.

Figura 44: APLICACIÓN WEB: Página de ejercicios (fallo)

Por último, hay que mencionar que en el apartado disponible para ampliar conocimientos (figura 45) nos encontraremos una serie de webs de referencia, donde algunas de ellas se han utilizado como bibliografía para la parte teórica de este trabajo y otras como el simulador de un computador cuántico, el IBM Quantum Experience, sirven al usuario como material adicional para practicar todo lo aprendido en esta aplicación web didáctica.

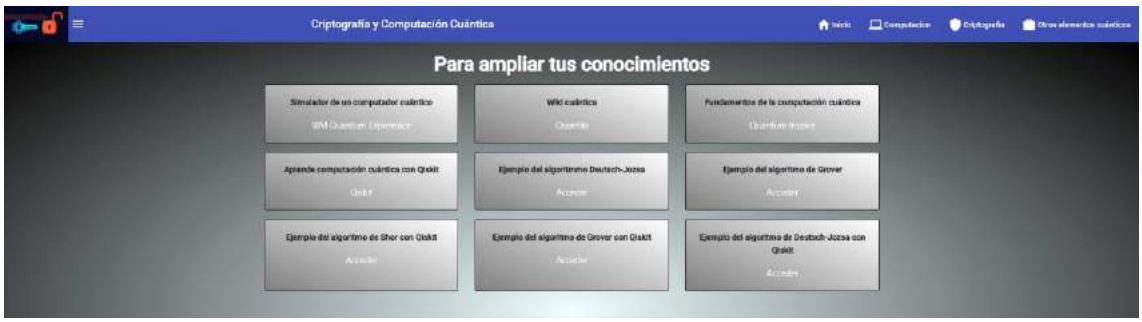

Figura 45: APLICACIÓN WEB: Página con contenido adicional

5. Conclusiones

Una vez llevada a cabo la implementación, ya disponemos de una aplicación web didáctica orientada a la enseñanza de la computación y criptografía cuántica para el alumnado del grado en Ingeniería Informática o de cualquier usuario con conocimientos avanzados en informática.

Con el desarrollo de este trabajo, se ha pretendido aclarar toda la información que hay en torno a la computación y criptografía cuántica, ya que es un campo aún muy nuevo y del que no hay excesiva documentación disponible. De esta manera, se ha agrupado toda la información en un único lugar y ha sido explicado de una manera que pueda entender cualquier persona con conocimientos en computación y criptografía. Dejando a un lado las demostraciones y fórmulas matemáticas que corresponden al campo de la física.

Finalmente, se han llevado a cabo todos los requisitos establecidos y se han cumplido todos los objetivos deseados. Como futuras ampliaciones a este trabajo se podría plantear la implementación de un sistema de interacción entre usuarios a modo de foro, donde los usuarios, en este caso estudiantes, puedan intercambiar opiniones, abrir debates sobre temas concretos, etc. Así como ampliar la batería de ejercicios propuestos.

6. Bibliografía y referencias

[1] M. Planck, Teoría de la ley de distribución de energías del espectro normal.

[2] «Xataka,» [En línea]. Available: https://www.xataka.com/ordenadores/computacion-cuantica-que-es-de-donde-viene-y-que-ha-conseguido .

[3] Dirac, P.A.M, «The principles of Quantum Mechanics,» Oxford University Press, 1930/1958.

[4] R. Restrepo-Villegas, «Entrelazamiento -Un rompecabezas cuántico para todo el mundo.,» 2014.

[5] M. R. Álvarez, «Evolución de la computación cuántica y los retos para la seguridad de la información,» 2016.

[6] «techedge,» [En línea]. Available: https://www.techedgegroup.com/es/blog/introduccion-computacion-cuantica-i.

[7] [En línea]. Available: https://conceptodefinicion.de/computacion/.

[8] L. F. Menabrea, Sketch of the Analytical Engine invented by Charles Babbage, 1843.

[9] [En línea]. Available: https://history-computer.com/.

[10] [En línea]. Available: https://plato.stanford.edu/archives/fall2008/entries/computing-history/#Bab.

[11] [En línea]. Available: https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/.

[12] «BBVA,» [En línea]. Available: https://www.bbva.com/es/computacion-cuantica-en-que-se-diferencia-de-la-computacion-clasica/.

[13] «FayerWayer,» [En línea]. Available: https://www.fayerwayer.com/2013/09/qubit-la-unidad-fundamental-del-futuro-informatico-y-tecnologico/.

[14] S. H. W. Dür, What we can learn about quantum physics from a single qubit, 2013.

[15] «Quantiki,» [En línea]. Available: https://www.quantiki.org/wiki/qubit.

[16] «Quantiki,» [En línea]. Available: https://www.quantiki.org/wiki/shors-factoring-algorithm.

[17] [En línea]. Available: http://www.criptored.upm.es/crypt4you/temas/cuantica/leccion3/leccion03.html.

[18] Bennett C.H. Bernstein E, Brassard G, Vazirani U., The strengths and weaknesses of quantum computation, 1997.

[19] G. L.K., A fast quantum mechanical algorithm for database search, 1996.

[20] G. L.K., From Schrodinger's equation to quantum search algorithm, 2001.

[21] [En línea]. Available: https://www.quantum-inspire.com/kbase/grover-algorithm/.

[22] R. J. David Deutsch, Rapid solutions of problems by quantum computation, 1992.

[23] A. W. Cross, L. S. Bishop, J. A. Smolin y J. M. Gambetta, Open Quantum Assembly Language.

[24] [En línea]. Available: https://qiskit.org/documentation/.

[25] [En línea]. Available: https://docs.microsoft.com/en-us/azure/quantum/overview-what-is-qsharp-and-qdk.

[26] T. G. M. B. M. V. Benjamin Bichsel, Silq: A High-Level Quantum Language with Safe Uncomputation and Intuitive Semantics.

[27] [En línea]. Available: https://www.tecnologia-informatica.com/que-es-la-criptografia/.

[28] M. J. L. López, Criptografía y Seguridad en Computadores.

[29] C. E. Shannon, «Communication Theory of Secrecy Systems,» *The Bell System Technical Journal ,* 1949.

[30] Y. Z. Hoi-Kwong Lo, Quantum Cryptography.

[31] M. Baig, Criptografía Cuántica.

[32] G. V. A. P. Navez, A method for secure transmission: Quantum, 2002.

[33] A. A. G. R. N. G. V. Scarani, “Quantum Cryptography Protocols Robus against Photon Number Splitting Attack for Weak Laser Pulse Implementations, 2004.

[34] S. J. LomonacoJr., A Talk on Quantum Cryptography or How Alice Outwits Eve.

[35] S. F. B. K. H. Z. G. R. N. Gisin, Trojan Horse attacks on Quantum Key Distribution systems, 2008.

[36] D. Mayers, “Unconditional Security in Quantum Cryptography, 1998.

[37] T. M. Eli Biham, Bounds on Information and the Security of Quantum Cryptography, 1997.

[38] B. Q. K. T. H.-K. L. Chi-Hang Fred Fung, Phase-Remapping Attack in Practical Quantum Key Distribution Systems.

[39] H. -K. L. N. L. J. P. D. Gottesman, Security of quantum key distribution with imperfect devices, 2004.

[40] [En línea]. Available: https://www.quantiki.org/wiki/quantum-gates.

[41] V. M. Bonillo, Principios fundamentales de computación cuántica, 2013.

[42] E. S. d. Cabezón, «Introducción a la computación cuántica (sin la física),» La Rioja, 2019.

[43] D. P. DiVincenzo, The Physical Implementation of Quantum Computation, 2008.

[44] J. R. S. C. W. K. L. A. E. W. I. C. T. N. B. L. A. R. K. H. S. Weidt, Trapped-ion quantum logic with global radiation fields.

[45] [En línea]. Available: https://www.sacyr.com/-/asi-se-construye-un-computador-cuantico.

[46] [En línea]. Available: https://cambridgequantum.com/technology/.

[47] [En línea]. Available: https://www.microsoft.com/en-us/research/project/language-integrated-quantum-operations-liqui/.

[48] C. G. H. D. A. J. M. Maricelly Gómez Vargas, El estado del arte: una metodología de investigación, 2015.

7. Anexos

Mockups de la aplicación web:

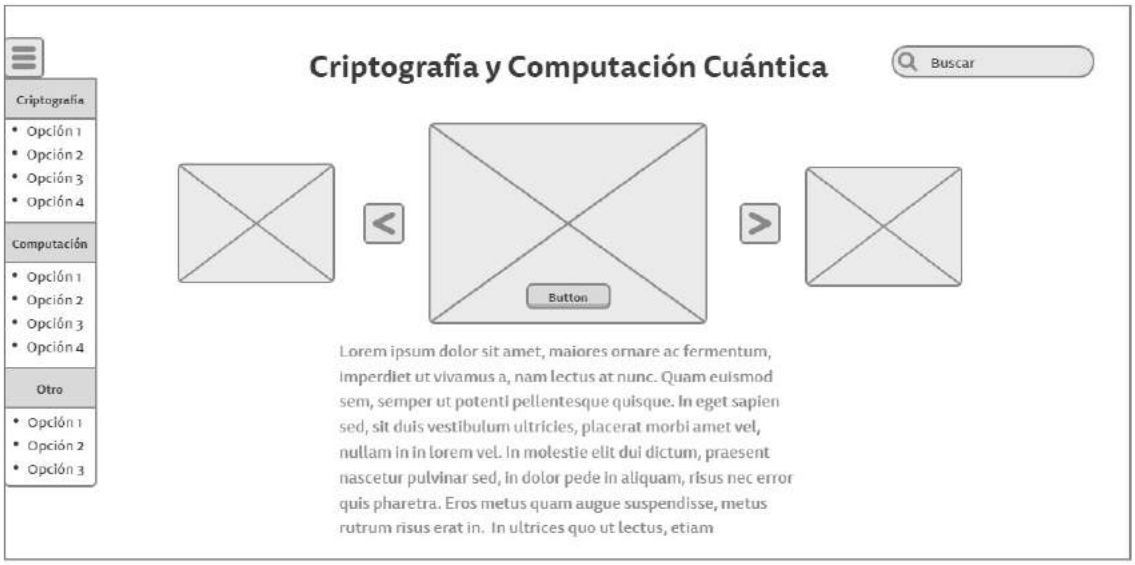

Figura 46: MOCKUPS: Pantalla de inicio

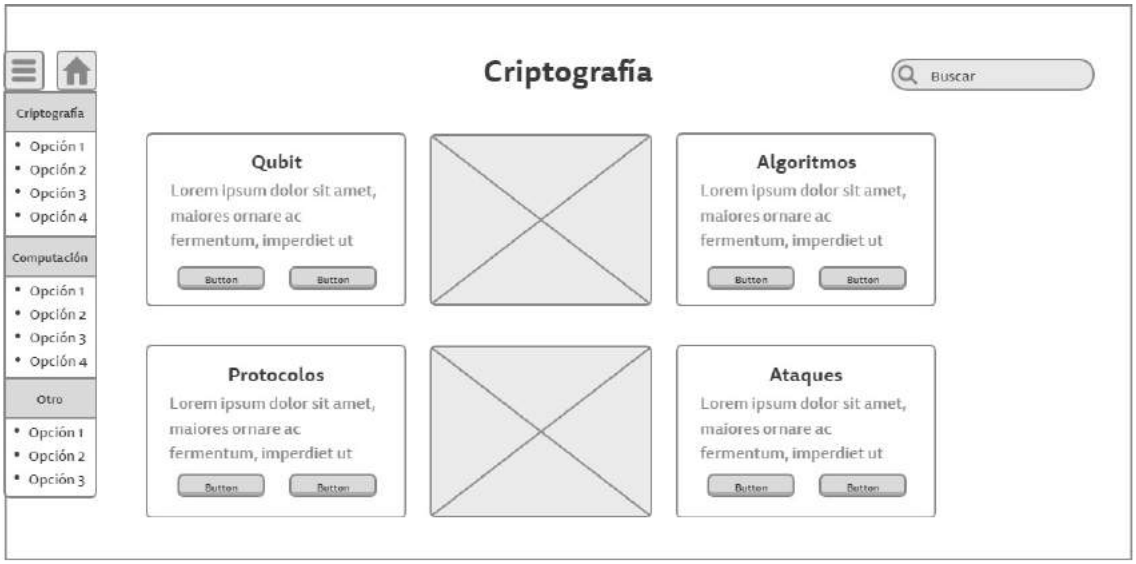

Figura 47: MOCKUPS: Pantalla Criptografía

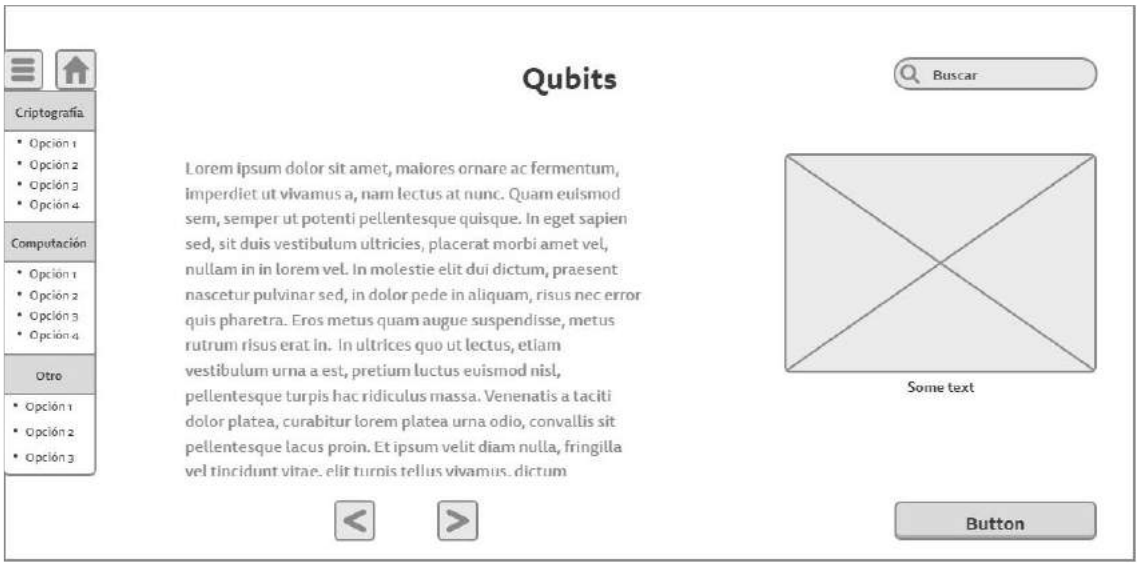

Figura 48: MOCKUPS: Pantalla Qubits

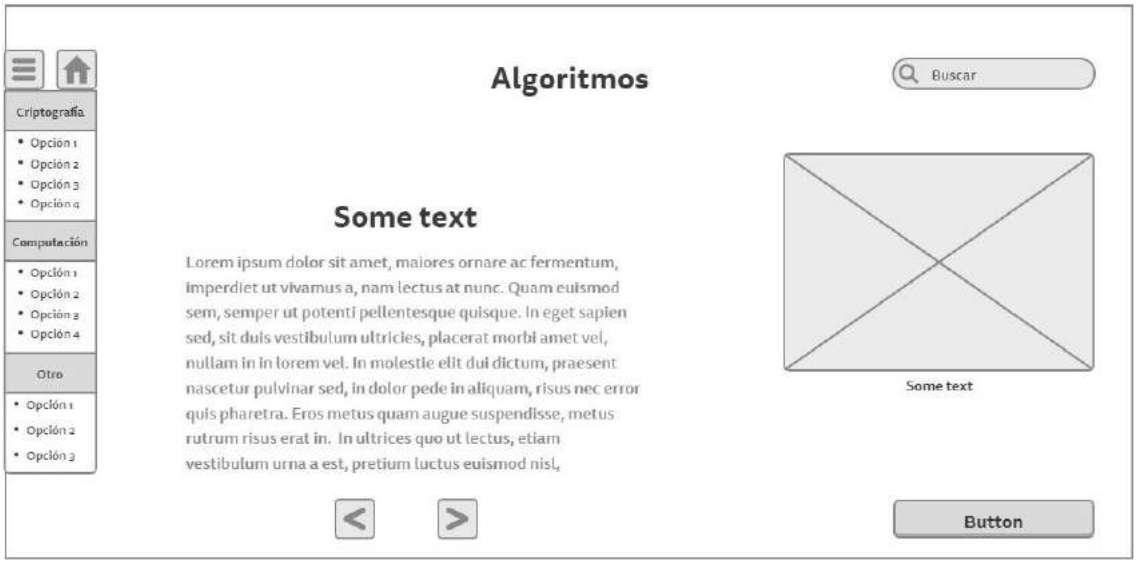

Figura 49: MOCKUPS: Pantalla Algoritmos

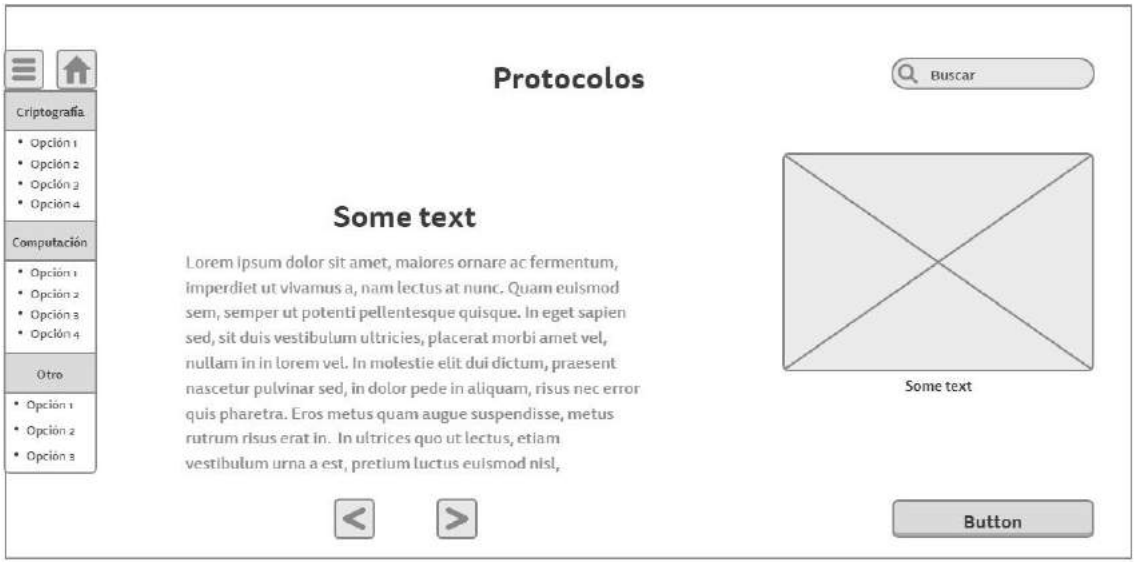

Figura 50: MOCKUPS: Pantalla Protocolos

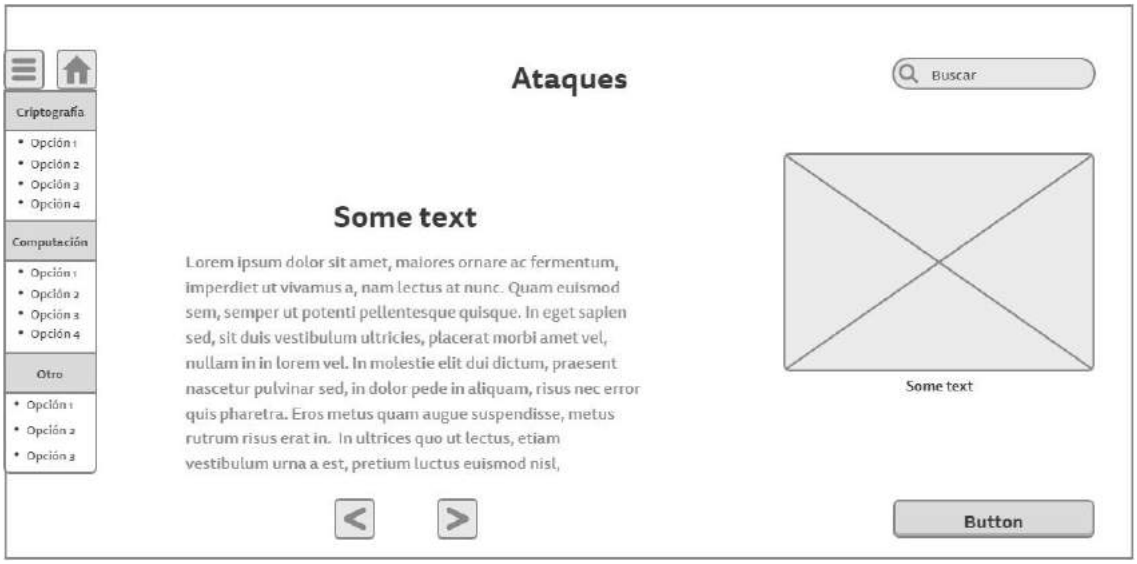

Figura 51: MOCKUPS: Pantalla Ataques

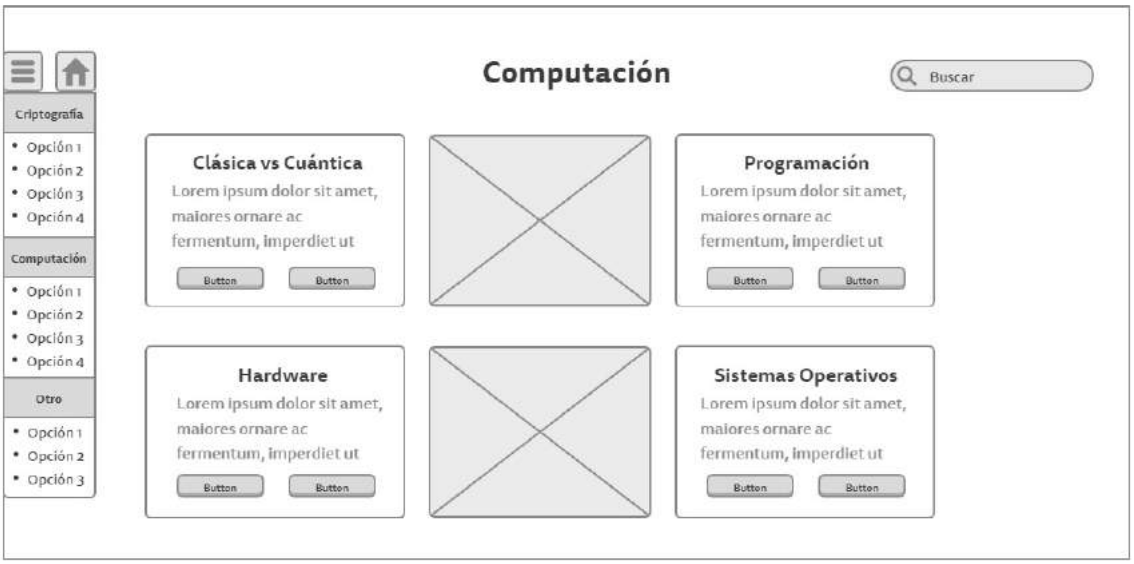

Figura 52: MOCKUPS: Pantalla Computación

Figura 53: MOCKUPS: Pantalla Clásica vs Cuántica

Figura 54: MOCKUPS: Pantalla Programación

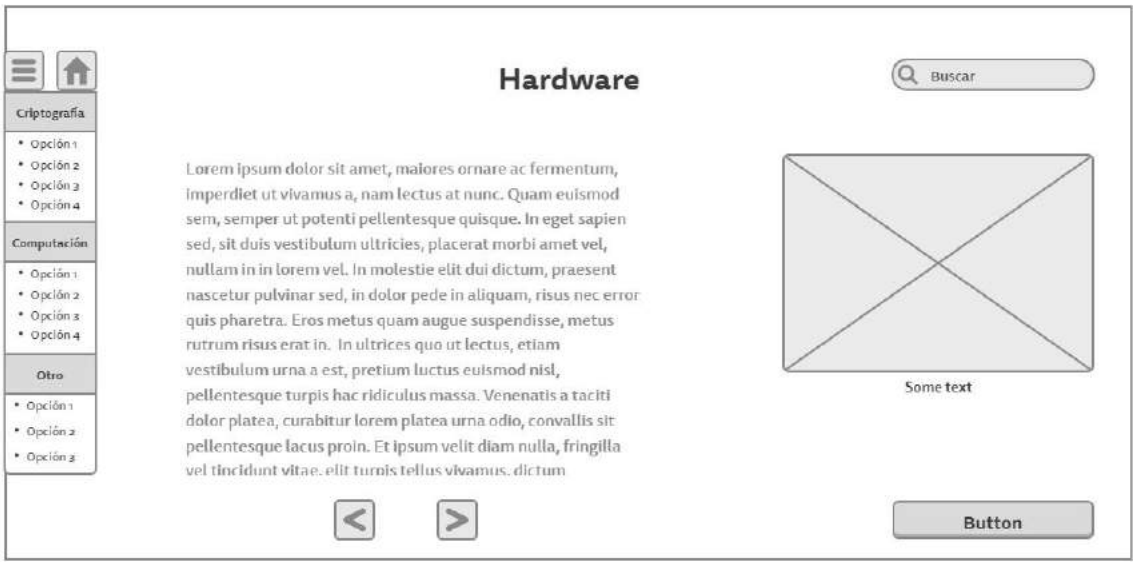

Figura 55: MOCKUPS: Pantalla Hardware

Figura 56: MOCKUPS: Pantalla Sistemas Operativos

Figura 57: MOCKUPS: Pantalla Otros Elementos Cuánticos

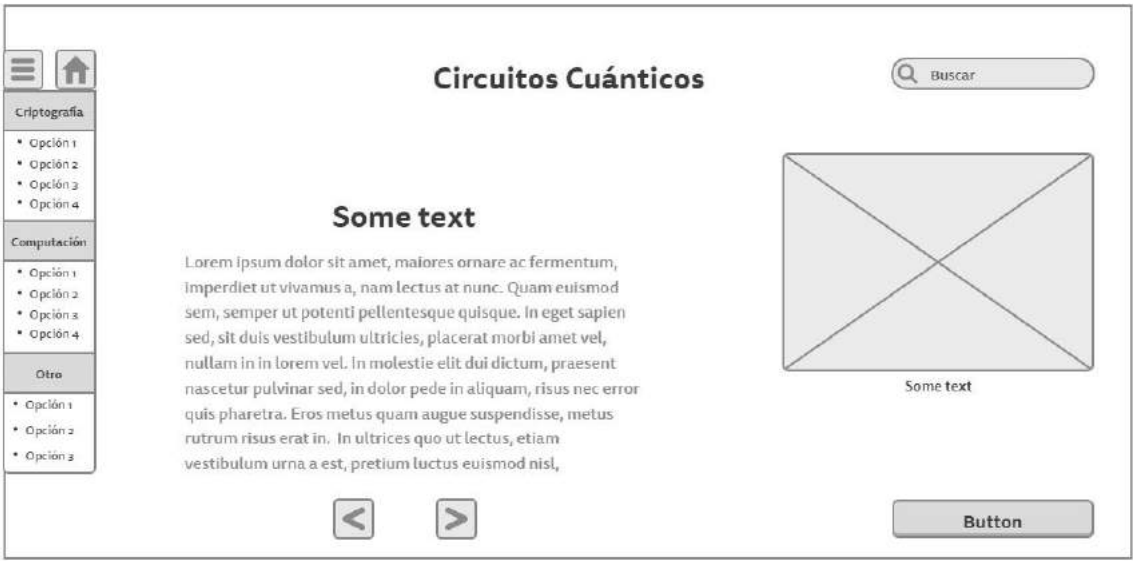

Figura 58: MOCKUPS: Pantalla Circuitos Cuánticos

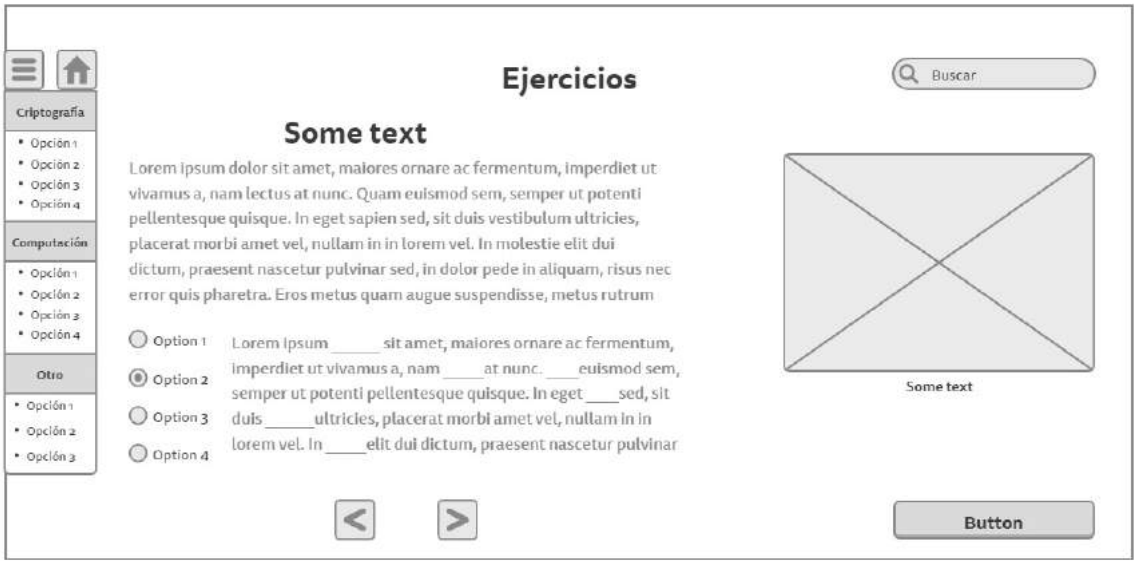

Figura 59: MOCKUPS: Pantalla Ejercicios

Figura 60: MOCKUPS: Pantalla Bibliografía

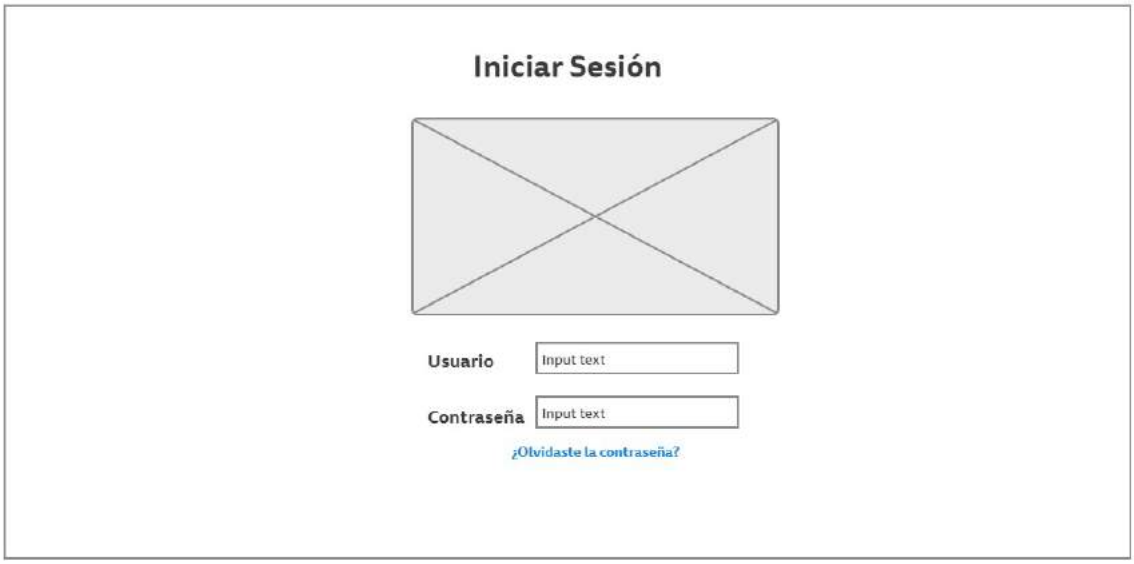

Figura 61: MOCKUPS (posible ampliación): Pantalla Inicio de Sesión

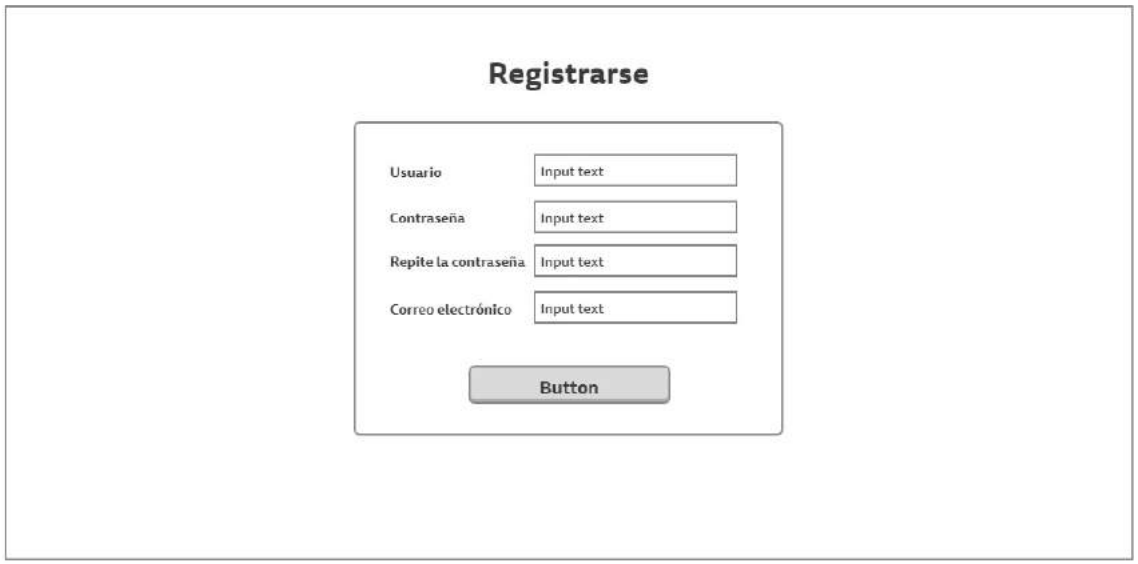

Figura 62: MOCKUPS (posible ampliación): Pantalla Registro Usuario