



Universidad del  
**Rosario**

*Escuela de Ciencias e  
Ingeniería*

**APLICACIÓN DEL ALGORITMO DE GROVER EN  
BÚSQUEDAS NO ESTRUCTURADAS: ANÁLISIS Y  
COMPARACIÓN DE LA EFICIENCIA DE ALGORITMOS  
DE BÚSQUEDA CON ALGORITMOS CUÁNTICOS.**

Presentado para obtener el título de:

**MAGÍSTER EN INGENIERÍA, CIENCIA Y TECNOLOGÍA**

**AUTOR**

Stiven Paloma Alarcon

Dirección:

**Ph.D. Nicolás Avilán Vargas**

UNIVERSIDAD DEL ROSARIO

ESCUELA DE CIENCIAS E INGENIERÍA

MAESTRÍA EN INGENIERÍA, CIENCIA Y TECNOLOGÍA

BOGOTÁ

2025

# Dedicatoria

Primero, a la Vida y al Universo, que me guían en cada paso que avanzo en esta vida profesional y personal; a mi familia, por brindarme los únicos y grandes consejos que hicieron de mí un mejor ser humano; y especialmente a mi amada novia Tatiana, quien fue mi compañía y mi gran motivación para continuar con este desafío a pesar de las dificultades, lo cual permitió culminar con éxito la presente investigación.

# Agradecimientos

A los integrantes del Semillero de Computación Cuántica de la Universidad del Rosario, por su valioso acompañamiento, el intercambio de ideas y el entusiasmo que fortaleció el desarrollo de esta investigación. Al profesor Nicolás Avilán, Doctor de la Universidad del Rosario, por su orientación académica, su constante disposición para guiar este trabajo y por inspirar con su compromiso y conocimiento el avance de la computación cuántica en nuestro entorno académico. Finalmente, a todas las personas que, de una u otra forma, contribuyeron con su tiempo, sus palabras y su ánimo para que este proceso culminara con éxito.

# Resumen

El objetivo de esta investigación es proponer una implementación del algoritmo de Grover, un método cuántico que permite reducir significativamente el tiempo de búsqueda en bases de datos no estructuradas. Se aplicó una metodología que integró diferentes experimentos en simuladores y una computadora cuántica en el cual se incluyó el análisis del algoritmo cuántico, y se compara con los algoritmos clásicos deterministas en diferentes escenarios de búsquedas. Se estructura la arquitectura del algoritmo ajustando los parámetros que lo constituyen y tipo de búsqueda. Por ende, dando una versión de los fundamentos de la computación cuántica, destacando los principios de superposición y entrelazamiento como elementos clave para abordar problemas de búsquedas en listas no ordenadas con tecnologías cuánticas. Asimismo, se introducen los conceptos de qubits, compuertas cuánticas, circuitos cuánticos, y se describe la interpretación geométrica de su acción sobre los estados cuánticos mediante la esfera de Bloch. Se detalla el funcionamiento del algoritmo de Grover con la utilización de simuladores cuánticos y su escalabilidad en la computadora cuántica *ibm\_brisbane*, para la estructuración de la solución a problemas de búsqueda en listas no ordenadas, explicando cómo logra una aceleración cuadrática al reducir la complejidad de búsqueda de  $O(N)$  a  $O(\sqrt{N})$ . Esta investigación siguió un enfoque cuantitativo, experimental y comparativo, visualizando como este algoritmo representa una ventaja computacional significativa en la resolución de problemas de búsqueda en estructuras de datos no organizadas y con el objeto experimental del tamaño de la búsqueda.

# Abstract

The objective of this research is to propose an implementation of Grover's algorithm, a quantum method that significantly reduces search time in unstructured databases. A methodology was applied that integrated different experiments on simulators and a quantum computer, which included the analysis of the quantum algorithm, comparing it with classical deterministic algorithms in different search scenarios. The algorithm's architecture is structured by adjusting its constituent parameters and the search type. Consequently, the study presents an overview of the fundamentals of quantum computing, highlighting the principles of superposition and entanglement as key elements for addressing search problems in unordered lists using quantum technologies. Likewise, it introduces the concepts of qubits, quantum gates, and quantum circuits, and describes the geometric interpretation of their action on quantum states through the Bloch sphere. The operation of Grover's algorithm is detailed using quantum simulators and its scalability on the quantum computer *ibm\_brisbane*, to structure solutions to search problems in unordered lists, explaining how it achieves quadratic speedup by reducing search complexity from  $O(N)$  to  $O(\sqrt{N})$ . This research followed a quantitative, experimental, and comparative approach, visualizing how this algorithm represents a significant computational advantage in solving search problems in unorganized data structures, with the experimental object being the size of the search.

# TABLA DE CONTENIDO

Lista de Tablas . . . . .	7
Lista de Figuras . . . . .	8
<b>1 INTRODUCCIÓN</b>	<b>10</b>
<b>2 OBJETIVOS</b>	<b>12</b>
2.1 Objetivo General . . . . .	12
2.2 Objetivos Específicos . . . . .	12
<b>3 PROBLEMA Y JUSTIFICACIÓN</b>	<b>13</b>
3.1 Limitación en la búsqueda en bases de datos no estructuradas	13
3.2 3-SAT: Un Caso de Estudio para la Búsqueda en Datos no Estructurados . . . . .	14
3.3 Algoritmo de búsqueda cuántico y la ventaja computacional .	15
3.4 Hipótesis y Alcance . . . . .	18
<b>4 Marco Teórico y Estado del Arte</b>	<b>19</b>
4.1 El Problema de la Búsqueda No Estructurada . . . . .	20
4.2 Computación Cuántica . . . . .	20
4.2.1 Entrelazamiento y Compuertas Cuánticas . . . . .	20
4.2.2 Interpretación Geométrica . . . . .	22
4.2.3 Circuito cuántico compuesto por las compuertas Hadamard, CNOT y Z . . . . .	23
4.3 Algoritmo de Grover . . . . .	26
4.4 Búsqueda de Grover . . . . .	26
4.4.1 Implementaciones y Verificaciones Experimentales . . . . .	34
<b>5 Metodología</b>	<b>37</b>
5.1 Implementación de una búsqueda de Grover . . . . .	37
5.1.1 Inicialización . . . . .	37
5.1.2 Oráculo Elemental . . . . .	42
5.1.3 Difusor . . . . .	43

5.1.4	Ensamble del Circuito de Grover . . . . .	45
5.1.5	Ejecución del Circuito en Computadora Cuántica . . . . .	46
5.1.6	Experimento en Simulador de Qiskit para un problema 3-SAT . . . . .	47
<b>6</b>	<b>Resultados y Discusión</b>	<b>50</b>
6.0.1	Computador Cuántico <i>ibm_brisbane</i> . . . . .	50
6.0.2	Simulador Cuántico <i>AerSimulator</i> , Tiempo y Búsqueda Clásica . . . . .	54
<b>7</b>	<b>Conclusiones y Recomendaciones</b>	<b>59</b>
<b>8</b>	<b>Anexos</b>	<b>65</b>

# LISTA DE TABLAS

4.1	Compuertas cuánticas básicas . . . . .	21
5.1	Resumen de compuertas cuánticas nativas utilizadas en la implementación	40
6.1	Resultados obtenidos del algoritmo cuántico en <i>ibm_brisbane</i> . . . . .	50
6.2	Resultados de la simulación del algoritmo cuántico <i>ibm_quantum_simulator</i>	51

# Lista de Figuras

3.1	<i>Mapa conceptual de las áreas donde la computación cuántica es aplicada. Elaboración propia.</i>	17
3.2	<i>Modelo Conceptual para la Evaluación Comparativa del Algoritmo de Grover en Búsquedas No Estructuradas: Variables Causales, Moderadoras (<math>x_1</math>) y Relación con Métodos Clásicos de Búsqueda. Elaboración propia.</i>	18
4.1	<i>Esfera de Bloch. Basado en documentación Qiskit.</i>	23
4.2	<i>Circuito Hadamard, CNOT y Z. Basado en documentación Qiskit.</i>	24
4.3	<i>Inicialización del Algoritmo de Grover. Elaboración basada en Qiskit [27, 28].</i>	27
4.4	<i>Oráculo Actuando sobre <math> \omega\rangle</math>. Elaboración basada en Qiskit [27, 28].</i>	28
4.5	<i>Reflejar Amplitud Sobre la Amplitud Media. Elaboración basada en Qiskit [27, 28].</i>	31
4.6	<i>Circuito cuántico de Grover, diseñado para verificar las restricciones de un problema CSP-I. El circuito marca un estado como “solución” (aplicando un cambio de fase) solo si cumple todas las restricciones. Tomado de Vinod &amp; Shaji [29].</i>	35
5.1	<i>Topología y conectividad del procesador cuántico ibm_brisbane 127 qubits. Imagen exportada desde IBM Quantum Platform, vista de calibración del backend. Fecha de acceso: 20 de octubre de 2025.</i>	40
5.2	<i>Calibración del procesador cuántico ibm_brisbane 127 qubits. Imagen exportada desde IBM Quantum Platform, vista de calibración del backend. Fecha de acceso: 20 de octubre de 2025.</i>	40
5.3	<i>Circuito de Grover Estructurado para ibm_brisbane. Basado en documentación de Qiskit [38].</i>	44
5.4	<i>Primera parte del Circuito de Grover Estructurado para un problema 3-SAT. Diagrama de resultado [44].</i>	48
5.5	<i>Segunda parte del Circuito de Grover Estructurado para un problema 3-SAT. Diagrama de resultado [44].</i>	49

6.1	<i>Histograma para 9 qubits en ibm_brisbane. Resultados del trabajo [47].</i>	52
6.2	<i>Histograma para 7 qubits en ibm_brisbane. Resultados del trabajo [47].</i>	52
6.3	<i>Histograma para 4 qubits en ibm_brisbane. Resultados del trabajo [47].</i>	53
6.4	<i>Histograma Simulador Qiskit para 9 qubits en AerSimulator. Resultados del trabajo [47]. . . . .</i>	53
6.5	<i>Histograma Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo [47]. . . . .</i>	54
6.6	<i>Matriz de Densidad Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo [47]. . . . .</i>	55
6.7	<i>City Plot de Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo para 1 iteración [47]. . . . .</i>	56
6.8	<i>City Plot de Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo para 3 iteraciones [47]. . . . .</i>	56
6.9	<i>Imagen IBM Quantum Platform para todos los experimentos ibm_brisbane. Resultados del trabajo [47]. . . . .</i>	57
6.10	<i>Imagen Tiempo Búsqueda Clásica. Resultados del trabajo [47]. . . . .</i>	58

## Capítulo 1

# INTRODUCCIÓN

El estado actual de la Computación Cuántica representa un hito en el desarrollo de tecnologías orientadas a la resolución de problemas complejos de procesamiento y optimización que, en el contexto de la computación clásica, presentan diversas limitaciones en eficiencia y capacidad de cálculo [1]. A través de principios cuánticos, como la superposición y el entrelazamiento, es posible abordar problemas de alta complejidad de manera más eficiente, ampliando las posibilidades en áreas donde la velocidad de procesamiento son cruciales [2].

Un desafío en la teoría de la computación radica en la ausencia de un orden subyacente en ciertos problemas, algo que limita la aplicación de técnicas clásicas de optimización que dependen de estructuras jerárquicas o definidas [3]. Desde una perspectiva conceptual se puede vincular en sistemas donde el comportamiento de ordenamiento observable de un sistema surge de una estructura más profunda y no evidente. Partiendo de lo anterior y acotando los desafíos que más persisten en la computación es la búsqueda en bases de datos no estructuradas, algo que consiste en encontrar un elemento específico dentro de una lista que carece o tiene ausencia de este orden predefinido. Si lo visualizamos desde una perspectiva clásica o determinista la solución al problema está intrínsecamente limitada, una búsqueda en una lista en desorden una vez ésta crece de forma lineal el nivel de operaciones se vuelven cada vez más complejas topándose con una limitante de las tecnologías clásicas ya que sin un índice o una estructura guiada el método más eficiente requerirá explorar todos los elementos  $O(N)$  en el peor de los casos, lo que equivale a la cantidad de operaciones que realiza la búsqueda en el tamaño de la base de datos, esto se refleja cada vez más en medida de tamaño de datos en billones de registros, volviendo imposible en términos de tiempo y recursos computacionales. Esta limitante explorada por investigadores como Knuth [4] ó Cormen [5] demuestran una barrera para los algoritmos clásicos.

En este contexto, el algoritmo de Grover, desarrollado por Lov Grover en 1996 y siendo demostrada su ventaja computacional en 1997, se destaca por su capacidad para reducir el tiempo de búsqueda de  $O(N)$  a  $O(\sqrt{N})$ , ofreciendo una aceleración cuadrática frente a los algoritmos clásicos utilizados para la solución de búsquedas en listas no ordenadas y bases de datos no estructuradas [6, 7]. Esta ventaja cuántica, ya que logra una aceleración cuadrática frente a los algoritmos clásicos, tiene implicaciones significativas en problemas altamente complejos siendo demostrables y óptimos en cualquier contexto clásico [8].

A diferencia de las búsquedas clásicas, ampliamente estudiadas por investigadores

como Knuth, Cormen, Vitter y Flajolet [4, 5, 9], el algoritmo de Grover aprovecha propiedades únicas de la computación cuántica, como la superposición y el entrelazamiento, para amplificar probabilidades de éxito. Grover aplica operaciones unitarias sobre un estado puro, seguidas de una medición, explotando así propiedades únicas de la computación cuántica [7]. El proceso inicia realizando una superposición de estados cuánticos sobre todos los elementos de la base de datos, con ayuda de compuertas cuánticas, seguido se aplica un Oráculo que se encarga de marcar o revelar el estado objetivo con el fin de que en el siguiente paso se pueda amplificar su probabilidad de ser encontrado gracias al operador Difusión. Esto se logra con la iteración óptima del circuito cuántico, dada por  $t \approx \frac{\pi}{4}\sqrt{N}$  que al final gracias al Difusor ampliamos la probabilidad de medir el estado correcto, con esto, logrando una ventaja cuadrática sobre una búsqueda no estructurada o lista en desorden.

Variedad de algoritmos cuánticos relevantes como el de Shor, que resuelven operaciones y problemas de gran complejidad, son probabilísticos, este algoritmo también exhibe dicho comportamiento [10, 11]. Avances de este nivel no solo representa un logro teórico, sino también un cambio de paradigma con aplicaciones prácticas en dominios donde la eficiencia computacional es crítica.

Durante las últimas décadas se ha implementado la búsqueda de Grover en diversos simuladores para el desarrollo de computación cuántica, siendo estos gestionados por diferentes empresas como Google, Microsoft, IBM entre otros [12, 13, 14], aunque variando el tamaño de búsqueda  $N$  en diferentes propuestas siendo pequeños estos experimentos. Existen estudios en los que se proponen pruebas experimentales en  $n = 8$  qubits, con probabilidades de éxito comparadas a la clásica no demostrables [12], como otras en las que se demuestra una probabilidad de éxito con  $n \leq 5$  qubits [15].

Esta investigación busca analizar, implementar, comparar y evaluar el Algoritmo de Grover especialmente en búsquedas no estructuradas, con la utilización de herramientas de simulación como Qiskit, la cual nos permite diseñar el circuito y poder correr experimentos con la estructura del algoritmo de Grover, con esto poder observar de cerca la forma de solucionar estos algoritmos y poder realizar una comparativa con algoritmos clásicos deterministas, además de explorar los desafíos inherentes a la implementación del algoritmo en la actualidad, incluyendo el diseño del oráculo cuántico específico para una búsqueda no estructurada, la preservación de la coherencia cuántica durante múltiples iteraciones y cómo estas afectan una vez se parametrizan en el hardware cuántico actual. La propuesta no solo contribuirá al desarrollo de métodos algorítmicos más eficientes, sino que también ofrecerá una comprensión más profunda del impacto de la computación cuántica en la resolución de problemas de alta complejidad.

## Capítulo 2

# OBJETIVOS

### 2.1. Objetivo General

Comparar la eficiencia del algoritmo de Grover con algoritmos clásicos para la solución de búsquedas en bases de datos no estructuradas, con apoyo de entornos para la computación cuántica de Qiskit. Consolidando el desarrollo, entendimiento de sus principios teóricos y profundizando en aplicaciones prácticas con circuitos cuánticos en arquitecturas específicas.

### 2.2. Objetivos Específicos

- Identificar los conceptos fundamentales de la computación cuántica para comprender sus principios teóricos, como la superposición, el entrelazamiento y las operaciones con qubits, y su impacto en la resolución de problemas de búsqueda en bases de datos no estructuradas.
- Examinar la complejidad computacional de los algoritmos de búsqueda en bases de datos no estructuradas en la computación clásica, concibiendo su escalabilidad y eficiencia en el contexto clásico.
- Evaluar el desempeño del algoritmo cuántico de Grover como una alternativa de solución a problemas de búsqueda en bases de datos no ordenadas, considerando aspectos como el diseño del circuito cuántico, la complejidad del desarrollo de su estructura lógica y su comparación con algoritmos clásicos equivalentes.
- Realizar experimentos implementando el algoritmo de grover en una computadora cuántica de Qiskit para estimar su comportamiento en la resolución de un problema de búsqueda en una base de datos no estructurada.

## PROBLEMA Y JUSTIFICACIÓN

### 3.1. Limitación en la búsqueda en bases de datos no estructuradas

Los algoritmos de búsqueda en tecnologías deterministas tienen una limitante computacional que ha sido demostrada mediante un riguroso análisis matemático, esto se logra definiendo un algoritmo de búsqueda secuencial que al medir su eficiencia mediante procesos cuantizados demuestra que para una lista no estructurada del tamaño  $N$ , el peor escenario será fallido o requerirá explorar  $N$  comparaciones, creciendo este mismo linealmente de acuerdo al tamaño de entrada, estableciendo así una complejidad  $O(N)$ [4].

En obras como “Introduction to Algorithms” de Thomas H. Cormen, Charles E. Leiserson [5], establecen este tipo de barreras computacionales para búsquedas en datos no estructuradas, se analizan algoritmos de búsqueda lineal y demuestran que para encontrar un elemento en una secuencia de datos en un tamaño  $N$  el número de operaciones que debe realizar es directamente proporcional a este  $N$ , lo que exhibe de nuevo una complejidad temporal  $O(N)$ .

Contrastando el tipo de búsquedas que se analizan en esta investigación nos encontramos con un tipo de búsqueda que son eficientes en el contexto determinista, como lo son las búsquedas binarias (binary search), en la misma obra se introduce este concepto siendo partidario de una mejora en el campo determinista, que este tipo de búsqueda es mucho más rápida cuya complejidad es  $\Theta(\log N)$  pero teniendo una casuística, el conjunto de datos en un tamaño  $N$  debe estar estrictamente ordenado [5].

En la computación clásica determinista, la búsqueda de un elemento  $x$  en un conjunto de datos no estructurado, tal que  $S = \{s_1, s_2, \dots, s_N\}$ , requiere en el peor de los casos poder recorrer todos sus elementos sin tener una información previa sobre su estructuración u ordenación. Podemos definir el problema como se observa en la siguiente ecuación:

$$search(S, x) = \begin{cases} i & \text{si } \exists i \text{ tal que } s_i = x, \\ \emptyset & \text{en otro caso.} \end{cases} \quad (3.1)$$

Con una complejidad en el peor caso  $T(N) = O(N)$ , entendemos a  $T$  como ese factor de tiempo que afecta el proceso de búsqueda una vez el tamaño de búsqueda crece linealmente en el ámbito de entrada. Podemos determinar de acuerdo a estas investigaciones que en la computación clásica determinista, no es posible reducir este

costo sin información adicional sobre la estructura del conjunto de los datos [4].

## 3.2. 3-SAT: Un Caso de Estudio para la Búsqueda en Datos no Estructurados

Una instancia del problema 3-SAT es una variante específica de los problemas de satisfacibilidad booleana (SAT), dado que conforma una clase fundamental de la complejidad computacional y se define como un problema de decisión que determina si existe una asignación de valores de verdad que satisfaga una fórmula booleana dada, como problema NP-Completo, es un caso representativo dentro de esta clase de complejidad siendo el primer problema para el cual se demostró completitud NP [16]. De modo que, la instancia 3-SAT se estructura mediante una fórmula  $F$  que es una conjunción ( $\wedge$ ) de  $m$  cláusulas ( $C_1 \wedge C_2 \wedge \dots \wedge C_m$ ) en Forma Normal Conjuntiva (FNC). Cada cláusula  $C_j$  debe contener exactamente tres literales conectados por disyunciones ( $\vee$ ). Un literal ( $l_{ji}$ ) es la expresión lógica de una variable de la instancia  $v_i$  o su negación ( $\neg v_i$ ), un literal pertenece al conjunto  $\{\mathbf{v}_i, \neg \mathbf{v}_i\}$  [17]. Cada literal representa una asignación de valor de verdad específica para una variable, su conjunto como en 3.2 define el espacio de condiciones elementales y esta combinación en 3-SAT forma cláusulas de longitud tres, estas cláusulas es la disyunción lógica que agrupan los tres literales como en la ecuación 3.3 por lo cual estructura el caso específico 3-SAT.

Por ejemplo, sea el conjunto de variables:

$$V = \{x, y, z, w\} \quad (3.2)$$

Y la siguiente instancia del problema 3-SAT:

$$F = (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (\neg x \vee \neg y \vee \neg w) \wedge (y \vee z \vee w) \quad (3.3)$$

La instancia 3.3 es distinta a SAT porque éstas pueden tener entre 1 o más literales, también dependiendo de las que se definan sin limitación, algo que en 3-SAT son restringidas, solo pueden escribirse cláusulas de tres literales sin dejar de ser un NP-Completo.

La clase NP comprende aquellos problemas de decisión cuyas instancias positivas pueden verificarse en tiempo polinomial. En el contexto específico de 3-SAT, dado un certificado consistente en una asignación de valores de verdad para las variables, es posible verificar su validez comprobando que satisface todas las cláusulas del problema [18]. El proceso de verificación requiere evaluar cada cláusula individualmente, demostrando cómo propiedades globales de satisfacibilidad emergen de la composición de verifica-

ciones locales. Esta verificación puede realizarse en tiempo polinomial proporcional al número de cláusulas, específicamente en  $O(|C|)$  donde  $|C|$  representa el número total de cláusulas, dado que cada cláusula contiene exactamente tres literales y su evaluación requiere tiempo constante [16].

Para instancias negativas de 3-SAT, no existe certificado alguno que permita su verificación positiva; el verificador rechazará cualquier certificado propuesto. Por consiguiente, 3-SAT pertenece a la clase NP, ya que las instancias positivas pueden verificarse en tiempo polinomial [19]. El objetivo fundamental de este problema es encontrar una asignación de valores de verdad a las variables que satisfaga todas las cláusulas simultáneamente, es decir, que haga verdadera la fórmula booleana completa. Este problema es uno de los más representativos y de mayor complejidad para una computadora clásica, ya que es NP-completo, lo que implica que no se conocen métodos eficientes, en tiempo polinomial, para resolverlo cuando el número de variables crece. De esta manera, dado que el algoritmo de Grover puede usar la superposición cuántica este permite encontrar de una forma eficiente las asignaciones de valores que satisfacen la ecuación booleana. Esto brinda una aceleración cuadrática en la búsqueda de soluciones para un problema 3-SAT, reduciendo la complejidad de la búsqueda exhaustiva  $O(2^N)$  a  $O(2^{N/2})$  [20].

### 3.3. Algoritmo de búsqueda cuántico y la ventaja computacional

Lov Grover demostró que la búsqueda en un conjunto no estructurado puede resolverse en  $O(\sqrt{N})$  pasos, siendo este  $N$  el total de elementos en el espacio de búsqueda no estructurada, en el cual logra reduciendo el número de operaciones necesarias. Las cotas inferiores establecidas son:

- **Clásica:**  $\Omega(N)$ , el límite inferior para la búsqueda clásica el cual se requieren  $N$  evaluaciones.
- **Cuántica:**  $\Omega(\sqrt{N})$  probada por Charles Bennett, Ethan Bernstein, Gilles Brassard y Umesh Vazirani (BBBV), demostraron que cualquier algoritmo cuántico que intente resolver un problema de búsqueda no estructurada debe realizar este mínimo de consultas al oráculo [8].

BBBV explican detalladamente que el algoritmo de Grover es óptimo y demostraron mediante teorema de complejidad cuántica que la aceleración cuadrática que ofrece este algoritmo no puede ser superada por otro procedimiento cuántico, lo que quiere decir que no se puede crear otro algoritmo cuántico que sea fundamentalmente más rápido que

el de Grover [8]. Lo establecen con una cota inferior  $\Omega(\sqrt{N})$  para el problema y Grover da un límite superior  $O(\sqrt{N})$ , concluyendo que el algoritmo es óptimo comparado con cualquier otro existente.

Mientras que, el algoritmo de Grover reduce la complejidad de búsqueda en bases de datos no estructuradas de  $O(N)$  a  $O(\sqrt{N})$ , logrando una aceleración cuadrática. Podemos representar el espacio de búsqueda como un conjunto de  $N = 2^n$  elementos, donde cada elemento está codificado mediante un estado base de  $n$  qubits. El estado inicial es una superposición aplicada a todos nuestros datos dada la siguiente ecuación:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \quad (3.4)$$

Donde, el estado solución o el estado objetivo:  $|\omega\rangle$  cumple  $f(\omega) = 1$ . Si hay  $M$  soluciones, la siguiente Ecuación representa los dos subespacios ortogonales:

$$|s\rangle = \sqrt{\frac{M}{N}} |\omega\rangle + \sqrt{\frac{N-M}{N}} |\omega^\perp\rangle \quad (3.5)$$

El algoritmo aplica dos operadores clave que son los que construyen al algoritmo en si. El Oráculo  $U_f$  invierte la amplitud del estado solución, como representa en la ecuación:

$$U_f |s\rangle = (I - 2|w\rangle\langle w|) |s\rangle \quad (3.6)$$

El operador de difusión refleja respecto a  $|s\rangle$ , amplificando la probabilidad de encontrar  $|\omega\rangle$ , podemos verlo representado con la siguiente ecuación:

$$V \cdot U_f \cdot |s\rangle = (2|s\rangle\langle s| - I) \cdot U_f \cdot |s\rangle \quad (3.7)$$

El Algoritmo de Grover opera en un subespacio bidimensional a través de una secuencia de rotaciones. Inicialmente, el estado de superposición uniforme  $|s\rangle$  se posiciona con un ángulo  $\theta = \arcsin\sqrt{\frac{M}{N}}$  respecto al subespacio de estados no marcados  $|\omega^\perp\rangle$ . Cada iteración completa de Grover  $G = D \cdot O$  actúa como un operador de amplificación de amplitud, produce una rotación de  $2\theta$  en dirección al estado objetivo  $|\omega\rangle$ . Por lo tanto, tras  $r$  iteraciones, la condición  $2r\theta \approx \frac{\pi}{2}$  corresponde a un máximo de la función  $\sin \theta$ , que es el punto óptimo geométrico en el algoritmo de Grover, donde la amplitud de probabilidad del estado objetivo se maximiza. De esta relación se deduce que el número óptimo de iteraciones es aproximadamente:

$$2r\theta \approx \frac{\pi}{2} \Rightarrow r \approx \frac{\pi}{4\theta} = \frac{\pi}{4} \sqrt{\frac{N}{M}} \quad (3.8)$$



### 3.4. Hipótesis y Alcance

La presente investigación se centra en la explicación, análisis y un fuerte componente descriptivo de la computación cuántica y las búsquedas no estructuradas. El trabajo permitirá ver de una forma descriptiva como funcionan estos tipos de algoritmos y comparar su estructura con algoritmos deterministas, con el fin de podernos responder preguntas de como influyen estos circuitos cuánticos en los rendimientos de los algoritmos, como el de Grover, en comparación de algoritmos deterministas que se ejecutan en hardware de tecnologías clásicas. De igual forma, ver si el tamaño de búsqueda, la complejidad del diseño del oráculo son factores fundamentales para el éxito de una búsqueda. Una forma de ver como abordamos este problema podemos observar en la figura 3.2.

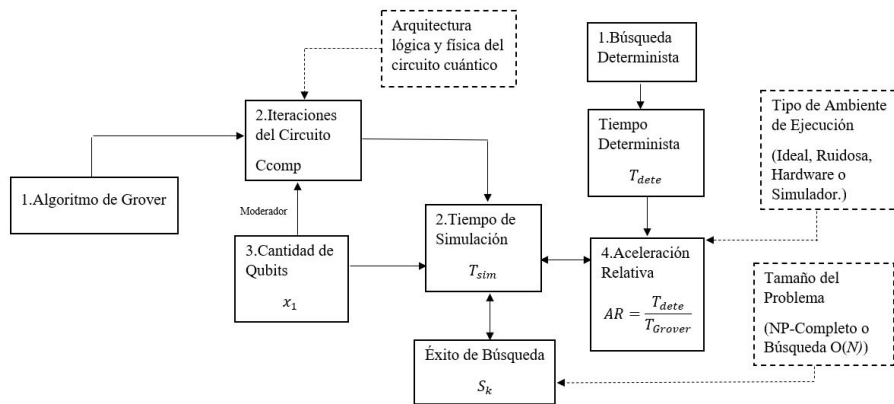


Figura 3.2: *Modelo Conceptual para la Evaluación Comparativa del Algoritmo de Grover en Búsquedas No Estructuradas: Variables Causales, Moderadoras ( $x_1$ ) y Relación con Métodos Clásicos de Búsqueda. Elaboración propia.*

H0: El rendimiento del algoritmo de Grover en la solución de una búsqueda no estructurada, cuando se mide en términos de tasa de éxito, se ve significativamente afectado por la cantidad de qubits utilizados. Siendo esta relación moderada el tamaño de la búsqueda y el número de iteraciones del circuito: además, la comparación con la búsqueda determinista revela que Grover solo supera en eficiencia cuando el tamaño del problema es suficientemente grande y, la implementación del circuito y sus estructuras está acoplada correctamente a la tipología de la computadora sin afectar el tiempo de simulación.

# MARCO TEÓRICO Y ESTADO DEL ARTE

Una de las tecnologías que más ha ayudado al hombre a evolucionar han sido las ciencias de la computación, ha sido construida bajo solidadas bases matemáticas, físicas y químicas, contribuyendo a la innovación y a la escala actual del Hombre del siglo XXI. Con el tiempo grandes personajes de la ciencia, como lo es el cofundador de Intel, predijeron que, con el paso del tiempo, aumentaría la cantidad de transistores integrados en un solo chip, estableciendo así la llamada **Ley de Moore**. Este crecimiento exponencial, aunque impulsó grandes avances, ha comenzado a mostrar limitaciones físicas y teóricas que plantean una barrera para la escalabilidad de la computación clásica [21].

Con el tiempo, las tecnologías clásicas van haciendo más visibles sus asuntos de optimización y en la construcción de soluciones a problemas de gran complejidad, viendo como una de las principales alternativas la computación cuántica [1]. Partiendo de este contexto, el premio Nobel de Física Richard Feynman (1982) dio una indicación clara: si la única respuesta a los problemas de gran complejidad es una computadora cuántica, entonces esta debe construirse, ya que las simulaciones realizadas con tecnologías clásicas no lograban reproducir adecuadamente los fenómenos cuánticos [22]. Dando inicio a la computación cuántica.

Esta nueva visión ha transformado significativamente nuestra comprensión de tecnologías cuánticas al ofrecernos ventajas para poder resolver problemas de un gran nivel de complejidad, gracias a los principios como la superposición o el entrelazamiento cuántico podemos realizar múltiples procesos al tiempo, se pueden abordar problemas que con las tecnologías clásicas son casi imposibles de llevar [1]. Una de las mayores diferencias entre la computación cuántica y la clásica es que esta usa propiedades únicas de la Mecánica Cuántica y emplea el uso de qubits o bits cuánticos como unidad fundamental de información, puede encontrarse simultáneamente en una combinación lineal de estados cuánticos entre 0 y 1, permitiendo explotar propiedades de espacios computacionales superiores a los de un sistema clásico. Un uso coherente de estas propiedades define la base de la ventaja cuántica, dado que puede resolver problemas que nunca los podría resolver una tecnología determinista, dejándolos como única alternativa para sin fin de soluciones a problemas del siglo actual [23].

Este potencial, se ve confirmado en algoritmos cuánticos como el algoritmo de Grover, diseñado para la búsqueda en bases de datos no estructuradas, el cual contempla un ejemplo de la ventaja cuántica demostrable, lo que representa un cambio sustancial en la complejidad computacional para abordar problemas de búsqueda [24]. Como respuesta académica, la presente investigación se orienta al análisis descriptivo del algoritmo

de Grover, considerando tanto sus fundamentos teóricos como los desafíos prácticos de su ejecución, con el fin de lograr una comparativa clara con tecnologías clásicas y poder ver si sus fundamentos teóricos responden a nuestras preguntas, todo esto estando presentes en la actual era de la “computación cuántica ruidosa a escala intermedia” en inglés Noisy intermediate-scale quantum (NISQ)[25].

## 4.1. El Problema de la Búsqueda No Estructurada

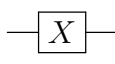
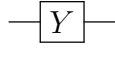

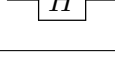
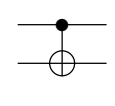
En el paradigma de la computación clásica determinista, la búsqueda de un elemento específico dentro de un conjunto de datos no estructurado de tamaño  $N$  se rige por un límite fundamental de complejidad. En el peor de los casos, para garantizar el hallazgo del elemento o la conclusión de que no existe, el algoritmo se ve obligado a inspeccionar, en promedio, la mitad de los elementos, lo que se traduce asintóticamente en la necesidad de recorrer todos los  $N$  elementos. Esto establece una complejidad temporal lineal de  $O(N)$  consultas al oráculo o a la base de datos. Este límite, que establece a la búsqueda secuencial como la estrategia estándar, ha sido rigurosamente demostrado en la teoría de la complejidad clásica, siendo un tema central en obras fundamentales como *Introduction to Algorithms* de Cormen [5]. Lo inevitable de esta cota inferior, subraya que ningún algoritmo clásico puede hacer un uso más eficiente de la información en ausencia de estructura. Aunque existen alternativas mucho más rápidas, como la búsqueda binaria, que ostenta una complejidad logarítmica  $\Theta(\log N)$ , su aplicabilidad está estrictamente restringida a conjuntos de datos que han sido previamente ordenados, lo que a su vez requiere una inversión de tiempo de  $O(N \log N)$  para el preprocesamiento, confirmando que la barrera lineal  $O(N)$  es propio al problema de búsqueda cuando la estructura no está disponible. En consecuencia, esta barrera de  $O(N)$  es el punto de referencia de la computación clásica y la razón precisa por la cual el algoritmo de Grover representa una ventaja cuántica. Al reducir la complejidad del mismo problema de  $O(N)$  a  $O(\sqrt{N})$ , la computación cuántica logra sortear esta limitación fundamental clásica al explotar los principios de superposición y entrelazamiento.

## 4.2. Computación Cuántica

### 4.2.1. Entrelazamiento y Computas Cuánticas

La computación cuántica emplea qubits, unidad básica de información de la computación cuántica, que pueden estar en una superposición de 0 y 1 con ciertas probabilidades, a diferencia de los bits clásicos que solo pueden estar en 1 o en 0. Un estado cuántico es un vector normalizado que define la configuración de un qubit [2, cap. 1,

1.2.1]. Las compuertas cuánticas son transformaciones unitarias que preservan la norma del vector y modifican su orientación en el espacio de Hilbert de dimensión 2. Algunas de ellas son las matrices de Pauli, la compuerta Hadamard y la CNOT, como se muestra en la tabla 4.1.

Nombre	Símbolo	Forma matricial	Descripción
Pauli-X / NOT		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	Invierte el estado: $0 \leftrightarrow 1$ .
Pauli-Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	Rotación con fase compleja.
Pauli-Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	Aplica una fase de $-1$ a $1$ .
Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	Crea una superposición.
CNOT		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	Aplica NOT al qubit objetivo si el qubit de control es $1$ .

Cuadro 4.1: Compuertas cuánticas básicas

Decimos que, un conjunto de qubits está en estado producto si su estado puede ponerse como producto tensorial de los estados de sus componentes. En caso contrario, se dice que está en un estado de entrelazamiento [2, cap 1, 1.3.6].

**Ejemplo:** Los estados de Bell son un conjunto fundamental de cuatro estados cuánticos de dos qubits que exhiben el máximo nivel de entrelazamiento. La característica de estos estados es que no pueden expresarse como el producto tensorial de dos estados de qubit individuales (es decir, no pueden ser factorizados como  $|\psi_A\rangle \otimes |\phi_B\rangle$ ). Esta propiedad implica que la medición de uno de los qubits afecta instantáneamente el estado del otro, sin importar la distancia, un fenómeno que Einstein, Podolsky y Rosen (EPR) denominaron la “acción fantasma a distancia”. Por ende, vamos a considerar el estado de Bell que, también se conoce como par EPR  $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ .

Para verificar si este estado está entrelazado, se puede intentar expresar como el producto tensorial de dos qubits individuales. Supongamos que es separable; es decir, que existen amplitudes  $\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathbb{C}$  tales que:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle) \quad (4.1)$$

Al comparar coeficientes en la base computacional se obtiene el sistema

$$\alpha_0\beta_0 = \frac{1}{\sqrt{2}} \quad (4.2)$$

$$\alpha_0\beta_1 = 0 \quad (4.3)$$

$$\alpha_1\beta_0 = 0 \quad (4.4)$$

$$\alpha_1\beta_1 = \frac{1}{\sqrt{2}} \quad (4.5)$$

De (4.3):  $\alpha_0 = 0$  o  $\beta_1 = 0$ . De (4.4):  $\alpha_1 = 0$  o  $\beta_0 = 0$ .

- Si  $\alpha_0 = 0$ , entonces por (4.2):  $0 = \frac{1}{\sqrt{2}}$
- Si  $\beta_1 = 0$ , entonces por (4.5):  $0 = \frac{1}{\sqrt{2}}$

Cualquier elección que satisfaga ambas igualdades anula alguno de los productos no nulos en (4.2) o (4.5). Por tanto, no existe factorización y el estado  $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$  es **entrelazado**.

De todos modos, las compuertas cuánticas, para ser físicamente realizables, deben preservar la normalización de los estados cuánticos sobre los cuales actúan, esto quiere decir que si una compuerta cuántica actúa sobre un estado  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  que cumple  $|\alpha|^2 + |\beta|^2 = 1$ , y el resultado es  $|\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ , se debe verificar también  $|\alpha'|^2 + |\beta'|^2 = 1$  [2, cap 2].

Esto se traduce en que la puerta cuántica debe ser **unitaria**. La condición es:

$$U^\dagger U = U U^\dagger = I \quad (4.6)$$

donde  $I$  es la matriz identidad,  $U$  la representación matricial de la puerta y  $U^\dagger$  su matriz adjunta.

La demostración es directa:

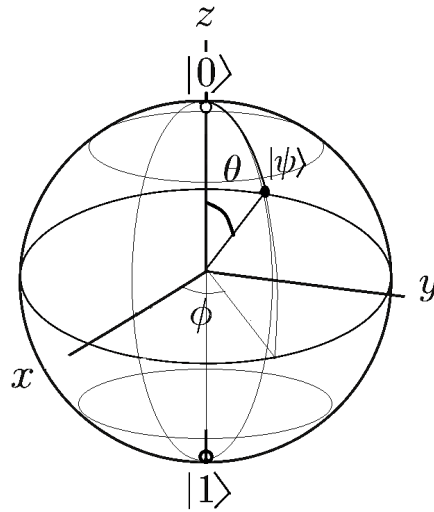
$$\langle\psi'|\psi'\rangle = \langle\psi|U^\dagger U|\psi\rangle = \langle\psi|I|\psi\rangle = \langle\psi|\psi\rangle = 1 \quad (4.7)$$

### 4.2.2. Interpretación Geométrica

La esfera de Bloch Figura 4.2 y su Ecuación 4.8 es una representación geométrica de los estados cuánticos de un qubit en una esfera unitaria, permitiendo visualizar las transformaciones inducidas por compuertas cuánticas como rotaciones de los vectores de estado.

- **Compuerta X (NOT cuántica)**: rota el qubit  $\pi$  radianes alrededor del eje  $x$ .
- **Compuerta Y**: similar a X, pero rota sobre  $y$  con una fase adicional de  $\pi/2$ .

- **Compuerta Z:** refleja el estado  $|1\rangle$  respecto al eje  $z$ , dejando invariante  $|0\rangle$ .
- **Compuerta Hadamard:** genera una superposición equilibrada de los estados base, ubicando el vector de estado en una posición intermedia. Mueve el vector de estado desde los polos (eje Z) hacia el ecuador (eje X). Geométricamente, H realiza una rotación de  $180^\circ$  alrededor del eje diagonal en el plano XZ, equivalente a una reflexión entre los ejes Z y X.
- **Compuerta CNOT:** opera sobre dos qubits; el primero actúa como control y, si está en  $|1\rangle$ , aplica la compuerta X sobre el segundo qubit.



$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \quad (4.8)$$

Figura 4.1: *Esfera de Bloch. Basado en documentación Qiskit.*

### 4.2.3. Circuito cuántico compuesto por las compuertas Hadamard, CNOT y Z

Un circuito cuántico es una secuencia de compuertas cuánticas que transforman los qubits de entrada, para una ejemplificación clara, se mostrará mediante un circuito de dos qubits cómo estos estados cambian mientras se recorre el circuito. De igual forma, es un método para enredar dos qubits y obtener un estado de Bell.

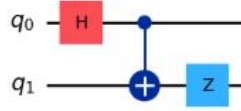


Figura 4.2: *Circuito Hadamard, CNOT y Z. Basado en documentación Qiskit.*

Estados básicos (2 qubits)

$$\begin{aligned}
 |00\rangle &= \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} &
 |01\rangle &= \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} &
 |10\rangle &= \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} &
 |11\rangle &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}
 \end{aligned} \tag{4.9}$$

Hadamard en el primer qubit y la identidad en el segundo qubit

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad
 I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad
 H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \tag{4.10}$$

Aplicado a  $|00\rangle$ , como primera compuerta del circuito:

$$\begin{aligned}
 (H \otimes I)|00\rangle &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
 &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\
 &= \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle)
 \end{aligned} \tag{4.11}$$

Estado inicial y aplicación de CNOT sobre el estado:

$$\frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \tag{4.12}$$

Aplicación de la CNOT al estado:

$$\text{CNOT } \psi_0 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \quad (4.13)$$

Aplicación de  $Z$  usando producto de Kronecker [2, cap 2, 2.1.7]

$$I \otimes Z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (4.14)$$

$$(I \otimes Z) \left( \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.15)$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle)$$

Estado entrelazado de Bell, entrelazado porque si afecta un qubit, afecta a todo el sistema de la ecuación o qubits:

$$|00\rangle \xrightarrow{H \otimes I} \frac{1}{\sqrt{2}} (|00\rangle + |10\rangle) \xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) \xrightarrow{I \otimes Z} \frac{1}{\sqrt{2}} (|00\rangle - |11\rangle) \quad (4.16)$$

Este resultado ilustra de forma concreta cómo, mediante una secuencia de puertas cuánticas, es posible preparar un estado de Bell entrelazado, específicamente el estado  $\frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$ . Este proceso, no solo resalta la potencia del entrelazamiento cuántico como principio fundamental en la computación cuántica, sino que también evidencia cómo permiten generar correlaciones no deterministas entre qubits. Este tipo de estructura resulta esencial para el diseño de algoritmos cuánticos, y tareas de verificación cuántica, siendo a los estados de Bell una base en la arquitectura de la computación cuántica.

### 4.3. Algoritmo de Grover

Las computadoras cuánticas, son una de las máquinas más complejas jamás creada por el hombre y que nos permite quitarnos la frustración de la necesidad de más recursos computacionales [2, cap 4, 4.1].

Recientemente, las computadoras cuánticas ofrecen la capacidad de resolver problemas de gran complejidad gracias a algoritmos cuánticos desarrollados para suplir los desafíos inherentes en la humanidad en ámbitos específicos. Uno de esos retos ha sido las búsquedas en bases de datos que no cuentan con un orden específico.

El algoritmo de Grover, uno de los algoritmos de computación cuántica más estudiados de la época y que cuenta con la capacidad de buscar una entidad con alta probabilidad utilizando únicamente evaluaciones de orden  $O(\sqrt{N})$ [26].

Como lo menciona Grover: «el algoritmo está dentro de un pequeño factor constante del algoritmo mecánico cuántico más rápido posible» [6, p. 1]. Esta investigación cita los estudios de más relevancia donde aplican el algoritmo de Grover a tal detalle, con el fin de mostrar esta ventaja, que en primera instancia fue demostrada por Bennet en 1997, afirmando que ningún otro algoritmo puede resolver problemas de búsquedas no estructuradas con mayor eficiencia que el de Grover, respaldado por la cota inferior  $\Omega(\sqrt{N})$ [8].

### 4.4. Búsqueda de Grover

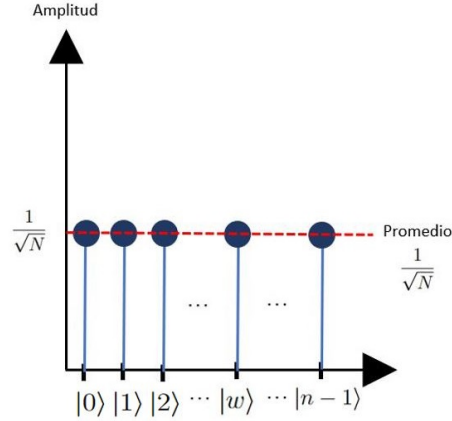
El algoritmo de Grover es un algoritmo cuántico fundamental que proporciona una aceleración cuadrática para problemas de búsqueda no estructurada. A diferencia de los métodos clásicos que requieren un promedio de  $O(N)$  consultas para encontrar un ítem en una base de datos de  $N$  elementos, Grover localiza la solución en aproximadamente  $O(\sqrt{N})$  pasos. El algoritmo logra esto iniciando el sistema en una superposición uniforme de todos los estados; seguidamente, un oráculo cuántico “marca” la solución correcta mediante una inversión de fase. Tras esto, un operador de difusión, descrito como “inversión sobre la media”, explota la interferencia cuántica para amplificar sistemáticamente la amplitud de probabilidad del estado marcado. Este proceso de oráculo-difusión se repite  $O(\sqrt{N})$  veces hasta que la medición final colapsa al estado deseado con alta probabilidad, un límite de velocidad que se ha demostrado teóricamente óptimo [2].

**Inicialización y Superposición Uniforme:** Para lograr una superposición uniforme aplicamos la compuerta Hadamard ( $H$ ) a cada qubit inicializados en  $|000\rangle$ .

- Con 3 qubits podemos representar  $2^3 = 8$

- El estado a buscar es el  $|101\rangle$  que corresponde al número 5
- Asumimos, que el estado objetivo está dentro de los 8 estados base, que nuestros 3 qubits pueden representar

$$|\psi_0\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle = |000\rangle \quad (4.17)$$



$$|s\rangle = H^{\otimes n} |0\rangle^{\otimes n} \quad (4.18)$$

Figura 4.3: *Inicialización del Algoritmo de Grover. Elaboración basada en Qiskit [27, 28].*

Aplicamos  $H$  a los 3 qubits en el estado  $|000\rangle$ :

$$H^{\otimes 3} = H \otimes H \otimes H, \quad |\psi_1\rangle = H^{\otimes 3}|000\rangle = (H|0\rangle) \otimes (H|0\rangle) \otimes (H|0\rangle) \quad (4.19)$$

$$|\psi_1\rangle = \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \quad (4.20)$$

Expansión del producto tensorial:

$$|\psi_1\rangle = \frac{1}{2\sqrt{2}} (|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \quad (4.21)$$

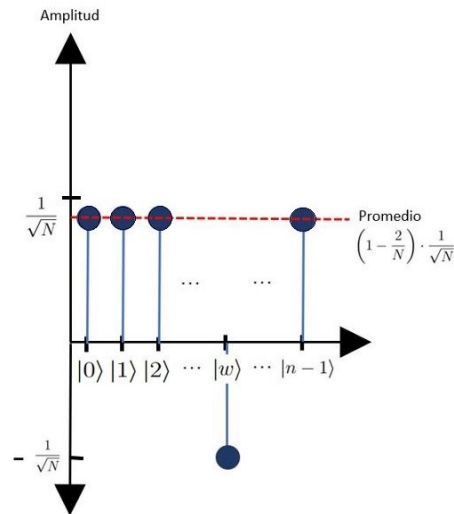
**Estado uniforme:** El estado  $|s\rangle$  representa una distribución de probabilidad uniforme sobre los 8 posibles valores de la base computacional.

$$|s\rangle = \frac{1}{(\sqrt{2})^3} \sum_{x \in \{0,1\}^3} |x\rangle = \frac{1}{\sqrt{8}} (|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \quad (4.22)$$

$$|s\rangle \equiv |\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \quad (4.23)$$

*Paralelismo cuántico, se define como la capacidad de un sistema cuántico de evaluar simultáneamente múltiples valores de una función gracias al uso de superposición cuántica.*

**Oráculo de Grover:** Esta parte es fundamental en el desarrollo de algoritmo ya que nos permite marcar el estado objetivo, la Ecuación 4.24 describe la acción del oráculo  $U_f$  en el algoritmo de Grover. Aquí,  $U_f$  es el operador unitario que marca el estado objetivo  $|w\rangle$ , donde  $w$  se refiere a  $\omega$ , el estado solución que satisface la condición de búsqueda. El término  $I - 2|w\rangle\langle w|$  representa una reflexión sobre el estado  $|w\rangle$ , lo que invierte el signo de su amplitud en el estado de superposición uniforme  $|s\rangle$ , mientras mantiene inalteradas las amplitudes de los demás estados. Esta operación es indispensable para amplificar la probabilidad de medir  $|w\rangle$  en iteraciones posteriores del algoritmo.



$$U_f |s\rangle = (I - 2 |w\rangle\langle w|) |s\rangle \quad (4.24)$$

Figura 4.4: Oráculo Actuando sobre  $|\omega\rangle$ . Elaboración basada en Qiskit [27, 28].

$$U_f |x\rangle = (-1)^{f(x)} |x\rangle \quad (4.25)$$

$$f(x) = \begin{cases} 1, & \text{si } x = w, \\ 0, & \text{si } x \neq w. \end{cases} \quad (4.26)$$

$$U_f \left( \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \right) = \frac{1}{\sqrt{8}} \sum_{x=0}^7 (-1)^{f(x)} |x\rangle \quad (4.27)$$

$$U_f = I - 2|w\rangle\langle w| \quad (4.28)$$

Los vectores de la base computacional  $|i\rangle$  satisfacen una propiedad fundamental conocida como ortonormalidad, expresada mediante la delta de Kronecker:

$$\langle i|j\rangle = \delta_{ij} = \begin{cases} 0, & \text{si } i \neq j \\ 1, & \text{si } i = j \end{cases} \quad (4.29)$$

*Delta de Kronecker*

El operador  $U_f$  actúa como una reflexión alrededor del estado solución  $|w\rangle$  y se define como:

$$U_f = I - 2|w\rangle\langle w| \quad (4.30)$$

Al aplicar  $U_f$  al estado uniforme  $|s\rangle$ , se obtiene:

$$U_f|s\rangle = (I - 2|w\rangle\langle w|)|s\rangle = I|s\rangle - (2|w\rangle\langle w|)|s\rangle \quad (4.31)$$

Dado que  $|s\rangle$  es una superposición uniforme de todos los estados base en el espacio de búsqueda:

$$I|s\rangle = |s\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \quad (4.32)$$

Para el término  $(2|w\rangle\langle w|)|s\rangle$ , es crucial recordar que  $\langle w|s\rangle$  es un escalar (producto interno entre  $|w\rangle$  y  $|s\rangle$ ):

$$(2|w\rangle\langle w|)|s\rangle = 2|w\rangle(\langle w|s\rangle) \quad (4.33)$$

Calculamos  $\langle w|s\rangle$ . Suponiendo  $|w\rangle = |5\rangle$  y usando la ortonormalidad:

$$\langle w|s\rangle = \langle 5| \left( \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \right) = \frac{1}{\sqrt{8}} \sum_{x=0}^7 \langle 5|x\rangle = \frac{1}{\sqrt{8}} \langle 5|5\rangle = \frac{1}{\sqrt{8}} \quad (4.34)$$

Esto se debe a que  $\langle 5|x\rangle = \delta_{5,x}$ , donde  $\delta_{5,x}$  es la delta de Kronecker.

Sustituyendo en la ecuación 4.33:

$$(2|w\rangle\langle w|)|s\rangle = 2|w\rangle \cdot \frac{1}{\sqrt{8}} = \frac{2}{\sqrt{8}}|w\rangle \quad (4.35)$$

Combinando los resultados en la ecuación 4.31:

$$U_f|s\rangle = |s\rangle - \frac{2}{\sqrt{8}}|w\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle - \frac{2}{\sqrt{8}}|5\rangle \quad (4.36)$$

Definimos el estado después de aplicar el oráculo como  $|\psi_2\rangle$ :

$$|\psi_2\rangle = U_f|s\rangle \quad (4.37)$$

Por lo tanto, de la ecuación 4.36:

$$|\psi_2\rangle = |s\rangle - \frac{2}{\sqrt{8}}|5\rangle \quad (4.38)$$

Expresando  $|s\rangle$  explícitamente y simplificando:

$$|\psi_2\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle - \frac{2}{\sqrt{8}}|5\rangle = \frac{1}{\sqrt{8}} \sum_{x \neq 5} |x\rangle - \frac{1}{\sqrt{8}}|5\rangle \quad (4.39)$$

Efecto sobre amplitudes individuales

$$\sum_{x=0}^7 |x\rangle = |0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle \quad (4.40)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{8}} \left( |000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle - |101\rangle + |110\rangle + |111\rangle \right) \quad (4.41)$$

$$|\psi_2\rangle = \frac{1}{\sqrt{8}}|0\rangle + \frac{1}{\sqrt{8}}|1\rangle + \frac{1}{\sqrt{8}}|2\rangle + \frac{1}{\sqrt{8}}|3\rangle + \frac{1}{\sqrt{8}}|4\rangle - \frac{1}{\sqrt{8}}|5\rangle + \frac{1}{\sqrt{8}}|6\rangle + \frac{1}{\sqrt{8}}|7\rangle \quad (4.42)$$

Si las sumas de las probabilidades es 1, el estado cuántico está bien definido o normalizado

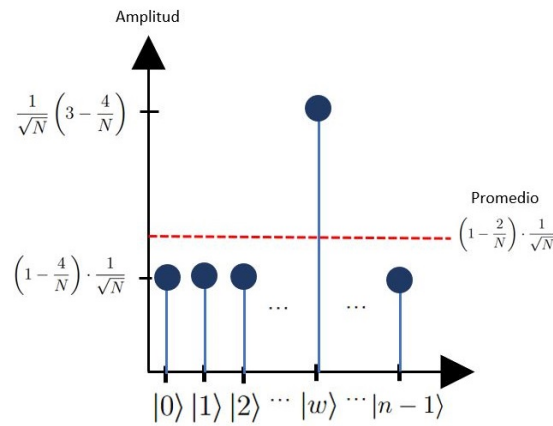
$$|\psi_2\rangle = \sum_{x=0}^7 c_x |x\rangle, \quad c_5 = -\frac{1}{\sqrt{8}}, \quad c_x = \frac{1}{\sqrt{8}} \quad (x \neq 5) \quad (4.43)$$

$$\sum_{x=0}^7 |c_x|^2 = |c_5|^2 + \sum_{x \neq 5} |c_x|^2 = \frac{1}{8} + 7 \cdot \frac{1}{8} = \frac{8}{8} = 1 \quad (4.44)$$

El estado  $|\psi_2\rangle$  está correctamente normalizado, pues la suma de todas las probabilidades es 1.

### Operador de Grover o Difusor

$$V = 2|s\rangle\langle s| - I \quad (4.45)$$



$$V \cdot U_f \cdot |s\rangle = (2|s\rangle\langle s| - I) \cdot U_f \cdot |s\rangle \quad (4.46)$$

Figura 4.5: *Reflejar Amplitud Sobre la Amplitud Media. Elaboración basada en Qiskit [27, 28].*

Calculamos la Ecuación (4.46), dado el resultado una vez aplicado el oráculo

Sustituimos los valores de  $f(x)$ :

- Para  $x = 5$ :  $(-1)^{f(5)} = (-1)^1 = -1$ .
- Para  $x \neq 5$ :  $(-1)^{f(x)} = (-1)^0 = 1$ .

$$\sum_{x=0}^7 (-1)^{f(x)} = 7 \cdot 1 + 1 \cdot (-1) = 6 \quad (4.47)$$

$$\langle s|U_f|s\rangle = \frac{6}{8} = \frac{3}{4} \quad (4.48)$$

Y según la definición del Difusor (4.45), lo aplicamos al estado  $U_f |s\rangle$  sustituyendo la expresión (4.48):

$$V U_f |s\rangle = (2 |s\rangle \langle s| - I) U_f |s\rangle = 2 |s\rangle \langle s| U_f |s\rangle - U_f |s\rangle \quad (4.49)$$

$$V \cdot U_f \cdot |s\rangle = 2 |s\rangle \cdot \frac{3}{4} - U_f |s\rangle = \frac{3}{2} |s\rangle - U_f |s\rangle \quad (4.50)$$

Para mantener la notación de la Ecuación (4.46) hablaremos de la Ecuación (??) como  $U_f |s\rangle$  y mis estados ortogonales como  $|s\rangle$ , así que sustituimos las expresiones de  $|s\rangle$  y  $U_f |s\rangle$ :

$$|s\rangle = \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle \quad U_f |s\rangle = \frac{1}{\sqrt{8}} \left( \sum_{x \neq 5}^7 |x\rangle - |5\rangle \right) \quad (4.51)$$

Expresando en términos de las sumas de estados:

$$V \cdot U_f \cdot |s\rangle = \frac{3}{2} \cdot \frac{1}{\sqrt{8}} \sum_{x=0}^7 |x\rangle - \frac{1}{\sqrt{8}} \left( \sum_{x \neq 5} |x\rangle - |5\rangle \right) \quad (4.52)$$

Debemos factorizar  $\frac{1}{\sqrt{8}}$ :

$$= \frac{1}{\sqrt{8}} \left[ \frac{3}{2} \sum_{x=0}^7 |x\rangle - \sum_{x \neq 5} |x\rangle + |5\rangle \right] \quad (4.53)$$

Descomponiendo  $\sum_{x=0}^7 |x\rangle = \sum_{x \neq 5} |x\rangle + |5\rangle$ :

$$= \frac{1}{\sqrt{8}} \left[ \frac{3}{2} \left( \sum_{x \neq 5} |x\rangle + |5\rangle \right) - \sum_{x \neq 5} |x\rangle + |5\rangle \right] \quad (4.54)$$

$$= \frac{1}{\sqrt{8}} \left[ \frac{3}{2} \sum_{x \neq 5} |x\rangle + \frac{3}{2} |5\rangle - \sum_{x \neq 5} |x\rangle + |5\rangle \right] \quad (4.55)$$

$$= \frac{1}{\sqrt{8}} \left[ \left( \frac{3}{2} - 1 \right) \sum_{x \neq 5} |x\rangle + \left( \frac{3}{2} + 1 \right) |5\rangle \right] \quad (4.56)$$

$$= \frac{1}{\sqrt{8}} \left[ \frac{1}{2} \sum_{x \neq 5} |x\rangle + \frac{5}{2} |5\rangle \right] \quad (4.57)$$

$$= \frac{1}{2\sqrt{8}} \sum_{x \neq 5} |x\rangle + \frac{5}{2\sqrt{8}} |5\rangle \quad (4.58)$$

**Resultado Final, el estado después de aplicar el operador de difusión es:**

$$V \cdot U_f \cdot |s\rangle = \frac{5}{2\sqrt{8}}|5\rangle + \frac{1}{2\sqrt{8}} \sum_{x \neq 5} |x\rangle \quad (4.59)$$

### Complejidad Computacional

- Cada iteración de Grover  $G = V \cdot U_f$  es una rotación de  $2\theta_0$  respecto al vector inicial en el subespacio

$$\{\omega^\perp, w\}. \quad (4.60)$$

- Estado tras  $t$  iteraciones:

$$|\psi_t\rangle = (VU_f)^t |s\rangle = \cos((2t+1)\theta_0)|\omega^\perp\rangle + \sin((2t+1)\theta_0)|w\rangle \quad (4.61)$$

El operador completo de Grover  $G = V \cdot U_f$  actúa como una rotación en el subespacio de dimensión 2 generado por los vectores  $|w\rangle$  (estado solución) y  $|\omega^\perp\rangle$  (ortogonal a la solución).

Al comenzar en el estado uniforme  $|s\rangle$ , que forma un ángulo  $\theta_0$  con  $|w\rangle$ , la primera aplicación de  $G$  rota el estado  $2\theta_0$  hacia  $|w\rangle$ .

Luego de  $t$  iteraciones, la rotación acumulada es  $(2t+1)\theta_0$ , porque se cuenta también el ángulo inicial entre  $|s\rangle$  y  $|w\rangle$ .

- Probabilidad de medir el estado solución:

$$P_t = \sin^2((2t+1)\theta_0), \quad P_0 = |\langle w|s\rangle|^2 = \frac{1}{N} \quad (4.62)$$

El algoritmo busca maximizar la amplitud de  $|w\rangle$ . El ángulo inicial satisface

$$\theta_0 = \arcsin(|\langle s|w\rangle|) = \arcsin\left(\sqrt{\frac{1}{N}}\right) \quad (4.63)$$

Luego de  $t$  iteraciones, el ángulo acumulado es

$$\theta_t = (2t+1)\theta_0 \quad (4.64)$$

Para maximizar la amplitud (hacer  $\sin((2t+1)\theta_0) \approx 1$ ):

$$\frac{\pi}{2} = (2t_{\text{óptimo}} + 1)\theta_0 \quad (4.65)$$

Cuando  $N$  es grande,  $\sin(\theta_0) \approx \theta_0$  y  $\theta_0 \approx \frac{1}{\sqrt{N}}$  Sustituyendo en (4.65):

$$(2t + 1) \frac{1}{\sqrt{N}} \approx \frac{\pi}{2} \implies t \approx \frac{\pi}{4} \sqrt{N} - \frac{1}{2} \quad (4.66)$$

Por lo tanto, el número de iteraciones crece como

$$\boxed{O(\sqrt{N})} \quad (4.67)$$

#### 4.4.1. Implementaciones y Verificaciones Experimentales

Observaremos las aplicaciones más recientes en el presente trabajo para abordar los conceptos fundamentales al querer implementar el algoritmo de Grover.

De hecho, en los enfoques clásicos para resolver problemas de satisfacibilidad booleana (SAT), se examinan todas las posibles asignaciones de valores de verdad (entradas) a las variables booleanas del problema, con el fin de verificar si alguna de ellas satisface todas las cláusulas de la fórmula lógica, es decir, cumple las desigualdades lógicas que representan las restricciones del problema. Este procedimiento, conocido como búsqueda exhaustiva, se vuelve ineficiente a medida que aumenta el número de variables, ya que el espacio de soluciones crece exponencialmente. De modo que, si hay  $N$  combinaciones que validar para un problema con una única solución, este enfoque determinista implica revisar una por una combinación dada hasta poder encontrarla. En el peor de los casos, se deben validar todos los  $N$  procesos donde un lugar indeterminado es la solución [29]. Existen otros enfoques en donde la búsqueda aprovecha para tener éxito en una escala  $O(N)$ , pero estos casos son netamente estructurados [30].

Al momento de resolver problemas de gran complejidad, las computadoras cuánticas entran como alternativa más eficiente que una computadora clásica, como primer ejemplo uno de los primeros algoritmos como lo es el algoritmo de Deutsch-Jozsa [31] determinando si una función es constante o tiene un balance en una sola consulta.

Otro tipo de problemas de gran complejidad, como lo son las búsquedas en listas o base de datos no estructuradas que se resuelven con mayor precisión dando uso del algoritmo de Grover gracias a que ofrecen una aceleración cuadrática frente a los clásicos, se impone como única alternativa eficiente para la resolución de dichos problemas [7].

Una de las implementaciones más completas desarrolladas hasta el momento para abordar problemas de satisfacibilidad se ha centrado en el Problema de Satisfacibilidad de Restricciones con Enteros (CSP-I), el cual extiende el SAT booleano a un caso más complejo al permitir que las variables tomen valores enteros en lugar de solo booleanos. Una notable implementación de 20 qubits, cuya arquitectura se representa en la figura 4.6, propone un algoritmo cuántico híbrido. Este método combina una búsqueda clásica, para estructurar el problema y con el algoritmo de Grover, el cual se utiliza como

subrutina para acelerar la búsqueda de soluciones [29].

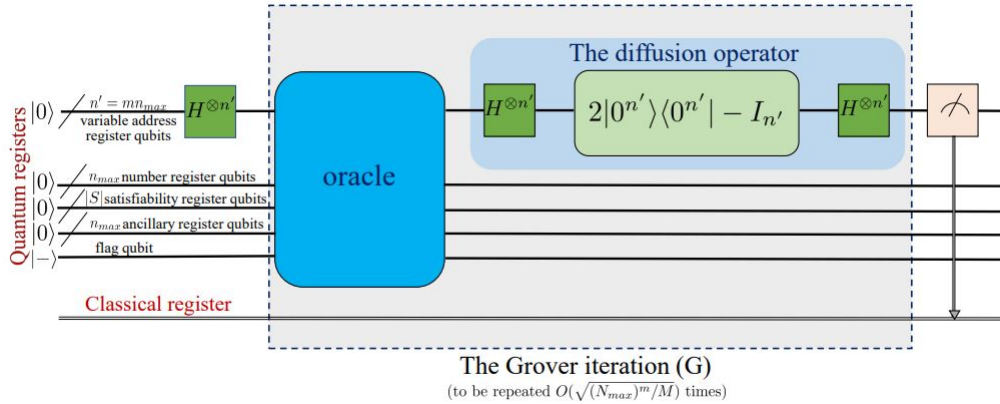


Figura 4.6: Circuito cuántico de Grover, diseñado para verificar las restricciones de un problema CSP-I. El circuito marca un estado como “solución” (aplicando un cambio de fase) solo si cumple todas las restricciones. Tomado de Vinod & Shaji [29].

Si bien existen implementaciones avanzadas como la de la figura 4.6 [29], que utiliza un método híbrido para el complejo caso de CSP-I, nuestro diseño de circuito sigue un enfoque diferente y fundamental.

En lugar de un modelo híbrido, esta investigación se centra en la construcción completa y unitaria de un oráculo específico para un problema 3-SAT. Para ello, aunque tomamos como base conceptual la lógica de satisfacibilidad booleana, nuestro diseño implementa el oráculo desde sus cimientos, utilizando estructuras de cómputo para las cláusulas en FNC y aplicando rigurosamente la descomputación, una técnica esencial en computación cuántica que consiste en revertir de manera controlada las operaciones realizadas sobre qubits auxiliares una vez que han cumplido su propósito en un cálculo intermedio. Este proceso garantiza que dichos qubits regresen a su estado  $|0\rangle$  eliminando así correlaciones cuánticas residuales que podrían introducir interferencias no deseadas en la evolución del sistema. En algoritmos como el de Grover, la descomputación se emplea durante la implementación del oráculo para preservar la unitariedad global de las transformaciones y mantener la integridad del registro principal de búsqueda [32, 33].

De igual forma, vemos otras implementaciones de gran nivel en la academia como la realizada por Vemula, Dinesh Reddy and Konar, Debanjan and Satheesan, Sudeep and Kalidasu, Sri Mounica and Cangi, Attila [26]. Se exploran circuitos escalables dando uso de compuertas Toffolis y CNOTs haciendo de este circuito menos profundo al usar menos compuertas pero para 5 o 6 qubits, limitando la observación de una base de datos no estructurada de mayor tamaño.

Otro de los trabajos más destacados de la época, es la que muestra el funcionamiento

del oráculo de Grover he implementada por Kumar [34]. En esta implementación en específico se ven reflejado la sensibilidad de las computadoras cuánticas en la era NISQ, como la *ibmq\_santiago*, degradando en lo que aumenta cada qubit de implementación en hardware ruidoso.

Con lo anterior, vemos un panorama claro de implementaciones en las computadoras cuánticas en la era NISQ y cuyos desarrollos reflejan todavía grandes desafíos dado la complejidad de mantener la coherencia de la teoría de Grover en ejecuciones de tamaños acordes a aplicaciones prácticas reales. Así mismo, esta investigación trata de dar profundidad a las implementaciones revisadas, aprovechar algunos de sus conceptos lógicos y ver una relación coherente comparando tecnologías clásicas con la computación cuántica.

## Capítulo 5

# METODOLOGÍA

El presente trabajo, es de tipo explicativo con un componente analítico y descriptivo, el cual se realiza a través de una investigación experimental enfocada en la definición del algoritmo de Grover y su funcionalidad frente a algoritmos clásicos, empleando el simulador de Qiskit, para ver cómo logra el algoritmo de Grover encontrar una solución en un espacio desordenado y ver en realidad su ventaja cuántica, al momento de resolver un problema de una búsqueda no estructurada.

En el desarrollo de esta investigación, se llevaron a cabo una serie de experimentos orientados a cumplir los objetivos propuestos. En primer lugar, se diseñaron circuitos cuánticos básicos que permitieron explorar los principios fundamentales de la computación cuántica, como la superposición, el entrelazamiento y las operaciones sobre qubits. Posteriormente, se contrastaron los algoritmos clásicos de búsqueda no estructurada con el algoritmo de Grover, analizando sus diferencias en complejidad computacional y escalabilidad. A continuación, se implementó el algoritmo de Grover tanto en simuladores cuánticos como en un backend real de IBM Quantum *ibm\_brisbane*, evaluando su rendimiento en escenarios con distintas cantidades de qubits, la distribución de resultados y el tiempo de ejecución. Finalmente, se examinó la aplicabilidad del algoritmo para resolver problemas de búsqueda, validando su desempeño como alternativa cuántica frente a métodos clásicos.

Para poder dar solución y estructurar adecuadamente el algoritmo de Grover se deben tener en cuenta el orden y las operaciones elementales del algoritmo, lo podemos demostrar dado el siguiente problema.

## 5.1. Implementación de una búsqueda de Grover

### 5.1.1. Inicialización

Para tener una correcta inicialización, debemos contar con una estructura bien definida de lo que vamos a implementar, para ello adecuaremos los qubits que se utilizaran en el circuito, el dato que debe marcar el oráculo debe ocultarse para que funcione como una caja negra dado que en otros ejemplos de ejecución se puede brindar al oráculo el dato objetivo anticipado pero, esto no sería correcto con la estructura lógica de un algoritmo de computación cuántica como el de Grover.

Como bien sabemos, el algoritmo de Grover opera sobre un espacio de búsqueda de  $N = 2^n$ , consideramos varios ejemplos en un conjunto de 3,4,5,6,7 y 9 qubits para

resolver un problema de búsqueda lo que se traduce como  $N = 2^9 = 512$  estados posibles y de la misma forma, para los ejemplos de tamaños más pequeños en estos estados, uno de sus datos va a representar una posible solución que queramos evaluar en el experimento.

Por lo consiguiente, se va a establecer un valor objetivo que lo consideraremos en adelante como el *target* que el algoritmo debe encontrar, para definir adecuadamente el oráculo y que el dato que queremos encontrar solo sea de verificación, éste debemos guardarlo en una función, tal como la mostramos en el código 5.1.

```
1 def f(x):  
2     return x == target_number
```

Listing 5.1: Función de comparación de un valor objetivo

Esto nos permite que, el oráculo pueda actuar para devolver True si  $x$  es la solución, y False si no lo es. Para un circuito cuántico, la representación de esta función permite marcar e invertir la fase de la solución sin revelar cuál es. Fuera de esta función, se definen otras variables en las que determinamos los “shots” (cantidad de repeticiones del experimento completo), uso de nube o local y la computadora cuántica *ibm\_brisbane* o el simulador *ibm\_quantum\_simulator*. Para el caso de las iteraciones, existe la forma de calcular el número de iteraciones óptimas dado el tamaño de búsqueda, algo que fue mencionado en el marco teórico pero, como requerimos hacer ejecuciones específicas para ver el comportamiento de los resultados las definimos manualmente en el código, igual en el experimento se deja la función para poderlos calcular. Debemos dar uso de cada variable que se estructura para ver el comportamiento del circuito, esto nos permite ver a detalle ver una comparativa del circuito de Grover en diferentes escenarios en una búsqueda no estructurada.

En la siguiente parte debemos definir los registros cuánticos y la adaptación a las compuertas de la computadora cuántica que deben ser compatibles con la *ibm\_brisbane* como lo mostramos en el código 5.2.

```
1 # COMPUERTAS NATIVAS DE BRISBANE  
2 def custom_h(qc, qubit):  
3     qc.rz(np.pi, qubit)  
4     qc.sx(qubit)  
5     qc.rz(np.pi, qubit)  
6     #qc.rz(-np.pi, qubit) para evitar fases globales que  
7     cancelan  
8 def custom_cx(qc, control, target):  
9     qc.rz(-np.pi/2, target)
```

```
10 qc.ecr(control, target)
11 qc.rz(np.pi/2, target)
```

Listing 5.2: Definición de compuertas nativas del backend `ibm_brisbane`

El Hardware cuántico *ibm\_brisbane* opera bajo un conjunto específico de compuertas base [`'ecr'`, `'id'`, `'rz'`, `'sx'`, `'x'`], estas compuertas deben descomponerse en la topología y su conectividad de la computadora cuántica representada en la figura 5.1, la figura 5.2 muestra el diagrama de calibración de la computadora cuántica *ibm\_brisbane*, cada fila corresponde a un qubit específico (del 0 al 126), y las columnas son las métricas que definen la calidad y el rendimiento de ese qubit en el momento de la calibración, la definición de cada compuerta la podemos ver en la tabla 5.1.

IBM Brisbane es un backend cuántico de IBM Quantum Platform con 127 qubits, basado en el procesador Eagle r3. Cuenta con una topología de acoplamiento heavy-hexagonal o heavy-hex lattice, es un diseño de red de acoplamiento de qubits en una estructura bidimensional hexagonal. Consiste en una malla donde la mayoría de los qubits tienen un grado de conectividad de 3 (aristas hexagonales), pero un subconjunto de qubits “pesado” (heavy qubits) en posiciones clave tienen hasta 6 conexiones, lo que optimiza la escalabilidad, reduce la profundidad de circuitos y facilita la corrección de errores en códigos, para entrelazamiento eficiente, soportando hasta 180 millones de operaciones CLOPS (Circuit Layer Operations Per Second). Lanzado en 2023 y actualizado en 2025, se usa para experimentos NISQ como algoritmos variacionales (VQE) y el algoritmo de Grover, con errores medios. Es accesible vía Qiskit para simulaciones y ejecuciones reales, ideal para pruebas de escalabilidad o para nuestro caso la ejecución de nuestro circuito estructurado para la solución de una búsqueda no estructurada [35].

Estas computadoras son ajustadas todos los días, al menos hasta la fecha (octubre del 2025), es por eso que los resultados coherentes que se exponen en esta investigación se repitieron hasta tener resultados consistentes con la hipótesis del trabajo. Además, la ecuación que valida como opera verdaderamente como una compuerta Hadamard es la Ecuación 5.1, lo mismo ocurre para la compuerta CNOT vista en la Ecuación 5.3 todas cumpliendo la estructura de las compuertas para una computadora cuántica y la que se usa en este trabajo [36, 37, 38, 39, 40].

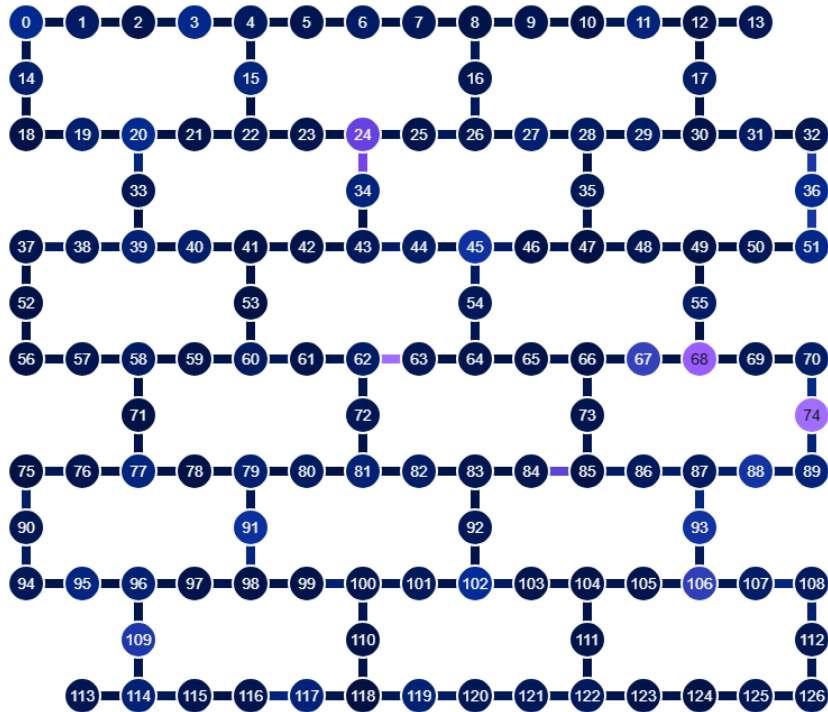


Figura 5.1: Topología y conectividad del procesador cuántico *ibm\_brisbane* 127 qubits. Imagen exportada desde IBM Quantum Platform, vista de calibración del backend. Fecha de acceso: 20 de octubre de 2025.

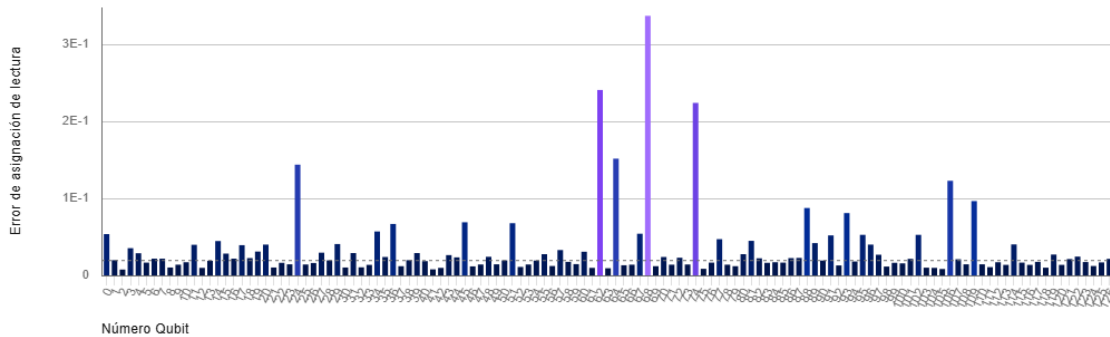


Figura 5.2: Calibración del procesador cuántico *ibm\_brisbane* 127 qubits. Imagen exportada desde IBM Quantum Platform, vista de calibración del backend. Fecha de acceso: 20 de octubre de 2025.

Cuadro 5.1: Resumen de compuertas cuánticas nativas utilizadas en la implementación

Compuerta	Tipo	Efecto	Uso en el algoritmo de Grover
ECR	2-qubit	Genera entrelazamiento con fase $i$	Base del <code>custom.cx</code> para multiCX (oráculo/difusor)
ID	1-qubit	Identidad, no modifica el estado	Usada automáticamente como relleno sin afectar el circuito
RZ( $\phi$ )	1-qubit	Rotación sobre eje Z (fase relativa)	Correcciones de fase en <code>custom.h</code> y <code>custom.cx</code>
SX	1-qubit	Rotación $\pi/2$ sobre eje X	Parte del Hadamard en <code>custom.h</code>
X	1-qubit	Pauli-X (NOT), intercambia $ 0\rangle \leftrightarrow  1\rangle$	Marca el target y aplica el difusor

La siguiente ecuación 5.1, representa la descomposición de la compuerta de Hadamard

$H$  utilizando solo compuertas nativas compatibles con los backends de IBM Quantum, específicamente **RZ** y **SX**. Esta técnica permite construir  $H$  sin introducir compuertas virtuales no nativas, lo cual mejora la fidelidad del circuito en implementaciones reales como *ibm\_brisbane*. Al aplicar una rotación  $R_z(\pi)$ , seguida de una **SX** (rotación de  $\pi/2$  sobre el eje X), y otra  $R_z(\pi)$ , se obtiene exactamente la matriz del operador Hadamard. Esta construcción se emplea en el diseño de la puerta `custom_h`, utilizada tanto para la superposición inicial como para el difusor de Grover.

$$R_z(\pi) \cdot SX \cdot R_z(\pi) = \begin{bmatrix} -i & 0 \\ 0 & i \end{bmatrix} \cdot \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} \cdot \begin{bmatrix} -i & 0 \\ 0 & i \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (5.1)$$

La compuerta ECR (Echoed Cross-Resonance) es nativa en procesadores IBM y se define como:

$$ECR = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & i & 0 \\ 1 & 0 & 0 & i \\ i & 0 & 0 & 1 \\ 0 & i & 1 & 0 \end{bmatrix} \quad (5.2)$$

La secuencia implementada transforma la compuerta ECR en una CNOT mediante la siguiente relación:

$$CNOT = (I \otimes R_z(\pi/2)) \cdot ECR \cdot (I \otimes R_z(-\pi/2)) \quad (5.3)$$

Las rotaciones  $R_z$  aplicadas al qubit *target* convierten la acción de ECR en la compuerta CNOT estándar, representada como:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.4)$$

La documentación de Qiskit dice formalmente que todo circuito realizado en simulación debemos traspilarlo para mapearlo a la ISA (Instruction Set Architecture) del backend, como la *ibm\_brisbane* para nuestro caso específico, es la que usaremos para realizar nuestros experimentos NISQ y comparativas mencionadas en esta investigación [36, 37].

### 5.1.2. Oráculo Elemental

El Oráculo de Grover es el factor fundamental de este algoritmo ya que nos permite marcar nuestro valor objetivo mediante la aplicación de una fase negativa, esto en un espacio vectorial de tamaño  $N$ . En el experimento de una búsqueda en una lista no estructurada para encontrar un dato definimos una función llamada *oracle\_brisbane* como se muestra en el código 5.3. La solución sigue un enfoque sistemático, se estructura como un circuito independiente para poder ser llamado múltiples veces durante la ejecución, esto de acuerdo al tamaño del problema, en el código 5.4 permitimos ocultar el *target* en lo que corre la ejecución cuántica ya que el oráculo no debe conocer la solución y gracias al *assert* garantizamos poder marcar exactamente un único objetivo.

```
1 def oracle_brisbane():
2     oracle_qc = QuantumCircuit(search_reg, ancilla, name="oracle
      ")
```

Listing 5.3: Definición del Oráculo *ibm\_brisbane*

```
1 # Usa f para encontrar target (oculto)
2 target_states = [i for i in range(N) if f(i)]
3 assert len(target_states) == 1, "f_debe_marcar_single_target
      para_Grover"
4 target = target_states[0] # Extrae sin hardcoded
5 target_bits = format(target, f'0{n}b')[::-1]
```

Listing 5.4: Definición del Oráculo para Encontrar Target *ibm\_brisbane*

La función del oráculo diseñada en este experimento hace que se permita ocultar el *target* objetivo, es una precomputación clásica que se realiza del lado del programa, no en el circuito cuántico. Por ende, permite ocultar y extraer de una forma dinámica el estado objetivo sin dárselo de forma directa al oráculo. Para el contexto del Algoritmo de Grover, el oráculo como sabemos bien es lo que nos permite aplicar una fase negativa sobre el estado marcado, no revela el estado, se entiende como una caja negra. Esta implementación, hace que la podamos usar en bases de datos de diferentes tamaños ya que se mantiene oculta la variable objetivo dado que, el oráculo se ejecuta una sola vez y no tiene que ver con iteraciones definidas o shots, por último, y no menos importante se formatea el *target* a un string binario para que la compuerta *multiCX* la pueda marcar. En el código 5.5 evidenciamos el núcleo cuántico del oráculo, definimos la lógica del oráculo como se apoya para marcar el dato objetivo permitiendo distinguir el *target*, sin esto el oráculo queda incompleto y es por eso que hace parte de toda la función *oracle\_brisbane* con el fin de al momento de pasar al difusor este entienda correctamente como amplificar el estado objetivo.

```

1 # Aplicar X donde bit es 0 (una vez, para marcar target)
2     for i, bit in enumerate(target_bits):
3         if bit == '0':
4             oracle_qc.x(search_reg[i])
5
6     # Fase -1: multiCX directo
7     for i in range(n):
8         custom_cx(oracle_qc, search_reg[i], ancilla[0])
9
10    # Revertir X (una vez)
11    for i, bit in enumerate(target_bits):
12        if bit == '0':
13            oracle_qc.x(search_reg[i])
14
15    return oracle_qc

```

Listing 5.5: Definición del Oráculo para Extraer el Target `ibm_brisbane`

### 5.1.3. Difusor

El Difusor, es la segunda parte fundamental del Algoritmo de Grover ya que se encarga de aumentar la amplitud del estado que marco el Oráculo que diseñamos para este experimento, dentro de la superposición cuántica el estado negativo se mantiene oculto, el difusor realiza una reflexión geométrica de todas las amplitudes alrededor del valor medio de la superposición uniforme. La función que se encarga de realizar este trabajo la llamamos *diffuser\_brisbane* implementa nuestra ecuación (4.46) para nuestros  $N$  qubits de búsqueda, código representado en listing 5.6 muestra como de manera eficiente usa las compuertas que brinda la computadora cuántica *ibm\_brisbane* para poder actuar sobre el espacio de vectorial, todas las iteraciones deben acoplarse con la computadora cuántica utilizada.

```

1 # --- DIFUSOR ---
2 def diffuser_brisbane():
3     qc_diff = QuantumCircuit(search_reg, name="diffuser")
4     for q in range(n):
5         custom_h(qc_diff, search_reg[q])
6         qc_diff.x(search_reg)
7
8     # Multi-CZ efectivo
9     custom_h(qc_diff, search_reg[-1])

```

```

10     for i in range(n - 1):
11         custom_cx(qc_diff, search_reg[i], search_reg[-1])
12     custom_h(qc_diff, search_reg[-1])
13
14     qc_diff.x(search_reg)
15     for q in range(n):
16         custom_h(qc_diff, search_reg[q])
17     return qc_diff

```

Listing 5.6: Definición del Difusor `ibm_brisbane`

El circuito es reutilizable creando un nuevo `QuantumCircuit` solo con el registro `search_reg` para poder obtener una visualización como se ve en la figura 5.3, partiendo del diseño se definen unas funciones para que las compuertas tengan una rotación uniforme y teniendo compatibilidad con la computadora cuántica que empleamos, mapea con ayuda de los bucles estructurados para aplicar cada compuerta y en la parte del *Multi-CZ*, siendo el componente principal del Difusor de Grover definiendo un único estado objetivo y no se puede adaptar a otros estados objetivos, si se quiere se debe cambiar toda la lógica del circuito cuántico.

Para la estructura se aplican unas capas de compuertas  $H$  que fueron personalizadas con `custom_h`, seguido de unas compuertas  $X$  y la *multi-CZ* revirtiendo las capas de  $X$  aplicando de nuevo las `custom_h`. El código se adaptó lo más posible a la computadora cuántica *ibm\_brisbane* para tener una compatibilidad lo más cercana con la topología mostrada en la figura 5.1, algo que garantiza su correcta ejecución.

Con lo anterior, se grafican los diagramas del circuito diseñado para tener una visual y mejor entendimiento del diseño del circuito que nos ayudó a resolver las búsquedas en estructuras desordenadas visto en la figura 5.3, esta imagen se ajusta a un problema de búsqueda en donde se utilizan 2 qubits el cual solo requiere una iteración para encontrar el *target* objetivo, la misma estructura se utilizó y ajustó para los experimentos con estructuras de datos más grandes al punto que se pudo realizar con 9 qubits.

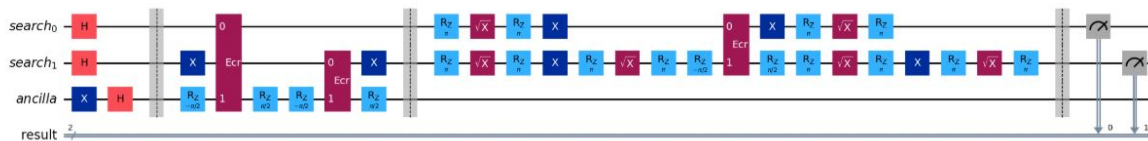


Figura 5.3: *Circuito de Grover Estructurado para `ibm_brisbane`. Basado en documentación de Qiskit [38].*

### 5.1.4. Ensamble del Circuito de Grover

Se realiza una arquitectura del circuito inicial con el código 5.7. Esto me permite integrar y realizar una combinación clara entre operaciones cuánticas y mediciones clásicas, sin antes preparar un estado inicial con ayuda del código 5.8 creando una superposición uniforme sobre todos los estados computacionales definidos para el problema, teniendo en cuenta que debemos facilitar la ejecución del algoritmo de la computadora cuántica junto con el análisis, la línea `qc.barrier` es fundamental para este proceso.

```
1 qc = QuantumCircuit(search_reg, ancilla, c_reg)
```

Listing 5.7: Inicialización `ibm_brisbane`

```
1 for q in range(n):
2     custom_h(qc, search_reg[q])
3 qc.x(ancilla[0])
4 custom_h(qc, ancilla[0]) # |->
5
6 qc.barrier()
```

Listing 5.8: Preparación `ibm_brisbane`

La construcción de los componentes del algoritmo se definen en este código 5.9, se construyen de forma independiente siendo reutilizables en diferentes iteraciones del circuito con la ventaja de ser validados por separado.

```
1 # Construir componentes
2 oracle = oracle_brisbane()
3 diffuser = diffuser_brisbane()
```

Listing 5.9: Componentes `ibm_brisbane`

El Algoritmo de Grover consiste en aplicar de forma iterativa el oráculo y el difusor. En esta implementación, se aplica el número de iteraciones definido en el código 5.10.

```
1 # Construir componentes
2 oracle = oracle_brisbane()
3 diffuser = diffuser_brisbane()
```

Listing 5.10: Iteraciones `ibm_brisbane`

En cada iteración se realiza de forma sistemática el cual se compone de Oráculo y seguido del Difusor, se aplica una compuerta  $Z$  en la ancilla para mantener la coherencia del estado cuántico. Seguimos usando las `qc.barrier` para aislar las iteraciones del circuito permitiendo una visualización del hardware real que se ve en el ambiente de

Qiskit Cloud. Una vez completada todas las iteraciones que se definen manualmente para ver el comportamiento se realiza una medición de todos los qubits del registro de búsqueda con el código 5.11, sin obviar la parte del *initial\_layout* que asigna los qubits lógicos a los físicos para el traspilador.

```
1 qc.measure(search_reg, c_reg)
2
3 initial_layout = list(range(n)) + [n//2 if not use_cloud else
4   63]
5 print(f"Layout inicial: {initial_layout}")
```

Listing 5.11: Medición *ibm\_brisbane*

### 5.1.5. Ejecución del Circuito en Computadora Cuántica

Durante la ejecución es necesario iniciar los servicios de Qiskit Runtime que me permiten conectar con mi token, canal e instancia. Esto se hace realizando el logeo en Qiskit Cloud y durante el registro se agregan los recursos para los cuales seleccionamos la computadora cuántica *ibm\_brisbane*, con esto Qiskit nos proporciona un entorno de ejecución en el que podemos ver la documentación de la computadora y todo el entorno de arquitectura para la ejecución. Qiskit nos indica que debemos realizar la traspilación del circuito cuántico acoplado a las características del hardware seleccionado como se muestra en el código 5.12 [36, 37, 40]. Existe una parte del resultado que muestra una ejecución usando 2 qubits para encontrar un número en un espacio de 4 estados, y se usa la librería *time* para medir el tiempo de ejecución de estos circuitos desde que se ejecuta en el notebook hasta que finaliza su ejecución en la computadora cuántica, el código sigue la misma arquitectura mostrada en la figura 5.3, concibiendo resultados de acuerdo a la lógica del Algoritmo de Grover y los parámetros seleccionados para cada experimento en cual los resultados del mismo se discutirán en el capítulo de los resultados.

```
1 service = QiskitRuntimeService(channel="ibm_cloud")
2 backend = service.backend("ibm_brisbane")
3 print(f"Backend: {backend.name}")
4 print(f"Status: {backend.status()}")
5
6 t_qc = transpile(qc, backend=backend, initial_layout=
7   initial_layout, optimization_level=3)
8 print(f"Profundidad post-transpile: {t_qc.depth()}")
9 print(f"Qubits mapeados: {len(t_qc.qubits)} nicos ")
10
```

```

10 Resultado Notebook
11 Backend: ibm_brisbane
12 Status: <qiskit_ibm_runtime.models.backend_status.BackendStatus
      object at 0x0000024DFCBEC050>
13 Profundidad post-transpile: 162
14 Qubits mapeados: 127 nicos
15 Ejecutando en Brisbane...
16 Job ID: d401avujbaoc73cqmtg - Track: https://quantum.cloud.ibm.
      com/
17 Completado.
18 Target '01': 30.97% (314/1014)
19 Top 5: {'11': 392, '01': 314, '10': 168, '00': 140}
20 Tiempo total de ejecución (cliente a resultado): 4.44 segundos

```

Listing 5.12: Backend IBM `ibm_brisbane`

### 5.1.6. Experimento en Simulador de Qiskit para un problema 3-SAT

Para un mejor entendimiento del funcionamiento del algoritmo de Grover y el uso del ambiente de Qiskit se procedió con una implementación usando el simulador de *aer\_simulator* aplicándolo a la solución de un problema 3-SAT. El proceso metodológico se centro en la construcción de un oráculo cuántico reversible, capaz de evaluar una Forma Normal Conjuntiva (CNF) en superposición. La arquitectura del oráculo se diseñó de forma que, utilizando registros cuánticos separados para las variables (`var_reg`), las cláusulas (`clause_reg`) y los qubits de trabajo (`g_reg`, `phase_reg`) facilitan la depuración y alineándose con las prácticas de diseño para hardware cuántico. El diseño de un oráculo robusto y reversible, como el implementado para este problema 3-SAT, representa el desafío central en la computación cuántica orientada a aplicaciones complejas. Tal como lo demuestran trabajos de benchmarking como el de Lubinski [41], el rendimiento real de los algoritmos cuánticos no depende de métricas aisladas, sino de la eficiencia y fidelidad con que se pueden ejecutar en situaciones específicas del problema.

El núcleo del oráculo consistió en un proceso de dos etapas: primero, la evaluación de cada cláusula individual (una operación OR) mediante la función `compute_clause_or`, que almacena de forma reversible el resultado en un qubit ancilla. Para manejar los literales negados ( $\neg x$ ) unidos al 3-SAT, se implementó una lógica de polaridad variable (`mcx_with_polarities`) que aplica compuertas  $X$  temporales a los controles correspondientes. Segundo, una compuerta `mcx` global actuó sobre todos los qubits de cláusula para computar el AND lógico de toda la fórmula, indicando si una asignación de va-

riables es una solución válida. La marcación de dicha solución se logró mediante el mecanismo de *phasekickback*, aplicando una compuerta CZ entre el qubit de trabajo global y el qubit de fase, previamente preparado en el estado. Finalmente, para preservar la coherencia y la unitariedad, se ejecutó un paso de descomputación riguroso, revirtiendo todas las operaciones de los pasos 1 y 2 para restaurar todos los qubits ancilla a su estado  $|0\rangle$  antes de la aplicación del operador de difusión estándar [15].

Es importante notar que la metodología de oráculo 3-SAT implementada en este trabajo, si bien es lógicamente correcta y funcional en simulación, no representa una implementación optimizada para el hardware cuántico actual (NISQ). El enfoque modular adoptado, que utiliza un qubit ancilla por cada cláusula y qubits de trabajo adicionales, genera una sobrecarga de recursos significativa. Esta implementación prioriza la claridad sobre la eficiencia. Investigaciones de gran nivel académico se centran en resolver la eficiencia de algoritmos como el Grover para recursos específicos [42].

La construcción de un oráculo específico para resolver un problema 3-SAT es una forma de ver la gran complejidad de abordar una investigación de este tipo, es por eso que nos centramos solo en el simulador para realizar este experimento debido a la complejidad de llevarlo a las computadoras cuánticas que ofrecía de forma gratuita IBM y teniendo estas limitaciones en compuertas y la arquitectura que no nos permitía implementar adecuadamente este resultado de simulación, de igual forma en medio de estas limitantes buscamos la mejor metodología para diseñar el circuito en el simulador, teniendo presentes trabajos de Al-Ghamdi [43], simplificando el algoritmo de grover a un gran nivel, algo que intentamos con este ejercicio en el simulador y que se visualiza en la figura 5.4 y la figura 5.5 [44].

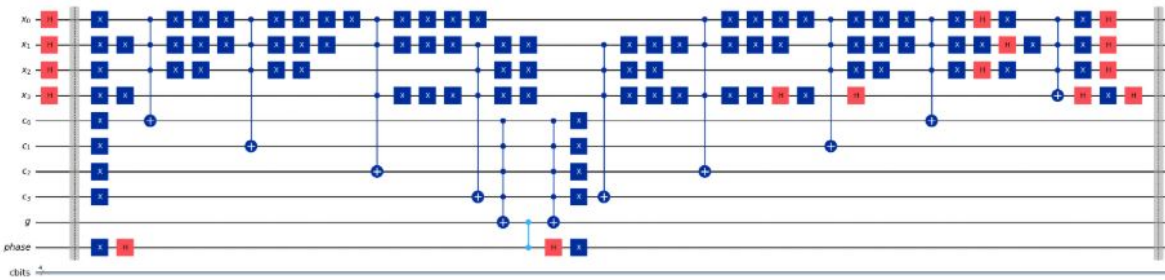


Figura 5.4: *Primera parte del Circuito de Grover Estructurado para un problema 3-SAT. Diagrama de resultado [44].*

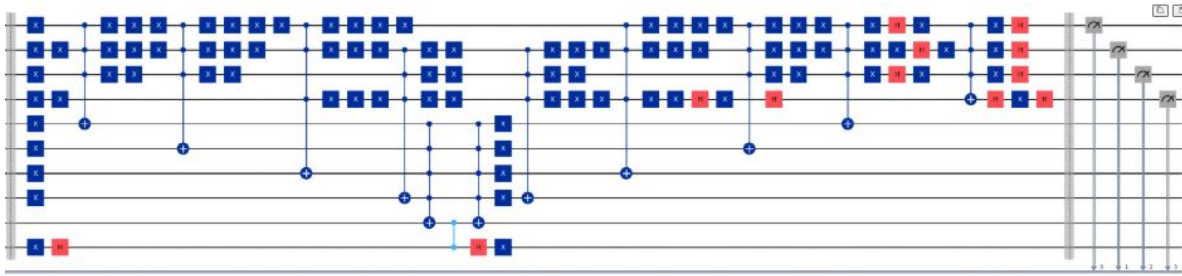


Figura 5.5: Segunda parte del Circuito de Grover Estructurado para un problema 3-SAT. Diagrama de resultado [44].

Como complemento, se procedió con la validación del circuito cuántico ejecutado en el simulador y el análisis de los datos resultantes evidenciando la profundidad y nivel de detalle en cada paso al crear un circuito que resuelve un problema 3-SAT ya que al pertenecer a un problema NP-Complejo requiere estructurar adecuadamente el circuito para lograr que el resultado satisfaga la ecuación, algo que se evidencia en los resultados del simulador.

## Capítulo 6

# RESULTADOS Y DISCUSIÓN

### 6.0.1. Computador Cuántico *ibm\_brisbane*

Con el fin, de tener en cuenta la importancia de elegir apropiadamente el método de análisis a emplear, el cual debe estar alineado con la pregunta de investigación a la cual se quiere dar solución [45], para este estudio se aprovecho el análisis comparativo de tendencias y comportamiento global, donde nos permitió encontrar un patrón característico consistente para una serie de qubits, iteraciones y shots configurados en el experimento.

Una vez diseñado el circuito que contempla toda la estructura lógica del algoritmo de Grover para una búsqueda en una lista no ordenada, se realizan una serie de ejecuciones teniendo un comportamiento que se distingue una vez aumentan la cantidad de qubits, estas comparativas necesarias de acuerdo a los parámetros contemplados las podemos visualizar en el cuadro 6.1, los notebooks de la implementación computacional están en el repositorio enlace del repositorio.

Cuadro 6.1: Resultados obtenidos del algoritmo cuántico en *ibm\_brisbane*

Cantidad de Qubits	Cantidad de Estados	Número de Iteraciones	Target Objetivo	Otros Amplificados	Encontrado	Shots
2	4	1	01	11	Sí Óptima	512
3	8	2	101	111, 001	Sí Óptima	1024
3	8	3	101	010,011,000,111,110,001	Sí	1024
4	16	1	0101	1011,0000,0010	Sí	512
4	16	2	0101	0111,0000,0011,0100	Sí	512
4	16	3	0101	1111,0001,1010	Sí	512
4	16	4	0101	0111,0010,1001	Sí Óptima	512
5	32	3	10000	00111,11111	Sí	512
5	32	4	10000	00111,11111	Sí	512
5	32	4	10000	10001	Sí	512
5	32	5	10000	10001,00000,11001	Sí Óptima	512
5	32	6	10000	00110,10100	Sí	512
6	64	2	010111	111111,110001	Sí	1024
6	64	3	010111	011011,011110,001010	Sí	1024
6	64	4	010111	111110,001010	Sí	1024
6	64	5	010111	001011	Sí	1024
7	128	5	0011000	Ruido	No	512
7	128	6	0011000	Ruido	No	512
7	128	7	0011000	Ruido	No	1024
7	128	8	0011000	Ruido	No	4048
7	128	9	0011000	Ruido	No	512
7	128	13	0011000	Ruido	No	512
9	512	8	000110101	Ruido	No	1024
9	512	10	000110101	Ruido	No	512
9	512	14	000110101	Ruido	No	1024
9	512	17	000110101	Ruido	No	1024

De lo anterior, y como se visualiza en la tabla de resultados 6.1, se destacan efectos cuánticos siendo uno de ellos al momento de encontrar el *target* objetivo, cuando el estado es un "Sí Óptimo", es porque se encuentra el dato y es el más amplificado de todos los que se amplifican así hayan otros pero no superando este, para el caso de Sí se

refiere a que se encuentra amplificado pero su amplificación no es un resultado óptimo y hay otros que no corresponde al target objetivo en la medición, estos se entienden como una dispersión probabilística y es una connotación clara del algoritmo de Grover ya que en medio de la rotación existe una fuga de probabilidad dada las iteraciones del circuito [46].

Cuadro 6.2: Resultados de la simulación del algoritmo cuántico *ibm\_quantum\_simulator*

Cantidad de Qubits	Cantidad de Estados	Número de Iteraciones	Target Objetivo	Otros Amplificados	Encontrado	Shots
2	4	2	01	11,00,10	Sí Óptima	512
3	8	2	101	110,000,011,010,100,001,111	Sí	1024
3	8	3	101	010,011,000,111,110,001	Sí	1024
3	8	4	101	110,010,001	Sí Óptima	1024
4	16	4	0101	1101,0010,1010	Sí Óptima	512
5	32	4	10000	00000,01111,11111	Sí	512
5	32	5	10000	Ruido	No	512
5	32	6	10000	Ruido	Sí	512
6	64	6	010111	Ruido	No	1024
6	64	13	010111	101000,001000,110111	Sí Óptima	1024
7	128	9	0010111	Ruido	No	1024
7	128	13	0010111	1010111,1101000,0101000	Sí Óptima	1024
8	256	13	00011000	11100111,01100111,10011000	Sí Óptima	1024
9	512	18	000110101	Ruido	No	1024
9	512	19	000110101	111001010,100110101,011001010	Sí	2048

De igual forma, se comparan resultados obtenidos en el simulador cuántico *Aer Simulator* y se observa un comportamiento similar, aunque en los casos de 6 a 8 qubits encuentra el *target* sin ruido, cuando este se aumenta a 9 qubits se comporta de la misma forma que una computadora cuántica, generando ruido en el éxito de la búsqueda sin encontrar el target objetivo pero, en esos casos específicos tiene un comportamiento acorde a lo esperado de la búsqueda de grover para el simulador y lo podemos ver en el cuadro de resultados 6.2.

Del mismo modo, los resultados que se observan en el histograma visto en la figura 6.1, el cual corresponde a una búsqueda de Grover en una base de datos no estructurada donde se busca un dato objetivo (target) en diferentes tamaños de búsqueda parametrizando cada experimento para ver el comportamiento una vez este aumenta. Este resultado, nos muestra desviaciones de resultados por hardware NISQ, se deben a factores en los que necesariamente por el tamaño de la búsqueda se aumentan qubits pero estos pueden contener errores de compuerta, de lectura, el qubit puede perder información durante el tiempo de la ejecución y claramente la topología de la computadora cuántica utilizada no brinda fiabilidad dado que aumenta la profundidad del circuito y el ruido se acumula. Asimismo, ajustar las iteraciones, la cantidad de shots y ajustar el circuito no ayuda en gran parte a encontrar el target objetivo debido al nivel de operaciones que este realiza, en la figura 6.2 en la descripción de la figura se indica la cantidad de iteraciones que se parametrizaron al igual que los shots, aun así genera ruido y el *target* no se encuentra en el top 5 de conteos cuando finaliza la ejecución.

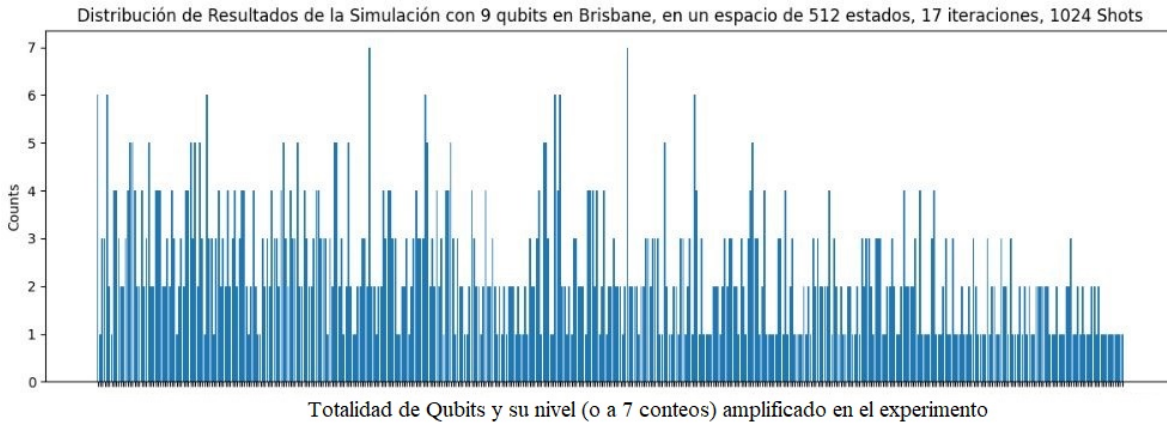


Figura 6.1: *Histograma para 9 qubits en ibm\_brisbane. Resultados del trabajo [47].*

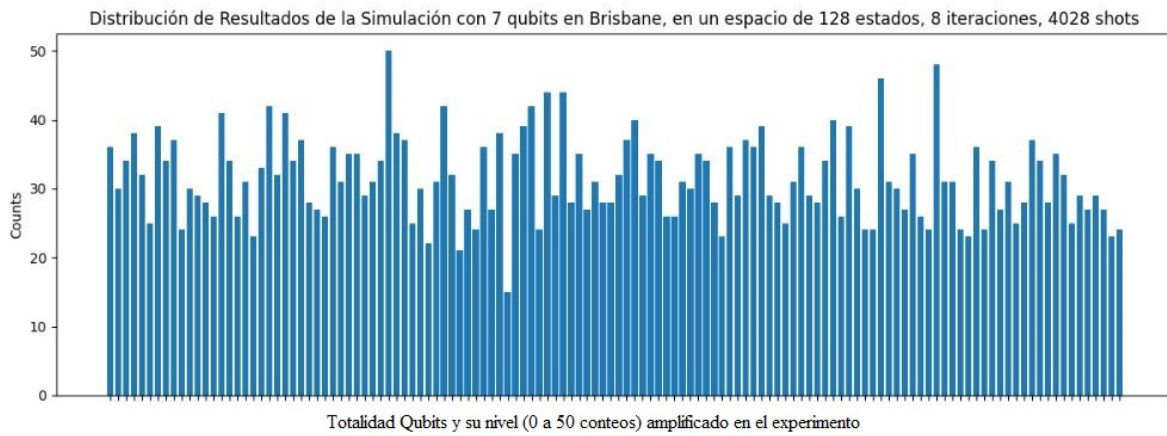


Figura 6.2: *Histograma para 7 qubits en ibm\_brisbane. Resultados del trabajo [47].*

Para un tamaño de búsqueda reducida, la búsqueda es efectiva para el circuito diseñado, en la figura 6.3 se observa el dato 0101 el cual es el *target* objetivo y cuyos conteos superan los 80 basándose en los counts de la izquierda en la figura los cuales van de 10 en 10. Con lo anterior, el objetivo Sí es óptimo teniendo un comportamiento diferente y más preciso en su ejecución, al momento de que se disminuye la cantidad de qubits podemos ver el comportamiento ideal para una búsqueda en datos desordenados.

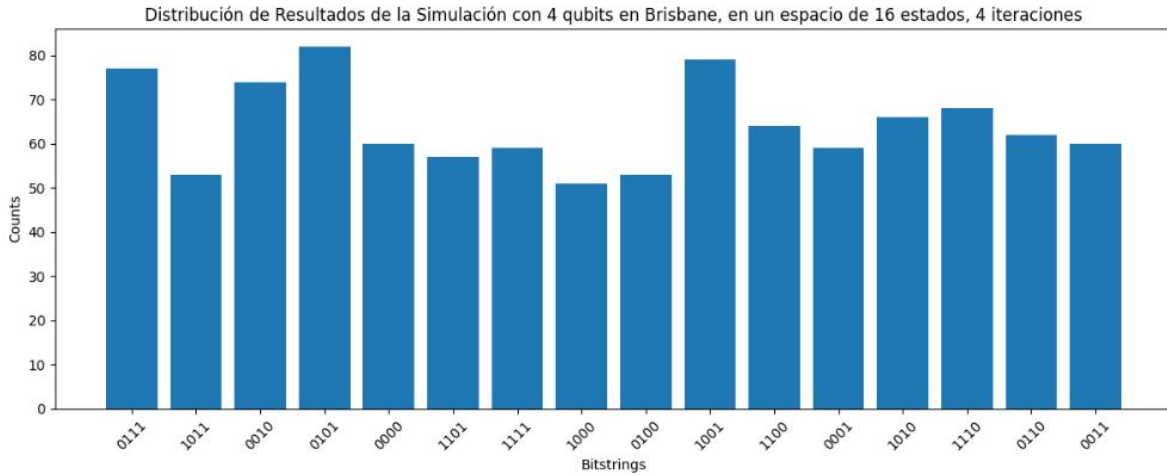


Figura 6.3: *Histograma para 4 qubits en ibm\_brisbane. Resultados del trabajo [47].*

Por el contrario, se evidencia que en los experimentos realizados con el simulador *AerSimulator* usando la misma estructura del circuito se tiene un mejor comportamiento, esto nos garantiza que el diseño del circuito cumple la funcionalidad, en el top 5 de conteos de datos el *target* objetivo para una búsqueda en 512 estados con 10 iteraciones mínimas aparece unas 252 veces en 1014 shots, es una probabilidad del 24,85% esto dado que la probabilidad se reparte en el top 5 de los conteos realizados y más altos de la medición, los podemos ver en la figura 6.4.

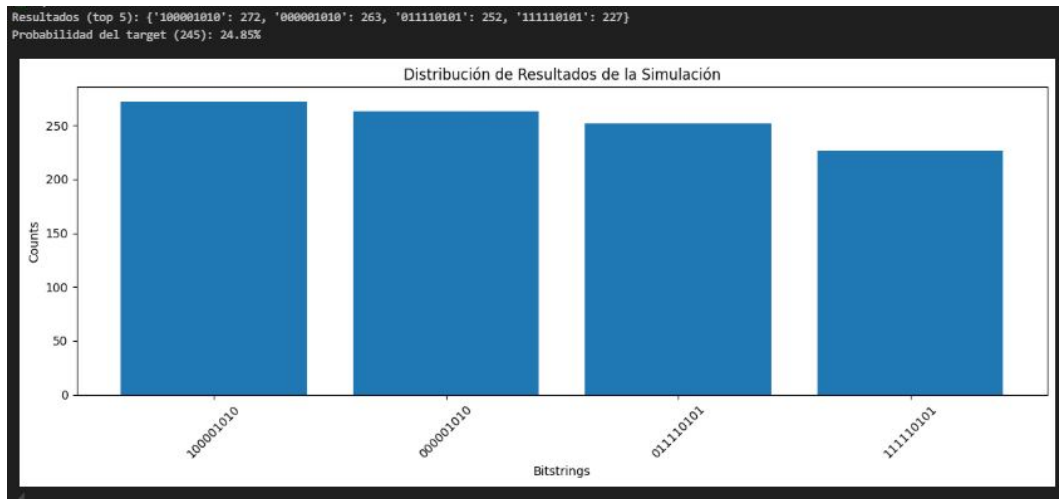


Figura 6.4: *Histograma Simulador Qiskit para 9 qubits en AerSimulator. Resultados del trabajo [47].*

## 6.0.2. Simulador Cuántico *AerSimulator*, Tiempo y Búsqueda Clásica

En cuando a los experimentos realizados en el simulador *AerSimulator* en el cual se realizo una búsqueda no estructurada para resolver un problema 3-SAT perteneciente a los NP-Complejos obtuvimos un mejor comportamiento de la simulación y los resultados dieron garantía, aunque requiere más verificaciones, las arquitecturas de simulación sirven para resolver un problema en este contexto, como lo muestra la figura 6.5.

Los resultados, demuestran una amplificación significativa de los estados cuánticos que satisfacen la instancia específica de 3-SAT implementada, consistente en cuatro cláusulas donde cada una contiene exactamente tres literales. Esta amplificación selectiva evidencia el correcto funcionamiento del oráculo cuántico, el cual marca precisamente las asignaciones de variables que cumplen simultáneamente todas las cláusulas de la fórmula booleana. El patrón de interferencia constructiva generado por las iteraciones de Grover aumenta drásticamente la probabilidad de medición de estos estados solución, confirmando la capacidad del algoritmo para resolver problemas de satisfacibilidad booleana mediante explotación de propiedades cuánticas [17]. Los resultados que los podemos distinguir en esta matriz de densidad de la figura 6.6 los cuales son amplificados aquellos que satisfacen mejor la ecuación.

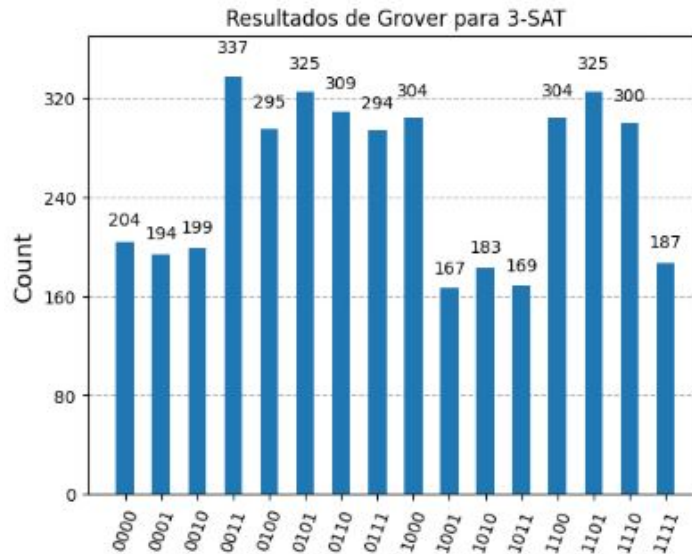


Figura 6.5: *Histograma Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo [47].*

La matriz de la figura 6.6 es un mapa de calor 2D que representa la matriz de densidad final de los 4 qubits de variables, mostrando el resultado de la simulación del algoritmo de Grover. Los ejes X e Y enumeran los estados posibles (desde  $|0000\rangle$  hasta

$|1111\rangle$ ), y el brillo de la diagonal principal revela la probabilidad de medición de cada estado.

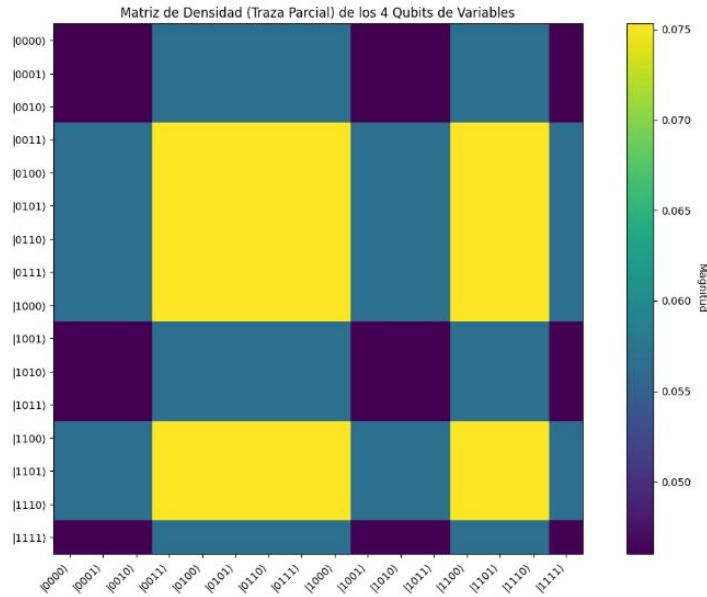


Figura 6.6: *Matriz de Densidad Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo [47].*

Otra forma de ver como se acumulan los resultados y que nos permite tener un visual mayormente descriptivo del resultado del experimento es un *CityPlop*, se genera a partir de la matriz de densidad y de acuerdo a los resultados consistentes se empiezan a reunir en un solo punto, esta se desfasa aumentando sus objetivos en los que más se amplifica dentro de la medición, lo podemos ver en la figura 6.8 con tres iteraciones y en la figura 6.7 para una iteración. Vemos que la ejecución del circuito en el simulador es óptima con tres iteraciones y con una sola se ven amplitudes que no satisfacen el problema 3-SAT.

### Matriz de Densidad de los 4 Qubits de Variables

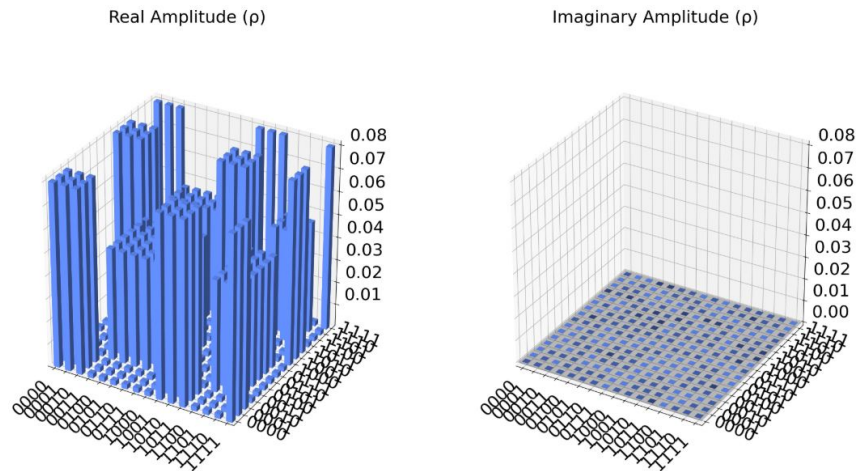


Figura 6.7: *City Plot de Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo para 1 iteración [47].*

### Matriz de Densidad de los 4 Qubits de Variables

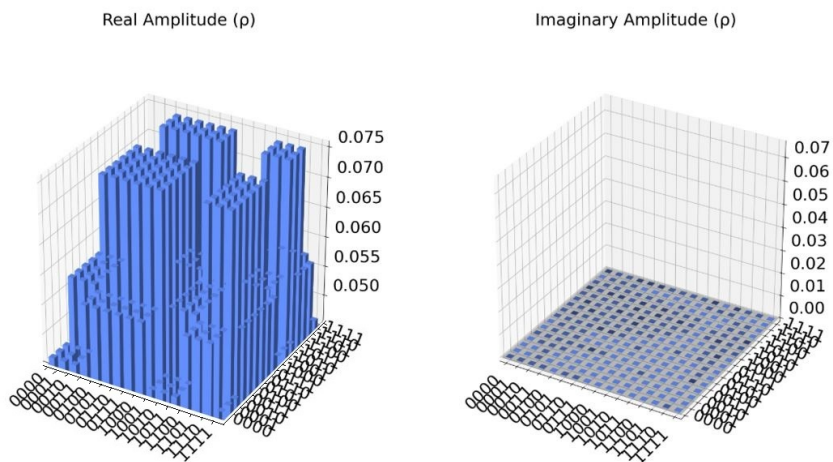


Figura 6.8: *City Plot de Simulador Qiskit para 3-SAT en AerSimulator. Resultados del trabajo para 3 iteraciones [47].*

Los resultados arrojados en comparación con el tiempo de ejecución tanto en una búsqueda clásica como para una búsqueda realizada en una computadora cuántica reflejan un escenario favorable en tiempo de ejecución, para una búsqueda cuántica no superan los 2 segundos de ejecución, se ve reflejado en la figura 6.9, donde por cada experimento de búsqueda tarda uno o dos segundos máximo en completar la búsqueda esto para los diferentes tamaños datos basados en casi 108 experimentos realizados. De

igual forma, se realizó el ejercicio aumentando la ejecución en 15 qubits que lo permitió la estructura del circuito y el resultado mantiene el mismo nivel de ruido en la búsqueda, obteniendo un tiempo de ejecución para una lista de un tamaño de 32768 elementos de 53 segundos. En una búsqueda clásica, este resultado lo arrojo en solo 17.1 microsegundos ( $\mu s$ ), podemos ver el resultado en la figura 6.10 también podemos observar el comportamiento de la búsqueda ya que en cada intento existe un pico de ruido y este va incrementando cada vez más en cada intento, como el tamaño de búsqueda es pequeño no se refleja esta deficiencia en la tecnología clásica notoriamente pero si el tamaño de búsqueda y el número de operaciones van aumentando esos ruidos cada vez más serán mayores según nuestro análisis y lo que refleja el resultado de la gráfica realizando la misma búsqueda para un solo dato en el mismo tamaño de datos.

Con ello, el análisis de los tiempos de ejecución obtenidos para los diferentes tamaños de datos evidencia un comportamiento acorde con las expectativas teóricas dentro del rango experimental al que tuvimos acceso. Sin embargo, es importante destacar que las limitaciones en la escala de los experimentos realizados no permiten una verificación empírica completa de los modelos de complejidad computacional asintótica [48].

The screenshot shows the IBM Quantum Platform interface. At the top, there's a search bar and navigation elements. Below that, the page title is 'Cargas de trabajo' (Jobs) with a filter for 'us-east'. A subtitle reads: 'Realice un seguimiento del estado y los resultados de las cargas de trabajo que haya ejecutado en las unidades de procesamiento cuántico de IBM.' Below this is a table with columns: ID, Estado, Instancia, Modo, Creado, Completado, QPU, Uso, Usuario, and Etiquetas. The table lists 10 jobs, all with 'Completado' status, using 'AlgoritmoGV\_01' on the 'ibm\_brisbane' QPU. The completion times range from 1s to 3s. At the bottom, there are pagination controls showing '11 páginas'.

ID	Estado	Instancia	Modo	Creado	Completado	QPU	Uso	Usuario	Etiquetas
d40112e60rg73...	Completado	AlgoritmoGV_01	Trabajo	27 oct 2025	27 oct 2025	ibm_brisbane	2s	Stiven Paloma Alarcon	:
d3up2spg33c7...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	3s	Stiven Paloma Alarcon	:
d3up1nc60rg73...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	2s	Stiven Paloma Alarcon	:
d3up0rcv6o9s73...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	2s	Stiven Paloma Alarcon	:
d3up066jbaoc73...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	2s	Stiven Paloma Alarcon	:
d3uovq460rg73...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	2s	Stiven Paloma Alarcon	:
d3uouuv6o9s73...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	2s	Stiven Paloma Alarcon	:
d3uou4hsg33c7...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	1s	Stiven Paloma Alarcon	:
d3uossujbaoc73...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	1s	Stiven Paloma Alarcon	:
d3uosspsg33c73...	Completado	AlgoritmoGV_01	Trabajo	25 oct 2025	25 oct 2025	ibm_brisbane	1s	Stiven Paloma Alarcon	:

Figura 6.9: Imagen IBM Quantum Platform para todos los experimentos *ibm\_brisbane*. Resultados del trabajo [47].

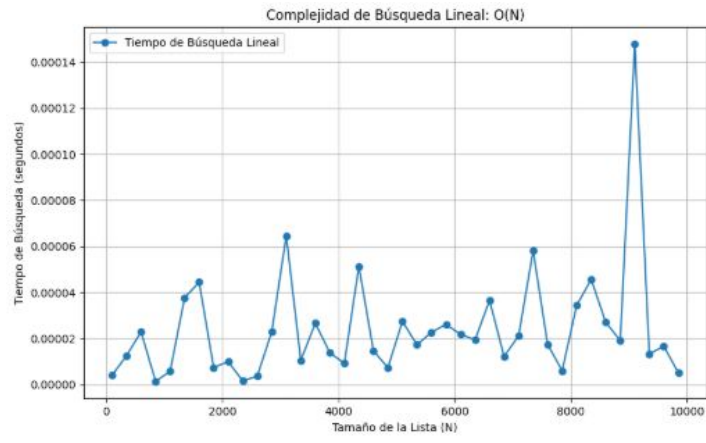


Figura 6.10: *Imagen Tiempo Búsqueda Clásica. Resultados del trabajo [47].*

# CONCLUSIONES Y RECOMENDACIONES

En el presente trabajo, se llevaron a cabo una serie de experimentos en los cuales se implementó el algoritmo de Grover para poder dar solución un problema de búsqueda en bases de datos no estructuradas. En la primera implementación, se diseñó, estructuró y ejecutó un circuito cuántico que resuelve un problema de búsqueda no estructurada con una arquitectura específica y conteniendo la lógica del algoritmo de Grover para el computador cuántico de IBM llamado *ibm\_brisbane*, se parametrizó en diferentes cantidades de qubits los cuales arrojaron diferentes comportamientos una vez estos aumentaban en su tamaño de datos, lo que muestra que las computadoras cuánticas en la era NISQ, para estos experimentos, presentan todavía ruido que desestabiliza las mediciones para este tipo de implementaciones una vez estas escalan a un tamaño de búsqueda mayor.

Dentro de la segunda fase de experimentos, se desarrollaron en el simulador cuántico de IBM *AerSimulator* en el cual se quiso dar solución a un problema de búsqueda no estructurada 3-SAT. En el cual, se diseñó un circuito cuántico específico haciendo uso de las compuertas nativas de IBM para el diseño de la estructura en este tipo de simulador cuántico, cabe destacar que se requieren más verificaciones para poder encontrar una solución óptima dando uso del algoritmo de Grover y tecnologías cuánticas, ya que el comportamiento en el simulador fue sobresaliente pero, para una implementación en backend real se debe tomar otras alternativas de diseño dada la complejidad del problema.

Los resultados de nuestro enfoque abordan ejecuciones exitosas con cantidades de qubits que superan las estructuradas en otras investigaciones, estas requieren mejoras, y por su puesto una implementación en otras computadoras cuánticas para poder salir del concepto teórico y entrar en la solución práctica de problemas de búsqueda de la actualidad.

## BIBLIOGRAFÍA

- [1] Laszlo Gyongyosi y Sandor Imre. «A Survey on Quantum Computing Technology». En: *Computer Science Review* 31 (2019), págs. 51-71. DOI: 10.1016/j.cosrev.2018.11.002.
- [2] Michael A. Nielsen e Isaac L. Chuang. *Quantum Computation and Quantum Information*. 10th Anniversary Edition. Cambridge, UK: Cambridge University Press, 2010. ISBN: 978-1107002173.
- [3] David Bohm. *Wholeness and the Implicate Order*. London: Routledge, 1980.
- [4] Donald E. Knuth. *The Art of Computer Programming, Volume 3: Sorting and Searching*. 2nd. Sección 6.1. Addison-Wesley, 1998.
- [5] Thomas H. Cormen et al. *Introduction to Algorithms*. 3rd. Cambridge, MA: MIT Press, 2009. ISBN: 978-0262033848.
- [6] Lov K. Grover. «A fast quantum mechanical algorithm for database search». En: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*. 1996, págs. 212-219. DOI: 10.1145/237814.237866. URL: <https://doi.org/10.1145/237814.237866>.
- [7] Lov K. Grover. *Quantum Mechanics helps in searching for a needle in a haystack*. Preprint. Bell Labs, 3C-404A, 600 Mountain Avenue, Murray Hill, NJ 07974. 1997. URL: <https://arxiv.org/abs/quant-ph/9706033>.
- [8] Charles H. Bennett et al. «Strengths and Weaknesses of Quantum Computing». En: *SIAM Journal on Computing* 26.5 (1997), págs. 1510-1523. DOI: 10.1137/S0097539796300933. URL: <https://epubs.siam.org/doi/10.1137/S0097539796300933>.
- [9] Jeffrey Scott Vitter y Philippe Flajolet. «Analysis of Algorithms and Data Structures». En: *Handbook of Theoretical Computer Science A* (1990), págs. 431-524.
- [10] Peter W. Shor. «Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer». En: *SIAM Journal on Computing* 26.5 (1997), págs. 1484-1509. DOI: 10.1137/S0097539795293172. URL: <https://doi.org/10.1137/S0097539795293172>.
- [11] Zhen Wang, Kun Zhang y Vladimir Korepin. «Near-deterministic quantum search algorithm without phase design». En: *arXiv* (2024). arXiv: 2407.10748 [quant-ph]. URL: <https://arxiv.org/abs/2407.10748>.

- [12] Tom Lubinski et al. «Application-Oriented Performance Benchmarks for Quantum Computing». En: *arXiv preprint arXiv:2110.03137* (2021). URL: <https://arxiv.org/abs/2110.03137>.
- [13] Sebastian Krinner et al. «Programmable Superconducting Processor with Native Three-Qubit Gates». En: *Physical Review Applied* 14.1 (2020), pág. 014072. DOI: 10.1103/PhysRevApplied.14.014072. URL: <https://journals.aps.org/prapplied/abstract/10.1103/PhysRevApplied.14.014072>.
- [14] Yingyue Zhu et al. «Characterizing Grover search algorithm on large-scale superconducting quantum computers». En: *Scientific Reports* 14.1 (2024), pág. 3792. DOI: 10.1038/s41598-024-80188-6. URL: <https://www.nature.com/articles/s41598-024-80188-6>.
- [15] Shouvanik Chatterjee et al. «Better-than-classical Grover search via quantum error detection and suppression». En: *arXiv preprint arXiv:2211.04543* (2022). URL: <https://arxiv.org/abs/2211.04543>.
- [16] Stephen A. Cook. «The complexity of theorem-proving procedures». En: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. New York, NY, USA: Association for Computing Machinery, 1971, págs. 151-158. DOI: 10.1145/800157.805047. URL: <https://doi.org/10.1145/800157.805047>.
- [17] Juan Francisco Díaz. *3SAT es NP Completo*. Profesor titular, Escuela de Ingeniería de Sistemas y Computación. Universidad del Valle. 2020. URL: <https://www.youtube.com/watch?v=ruiuOSkd-pE&list=LL&index=1&t=4s> (visitado 21-09-2025).
- [18] Sanjeev Arora y Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge, UK: Cambridge University Press, 2009.
- [19] Richard M. Karp. «Reducibility Among Combinatorial Problems». En: *Complexity of Computer Computations*. Boston, MA: Springer, 1972, págs. 85-103. DOI: 10.1007/978-1-4684-2001-2\_9. URL: [https://www.researchgate.net/publication/221580898\\_Reducibility\\_Among\\_Combinatorial\\_Problems](https://www.researchgate.net/publication/221580898_Reducibility_Among_Combinatorial_Problems).
- [20] Rolando D. Somma, Daniel Nagaj y Mária Kieferová. «A quantum algorithm for the 3-Satisfiability problem». En: *arXiv preprint quant-ph/0507200* (2005). Los autores presentan un algoritmo cuántico para 3-SAT que combina el algoritmo de Grover con una heurística clásica, confirmando la aceleración cuadrática esperada. URL: <https://arxiv.org/abs/quant-ph/0507200>.
- [21] Gordon E. Moore. «Cramming More Components onto Integrated Circuits». En: *Electronics* 38.8 (abr. de 1965). Artículo donde se enuncia la ley de Moore, págs. 114-117.

- [22] Richard P. Feynman. «Simulating Physics with Computers». En: *International Journal of Theoretical Physics* 21.6-7 (1982), págs. 467-488. DOI: 10.1007/BF02650179.
- [23] John Preskill. «Quantum computing and the entanglement frontier». Rapporteur talk at the 25th Solvay Conference on Physics (Brussels, Oct. 2011). 2012. arXiv: 1203.5813 [quant-ph]. URL: <https://arxiv.org/abs/1203.5813>.
- [24] Ashley Montanaro. «Quantum algorithms: an overview». En: *npj Quantum Information* 2 (2016), pág. 15023. DOI: 10.1038/npjqi.2015.23.
- [25] John Preskill. «Quantum Computing in the NISQ era and beyond». En: *Quantum* 2 (ago. de 2018), pág. 79. DOI: 10.22331/q-2018-08-06-79. arXiv: 1801.00862 [quant-ph].
- [26] Dinesh Reddy Vemula et al. «A Scalable 5,6-Qubit Grover’s Quantum Search Algorithm». En: (2022). arXiv: 2205.00117 [quant-ph]. URL: <https://arxiv.org/abs/2205.00117>.
- [27] Qiskit. *Writing and Running Quantum Programs — Part 2*. YouTube video. Playlist: Writing and Running Quantum Programs (Video 5). URL: <https://www.youtube.com/watch?v=iJX794qJIpY> (visitado 30-09-2025).
- [28] Qiskit. *Writing and Running Quantum Programs — Part 3*. YouTube video. Playlist: Writing and Running Quantum Programs (Video 6). URL: <https://www.youtube.com/watch?v=enYP9TaDeJI> (visitado 30-09-2025).
- [29] Gayathree M. Vinod y Anil Shaji. «Finding Solutions to the Integer Case Constraint Satisfiability Problem Using Grover’s Algorithm». En: *IEEE Transactions on Quantum Engineering* 2 (2021), págs. 1-13. DOI: 10.1109/TQE.2021.3120449.
- [30] Christos Papadimitriou. «Algorithms, Complexity, and the Sciences». En: *Proceedings of the National Academy of Sciences of the United States of America* 111.45 (2014), págs. 15881-15887. ISSN: 0027-8424. DOI: 10.1073/pnas.1416954111. URL: <https://www.pnas.org/doi/10.1073/pnas.1416954111> (visitado 29-09-2025).
- [31] David Deutsch y Richard Jozsa. «Rapid Solution of Problems by Quantum Computation». En: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439 (1992), págs. 553-558. DOI: 10.1098/rspa.1992.0167. URL: <https://royalsocietypublishing.org/doi/10.1098/rspa.1992.0167>.
- [32] Michael A. Nielsen e Isaac L. Chuang. «Quantum Computation and Quantum Information». En: Capítulo 4: Quantum Circuits. Cambridge University Press, 2010. Cap. 4, págs. 162-208.

- [33] Michael A. Nielsen e Isaac L. Chuang. «Quantum Computation and Quantum Information». En: Capítulo 6: Quantum Algorithms. Cambridge University Press, 2010. Cap. 6, págs. 244-282.
- [34] Tarun Kumar, Dilip Kumar y Gurmohan Singh. «Noise Analysis of Grover’s Quantum Search Algorithm». En: *Indian Journal of Pure & Applied Physics* 61.5 (2023), págs. 355-365. DOI: 10.56042/ijpap.v61i5.69090. URL: <https://doi.org/10.56042/ijpap.v61i5.69090>.
- [35] IBM Quantum Team. *IBM Brisbane 127-Qubit Processor*. <https://quantum.cloud.ibm.com/docs/guides/get-qpu-information>. Disponible vía Qiskit, topología heavy-hex, backend NISQ. 2023. URL: <https://quantum.cloud.ibm.com/docs/guides/get-qpu-information>.
- [36] IBM Quantum Team. *Introduction to Transpilation*. <https://quantum.ibm.com/docs/guides/transpilation-intro>. IBM Quantum Documentation, Accedido: 3 de septiembre de 2025. Sep. de 2025. (Visitado 03-09-2025).
- [37] IBM Quantum Team. *Transpiler API*. <https://quantum.ibm.com/docs/api/qiskit/transpiler>. IBM Quantum Documentation, Accedido: 3 de septiembre de 2025. Sep. de 2025. (Visitado 03-09-2025).
- [38] Qiskit Community. *Qiskit Documentation Mirror*. <https://qiskit.org/documentation/>. Espejo comunitario de Qiskit, Accedido: 3 de septiembre de 2025. Sep. de 2025. (Visitado 03-09-2025).
- [39] Qiskit Team. *Qiskit Documentation GitHub Repository*. <https://github.com/Qiskit/qiskit/tree/main/docs>. Fuente open-source de documentación Qiskit, Accedido: 3 de septiembre de 2025. GitHub, 2025. (Visitado 03-09-2025).
- [40] IBM Quantum Team. *Qiskit IBM Transpiler*. [https://github.com/Qiskit/qiskit-ibm-runtime/tree/main/qiskit\\_ibm\\_runtime/transpiler](https://github.com/Qiskit/qiskit-ibm-runtime/tree/main/qiskit_ibm_runtime/transpiler). Biblioteca para transpilación optimizada en IBM hardware, Accedido: 3 de septiembre de 2025. GitHub, 2025. (Visitado 03-09-2025).
- [41] T. Lubinski et al. «Application-Oriented Performance Benchmarks for Quantum Computing». En: *arXiv preprint* (2021). Disponible en: <https://arxiv.org/abs/2110.03137>. arXiv: 2110.03137 [quant-ph].
- [42] T. Roy et al. «Programmable Superconducting Processor with Native Three-Qubit Gates». En: *Physical Review Applied* 14.1 (2020), pág. 014072. DOI: 10.1103/PhysRevApplied.14.014072.

- [43] S. Al-Ghamdi et al. «A simplified and efficient quantum circuit for the oracle in Grover's algorithm». En: *Quantum Information Processing* 20.218 (2021). DOI: 10.1007/s11128-021-03165-2. URL: <https://link.springer.com/article/10.1007/s11128-021-03165-2>.
- [44] IBM Quantum Learning. *Grover's algorithm*. <https://learning.quantum.ibm.com/tutorial/grovers-algorithm>. Recuperado el 4 de octubre de 2025. 2024. (Visitado 04-10-2025).
- [45] F. Glassman. «MANOVA: A Procedure Whose Time Has Passed?» En: *Gifted Child Quarterly* 64.1 (2020), págs. 56-60.
- [46] Gui Lu Long et al. «Quantum searching and the optimal iteration number». En: *Physical Review A* 60.4 (1999), pág. 2742. DOI: 10.1103/PhysRevA.60.2742.
- [47] Stiven Paloma Alarcon. *Implementación del algoritmo de Grover para simulaciones cuánticas*. <https://github.com/Stivenalp/Algoritmo-de-grover>. Repositorio en GitHub, contiene los códigos y resultados de la investigación. 2025. (Visitado 28-10-2025).
- [48] Michael A. Nielsen e Isaac L. Chuang. «Asymptotic notation». En: *Quantum Computation and Quantum Information*. Capítulo 3: Introduction to Computer Science, Sección 3.1.2. Cambridge, UK: Cambridge University Press, 2010. Cap. 3.

## Capítulo 8

# ANEXOS

### Glosario de Términos

**Algoritmo de Grover** Algoritmo cuántico de búsqueda que reduce la complejidad de  $O(N)$  a  $O(\sqrt{N})$  para listas no estructuradas.

**Búsqueda Clásica** Algoritmo determinista que recorre cada elemento de una lista no estructurada, con complejidad temporal  $O(N)$ .

**Plataforma cuántica / Simulador** Su estructura puede ser ideal, ruidosa o basada en hardware cuántico. Existen múltiples simuladores; en este trabajo se utiliza Qiskit, desarrollado por IBM. Afecta la implementación, fidelidad y rendimiento.

**Arquitectura lógica y física del circuito cuántico** Número total de compuertas, traspilación del circuito y qubits requeridos para implementar Grover y resolver la búsqueda no estructurada.

**Tiempo de búsqueda** Tiempo requerido por el algoritmo para completarse en el simulador o en la computadora cuántica, determinado por la complejidad del problema.

**Éxito de búsqueda** Probabilidad de que el algoritmo ubique correctamente el elemento objetivo. El algoritmo de Grover es probabilístico y depende del número de iteraciones, nivel de ruido y construcción del circuito.

**Ventaja cuántica esperada** Aceleración teórica cuadrática del algoritmo sobre métodos clásicos bajo condiciones ideales. Es una métrica cuantitativa para comparar el rendimiento entre Grover y la búsqueda clásica. Se define como:

$$AR = \frac{T_{\text{clásico}}}{T_{\text{Grover}}}$$