Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

ALGORITMOS QUÂNTICOS E SUAS APLICAÇÕES: UMA REVISÃO SISTEMÁTICA

Autor: Nícolas Georgeos Mantzos
Orientador: Prof. Dr. Ronni Geraldo Gomes De Amorim

Brasília, DF
2023

Nícolas Georgeos Mantzos

ALGORITMOS QUÂNTICOS E SUAS APLICAÇÕES: UMA REVISÃO SISTEMÁTICA

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Ronni Geraldo Gomes De Amorim

Brasília, DF

2023

Nícolas Georgeos Mantzos
ALGORITMOS QUÂNTICOS E SUAS APLICAÇÕES: UMA REVISÃO SISTEMÁTICA/ Nícolas Georgeos Mantzos. – Brasília, DF, 2023-
95 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Ronni Geraldo Gomes De Amorim

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2023.

1. Computação Quântica. 2. Algoritmos Quânticos. I. Prof. Dr. Ronni Geraldo Gomes De Amorim. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. ALGORITMOS QUÂNTICOS E SUAS APLICAÇÕES: UMA REVISÃO SISTEMÁTICA

CDU 02:141:005.6

Errata

Nícolas Georgeos Mantzos

ALGORITMOS QUÂNTICOS E SUAS APLICAÇÕES: UMA REVISÃO SISTEMÁTICA

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 18 de dezembro de 2023:

Prof. Dr. Ronni Geraldo Gomes De Amorim
Orientador

Prof. Dr. Rendisley Aristóteles dos Santos Paiva
Convidado 1

Prof. Dr. Vinícius de Carvalho Rispoli
Convidado 2

Brasília, DF
2023

Este trabalho é dedicado aos tutores, amigos e amores do presente e ao Nícolas do futuro. Espero que tudo tenha corrido bem.

Agradecimentos

Ao professor Ronni Geraldo Gomes de Amorim, orientador e motivador deste trabalho, por ter se disposto a emprestar uma fração de seu tempo e ímpar inteligência ao seu desenvolvimento.

Aos professores da Universidade de Brasília, pela sua competência e comprometimento. Certamente são os gigantes a partir dos quais consigo ver mais longe.

Aos meus pais e amigos, especialmente a Eurípedes Nunes Braga, pelo apoio e torcida silenciosos.

"Seja fiel a ti mesmo e jamais serás falso com ninguém" (Hamlet, William Shakespeare)

Resumo

Este trabalho explora as aplicações promissoras e práticas da computação quântica, uma área de rápido crescimento que aproveita os princípios da física quântica. Primeiramente, são abordados os fundamentos teóricos necessários, incluindo os aspectos da matemática complexa e da álgebra linear. Em seguida, é explicado o método de Revisão Sistemática empregado para coletar a literatura relevante. A seção de Computação Quântica é desenvolvida contrastando com a computação clássica e descrevendo as bases para a implementação prática de programas quânticos. Finalmente, são apresentados os fundamentos de cinco principais algoritmos quânticos e seus recentes avanços e aplicações. Este estudo busca proporcionar uma visão geral sólida e acessível das possibilidades atuais e futuras da computação quântica.

Palavras-chave: computação quântica, álgebra linear, revisão sistemática, algoritmos quânticos, superposição, emaranhamento.

Abstract

This work explores the promising and practical applications of quantum computing, a rapidly growing field that leverages the principles of quantum physics. Initially, the necessary theoretical foundations are addressed, including aspects of complex mathematics and linear algebra. Next, the Systematic Review method employed to collect relevant literature is explained. The Quantum Computing section is developed by contrasting with classical computing and describing the basis for the practical implementation of quantum programs. Finally, the fundamentals of five major quantum algorithms and their recent advancements and applications are presented. This study seeks to provide a solid and accessible overview of the current and future possibilities of quantum computing.

Key-words: quantum computing, linear algebra, systematic review, quantum algorithms, superposition, entanglement.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

UFSCar	Universidade Federal de Santa Catarina
LaPES	Laboratório de Pesquisa em Engenharia de Software
StArt	State of the Art Through Systematic Review
CD	Ciência de Dados
IA	Inteligência Artificial
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CAFe	Comunidade Acadêmica Federada
WoS	Web of Science

Lista de símbolos

$\in$	Pertence
$\forall$	Para todo
∞	Infinito
$\mathbb{C}$	Cojunto dos números complexos
$\mathbb{R}$	Cojunto dos números reais
$\otimes$	Operador produto tensorial

Sumário

Introdução

Há muito, termos relacionados à física quântica como *superposição*, *emaranhamento* e *princípio da incerteza* deixaram de figurar somente em rebuscadas teses científicas, marcando presença na mídia, em ambientes corporativos de empresas como Google e IBM e até em produtos da cultura POP. Embora por vezes haja certa especulação e supertição nas tratativas menos rigorosas do tema, muitas das tecnologias basilares da computação moderna, tais como transistores e memórias flash, estão basedas em seus fundamentos teóricos. Outras aplicações, mesmo em estágios iniciais de maturação, têm se mostrado promissoras.

Uma delas, a Computação Quântica, já possui exemplares bem-sucedidos em áreas como inteligência artificial, ciência de dados, criptografia e até mesmo agronomia. No âmbito de CD e IA, por exemplo, a capacidade de computadores quânticos de lidar com estados de superposição e realizar cálculos em paralelo é aproveitada no assessoramento de modelos de aprendizado e na detecção de correlações complexas em conjuntos massivos de dados.

Este trabalho se destina à investigação do estado da arte das aplicações da computação quântica, bem como de suas principais perspectivas; subdividindo-se em quatro partes, cada qual com seu objetivo e contéudo. São elas:

1. **Fundamentos Teóricos**

   O objetivo desta seção é prover os fundamentos matemáticos para a compreensão dos principais algoritmos, servindo, nesse sentido, como uma ferramenta para a discussão de conceitos que serão constantemente referenciados ao longo do texto, mas não desenvolvidos, tais como o conjunto dos números complexos (definições, operações e plano) e as teorias elementares de álgebra linear (espaços vetoriais, base, dimensão, dentre outros tópicos), álgebra linear complexa e mecânica quântica (contexto, experimentos fundamentais e postulados).

   Como o foco está em instrumentalizar tais tópicos para uso posterior, demonstrações e considerações mais sofisticadas serão somente provocadas, ou mesmo deixadas de lado, em prol da apresentação das definições elementares acompanhadas de suas implicações mais imediatas.

   Boa parte do seu conteúdo está baseado em (IEZZI, 2013), (HOWARD, 2013) e (BARAVIERA; AMARAL; CUNHA, 2011).

2. **Materiais e Métodos**

Partindo de (KITCHENHAM, 2015) e (BIOLCHINI et al., 2005), sobretudo, será descrita a Revisão Sistemática - metodologia de pesquisa utilizada para a coleta da teoria sobre o tema - apresentando suas vantagens, desvantages e os principais problemas aos quais ela vem ao encalço. Uma vez justificada, sua utilização na pesquisa será demonstrada através da apresentação de protocolos, artigos coletados, dentre outros artefatos.

3. **Computação Quântica**

   A partir da base estabelecida em 1, a teoria elementar da computação quântica será desenvolvida, (a fim de solidificar o entendimento dos conceitos), de forma comparativa, isto é, contrastando a abordagem da computação dita *clássica*, de *bits* e portas lógicas, com o novo paradigma de *qubits* e portas quânticas.

   Por fim, uma breve discussão a respeito dos requisitos teóricos para a implementação prática de programas quânticos será realizada com foco nas linguagens que poderiam descrevê-los com precisão e nas atuais tecnologias que efetivamente os simulam, seja para fins científicos ou educacionais.

   Aqui, (YANOFSKY; MANNUCCI, 2008) e (IMRE; BALáZS, 2005) são as referências fundamentais.

4. **Algoritmos e Aplicações**

   O foco desta seção é utilizar a teoria de 3 para apresentar os fundamentos dos quatro principais algoritmos de computação quântica - Deutsch, Deutsch-Jozsa, Busca de Grover e Fatoração de Shor - e o material coletado em 2 para pontuar seus mais recentes aprimoramentos e aplicações, assim como destacar suas implementações através do SDK de código aberto Qiskit [1], disponibilizado pela IBM para a efetiva escrita e visualização de circuitos quânticos, manipulação de qubits, aplicação de portas, dentre outras funcionalidades.

   Outrossim, exemplos mais familiares ao contexto "clássico" serão reimplementados, discutindo o porquê de sua escolha e as consequências práticas e téoricas dessa reescrita.

[1] Mais informações na documentação oficial

Parte I

Fundamentos Teóricos

1 Fundamentos Matemáticos

Este capítulo explora os fundamentos matemáticos essenciais para a compreensão da mecânica quântica e sistemas dinâmicos. Inicia-se com uma abordagem detalhada sobre os Números Complexos, fundamentais para muitos conceitos em física e engenharia. Segue-se com uma exploração da Geometria dos Números Complexos, examinando suas propriedades e aplicações. A seção de Álgebra Linear introduz conceitos vitais como espaços e subespaços vetoriais, produto interno e tensorial, essenciais para o entendimento dos Espaços Vetoriais Complexos. Por fim, aborda-se Sistemas Dinâmicos e Cadeias de Markov, oferecendo uma visão poderosa sobre a evolução dos sistemas ao longo do tempo.

1.1 Números Complexos

1.1.1 Contexto

A resolução de equações algébricas é um problema que ocupa matemáticos há milênios. Apesar disso, até o século XVI somente equações de primeiro e segundo graus, isto é, expressões na forma $ax + b = 0$ e $ax^2 + bx + c = 0$ com $a \neq 0$, respectivamente, tinham estratégias de resolução bem conhecidas e documentadas.

Niccoló Fontana Tartaglia, matemático do período, demonstrou que expressões na forma $x^3 + px = q$, por sua vez, tinham solução dada por

$$x = \sqrt[3]{\sqrt{\left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2} + \frac{q}{2}} - \sqrt[3]{\sqrt{\left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2} - \frac{q}{2}}. \tag{1.1}$$

Ao aplicar (1.1) à igualdade $x^3 - 15x = 4$, Bombelli, companheiro de Tartaglia, notou que a solução resultante $(2 + \sqrt{-1}) + (2 - \sqrt{-1})$ possuía um termo desconhecido, $\sqrt{-1}$, que caso fosse operado como um número real, levaria ao cancelamento da soma $\sqrt{-1} + (-\sqrt{-1})$ e a uma das raízez corretas: 4.

Inicialmente denominado *unidade imaginária* ou *unidade fictítia*, o termo $\sqrt{-1}$ se tornou a base de um conjunto numérico cuja natureza algébrica e geométrica seria melhor descrita por Frederich Gauss e seus contemporâneos, três séculos depois.

Atualmente, o dito *conjunto dos complexos* estende o conjunto dos reais de forma a incluir a incluir soluções para equações que não têm solução dentro dele, sendo matematicamente definido como:

$$\mathbb{C} = \{a + bi : a, b \in \mathbb{R} \text{ e } i^2 = -1\},$$

onde cada elemento consiste em um par ordenado de números reais, o primeiro dos quais é a parte real e o segundo a parte imaginária.

1.1.2 Conjugado

Dado um número complexto $z_1 = a + bi$ seu *conjugado* é denotado por $\overline{z_1}$ ou z_1^* e definido como

$$\overline{z_1} = z_1^* = a - bi,$$

isto é, consiste na alteração do sinal da parte real.

1.1.3 Operações

Dados dois números complexos $z_1 = a + bi$ e $z_2 = c + di$, as quatro operações abaixo são definidas:

1.1.3.1 Adição e subtração

A adição e subtração são realizadas componente a componente

$$z_1 \pm z_2 = (a + bi) \pm (c + di) = (a \pm c) + (bi \pm di)(a \pm c) + (b \pm d)i.$$

1.1.3.2 Multiplicação

A multiplicação de números complexos utiliza a distributividade e o fato de que $i^2 = -1$, sendo dada por:

$$z_1 \cdot z_2 = (a + bi) \cdot (c + di) = ac + (ad)i + (bc)i + (bd)i^2 = (ac - bd) + (ad + bc)i$$

1.1.3.3 Divisão

A divisão é definida como:

$$\frac{z_1}{z_2} = \frac{a + bi}{c + di}. \tag{1.2}$$

Para simplificar essa expressão, multiplicamos o numerador e o denominador pelo conjugado do denominador. O conjugado de $c + di$ é $c - di$. Então, temos:

$$\frac{z_1}{z_2} = \frac{a + bi}{c + di} \cdot \frac{c - di}{c - di} = \frac{(a + bi)(c - di)}{(c + di)(c - di)}. \tag{1.3}$$

Aplicando a distributividade, temos:

$$\frac{z_1}{z_2} = \frac{ac - adi + bci - bdi^2}{c^2 - cdi + cdi - d^2i^2}. \tag{1.4}$$

Lembrando que $i^2 = -1$, podemos simplificar a equação para:

$$\frac{z_1}{z_2} = \frac{ac - adi + bci + bd}{c^2 + d^2} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i. \tag{1.5}$$

1.1.4 Geometria

Na análise dos números complexos, é útil introduzir o conceito do *plano complexo*, também conhecido como *plano Argand-Gauss*.

No *plano complexo*, um número complexo $z = a+bi$ é representado como um ponto (a, b) ou como um vetor que origina do ponto $(0, 0)$ até o ponto (a, b). O eixo horizontal é conhecido como o eixo real e o eixo vertical é o eixo imaginário.

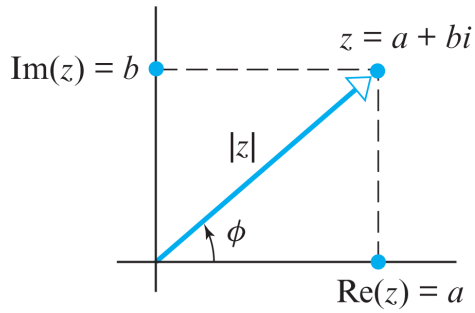

Figura 1 – Plano Complexo. Fonte: (HOWARD, 2013, p. 314)

O *módulo* de um número complexo $z = a + bi$, denotado por $|z|$, é a distância do ponto que representa o número complexo à origem do plano complexo e é matematicamente definido como $|z| = \sqrt{a^2 + b^2}$. Um número complexo, z, pode ser expresso na forma trigonométrica como:

$$z = r(\cos\phi + i\sin\phi)$$

. Onde r é o módulo e ϕ é o ângulo ou argumento do número complexo. Alternativamente, um número complexo pode ser expresso na forma exponencial:

$$z = re^{i\phi}$$

. Essa é uma consequência direta da fórmula de Euler: $e^{i\phi} = \cos\phi + i\sin\phi$.

1.2 Álgebra Linear

1.2.1 Espaços Vetoriais

Um espaço vetorial é um conjunto não vazio V, junto com duas operações – adição de vetores e multiplicação por escalares – que satisfazem os seguintes axiomas para todos $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w} \in V$ e $a, b \in \mathbb{F}$:

1. $\mathbf{u} + \mathbf{v} \in V$
2. $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
3. $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
4. $\exists \mathbf{0} \in V$ tal que $\mathbf{u} + \mathbf{0} = \mathbf{u}$ para todo $\mathbf{u} \in V$
5. $\forall \mathbf{u} \in V$, existe $-\mathbf{u} \in V$ tal que $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$
6. $a\mathbf{u} \in V$
7. $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
8. $(a + b)\mathbf{u} = a\mathbf{u} + b\mathbf{u}$
9. $a(b\mathbf{u}) = (ab)\mathbf{u}$
10. $1\mathbf{u} = \mathbf{u}$

Onde $\mathbb{F}$ é o conjunto dos números reais, $\mathbb{R}$, ou complexos, $\mathbb{C}$.

1.2.1.1 Base

Se V for um espaço vetorial qualquer e $S = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ for um conjunto finito de vetores em V, dizemos que S é uma base de V se valerem as duas condições a seguir:

(a) S é linearmente independente, isto é, nenhum vetor em S pode ser expresso como uma combinação linear dos outros vetores em S. Formalmente, para todos os escalares $a_1, a_2, ..., a_n$, se $a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + ... + a_n\mathbf{v}_n = \mathbf{0}$, então todos a_i devem ser zero.

(b) S gera V, isto é, todo vetor em V pode ser expresso como uma combinação linear dos vetores em S. Formalmente, para todo $\mathbf{u} \in V$, existem escalares $a_1, a_2, ..., a_n$ tais que $\mathbf{u} = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + ... + a_n\mathbf{v}_n$.

1.2.1.2 Dimensão

A dimensão de um espaço vetorial de dimensão finita V é denotada por $\dim(V)$ e é definida como o número de vetores numa base de V.

1.2.2 Subespaços Vetoriais

$S \subseteq V$ é um subespaço vetorial se satisfizer os seguintes axiomas:

1. $\mathbf{0} \in S$
2. $\forall \mathbf{u}, \mathbf{v} \in S, \mathbf{u} + \mathbf{v} \in S$
3. $\forall \mathbf{u} \in S, \forall c \in \mathbb{F}, c\mathbf{u} \in S$

1.2.3 Transformações Lineares

Seja $T : V \to W$ uma função de um espaço vetorial V para um espaço vetorial W. T é denominada transformação linear de V em W se as seguintes propriedades são satisfeitas para todos os vetores $\mathbf{u}, \mathbf{v} \in V$ e qualquer escalar k:

1. $T(k\mathbf{v}) = kT(\mathbf{v})$
2. $T(\mathbf{u} + \mathbf{v}) = T(\mathbf{u}) + T(\mathbf{v})$

1.2.4 Produto Interno

Dado um espaço vetorial V sobre um campo $\mathbb{F}$, um produto interno em V é uma função $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$ que associa a cada par de vetores $(\mathbf{v}, \mathbf{w})$ um escalar $\langle \mathbf{v}, \mathbf{w} \rangle$, e satisfaz as seguintes propriedades para todos os vetores $\mathbf{v}, \mathbf{w}, \mathbf{u} \in V$ e todos os escalares $c \in \mathbb{F}$:

1. $\langle c\mathbf{v} + \mathbf{w}, \mathbf{u} \rangle = c\langle \mathbf{v}, \mathbf{u} \rangle + \langle \mathbf{w}, \mathbf{u} \rangle$.
2. $\langle \mathbf{v}, \mathbf{w} \rangle = \overline{\langle \mathbf{w}, \mathbf{v} \rangle}$ se $\mathbb{F} = \mathbb{C}$, ou $\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{w}, \mathbf{v} \rangle$ se $\mathbb{F} = \mathbb{R}$.
3. $\langle \mathbf{v}, \mathbf{v} \rangle \geq 0$ e $\langle \mathbf{v}, \mathbf{v} \rangle = 0$ se, e somente se, $\mathbf{v} = \mathbf{0}$.

Observe como as propriedades derivam da definição de produto interno já apresentada.

1.2.5 Ortogonalidade e ortonormalidade

Dado um espaço vetorial V com produto interno e dois vetores $\mathbf{v}, \mathbf{w} \in V$, dizemos que $\mathbf{v}$ e $\mathbf{w}$ são ortogonais se o produto interno deles é zero, isto é, se $\langle \mathbf{v}, \mathbf{w} \rangle = 0$.

Um conjunto de vetores $\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n\}$ em V é chamado de conjunto **ortonormal** se para cada par de vetores distintos $\mathbf{v}_i$ e $\mathbf{v}_j$ do conjunto, eles são ortogonais, e o produto interno de cada vetor consigo mesmo é 1, ou seja,

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle = \delta_{ij}, \text{ com}$$

$$\delta_{ij} = \begin{cases} 0 & \text{se } i \neq j, \\ 1 & \text{se } i = j. \end{cases}.$$

δ_{ij} é o chamado *delta de Kronecker*.

1.2.6 Autovalores e autovetores

Seja A uma matriz quadrada e $\mathbf{v}$ um vetor não nulo. Dizemos que λ é um autovalor de A e que $\mathbf{v}$ é um autovetor correspondente a λ se a seguinte relação for satisfeita:

$$A\mathbf{v} = \lambda\mathbf{v}$$

Em outras palavras, a multiplicação de A pelo vetor $\mathbf{v}$ resulta em um vetor que é uma múltiplo escalar λ de $\mathbf{v}$.

1.3 Espaços vetoriais complexos

1.3.1 Definições básicas

O conjunto de todas as n-uplas $(v_1, v_2, ..., v_n)$, onde $v_i \in \mathbb{C}$, é chamado de *espaço n-dimensional complexo* ou, simplesmente, *espaço vetorial complexo*, e é denotado por $\mathbb{C}^n$.

Um vetor $\mathbf{v} \in \mathbb{C}^n$ pode ser escrito como

$$\begin{aligned} \mathbf{v} &= (v_1, v_2, \ldots, v_n) \\ &= (a_1 + b_1 i, a_2 + b_2 i, \ldots, a_n + b_n i) \\ &= (a_1, a_2, \ldots, a_n) + i(b_1, b_2, \ldots, b_n) \\ &= \mathrm{Re}(\mathbf{v}) + i\mathrm{Im}(\mathbf{v}). \end{aligned}$$

Logo, $\mathrm{Re}(\mathbf{v}) = (a_1, a_2, \ldots, a_n)$ e $\mathrm{Im}(\mathbf{v}) = (b_1, b_2, \ldots, b_n)$.

1.3.2 Conjugado

O vetor

$$\begin{aligned}\overline{\mathbf{v}} &= (\overline{v_1}, \overline{v_2}, \ldots, \overline{v_n}) \\ &= (a_1 - b_1 i, a_2 - b_2 i, \ldots, a_n - b_n i) \\ &= (a_1, a_2, \ldots, a_n) - i(b_1, b_2, \ldots, b_n) \\ &= \mathrm{Re}(\mathbf{v}) - i\mathrm{Im}(\mathbf{v})\end{aligned}$$

é o chamado ***conjugado complexo*** de $\mathbf{v}$ e obedece às seguintes propriedades, considerando $\mathbf{u} \in \mathbb{C}^n$ e $k \in \mathbb{R}$

1. $\overline{\overline{\mathbf{v}}} = \mathbf{v}$
2. $\overline{k\mathbf{v}} = \overline{k}\overline{\mathbf{v}} = k^*\overline{\mathbf{v}}$
3. $\overline{\mathbf{v} \pm \mathbf{u}} = \overline{\mathbf{u} \pm \mathbf{v}}$

1.3.3 Produto Interno

Como será explorado futuramente, o estado dos sistemas quânticos é descrito também por um vetor complexo e, para avaliá-los comparativamente, o número real resultante da igualdade 1.6, considerando $\mathbf{u}, \mathbf{v} \in \mathbb{C}^n$, é utilizado.

$$\mathbf{u} \cdot \overline{\mathbf{v}} = \langle \mathbf{u}, \overline{\mathbf{v}} \rangle = u_1\overline{v_1} + u_2\overline{v_2} + \ldots + u_n\overline{v_n}. \tag{1.6}$$

1.6 é o chamado *produto interno Euclidiano* de $\mathbb{C}^n$ e obedece às seguintes propriedades, com $k \in \mathbb{R}$

1. $\mathbf{u} \cdot \mathbf{v} = \overline{\mathbf{u} \cdot \mathbf{v}}$
2. $\mathbf{u} \cdot (\mathbf{v} + \mathbf{w}) = \mathbf{u} \cdot \mathbf{v} + \mathbf{u} \cdot \mathbf{w}$
3. $k(\mathbf{u} \cdot \mathbf{v}) = (k\mathbf{u}) \cdot \mathbf{v}$
4. $\mathbf{v} \cdot \mathbf{v} \geq 0 \Leftrightarrow \mathbf{v} = \mathbf{0}$

1.3.4 Norma

A norma de um vetor $\mathbf{v} \in \mathbb{C}^n$, denotada por $\|\mathbf{v}\|$, é definida a partir de 1.6 como

$$\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle} = \sqrt{|v_1|^2 + |v_2|^2 + |v_3|^2 + \ldots + |v_n|^2} \tag{1.7}$$

e obedece às seguintes propriedades

1. $\|\mathbf{0}\| = 0$
2. $\mathbf{u} \neq \mathbf{0} \Rightarrow \|\mathbf{u}\| > 0$
3. $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$
4. $\|k \cdot \mathbf{u}\| \leq \|k\| \times \|\mathbf{u}\|$

1.3.5 Distância

A distância entre dois vetores $\mathbf{v}, \mathbf{u} \in \mathbb{C}^n$, denotada por $d(\mathbf{v_1}, \mathbf{v_2})$, é definida a partir de 1.6 como

$$d(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\| = \sqrt{\langle \mathbf{u} - \mathbf{v}, \mathbf{u} - \mathbf{v} \rangle} \tag{1.8}$$

e obedece às seguintes propriedades

1. $\mathbf{u} \neq \mathbf{v} \Rightarrow d(\mathbf{u}, \mathbf{v}) > 0$
2. $d(\mathbf{u}, \mathbf{v}) \leq d(\mathbf{u}, \mathbf{w}) + d(\mathbf{w}, \mathbf{v})$
3. $d(\mathbf{u}, \mathbf{v}) = d(\mathbf{v}, \mathbf{w})$

1.4 Produto Tensorial

Dados dois espaços vetoriais, V_A e V_B, sobre $\mathbb{C}$ com dimensões n_A e n_B, respectivamente, podemos construir um novo espaço vetorial de dimensão $n_A n_B$ através do operador produto tensorial.

Para construir esse novo espaço, que denotaremos por $V_A \otimes V_B$ (lê-se "V_A tensorial V_B"), tomamos bases $\mathbf{u}$ para V_A e $\mathbf{v}$ para V_B. Declaramos então que os $n_A n_B$ elementos da forma

$$\mathbf{u} \otimes \mathbf{v}, u = 0, 1, \ldots, n_A, v = 0, 1, \ldots, n_B \tag{1.9}$$

formam uma base para $V_A \otimes V_B$. As seguintes condições são impostas:

1. Para um escalar arbitrário $a \in \mathbb{C}$ e elementos $\mathbf{u}$ de V_A e $\mathbf{v}$ de V_B,

$$a(\mathbf{u} \otimes \mathbf{v}) = (a\mathbf{u}) \otimes \mathbf{v} = \mathbf{u} \otimes (a\mathbf{v}).$$

2. Para $\mathbf{u}$ e $\mathbf{w}$ arbitrários em V_A e $\mathbf{v}$ em V_B,

$$(\mathbf{u} + \mathbf{w}) \otimes \mathbf{v} = \mathbf{u} \otimes \mathbf{v} + \mathbf{w} \otimes \mathbf{v}.$$

3. Para $\mathbf{u}$ arbitrário em V_A e $\mathbf{w}$ e $\mathbf{v}$ em V_B,

$$\mathbf{u} \otimes (\mathbf{w} + \mathbf{v}) = \mathbf{u} \otimes \mathbf{w} + \mathbf{u} \otimes \mathbf{v}.$$

1.5 Espaços de Hilbert

1.5.1 Sequência de Cauchy

Dado um espaço vetorial com produto interno definido (e, por consequência distância e norma), uma sequência de de vetores $\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_n}, \ldots$ é chamada de "sequência de Cauchy" se, para cada $\epsilon > 0$ existe um $n_0 \in \mathbb{N}$ tal que

$$\forall m, n \geq n_0, d(\mathbf{u_m}, \mathbf{u_n}) \leq \epsilon. \tag{1.10}$$

1.5.2 Completude

Um espaço vetorial $\mathbb{V}$ com produto interno definido é chamado de **completo** se, para qualquer sequência de Cauchy $\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_n}, \ldots$ em seu interior, existe um vetor $\overline{\mathbf{v}} \in \mathbb{V}$, tal que

$$\lim_{n \to \infty} (\mathbf{v_n} - \overline{\mathbf{v}}) = 0. \tag{1.11}$$

Espaços de Hilbert são espaços vetoriais complexos completos e, como todo espaço de dimensão finita com produto interno definido é completo, sempre se tratará de um Espaço de Hilbert.

1.6 Sistemas dinâmicos e cadeias de markov

1.6.1 Matrizes de Estado

Um sistema dinâmico é uma regra ou conjunto de regras que descreve como um sistema evolui ao longo do tempo. Matematicamente, as regras podem ser descritas por meio de variáveis agregadas em um *vetor de estados* e o sistema, naturalmente, pela união dessas regras em uma *matriz de estados* (HOWARD, 2013, p. 332).

Recorrentemente, o estado dessas variáveis não é conhecido com precisão, mas pode ser expresso por meio de probabilidades. Em um sistema com n estados possíveis, por exemplo, o vetor de estados em um dado momento t seria dado por

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix}. \tag{1.12}$$

Tais sistemas dinâmicos são conhecidos como *processos estocásticos* e obdecem aos seguintes axiomas de probabilidade na composição de seus vetores:

1. $x_i \geq 0,\ \forall i \in [1, n]$ pois, para qualquer evento A, $P(A) \geq 0$.

2. $\sum_{i=1}^{n} x_i(t) = 1$, pois $P(S) = 1$, onde S é o espaço amostral.

A matriz de estados, isto é, a representação do fluxo entre os vetores seria dada por

$$\begin{bmatrix} x_1(0) & x_1(1) & \cdots \\ x_2(0) & x_2(1) & \cdots \\ \vdots & \vdots & \ddots \\ x_n(0) & x_n(1) & \cdots \end{bmatrix}.$$

1.6.2 Cadeias de Markov

Processos estocásticos nos quais os vetores de estado estão relacionados por uma equação na forma

$$\mathbf{x}(k+1) = \mathbf{x}_{k+1} = P\mathbf{x}(k) = P\mathbf{x}_k, \tag{1.13}$$

onde $P = [p_{ij}]$ é uma matriz de estados e p_{ij} é a probabilidade de que o sistema estará no estado i quando $t = k + 1$ e no estado j quando $t = k$, são chamados de **Cadeias de Markov**.

Portanto, cada vetor em um dado momento $t + 1$ deriva da multiplicação do vetor em t pela matriz de estados.

Para um instante $t = k$, considerando um vetor inicial $\mathbf{x}(0)$, $\mathbf{x}_k$ pode-se derivar a partir de 1.13 que

$$\mathbf{x}_k = P^k\mathbf{x}(0). \tag{1.14}$$

1.6.3 Convergência

Seja P a matriz de transição de uma cadeia de Markov regular. Então:

1. Existe um vetor de probabilidade único q com entradas positivas tal que $Pq = q$.

2. Para qualquer vetor de probabilidade inicial x_0, a sequência de vetores de estado x_0, Px_0, ..., $P^k x_0$, ... converge para q.

3. A sequência P, P^2, P^3, ..., P^k, ... converge para a matriz Q, cujos vetores coluna são todos q.

1.6.4 Exemplo

Suponha que temos um sistema quântico com um conjunto discreto de níveis de energia, denominados nível 1, nível 2 e nível 3. Baseado nos dados hipotéticos sobre a distribuição de energia do sistema, a evolução do estado quântico pode ser modelada por uma cadeia de Markov com a seguinte matriz de transição:

$$P = \begin{bmatrix} 0.5 & 0.4 & 0.6 \\ 0.2 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.1 \end{bmatrix},$$

onde o elemento da matriz p_{ij} representa a probabilidade do estado quântico fazer a transição do nível j para o nível i para um intervalo de tempo t. Ou seja,

- $p_{11} = 0.5$ = probabilidade de que o estado quântico permaneça no nível 1 quando está no nível 1
- $p_{12} = 0.4$ = probabilidade de que o estado quântico faça a transição do nível 2 para o nível 1
- $p_{13} = 0.6$ = probabilidade de que o estado quântico faça a transição do nível 3 para o nível 1
- $p_{21} = 0.2$ = probabilidade de que o estado quântico faça a transição do nível 1 para o nível 2
- $p_{22} = 0.2$ = probabilidade de que o estado quântico permaneça no nível 2 quando está no nível 2
- $p_{23} = 0.3$ = probabilidade de que o estado quântico faça a transição do nível 3 para o nível 2
- $p_{31} = 0.3$ = probabilidade de que o estado quântico faça a transição do nível 1 para o nível 3
- $p_{32} = 0.4$ = probabilidade de que o estado quântico faça a transição do nível 2 para o nível 3
- $p_{33} = 0.1$ = probabilidade de que o estado quântico permaneça no nível 3 quando está no nível 3

Assumindo que o sistema inicialmente está no nível 2, isto é,

$$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},$$

podemos deduzir o fluxo dos níveis de energia ao longo do tempo através dos vetores de estado $\mathbf{x}(t)$. Para $1 \leq t \leq 6$, por exemplo, temos:

$$\mathbf{x}(1) = P\mathbf{x}(0) = \begin{bmatrix} 0.400 \\ 0.200 \\ 0.400 \end{bmatrix}, \quad \mathbf{x}(2) = P\mathbf{x}(1) = \begin{bmatrix} 0.520 \\ 0.240 \\ 0.240 \end{bmatrix}, \quad \mathbf{x}(3) = P\mathbf{x}(2) = \begin{bmatrix} 0.500 \\ 0.224 \\ 0.276 \end{bmatrix},$$

$$\mathbf{x}(4) = P\mathbf{x}(3) = \begin{bmatrix} 0.505 \\ 0.228 \\ 0.267 \end{bmatrix}, \quad \mathbf{x}(5) = P\mathbf{x}(4) \approx \begin{bmatrix} 0.504 \\ 0.227 \\ 0.269 \end{bmatrix}, \quad \mathbf{x}(6) = P\mathbf{x}(5) \approx \begin{bmatrix} 0.504 \\ 0.227 \\ 0.269 \end{bmatrix}.$$

Observe que $\mathbf{x}(5) = \mathbf{x}(6)$. Para $t \geq 7$, a convergência descrita pelo teorema da seção 1.6.3 permanece. Isto é, as probabilidades de o sistema estar nos níveis de energia 1, 2 ou 3, respectivamente, estabilizam em 0.504, 0.227 e 0.269.

2 Teoria quântica elementar

Este capítulo explora a teoria quântica elementar, uma área fundamental da física que revolucionou nossa compreensão do universo em escala microscópica. A jornada começa com um olhar sobre o contexto histórico que moldou a mecânica quântica, destacando como os avanços científicos do início do século XX desafiaram e eventualmente reformularam as noções clássicas de luz, tempo, espaço e gravidade. As seções seguintes abordam fenômenos específicos e conceitos-chave que são a pedra angular da teoria quântica. O efeito fotoelétrico, que ilustra a natureza quântica da luz, é examinado em detalhes, seguido por uma discussão sobre o conceito intrigante de spin, um aspecto quântico fundamental dos elétrons.

2.1 Contexto histórico

No alvorecer do século XX a mecânica clássica era o *status quo* da física. Nela, a luz é concebida como um conjunto contínuo de "pacotes de energia" que se propagam no espaço. Não obstante, desde Newton, o tempo e espaço eram pressupostos como grandezas absolutas e, portanto, indiferentes ao referencial adotato. A gravidade, por sua vez, era vista como uma força atrativa fundamentada na massa dos corpos e em suas distâncias. Diversos experimentos no decorrer nos anos 1900, no entanto, colocaram em xeque esssas e outras percepções estabelecendo novos paradigmas.

O efeito fotoelétrico (observado por Hertz in 1887), por exemplo, demonstrou que um átomo bombardeado por feixes de luz absorvia sua energia, causando a transição de seus elétrons para uma órbita mais excitada. Mais tarde, essa energia excedente era liberada na forma de luz, causando o retorno para uma órbita menor. A grande revelação de tal experimento foi que transições desse tipo ocorriam através de pacotes discretos de energia - os chamados "fótons".

Essa e outras evidências experimentais posteriores levaram à substituição da dualidade *onda-partícula* pela teoria que concebia a luz como detentora de uma natureza dual, podendo hora se comportar como onda, ora como partícula.

Não obstante, os artigos de Albert Einstein de 1905 - *o ano do milagre* - e as Teorias da Relatividade Geral e Restrita advindas deles conceberam a presença de massa e energia como curvadora do espaço-tempo, sendo a gravidade a manifestação desse fenômeno, além de demonstrarem a dependência do tempo do referencial adotado.

2.2 Efeito Fotoelétrico

O efeito fotoelétrico é um fenômeno observado quando a luz incide em uma superfície metálica, resultando na emissão de elétrons pela matéria. O fenômeno foi estudado em detalhes por Albert Einstein e é explicado pela natureza quântica da luz. De acordo com a teoria do efeito fotoelétrico, a luz é composta por pacotes de energia chamados fótons. Quando um fóton incide em um elétron na superfície metálica, pode transferir sua energia para o elétron, permitindo que ele escape do material. A energia dos fótons deve ser igual ou superior a um valor mínimo chamado de "energia de ionização" para que ocorra a emissão dos elétrons.

O efeito fotoelétrico teve um impacto significativo na percepção da física, pois desafiou conceitos clássicos estabelecidos. Em particular, contradisse a visão tradicional da luz como uma onda contínua, sustentada pela teoria eletromagnética clássica. Em vez disso, evidenciou que a luz possui uma natureza dual, exibindo tanto características de partículas quanto de ondas. Essa descoberta foi fundamental para o desenvolvimento da teoria quântica e para o estabelecimento do princípio fundamental da dualidade onda-partícula.

Além disso, o efeito fotoelétrico demonstrou a existência de uma relação direta entre a energia da luz (fótons) incidente e a energia cinética dos elétrons emitidos. Isso levou à formulação da equação de Einstein, que estabelece que a energia dos elétrons emitidos é proporcional à frequência da luz incidente e não à sua intensidade. Essa relação entre energia e frequência, em vez de intensidade, contrariava a compreensão clássica da interação da luz com a matéria.

2.3 Spin

O conceito de spin foi introduzido pela primeira vez como uma propriedade teórica para explicar observações experimentais relacionadas ao comportamento de elétrons em campos magnéticos. O experimento chave que originou o conceito de spin é conhecido como experimento de Stern-Gerlach, realizado por Otto Stern e Walther Gerlach em 1922.

No experimento de Stern-Gerlach (imagem 2), um feixe de átomos de prata é enviado através de um campo magnético não uniforme. Esperava-se que o feixe se dividisse em uma distribuição contínua, de acordo com as pectativas clássicas baseadas no momento magnético orbital dos elétrons. No entanto, para surpresa dos pesquisadores, o feixe foi dividido em apenas duas direções distintas.

Esse resultado experimental desafiou a compreensão clássica da natureza dos elétrons e exigiu uma explicação teórica adicional. Foi nesse contexto que o conceito de

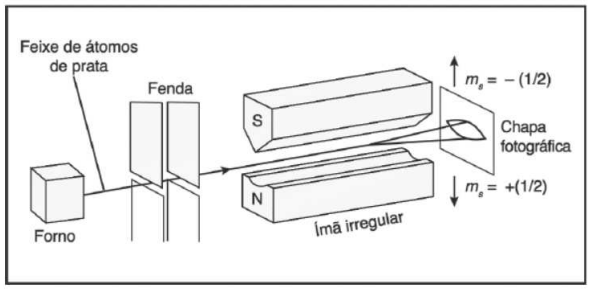

Figura 2 – Experimento de Stern-Gerlach. Fonte: (VITOR, 2016, p. 903)

spin foi proposto para descrever a propriedade intrínseca dos elétrons que determina sua interação com campos magnéticos.

É uma das propriedades mais estranhas e distintamente quânticas da matéria, visto que não possui um anáçogo clássico. Embora ele possa ser medido em direções específicas, tem um módulo e uma direção e possa ser somado em sistemas de várias partículas o spin de uma partícula é sempre a mesma quantidade (dependendo do tipo de partícula), não depende de qualquer movimento ou rotação externa, e seus valores possíveis são restritos pelos princípios da mecânica quântica.

As partículas podem ser divididas em duas categorias com base em seu spin: bósons, que têm spin inteiro (0, 1, 2, etc.), e férmions, que têm spin meio-inteiro (1/2, 3/2, etc.). Essa distinção é fundamental e tem implicações profundas. O Princípio da Exclusão de Pauli, que afirma que dois férmions não podem ocupar o mesmo estado quântico ao mesmo tempo, se origina da natureza do spin e é crucial para a estrutura e estabilidade da matéria.

Um exemplo de aplicação desse conceito é a denominada “spintrônica”, ou eletrônica de spin, uma tecnologia emergente que explora a propriedade fundamental do spin para o desenvolvimento de dispositivos eletrônicos. Como pontuado por (AWSCHALOM, 2002, p. 41), enquanto a eletrônica tradicional depende do transporte de carga elétrica em semicondutores, como o silício, para manipular informações, a spintrônica se beneficia do spin para criar dispositivos com maior eficiência energética, velocidade e densidade de dados. Esta abordagem inovadora tem o potencial de revolucionar a maneira como informações são armazenadas e processadas, abrindo caminho para uma nova geração de computadores e dispositivos de armazenamento.

Um exemplo notável de aplicação da spintrônica é a memória de acesso aleatório magnetorresistiva (MRAM), que utiliza o alinhamento magnético dos elétrons para armazenar dados. A MRAM promete ser mais rápida e eficiente do que as tecnologias de memória tradicionais, como DRAM e *flash*, pois não requer energia para manter as informações armazenadas e é capaz de reter dados após a remoção da energia. Além disso, a pesquisa em spintrônica está impulsionando o avanço na computação quântica, onde

o controle preciso do spin de partículas individuais pode levar à realização de qubits, as unidades básicas de informação quântica, para computadores quânticos poderosos e altamente eficientes.

2.4 Estados e Medições

Todo sistema quântico é associado a um espaço de estados E, que é um espaço vetorial complexo equipado com um produto interno. Dese modo, qualquer previsão sobre o sistema pode ser derivada a partir do conhecimento desse estado. Para fins de simplificação, é conveniente considerar que $\dim(E) = 2$ e representar estado do sistema por um vetor unitário em E.

2.5 Teste

Dada uma base ortonormal, um *teste* refere-se ao ato de decompor o vetor estado em relação a esta base. No contexto da mecânica quântica, isso é essencialmente o ato de medir o estado do sistema em relação aos vetores base. A ideia subjacente é que cada vetor base na base ortonormal representa uma possível saída ou resultado da medição.

O resultado de um teste quântico é probabilístico. A probabilidade de cada possível resultado é determinada pelo módulo ao quadrado do coeficiente do vetor de estado na decomposição em relação à base ortonormal. Uma vez realizada a medição, o sistema quântico "colapsa" no vetor base correspondente ao resultado da medição.

Deste modo, um *teste* em mecânica quântica é um procedimento para selecionar uma das alternativas possíveis que são representadas pelos vetores base na base ortonormal. Esta seleção não é determinística, mas sim probabilística, com as probabilidades determinadas pela estrutura o vetor estado e pela base ortonormal escolhida para a medição.

2.6 Postulados

2.6.1 Primeiro potulado

O primeiro postulado afirma que o estado de qualquer sistema físico fechado pode ser descrito através de um *vetor de estado* (vide 1.12) $\mathbf{v}$ que possua coeficientes complexos, norma (vide 1.7) unitária e pertença a um espaço de Hilbert, isto é um espaço complexo dotado de produto interno.

2.6.1.1 Notação de Dirac

A notação de Dirac é amplamente utilizada para representar o vetor de estado. Nessa notação, o vetor de estado $\mathbf{v}$ é representado por $|\mathbf{v}\rangle$, onde o símbolo $|\rangle$ é conhecido como *ket*. Além disso, o produto interno entre dois vetores de estado $\mathbf{u}$ e $\mathbf{v}$ é denotado por $\langle \mathbf{u}|\mathbf{v}\rangle$, onde o símbolo $\langle$ é conhecido como bra. A combinação do bra $\langle \mathbf{u}|$ e do ket $|\mathbf{v}\rangle$ representa o produto interno entre os dois vetores de estado.

2.6.2 Segundo postulado

O segundo postulado estabelece que quantidades observáveis como energia, *momentum*, posição etc, deve ser representados por operadores hermitianos, que, dada essa observabilidade, são transformaççoes lineares $\hat{\mathbf{O}} : H \to H$ que levam um vetor do espaço de Hilbert a outro vetor nesse mesmo espaço. Vale ressaltar, entretanto, que, na mecânica quântica, nem todos os operadores hermitianos são necessariamente transformações lineares.

Uma das principais características é que os operadores hermitianos garantem que as observáveis associadas a eles possuam valores reais. Isso é crucial para garantir que as medições dessas grandezas correspondam a resultados reais e consistentes com as previsões teóricas.

Além disso, os operadores hermitianos possuem um conjunto completo de autovetores que formam uma base para o espaço de Hilbert. Esses autovetores representam os estados próprios (ou estados estacionários) do sistema, nos quais as observáveis associadas aos operadores têm valores bem definidos. Essa propriedade é fundamental para a interpretação da mecânica quântica e para a obtenção de resultados mensuráveis por meio de experimentos.

Portanto, ao representar as quantidades observáveis por operadores hermitianos, a mecânica quântica assegura que as medições sejam consistentes, permite a obtenção de valores reais para as grandezas observadas e fornece uma estrutura matemática adequada para o estudo e a descrição dos sistemas quânticos.

2.6.3 Terceiro postulado

O terceiro postulado, por sua vez, afirma que a medição de uma grandeza física associada a um observável $\hat{\mathbf{O}}$ poderá fornecer como resultado somente um dos autovalores de $\hat{\mathbf{O}}$.

Parte II

Materias e métodos

3 Revisão Sistemática

3.0.1 Introdução

A escrita de trabalhos acadêmicos de longa duração, sejam eles monografias, dissertações ou teses, em geral segue um roteiro canônico. Primeiro, o autor, ou conjunto de autores, determina um tema a partir do contexto, conhecimento prévio, tempo para a escrita etc. Em seguida, o encaminha para uma figura de orientação a fim de delimitá-lo ponderando questões como ineditismo, nível de complexidade e relevância. Após isso, começa o pesaroso trabalho de revisão biliográfica através de artigos, livros, conferências, dentre outras fontes. Nesta etapa, principalmente quano há um grande volume de material, a auditabilidade, isto é, a possibilidade de o leitor conseguir reproduzir uma série de passos e chegar aos mesmos resultados, tende a ser de extrema relevância e complexa realização. Isso se deve sobretudo ao volume de questões com as quais ela deve se ocupar, tais como *quais bases foram escolhidas?*, *quais foram seus critérios de seleção?*, *quais foram as strings de busca e palavras-chave utilizadas?*.

As Revisões Sistemáticas como método de revisão biliográfica vêm ao encalço dessas questões.

> ....elas têm como objetivo apresentar uma avaliação criteriosa a respeito de um tópico de pesquisa, fazendo uso de uma metodologia de revisão que seja confiável, rigorosa e que permia auditagem. Kitchenham (2015)

3.0.2 Etapas

Segundo Biolchini et al. (2005), o processo de revisão consiste em um conjunto de três passos - Planejamento, Execução e Análise de Resultados - bem definidos e planejados de acordo com um protocolo previamente estabelecido.

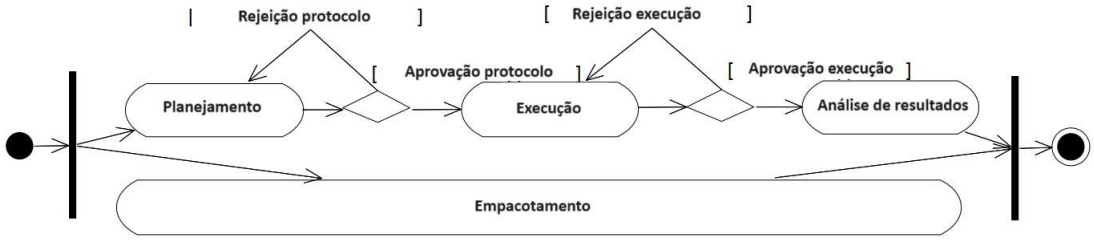

Figura 3 – Processo de Revisão Sistemática. Fonte: (BIOLCHINI et al., 2005, p. 10) (Adapt.)

3.0.2.1 Planejamento

O objetivo desta etapa é construir um protocolo que guiará o processo de revisão. Nele, os seguintes tópicos devem ser abordados:

1. Objetivo(s) da revisão;
2. Pergunta principal;
3. Palavras-chave que serão utilizadas e seus sinônimos;
4. Critérios de seleção das bases de dados (indexação, idioma etc);
5. Linguagem dos estudos que serão coletados;
6. Método de pesquisa nas bases de dados;
7. Bases de dados escolhidas;
8. Critérios de inclusão-exclusão;
9. Tipos de estudo (primário ou secundário);
10. Método inicial de seleção;
11. Campos de extração das informações.

3.0.2.2 Execução

Nesta etapa, os estudos são identificados nas bases de 7 a partir do método especificado em 6. Após isso, a estratégia de 10 é utilizada em conjunto com 8 para a primeira filtragem do resultado da busca, processo denominado *seleção primária*. Por fim, os resultados dessa etapa são lidos na íntegra a fim de se realizar uma segunda peneiração, a dita *seleção secundária*.

Após ambas as seleções, as informações dos campos especificados em 11 são coletadas. Ou seja, o propósito final da etapa de execução é extrair uma lista de campos especificados no protocolo com base em duas seleções, uma mais objetiva e superficial e outra mais aprofundada.

3.0.2.3 Análise de Resultados

Também chamade de “Sumarização de Resultados”, o objetivo desta etapa é efetivamente “resumir” o material resultante da etapa anterior identificando convergências, divergências e áreas inexploradas.

3.1 Ferramenta StArt

Como destacado na seção 3.0.1, a Revisão Sistemática provê um método cartesiano para a coleta e análise do estado da arte de um tópico. No entanto, dada a quantidade de passos e requisitos que deve cumprir, seguí-la sem erros pode ser uma tarefa desafiadora, sobretudo considerando o alto volume de artigos recorrentemente gerado após sua aplicação.

Cientes disso, pesquisadores do LaPES da UFSCar produziram a *StArt*, ferramenta gratuita voltado à facilitação do processo de Revisão Sistemática que permite realizar todas as suas etapas, exportar o estado atual delas, visualizar estatísticas a respeito dos materiais coletados, integrar com diversas bases de dados, dentre outras funcionalidades.

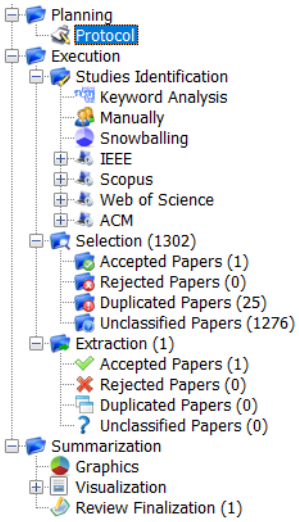

Figura 4 – Etapas Revisão Sistemática - StArt. Fonte: Coletada pelo autor

Observe na imagem 4 como constam na raiz as três etapas com suas respectivas tarefas: *Planning*, *Execution* e *Summarization*.

3.2 Aplicação

3.2.1 Protocolo

Objetivo	Determinar quais são os principais algoritmos quânticos e suas implementações
Palavras-chave e sinônimos	algoritmos quântcos aplicações applications computação quântica graduação quantum algorithms quantum computing revisão sistemática systematic review undergraduate
Critérios de definição das fontes	Indexar estudos na área de física ou computação Exportar o que foi encontrado em .ris ou .csv

Idiomas dos estudos	Inglês Português Espanhol
Método de pesquisa	Construir uma string de busca com base nas keywords definidas e depois exportar os artigos recuperados nos formatos aceitos pela *StArt*
Lista de Bases	IEEE Web of Science Scopus ACM
Critérios de Inclusao dos Estudos	O estudo descreve o funcionamento de algum algoritmo quântico O estudo apresenta uma visão introdutória sobre computação quântica O estudo descreve uma aplicação de algum dos cinco algoritmos
Critérios de Exclusão dos Estudos	O estudo não possui abordagem prática O estudo não está nos idiomas selecionados
Definição dos Tipos de Estudo	Primários Secundários
Estratégia Inicial de Seleção	Com base na leitura do título e abstract dos artigos e aplicando os clitérios de inclusão-exclusão definidos.
Campos a serem extraídos	Algoritmo descrito Breve descrição da aplicação Traz alguma nova perspectiva para o algoritmo em questão

Tabela 1 – Protocolo de Revisão Sistemática

3.2.2 Execução

As bases e coleções foram acessadas por meio da Rede CAFe [1] disponibilizada pela CAPES para a comunidade acadêmica da UnB.

A string `(quantum algorithms OR quantum computing OR implementation) AND (DEUTSCH OR DEUTSCH-JOZSA OR SIMON OR GROVER OR SHOR)` foi aplicada aos campos de busca das bases especificadas no protocolo. Em seguida, os resultados foram limitados a até 20 e ordenados por relevância e data de publicação, retornando os *pappers* da tabela 2.

Após essa pesquisa, houve a importação em `.csv` para a ferramente StArt e a utilização da estratégia inicial de seleção estabelecida no protocolo para produzir as estatísticas dos artigos.

[1] Mais informações no site

Id	Artigo	Autores	Ano	Base(s)
1	Quantum Machine Learning: A Case Study of Grover's Algorithm	Khanal, Bikram and Rivas, Pablo and Orduz, Javier and Zhakubayev, Alibek	2021	IEEE
2	Quantum cryptography based on Grover's algorithm	Sakhi, Z. and Kabil, R. and Tragha, A. and Bennai, M.	2012	IEEE
3	Experimental Implementation of Shor's Quantum Algorithm to Break RSA	Albuainain, Aminah and Alansari, Jana and Alrashidi, Samiyah and Alqahtani, Wasmiyah and Alshaya, Jana and Nagy, Naya	2022	IEEE
4	Quantum Algorithms: Overviews, Foundations, and Speedups	Wang, Shuangbao Paul and Sakk, Eric	2021	IEEE
5	Using Quantum computers to speed up dynamic testing of software	Miranskyy, A. ; Pecorelli F. ; Barletta V.S. ; Serrano M.A.	2022	Scopus
6	Solving the Shortest Path Problem with QAOA	Fan, Z. ; Xu, J. ; Shu, G. ; Ding, X. ; Lian, H. ; Shan, Z.	2023	Scopus
7	Quantum-resistance in blockchain networks	Allende, M. ; Leon, D.L. ; Ceron, S. ; Pareja, A. ; Pacheco, E. ; Leal, A. ; Da Silva, M. ;	2023	Scopus
8	Quantum technology to expand soft computing	Werbos, P.J.	2022	Scopus
9	SPEEDY Quantum Circuit for Grover's Algorithm	Song, Gyeongju and Jang, Kyoungbae and Kim, Hyunjun and Eum, Siwoo and Sim, Minjoo and Kim	2022	WoS
10	Shor's Algorithm for Quantum Numbers Using MATLAB Simulator	Nagaich, Shweta and Goswami, Y.C.	2015	WoS/IEEE
11	Why Haven't More Quantum Algorithms Been Found?	Shor, Peter W.	2003	ACM
12	When Post-Quantum Cryptography Meets the Internet of Things: An Empirical Study	Chung, Chia-Chin and Pai, Chu-Chi and Ching, Fu-Shiang and Wang, Chao and Chen, Ling-Jyh	2022	ACM
13	When Machine Learning Meets Quantum Computers: A Case Study	Jiang, Weiwen and Xiong, Jinjun and Shi, Yiyu	2021	ACM
14	Quantum Related-Key Attack Based on Simon's Algorithm and Its Applications	Zhang, P.	2023	ACM

Tabela 2 – Estudos Identificados - IEEE/Scopus/WoS/ACM

Parte III

Computação Quântica

4 Bits e qubits

4.1 Bits

Um **bit** é uma unidade de informação que descreve uma sistema clássico de duas dimensões/estados. Alguns exemplos emblemáticos de bits são:

- O estado da corrente atravessando um circuto (*alto* e *baixo*);
- Uma maneira de denotar *true* ou *false*;
- Um botão no estado *ligado* ou *desligado*.

Recorrendo à notação matricial, 0 e 1 podem ser representados como os estados $|0\rangle$ e $|1\rangle$ na forma das seguintes matrizes 2x1:

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

Esse pragmatismo embutido no bit (ou $|0\rangle$ ou $|1\rangle$) é perfeito para representar situações clássicas.

4.2 Qubits

No mundo quântico há contextos nos quais um agente envolvido no sistema está em ambos os estados, 1 e 0, ao mesmo tempo, o que leva à definição da contraparte quântica do *bit* - o *qubit*. Isto é, *uma unidade de informação que descreve um sistema quântico bidimensional*

Matematicamente, um qubit é representado como um vetor em um espaço de Hilbert complexo de dimensão 2. Esses vetores de estado são geralmente denotados como $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, onde α e β são coeficientes complexos e $|0\rangle$ e $|1\rangle$ são os estados de base que formam uma base ortonormal para o espaço de Hilbert. Esses coeficientes α e β são as chamados *amplitudes de probabilidade* e estão sujeitos à restrição de que $|\alpha|^2 + |\beta|^2 = 1$ para garantir a normalização do vetor de estado.

Ele pode ser representado como uma matriz 2x1 na qual os elementos são números complexos c_0 e c_1 com $|c_0|^2 + |c_1|^2 = 1$, isto é

$$\begin{bmatrix} c_0 \\ c_0 \end{bmatrix}.$$

Ou seja, o bit clássico pode ser interpretado como um tipo especial de qubit. Logo, $|c_0|^2$ e $|c_1|^2$ seriam as probabilidades de, após realizada a sua medição, ele estar no estado $|0\rangle$ e $|1\rangle$, respectivamente. Note que

$$\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = c_0 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + c_1 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = c_0|0\rangle + c_1|1\rangle \tag{4.1}$$

Como foi discutido, o produto tensorial é usado para "combinar" sistemas quânticos. Um byte 01101011, por exemplo, é representado pela equação

$$|0\rangle \otimes |1\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |0\rangle \otimes |1\rangle \otimes |1\rangle \tag{4.2}$$

que é um elemento de 4.3, um espaço vetorial de dimensão $2^8 = 256$.

$$\mathbb{C} \otimes \mathbb{C} \otimes \mathbb{C} \otimes \mathbb{C} \otimes \mathbb{C} \otimes \mathbb{C} \otimes \mathbb{C} \otimes \mathbb{C} \tag{4.3}$$

4.2.1 Superposição

A superposição é um conceito fundamental na computação quântica, que permite que um qubit exista em múltiplos estados simultaneamente. Isso significa que um qubit pode ser representado por uma combinação linear dos estados $|0\rangle$ e $|1\rangle$, como por exemplo $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Essa capacidade de estar em vários estados ao mesmo tempo permite que a computação quântica realize cálculos em paralelo, explorando todas as possibilidades de forma eficiente. Através das portas quânticas, é possível criar superposições e manipular os estados dos qubits para realizar operações complexas.

4.2.2 Emaranhamento

O emaranhamento é um fenômeno quântico no qual dois ou mais qubits se tornam correlacionados de forma intrincada, de modo que as propriedades de um qubit dependem das propriedades dos outros qubits emaranhados. Por exemplo, dois qubits emaranhados podem estar em um estado entrelaçado conhecido como "EPR pair"ou par de Einstein-Podolsky-Rosen, dado por $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Quando um qubit é medido, o estado do outro qubit é instantaneamente determinado, independentemente da distância que os separa. O emaranhamento desempenha um papel crucial em muitos algoritmos quânticos, permitindo a realização de operações e comunicações quânticas mais poderosas.

4.2.3 Medição

A medição em um sistema quântico é um processo probabilístico que permite extrair informações dos qubits. Quando um qubit é medido, seu estado quântico colapsa

em um estado específico correspondente ao resultado da medição. Por exemplo, se um qubit está em um estado superposto $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e é medido, ele colapsará em um dos dois estados possíveis, $|0\rangle$ ou $|1\rangle$, com uma probabilidade associada a cada estado determinada pelas amplitudes do estado superposto. A medição é uma etapa essencial nos algoritmos quânticos, pois fornece a saída final do cálculo quântico.

4.3 Esfera de Bloch

A Esfera de Bloch, como mostrado na figura 5, é uma representação geométrica do estado de um qubit em um espaço de Hilbert bidimensional. Na esfera, os polos representam os estados clássicos $|0\rangle$ e $|1\rangle$, e qualquer outro ponto na superfície corresponde a uma superposição desses estados. O estado de um qubit $|\Psi\rangle$ pode ser representado por um vetor na esfera, dado por:

$$|\Psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle,$$

onde θ e ϕ são ângulos que definem a posição do vetor na esfera. O ângulo θ (medido a partir do polo z positivo) e o ângulo ϕ (medido no plano xy a partir do eixo x) determinam as probabilidades e as fases relativas dos estados básicos, respectivamente.

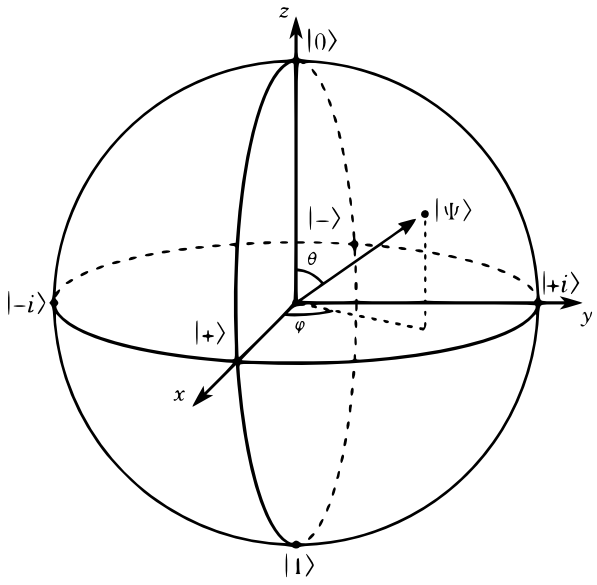

Figura 5 – Esfera de Bloch. Fonte: Criado pelo autor.

5 Portas Quânticas

5.1 Introdução

As portas quânticas são operadores unitários que atuam nos qubits para realizar transformações na informação quântica. Matematicamente, elas são representadas como matrizes unitárias que operam no espaço de Hilbert do qubit. Essas matrizes são conhecidas como *operadores de porta*. Cada porta quântica realiza uma transformação específica no estado do qubit, alterando as amplitudes de probabilidade e, consequentemente, as probabilidades de observação dos diferentes estados. As portas quânticas são fundamentais para a manipulação e processamento de informações quânticas.

Podem ser vistas também como transformações lineares no espaço de Hilbert do qubit e representadas por matrizes unitárias, que preservam a norma e a ortogonalidade dos vetores de estado. A combinação de portas quânticas em uma sequência de operações permite a implementação de algoritmos e a realização de cálculos quânticos. Essas sequências de portas quânticas podem ser descritas como composições de operadores de porta, onde a ordem das operações é importante e afeta o resultado final.

5.2 Operações Fundamentais em Qubits Individuais

As operações de um único qubit são a base para a manipulação de estados quânticos. Ao aplicar portas quânticas a um único qubit, pode-se realizar uma variedade de tarefas computacionais, desde a simples inversão de um estado até a criação de superposições e a introdução de fases. As portas de um único qubit são representadas por matrizes 2×2 e podem ser visualizadas geometricamente na Esfera de Bloch, uma representação que facilita o entendimento de suas funções.

5.2.1 Porta X (NOT quântico)

A porta X, também conhecida como "NOT quântico", inverte o estado de um qubit. Dessse modo, quando é aplicada ao estado $|0\rangle$, o transforma no estado $|1\rangle$ e vice-versa. Trata-se, portanto, de uma rotação de 180° no eixo z da esfera de Bloch. Matematicamente, a porta X é representada pela matriz

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

e graficamente pela figura abaixo:

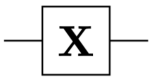

Figura 6 – Porta Quântica X. Fonte: Criado pelo autor.

Observe que ao aplicar a porta X ao estado $|0\rangle$, obtem-se:

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

E ao aplicar a porta X ao estado $|1\rangle$, obtem-se:

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

5.2.2 Porta Y

A porta Y realiza uma rotação de 180° no eixo y da esfera de Bloch. Ela mapeia o estado $|0\rangle$ em $i|1\rangle$ e o estado $|1\rangle$ em $-i|0\rangle$, o que modifica a amplitude original. Matematicamente, ela é definida pela matriz

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

e possui a representação gráfica da figura 7.

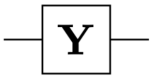

Figura 7 – Porta Quântica Y. Fonte: Criado pelo autor.

5.2.3 Porta Z

A porta Z realiza uma inversão de fase do estado $|1\rangle$. Ela deixa o estado $|0\rangle$ inalterado. Matematicamente, a porta Z é representada pela matriz:

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

5.2.4 Porta Hadamard

A porta Hadamard cria uma superposição equilibrada entre os estados $|0\rangle$ e $|1\rangle$. Ela mapeia o estado $|0\rangle$ para $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e o estado $|1\rangle$ para $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Matematicamente, a porta Hadamard é representada por:

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

5.2.5 Porta S

A porta S é uma porta de fase que introduz uma fase de 90 graus ($\frac{\pi}{2}$) no estado $|1\rangle$. Ela deixa o estado $|0\rangle$ inalterado. A definição matemática da porta S é:

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

5.2.6 Porta T

A porta T é uma extensão da porta S que introduz uma fase de 45 graus ($\frac{\pi}{4}$) no estado $|1\rangle$. Ela deixa o estado $|0\rangle$ inalterado. Matematicamente, a porta T é representada por:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}.$$

5.2.7 Porta CNOT (Controlled-NOT)

A porta CNOT é uma porta de dois qubits que aplica uma porta X ao segundo qubit somente quando o primeiro qubit está no estado $|1\rangle$. Ela mantém o segundo qubit inalterado quando o primeiro qubit está no estado $|0\rangle$. Matematicamente, a porta CNOT é definida pela matriz:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Essas são apenas algumas das portas quânticas mais comuns, mas existem muitas outras que desempenham papéis importantes na computação e processamento quântico. Cada porta quântica realiza uma transformação específica nos qubits, alterando seus estados e permitindo a realização de diferentes operações e cálculos quânticos.

5.3 Operações Fundamentais em Múltiplos Qubits

As operações em múltiplos qubits são cruciais na computação quântica para a criação de estados emaranhados e a execução de algoritmos complexos. Portas como CNOT,

CZ e outras são responsáveis por introduzir interações entre qubits, o que é essencial para a realização de cálculos paralelos e a manipulação avançada de informações quânticas. Segue uma descrição de algumas das portas mais fundamentais para múltiplos qubits, junto com suas matrizes representativas, tabelas de verdade e uma breve descrição de cada uma.

5.3.1 Porta CNOT (Controlled-NOT)

A porta CNOT é uma das portas quânticas mais fundamentais em sistemas de múltiplos qubits. É uma operação condicional que aplica a porta X (NOT) em um qubit-alvo somente se o qubit de controle estiver no estado $|1\rangle$.

Matematicamente, é representada pela matriz:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Sua tabela de verdade é a seguinte:

Controle	**Alvo**	**Alvo de Saída**
0	0	0
0	1	1
1	0	1
1	1	0

E ela é representada graficamente pela figura 8.

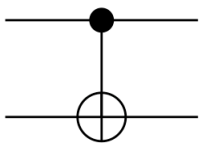

Figura 8 – Porta Quântica CNOT. Fonte: Criado pelo autor.

5.3.2 Porta CZ (*Controlled Phase*)

A porta CZ, também conhecida como a porta de Fase Controlada, aplica uma inversão de fase no qubit-alvo apenas se o qubit de controle estiver no estado $|1\rangle$. Essa porta é particularmente útil para criar emaranhamento entre qubits e realizar operações condicionais complexas.

Matematicamente, a porta CZ é representada pela matriz:

$$\text{CZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

A tabela de verdade para a porta CZ é:

Controle	**Alvo**	**Alvo de Saída**
0	0	0
0	1	1
1	0	0
1	1	1 (com inversão de fase)

Assim como a porta CNOT, a porta CZ não altera o estado do qubit de controle, mas aplica uma operação condicional no qubit alvo. A diferença principal é que, em vez de aplicar a operação NOT, a porta CZ aplica uma inversão de fase.

A representação gráfica da porta CZ pode ser ilustrada como segue:

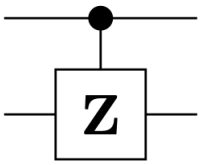

Figura 9 – Porta Quântica CZ. Fonte: Criado pelo autor.

5.3.3 Porta CCNOT/CCX/Toffoli

A porta Toffoli, também conhecida como porta CCNOT (*Controlled-Controlled-NOT*) ou porta CCX, é uma extensão da porta CNOT para três qubits. Ela realiza uma operação NOT no terceiro qubit (qubit-alvo) apenas quando os dois primeiros qubits (qubits de controle) estão ambos no estado $|1\rangle$. Essa porta é essencial em computação quântica para implementar operações lógicas clássicas, como o AND, em um contexto quântico.

Matematicamente, a porta Toffoli é representada pela seguinte matriz:

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

Sua tabela de verdade é:

Controle 1	**Controle 2**	**Alvo de Saída**
0	0	Inalterado
0	1	Inalterado
1	0	Inalterado
1	1	Inverte

A porta Toffoli é um componente chave para a construção de circuitos quânticos universais, pois permite a implementação de qualquer função lógica clássica. Ela é um exemplo de como a lógica clássica pode ser incorporada em sistemas quânticos.

A representação gráfica da porta Toffoli pode ser ilustrada como segue:

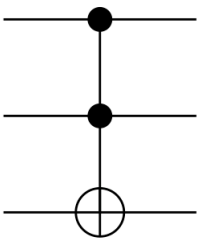

Figura 10 – Porta Quântica Toffoli. Fonte: Criado pelo autor.

Parte IV

Algoritmos e aplicações

5.4 Introdução

Os algoritmos são frequentemente desenvolvidos muito antes das máquinas em que se supõe que devem funcionar. Algoritmos clássicos precedem computadores clássicos por milênios e, de maneira semelhante, existem vários algoritmos quânticos antes que qualquer computador quântico de grande escala tenha visto a luz do dia. Esses algoritmos manipulam qubits para resolver problemas e, em geral, resolvem essas tarefas com mais eficiência do que os computadores clássicos.

A Seção 5.5 descreve o algoritmo de Deutsch, que determina uma propriedade das funções de $0, 1$ para $0, 1$. Na Seção 5.6, este algoritmo é generalizado para o algoritmo Deutsch-Jozsa, que lida com uma propriedade semelhante para funções de $0, 1^n$ para $0, 1$.

5.5 Algoritmo Deutsch

O algoritmo de Deutsch, também conhecido como o problema de Deutsch, é um dos primeiros e mais simples exemplos demonstrando a superioridade da computação quântica sobre a clássica em resolver certos problemas específicos. Ele aborda o problema de determinar se uma função quântica desconhecida $f : \{0, 1\} \to \{0, 1\}$ é constante ou balanceada. Uma função é considerada constante se ela retorna o mesmo valor para todas as entradas, e balanceada se retorna 0 para uma entrada e 1 para a outra.

O algoritmo de Deutsch pode determinar essa propriedade com uma única avaliação da função f, usando um circuito quântico e explora a interferência quântica, a superposição de estados e o emaranhamento, que são fenômenos centrais para o poder da computação quântica.

Ele é composto por quatro passos principais:

1. Preparação do estado inicial: Dois qubits são inicializados, um em estado $|0\rangle$ e outro em estado $|1\rangle$. Isso cria o estado inicial $|\psi_0\rangle = |0\rangle \otimes |1\rangle$, onde o símbolo $\otimes$ denota o produto tensorial, indicando que os dois qubits estão em um estado combinado que é o produto de seus estados individuais.

2. Aplicação da porta Hadamard: Uma porta Hadamard é aplicada ao primeiro qubit para colocá-lo em uma superposição de estados. A porta Hadamard transforma o estado $|0\rangle$ no estado $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e o estado $|1\rangle$ no estado $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Após aplicar a porta Hadamard, obtemos o estado $|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle$.

3. Consulta à função f: A função f é aplicada ao segundo qubit de maneira controlada pelo primeiro qubit. Isso é geralmente alcançado por uma operação quântica controlada, como a porta CNOT, que entrelaça os dois qubits. A função f é representada

por uma porta quântica U_f tal que $U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$. Após a aplicação de U_f, o sistema evolui para o estado $|\psi_2\rangle$.

4. Medição: A medição do primeiro qubit nos dirá se a função é constante ou balanceada. Se o primeiro qubit for medido no estado $|0\rangle$, então sabemos que a função é constante. Se for medido no estado $|1\rangle$, a função é balanceada.

Sua aplicação mostra que apenas uma consulta à função f é suficiente para determinar se ela é constante ou balanceada. Esse é um exemplo claro de como a computação quântica pode ser mais eficiente do que a computação clássica em certos cenários.

5.5.1 Exemplos

5.5.1.1 $f : \{0,1\} \to \{0,1\}$

Considere a função $f : \{0,1\} \to \{0,1\}$, onde:

$$f(0) = 0, \quad f(1) = 1.$$

Nosso objetivo é determinar se a função f é constante ou balanceada. Para isso, utilizaremos o algoritmo de Deutsch.

Iniciamos com os qubits no estado $|\psi_0\rangle = |0\rangle \otimes |1\rangle$. A representação vetorial deste estado inicial é:

$$|\psi_0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \cdot 0 \\ 1 \cdot 1 \\ 0 \cdot 0 \\ 0 \cdot 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

A porta Hadamard é definida pela seguinte matriz:

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

Aplicamos a porta Hadamard H ao primeiro qubit para obter uma superposição de estados. A operação de tensor product entre a matriz Hadamard e o vetor do primeiro qubit é dada por:

$$|\psi_1\rangle = H \otimes I \cdot |\psi_0\rangle,$$

onde I é a matriz identidade de dimensão 2, representando a operação no segundo qubit (que permanece inalterado). Realizando a multiplicação, temos:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \otimes I \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \otimes I \\ 1 \otimes I \\ 1 \otimes (-I) \\ 1 \otimes (-I) \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

Expandindo o produto tensorial e realizando a multiplicação, obtemos:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \cdot 0 + 1 \cdot 1 \\ 1 \cdot 0 + 1 \cdot 0 \\ 1 \cdot 0 - 1 \cdot 1 \\ 1 \cdot 0 - 1 \cdot 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle.$$

Continuamos nossa análise matemática após a aplicação da porta Hadamard. Agora, aplicamos a porta quântica controlada U_f, que implementa a função f no nosso sistema quântico. A matriz representativa de U_f, dada a nossa função específica $f(0) = 0$ e $f(1) = 1$, é:

$$U_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Agora, aplicamos esta matriz ao nosso estado superposto $|\psi_1\rangle$. A multiplicação dessa matriz pelo estado $|\psi_1\rangle$ é calculada como:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}} U_f(|0\rangle \otimes |1\rangle + |1\rangle \otimes |1\rangle).$$

Vamos expandir esta operação e calcular a aplicação de U_f termo a termo. O primeiro termo é $U_f|0\rangle \otimes |1\rangle$, que resulta em $|0\rangle \otimes |1\rangle$ já que a operação CNOT não altera o estado quando o qubit de controle está no estado $|0\rangle$. O segundo termo é $U_f|1\rangle \otimes |1\rangle$, que resulta em $|1\rangle \otimes |0\rangle$ já que a operação CNOT inverte o qubit alvo quando o qubit de controle está no estado $|1\rangle$. Portanto:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |1\rangle + |1\rangle \otimes |0\rangle).$$

Em termos de vetores, a multiplicação de U_f com cada termo é:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right)$$

Expandindo a multiplicação dos vetores pelos blocos da matriz U_f, obtemos o estado final:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}} \left(\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \right) = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle).$$

O estado $|\psi_2\rangle$ é uma superposição igual dos estados $|01\rangle$ e $|10\rangle$. Em termos de medição quântica, quando realizamos uma medição no primeiro qubit, estamos projetando este qubit nos estados de base $|0\rangle$ e $|1\rangle$.

Dado que a superposição é igual, as probabilidades de projeção são determinadas pelos quadrados das amplitudes de cada componente da superposição. Para o estado $|\psi_2\rangle$, as amplitudes são ambas $\frac{1}{\sqrt{2}}$, portanto as probabilidades são:

$$P(|0\rangle) = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2},$$
$$P(|1\rangle) = \left| \frac{1}{\sqrt{2}} \right|^2 = \frac{1}{2}.$$

Estas probabilidades são iguais porque a amplitude de cada estado na superposição é a mesma. Assim, a medição do primeiro qubit resultará em $|0\rangle$ com probabilidade $\frac{1}{2}$ e em $|1\rangle$ também com probabilidade $\frac{1}{2}$.

Em nosso caso específico, onde $f(0) = 0$ e $f(1) = 1$, após a aplicação da porta CNOT e a realização da medição, se o primeiro qubit for medido como $|0\rangle$, isso significa que a aplicação de U_f não alterou o segundo qubit, indicando que a função f é constante. No entanto, para a função dada, a medição do primeiro qubit resultará em $|1\rangle$, o que revela que a função é balanceada.

Portanto, o algoritmo de Deutsch nos permite determinar com uma única consulta à função f se ela é constante ou balanceada, um feito que destaca a eficiência superior dos computadores quânticos para certas classes de problemas em comparação com os computadores clássicos.

5.5.1.2 Oráculo quântico

Considere agora uma função $f : \{0, 1\} \to \{0, 1\}$ que é prometida ser ou constante ou balanceada, mas a forma da função não é explicitamente dada. Em vez disso, é nos fornecido um oráculo quântico U_f que implementa a função desconhecida f.

No contexto do algoritmo de Deutsch e de outros algoritmos quânticos, um "oráculo quântico" refere-se a uma operação de caixa preta que implementa uma função específica. O termo "oráculo" é usado em teoria da computação para descrever uma entidade que resolve um problema específico ou executa uma tarefa específica, mas cujo funcionamento

interno não é necessariamente conhecido ou relevante para o usuário do oráculo. Em nosso caso, o oráculo quântico U_f é uma representação da função desconhecida f no contexto da computação quântica.

O oráculo quântico U_f opera da seguinte maneira:

$$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle .$$

Aqui, $|x\rangle$ e $|y\rangle$ são qubits de entrada, e $f(x)$ é a saída da função desconhecida f quando aplicada à entrada x. O resultado da operação do oráculo é armazenado no qubit $|y\rangle$, que é transformado de acordo com o valor de $f(x)$. A operação $\oplus$ representa a soma módulo 2 (XOR).

O uso de um oráculo quântico é fundamental em vários algoritmos quânticos porque permite a incorporação de uma função específica no algoritmo sem a necessidade de compreender ou implementar detalhadamente a lógica da função. Isso torna o oráculo uma ferramenta poderosa para demonstrar as vantagens da computação quântica em certos problemas, mesmo sem conhecer os detalhes específicos da função f.

No algoritmo de Deutsch, o oráculo quântico U_f permite-nos determinar se a função desconhecida f é constante ou balanceada com apenas uma consulta ao oráculo, uma tarefa que seria impossível em um único passo na computação clássica.

Nosso objetivo continua sendo determinar se a função é constante ou balanceada, mas agora sem conhecimento prévio de f. Para isso, utilizaremos o algoritmo de Deutsch.

O estado inicial dos qubits é o mesmo do exemplo anterior:

$$|\psi_0\rangle = |0\rangle \otimes |1\rangle.$$

Após a aplicação da porta Hadamard ao primeiro qubit, chegamos ao estado:

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |1\rangle.$$

Em seguida, aplicamos o oráculo quântico U_f ao nosso estado superposto $|\psi_1\rangle$, resultando em:

$$|\psi_2\rangle = U_f|\psi_1\rangle.$$

O oráculo U_f realiza a seguinte transformação quântica em nosso sistema:

$$U_f \left|x\right\rangle \left|y\right\rangle = \left|x\right\rangle \left|y \oplus f(x)\right\rangle .$$

Expandindo o estado $|\psi_2\rangle$ sem conhecer explicitamente f, obtemos:

$$|\psi_2\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle \otimes |1 \oplus f(0)\rangle + |1\rangle \otimes |1 \oplus f(1)\rangle\right) .$$

Agora, sem saber f, analisamos as possíveis medições do primeiro qubit após aplicar novamente a porta Hadamard ao primeiro qubit:

$$|\psi_3\rangle = (H \otimes I)|\psi_2\rangle.$$

A medição do primeiro qubit nos dirá se f é constante ou balanceada. Se a função f é constante, as amplitudes para $|0\rangle$ e $|1\rangle$ no primeiro qubit interferirão construtivamente para $|0\rangle$ após a segunda aplicação da porta Hadamard, resultando em uma probabilidade de 1 para medir $|0\rangle$. Se f é balanceada, haverá interferência destrutiva para $|0\rangle$ e construtiva para $|1\rangle$, resultando em uma probabilidade de 1 para medir $|1\rangle$.

5.5.1.3 Qiskit

A implementação da função e do oráculo correspondente no Qiskit é apresentada a seguir:

```
import qiskit
from qiskit import QuantumCircuit, Aer, execute

# Funcao para criar o oraculo para uma funcao f dada
def criar_oraculo(f):
    # Criar um circuito quantico com 3 qubits
    oraculo = QuantumCircuit(3, name='oraculo')

    # Implementacao da funcao f como parte do oraculo
    if f == 'balanceada':
        oraculo.cx(0, 2) # CNOT controlado pelo primeiro qubit
        oraculo.cx(1, 2) # CNOT controlado pelo segundo qubit
    elif f == 'constante':
        pass # Funcao constante nao faz nada

    oraculo.to_gate()
    return oraculo

# Funcao implementando o algoritmo de Deutsch
def algoritmo_deutsch(f):
    # Criar o oraculo para a funcao dada
    oraculo = criar_oraculo(f)

    # Criar um circuito quantico com 3 qubits e 1 bit classico
```

```
    qc = QuantumCircuit(3, 1)

    # Preparar o estado inicial
    qc.x(2)
    qc.h(range(3))

    # Aplicar o oraculo
    qc.append(oraculo, range(3))

    # Aplicar portas Hadamard aos primeiros dois qubits
    qc.h(range(2))

    # Medir o primeiro qubit
    qc.measure(0, 0)

    # Executar o circuito
    backend = Aer.get_backend('qasm_simulator')
    job = execute(qc, backend, shots=1)
    result = job.result()
    counts = result.get_counts()

    # Interpretar o resultado
    bit_medido = list(counts.keys())[0]
    return 'constante' if bit_medido == '0' else 'balanceada'
```

5.5.1.4 Implementação Clássica do Algoritmo de Deutsch

Em uma abordagem clássica, o algoritmo de Deutsch requer duas avaliações da função f para determinar se ela é constante ou balanceada. A seguir, apresentamos um exemplo de implementação clássica do algoritmo de Deutsch em Python:

```
def deutsch_classico(f):
    # Avaliar a funcao para as duas entradas possiveis
    resultado_0 = f(0)
    resultado_1 = f(1)

    # Determinar se a funcao e constante ou balanceada
    if resultado_0 == resultado_1:
        return 'constante'
```

```
    else:
        return 'balanceada'

# Exemplo de funcoes
def funcao_constante(x):
    return 0

def funcao_balanceada(x):
    return x

# Testando o algoritmo
print(deutsch_classico(funcao_constante))  # Deve retornar 'constante'
print(deutsch_classico(funcao_balanceada)) # Deve retornar 'balanceada'
```

Nesta implementação, a função f é avaliada para ambas as entradas possíveis (0 e 1), e a natureza da função (constante ou balanceada) é determinada comparando esses dois resultados.

5.5.1.5 Implementação Quântica do Algoritmo de Deutsch no Qiskit

A implementação do algoritmo de Deutsch no Qiskit permite visualizar o funcionamento da computação quântica na prática. A seguir, apresentamos uma implementação detalhada desse algoritmo, com comentários explicativos para cada passo.

```
# Importando as bibliotecas necessarias do Qiskit
from qiskit import QuantumCircuit, Aer, execute
from qiskit.visualization import plot_histogram

# Definicao do oraculo para uma funcao f
# Esta funcao cria um circuito que implementa a funcao f
def criar_oraculo(f):
    oraculo = QuantumCircuit(2, name='oraculo')

    # Implementacao do oraculo dependendo se f e constante ou balanceada
    if f == 'constante':
        # Para uma funcao constante, nao e feita nenhuma operacao
        pass
    elif f == 'balanceada':
        # Para uma funcao balanceada, utilizamos a porta CNOT
        oraculo.cx(0, 1)
```

```
    oraculo.to_gate()
    return oraculo

# Funcao que implementa o algoritmo de Deutsch
def algoritmo_deutsch(f):
    # Criacao do circuito quantico com 2 qubits e 1 bit classico
    qc = QuantumCircuit(2, 1)

    # Preparacao do estado inicial
    qc.x(1) # Aplicando a porta X ao segundo qubit para iniciar em |1>
    qc.h(0) # Aplicando a porta Hadamard ao primeiro qubit
    qc.h(1) # Aplicando a porta Hadamard ao segundo qubit

    # Insercao do oraculo
    oraculo = criar_oraculo(f)
    qc.append(oraculo, [0, 1])

    # Aplicacao da porta Hadamard ao primeiro qubit apos o oraculo
    qc.h(0)

    # Medicao do primeiro qubit
    qc.measure(0, 0)

    # Executando o circuito no simulador
    backend = Aer.get_backend('qasm_simulator')
    job = execute(qc, backend, shots=1)
    result = job.result()
    counts = result.get_counts()

    # Interpretacao do resultado
    medido = list(counts.keys())[0] # Obtem o valor medido
    return 'constante' if medido == '0' else 'balanceada'

# Testando o algoritmo
print("Resultado para funcao constante:", algoritmo_deutsch('constante'))
print("Resultado para funcao balanceada:", algoritmo_deutsch('balanceada'
```

Neste código, o algoritmo de Deutsch é implementado em Qiskit passo a passo, desde a criação do oráculo que implementa a função f até a execução do circuito e a

interpretação do resultado. A função define o oráculo para simular a função f, seja ela constante ou balanceada. circuito é construído aplicando as portas necessárias e, finalmente, a medição do primeiro qubit determina se a função é constante ou balanceada.

5.5.2 Comparação entre as Abordagens Quântica e Clássica

O algoritmo de Deutsch é um exemplo notável que ilustra a vantagem da computação quântica sobre a clássica para certos problemas. Na abordagem clássica, duas avaliações da função f são necessárias para determinar se ela é constante ou balanceada. Em contraste, a abordagem quântica requer apenas uma única avaliação da função.

Esta diferença torna-se mais significativa à medida que o problema se expande para versões mais complexas, como o algoritmo de Deutsch-Jozsa, onde a eficiência da computação quântica se destaca ainda mais. Enquanto a computação clássica exigiria um número exponencialmente crescente de avaliações da função para resolver instâncias maiores do problema, a computação quântica consegue resolver todas as instâncias com uma única avaliação da função, independentemente do tamanho da entrada.

Assim, o algoritmo de Deutsch serve como uma prova de conceito para a superioridade da computação quântica em resolver certos tipos de problemas mais eficientemente do que é possível com a computação clássica.

5.6 Algoritmo Deutsch-Jozsa

O Algoritmo de Deutsch-Jozsa é uma extensão do Algoritmo de Deutsch que lida com funções mais gerais, nas quais é necessário determinar se a função é constante ou balanceada para um conjunto maior de entradas. Ao contrário do Algoritmo de Deutsch original, que requer apenas uma consulta à função, o Deutsch-Jozsa exige várias consultas. O objetivo é identificar o padrão geral da função em vez de avaliar entradas específicas.

O algoritmo começa com $n + 1$ qubits, onde n é o número de bits de entrada da função. Esses n qubits são inicializados no estado $|0\rangle$, e o último qubit é inicializado no estado $|1\rangle$. Em seguida, aplicamos a porta Hadamard em todos os qubits para criar uma superposição de todos os estados possíveis.

5.6.1 Exemplos

5.6.1.1 Bit mais significativo

Considere uma função $f(x)$ que retorna o bit mais significativo de x. Nosso objetivo é determinar se f é constante ou balanceada.

Agora, vamos seguir os passos do algoritmo Deutsch-Jozsa com essa função:

1. **Inicialização dos qubits:** Começamos com dois qubits de entrada inicializados no estado $|0\rangle$, e um qubit auxiliar inicializado no estado $|1\rangle$.

$$|\psi_0\rangle = |00\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix}.$$

2. **Aplicação da porta Hadamard:** Aplicamos a porta Hadamard em cada um dos três qubits para criar uma superposição de estados.

O estado inicial dos qubits é:

$$|\psi_0\rangle = |00\rangle \otimes |1\rangle.$$

Em termos de vetor, este estado é representado como:

$$|\psi_0\rangle = \begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix}.$$

A aplicação da porta Hadamard a cada qubit resulta em:

$$|\psi_1\rangle = (H \otimes H \otimes H)|\psi_0\rangle.$$

Calculamos esta operação passo a passo. Primeiro, expandimos o produto tensorial das três matrizes Hadamard:

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

Multiplicamos esta matriz pelo vetor de estado $|\psi_0\rangle$:

$$|\psi_1\rangle = (H \otimes H \otimes H)|\psi_0\rangle = \frac{1}{2\sqrt{2}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

A multiplicação resulta no estado $|\psi_1\rangle$, que é uma superposição de todos os possíveis estados dos três qubits.

3. **Aplicação da porta controlada pela função $f(x)$:**

Aplicamos a porta controlada pela função $f(x)$ no último qubit, controlado pelos dois primeiros qubits. No nosso exemplo, $f(x)$ retorna o bit mais significativo de x. Portanto, se o primeiro bit for 0, o último qubit não sofrerá alteração; caso contrário, o último qubit receberá uma fase negativa.

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

4. Aplicação da porta Hadamard novamente: Aplicamos a porta Hadamard nos dois primeiros qubits.

$$|\psi_3\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes \frac{1}{2}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Simplificando, temos:

$$|\psi_3\rangle = \frac{1}{2\sqrt{2}}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)$$

$$|\psi_3\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |100\rangle + |101\rangle)$$

Portanto, no nosso exemplo, a saída do algoritmo será um dos estados $|000\rangle$, $|001\rangle$, $|100\rangle$ ou $|101\rangle$, dependendo do valor do primeiro bit de entrada.

5.6.1.2 Qiskit

Consideramos uma função $f : \{0,1\}^3 \to \{0,1\}$ que é balanceada. A função retorna 1 se o número de bits 1 na entrada for ímpar, e 0 caso contrário.

```
import qiskit
from qiskit import QuantumCircuit, Aer, execute

# Definindo a funcao criar_oraculo
def criar_oraculo(quantum_circuit, qubits):
    # Aplica uma porta CNOT para cada qubit de entrada
    quantum_circuit.cx(qubits[0], qubits[-1])
    quantum_circuit.cx(qubits[1], qubits[-1])
    quantum_circuit.cx(qubits[2], qubits[-1])

# Numero de qubits de entrada
n = 3

qc = QuantumCircuit(n + 1, n)

# Inicializando o qubit auxiliar em |1>
qc.x(n)

# Aplicando portas Hadamard a todos os qubits
qc.h(range(n + 1))

# Criando e aplicando o oraculo
criar_oraculo(qc, range(n + 1))

# Aplicando portas Hadamard aos qubits de entrada novamente
qc.h(range(n))

# Medindo os qubits de entrada
qc.measure(range(n), range(n))
```

```
# Mostrando o circuito
print(qc.draw())

# Executando o circuito em um simulador quantico
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator).result()
counts = result.get_counts(qc)

# Exibindo os resultados
print(counts)
```

Se todos os resultados medidos forem $|0\rangle$, então f é constante. Se houver algum resultado $|1\rangle$, então f é balanceada.

5.7 Algoritmo de Busca de Grover

O algoritmo de busca de Grover, proposto por Lov Grover em 1996, é uma técnica de computação quântica para pesquisar elementos em um banco de dados não ordenado com a velocidade quadrática superior à dos melhores algoritmos clássicos. Este algoritmo é especialmente significativo porque demonstra uma das poucas maneiras em que a computação quântica supera de maneira clara e prática a computação clássica.

O algoritmo começa criando uma superposição uniforme de todos os possíveis estados de entrada, que representam os índices de todos os possíveis elementos em um banco de dados. Em seguida, o algoritmo usa a transformação de Grover, também conhecida como "operação de Grover" ou "iteração de Grover", que é uma combinação de operações quânticas projetadas para ampliar a amplitude do estado desejado (o "item de destino") e diminuir a amplitude de todos os outros estados. A operação de Grover é aplicada repetidamente - tipicamente cerca de $\sqrt{N}$ vezes, onde N é o número total de itens no banco de dados - para continuar amplificando a amplitude do item de destino até que a probabilidade de medir o estado quântico e encontrar o item de destino seja efetivamente maximizada.

Após a aplicação repetida da operação de Grover, o sistema quântico é medido. Com alta probabilidade, o resultado da medição será o índice do item de destino no banco de dados. Note-se que, embora o algoritmo de Grover acelere a busca em um banco de dados não ordenado, ele não fornece uma aceleração exponencial como alguns outros algoritmos quânticos (por exemplo, o algoritmo de fatoração de Shor). No entanto, o algoritmo de Grover mostra uma clara superioridade sobre os algoritmos clássicos para o mesmo problema, fornecendo uma aceleração quadrática. Isto, aliado à sua generalidade

e aplicabilidade a uma ampla gama de problemas, torna o algoritmo de Grover uma das descobertas mais importantes na computação quântica.

5.7.0.1 Implementação Qiskit do Algoritmo de Grover

Suponha que temos uma lista de números e queremos encontrar um número específico x nesta lista. A lista é representada por estados quânticos de n qubits, onde cada estado representa um número na lista.

```
import qiskit
from qiskit import QuantumCircuit, Aer, execute
import numpy as np

# Configuracao inicial
n = 3  # Numero de qubits
x = '101'  # O numero que queremos encontrar em formato binario

# Criar um circuito quantico
qc = QuantumCircuit(n+1, n)

# Aplicar a porta Hadamard para criar uma superposicao
for qubit in range(n):
    qc.h(qubit)

# Preparar o qubit auxiliar em estado |->
qc.x(n)
qc.h(n)

# Definindo o oraculo
# O oraculo inverte a fase do estado que estamos procurando
for i, char in enumerate(x):
    if char == '0':
        qc.x(i)
qc.mct(list(range(n)), n)  # Multi-controlled Toffoli
for i, char in enumerate(x):
    if char == '0':
        qc.x(i)

# Amplificacao de amplitude
```

```
for qubit in range(n):
    qc.h(qubit)
    qc.x(qubit)
qc.h(n-1)
qc.mct(list(range(n-1)), n-1)
qc.h(n-1)
for qubit in range(n):
    qc.x(qubit)
    qc.h(qubit)

# Medindo o circuito
qc.measure(range(n), range(n))

# Executando o circuito em um simulador
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator, shots=1024).result()
counts = result.get_counts()

# Exibindo os resultados
print(counts)
```

5.7.0.2 Implementação Clássica do Algoritmo de Grover

Embora o algoritmo de Grover seja intrinsecamente quântico, podemos considerar um análogo clássico que simula a busca em uma lista não ordenada. O seguinte código Python representa uma busca clássica simples:

```
def busca_classica(lista, item):
    for i, elemento in enumerate(lista):
        if elemento == item:
            return i
    return -1

# Exemplo de uso
lista = [3, 5, 7, 9, 11, 13]
item = 7
indice = busca_classica(lista, item)
```

Neste exemplo, o algoritmo percorre cada item da lista sequencialmente até encontrar o elemento desejado. A eficiência desse algoritmo é linear, ou seja, no pior caso,

ele precisa verificar cada elemento da lista uma vez.

5.7.1 Vantagens da Solução Quântica

O algoritmo de Grover oferece vantagens significativas sobre a abordagem clássica para a busca em uma lista não ordenada:

1. **Velocidade Quadraticamente Superior:** Enquanto o algoritmo clássico requer, em média, $N/2$ comparações (e no pior caso N comparações) para encontrar um item em uma lista de N elementos, o algoritmo de Grover encontra o item desejado em aproximadamente $\sqrt{N}$ operações. Esta é uma melhoria quadrática em relação ao algoritmo clássico.

2. **Superposição e Interferência Quântica:** O algoritmo de Grover utiliza a superposição de estados para processar informações sobre todos os elementos da lista simultaneamente. Através da interferência quântica, ele amplifica a probabilidade de encontrar o estado desejado.

3. **Generalidade:** O algoritmo de Grover pode ser aplicado a uma ampla gama de problemas de busca e otimização, tornando-o uma ferramenta valiosa para a computação quântica.

4. **Demonstração Prática da Superioridade Quântica:** O algoritmo de Grover é um dos poucos algoritmos que demonstram claramente a vantagem da computação quântica sobre os métodos clássicos, mesmo para problemas não relacionados à fatoração de números ou criptografia.

Apesar dessas vantagens, é importante notar que o algoritmo de Grover não fornece uma aceleração exponencial como o algoritmo de Shor para fatoração. No entanto, sua aceleração quadrática é significativa, especialmente para grandes conjuntos de dados, e destaca o potencial único da computação quântica.

5.8 Geração de UUIDs Aleatórios

UUIDs (Universally Unique Identifiers) são identificadores únicos que são amplamente utilizados em sistemas de computação para garantir que cada entidade seja única globalmente. A geração de UUIDs aleatórios desempenha um papel crítico em muitas aplicações, desde a identificação única de registros em bases de dados até o rastreamento de transações em sistemas distribuídos. A natureza única de cada UUID ajuda a evitar colisões, o que é especialmente importante em sistemas com grande volume de dados e alta concorrência. Isso é crucial para manter a integridade dos dados e a consistência em sistemas que operam em larga escala, como bancos de dados distribuídos, sistemas de arquivos e aplicações na nuvem.

Os UUIDs são geralmente compostos por 32 caracteres hexadecimais e são divididos em cinco grupos separados por hifens, seguindo o formato 8-4-4-4-12. Essa estrutura padronizada garante a geração de identificadores que são não apenas únicos, mas também uniformemente distribuídos e imprevisíveis. A geração aleatória de UUIDs é um desafio interessante, especialmente quando consideramos sistemas que exigem um alto grau de aleatoriedade e segurança. Em um cenário ideal, a geração de um UUID deve ser tal que a probabilidade de gerar o mesmo UUID mais de uma vez seja extremamente baixa, garantindo assim a unicidade e a confiabilidade necessárias para operações críticas em sistemas complexos.

5.8.1 Implementação Clássica

```
import random

def gerar_uuid_aleatorio_classico():
    # Gera uma sequencia de 128 bits aleatorios
    random_bits = [random.randint(0, 1) for _ in range(128)]

    # Converte a sequencia de bits em uma string hexadecimal
    hex_string = ''.join(f'{bit:0>4b}' for bit in random_bits)

    # Formata a string hexadecimal para o formato UUID (8-4-4-4-12)
    uuid_format = f'{hex_string[0:8]}-{hex_string[8:12]}-{hex_string[12:1

    return uuid_format

# Exemplo de uso
uuid_aleatorio_classico = gerar_uuid_aleatorio_classico()
print("UUID Aleatorio Classico:", uuid_aleatorio_classico)
```

5.8.2 Implementação em Qiskit

```
from qiskit import QuantumCircuit, Aer, execute
import uuid

def gerar_bits_aleatorios(n):
    qc = QuantumCircuit(n)
```

```
        qc.h(range(n))
        qc.measure_all()
        simulator = Aer.get_backend('qasm_simulator')
        job = execute(qc, simulator, shots=1)
        result = job.result()
        counts = result.get_counts()
        return list(counts.keys())[0]

    def gerar_uuid_aleatorio():
        bits = ''
        for _ in range(16):  # 16 vezes 8 bits para 128 bits
            bits += '{:08b}'.format(int(gerar_bits_aleatorios(8), 2))

        # Converter a string de bits em um UUID hexadecimal
        uuid_hex = '%032x' % int(bits, 2)
        return uuid.UUID(uuid_hex)

    uuid_aleatorio = gerar_uuid_aleatorio()
```

5.8.3 Comparação entre as Implementações Clássica e Quântica

A geração de UUIDs é um processo fundamental em muitos sistemas de computação, e a comparação entre as abordagens clássica e quântica para esta tarefa oferece insights interessantes sobre suas diferenças em termos de lógica, performance e abordagem.

5.8.3.1 Lógica e Método

- **Clássica**: A implementação clássica usa um gerador de números aleatórios (RNG) baseado em algoritmos para criar uma sequência de 128 bits, que é então formatada como um UUID. Este método depende da qualidade do algoritmo RNG para garantir a aleatoriedade e a imprevisibilidade dos UUIDs.

- **Quântica**: A implementação quântica utiliza princípios da mecânica quântica, especificamente a superposição de estados, para gerar bits aleatórios. Cada qubit em um estado de superposição tem uma probabilidade igual de ser medido como 0 ou 1, o que teoricamente oferece uma fonte de aleatoriedade mais pura.

5.8.3.2 Performance e Eficiência

- **Clássica**: A eficiência da implementação clássica depende fortemente do desempenho do algoritmo RNG e da velocidade do processador. Em geral, os RNGs clássicos são bastante rápidos e podem gerar UUIDs rapidamente, adequados para a maioria das aplicações.
- **Quântica**: A implementação quântica, embora inovadora, pode ser mais lenta devido à necessidade de inicializar e medir qubits, bem como à limitação de hardware quântico atual. No entanto, à medida que a tecnologia quântica amadurece, espera-se que a eficiência melhore significativamente.

5.8.3.3 Abordagem e Segurança

- **Clássica**: A segurança e a imprevisibilidade dos UUIDs gerados classicamente dependem da imprevisibilidade do algoritmo RNG. Algoritmos RNG comuns podem ser suscetíveis a previsibilidade em teoria, especialmente se o estado inicial (semente) for conhecido.
- **Quântica**: A aleatoriedade inerente à mecânica quântica oferece uma abordagem potencialmente mais segura para a geração de UUIDs. A natureza imprevisível das medidas quânticas teoricamente torna mais difícil prever ou reproduzir a sequência de bits gerada.

5.8.3.4 Aplicabilidade Prática

- **Clássica**: Sistemas clássicos são amplamente acessíveis e podem ser facilmente implementados em quase qualquer hardware moderno. Esta abordagem é prática e suficiente para a maioria das aplicações atuais.
- **Quântica**: Atualmente, a tecnologia quântica ainda está em desenvolvimento e não é tão acessível quanto a tecnologia clássica. Portanto, a implementação quântica de UUIDs é mais experimental e adequada para pesquisa ou para aplicações onde a segurança extrema é necessária.

5.8.3.5 Conclusão

Enquanto a implementação clássica oferece uma solução prática e rápida para a maioria das aplicações, a abordagem quântica introduz um novo nível de segurança e imprevisibilidade, embora atualmente seja mais teórica e menos acessível. À medida que a computação quântica continua a evoluir, pode-se esperar que suas aplicações em tarefas como a geração de UUIDs se tornem mais viáveis e eficientes.

Considerações Finais e Perspectivas

Este trabalho representou um esforço para compreender o estado da arte da computação quântica utilizando a revisão sistemática e uma estratégia retórica que contrastasse os algoritmos clássicos e quânticos implementados com a ferramente Qiskit. O catapultador inicial, foi a perspectiva de que seria possível visualizar diferenças significativas entre a performance das implementações para além do campo teórico e paradigmático. Isto é, para além das regras e ditâmes do "paradigma quântico" de qubits, registradores e portas quânticas. Dado o fato de que boa parte das atuais ferramentas para a escrita de tais algoritmos estão baseadas nas ferramentas clássicas, vide o fato de Qiskit ser escrita em C++ e não necessitar de um hardware específico, não foi possível atestar diferenças significativas e, quando ocorreram, poderiam muito bem ser atribuídas à aleatoridade de cada execução, visto a pouco distância entre elas.

Endossam esses pontos o pouco volume de bibliotecas com alta aplicabilidade presentes em gerenciadores de pacotes como PyPI [1] e a natureza puramente matemática de boa parte dos artigos coletados e avaliados. Há, no entanto, um movimento mais profícuo que objetiva levar a computação quântica não para a simples otimização do que já é conhecido há décadas pela computação tradicional, mas sim para ramos mais "modernos" como inteligência artificial e criptografia. Este último, cada vez mais necessário à medida que hardwares ficam mais rapidamente potentes com o passar dos anos e trivializam estratégias de força-bruta comumente utilizadas para a quebra de senhas. Em relação ao primeiro, inciativas promissoras como a biblioteca Qiskit Quantum kNN [2] também têm se mostrado como alternativas para assistência (não substituição) às implementações tradicionais.

Outro ponto a se considerar, é o nível de complexidade requerido na transição 1:1 das soluções clássicas para as quânticas, vide a diferença entre as seções 5.7.0.1 e 5.7.0.2. Sendo a popularização fator fundamental para o desenvolvimento de novas tecnologias, especialmente em Engenharia de Sofware, e assumindo que ainda não há ganhos reais na migração, a adição de camadas de abstração entre as atuais soluções e as suas eventuais contrapartes quânticas se faz necessária, dada a facilitação que poderia trazer. Ao migrar código estruturado para, por exemplo, código orientado a objetos, não se faz necessária a apreensão de uma grande gama de novos conceitos; o que não procede com a computação quântica e acaba tornado-a mais árida ao grande público.

[1] Mais informações na documentação oficial
[2] Mais informações no repositório oficial

Referências

AWSCHALOM, D. L. e. N. S. D. D. *Semiconductor Spintronics and Quantum Computation*. [S.l.]: Springer, 2002. ISBN 978-3-642-07577-3. Citado na página 47.

BARAVIERA, A. T.; AMARAL, B.; CUNHA, M. O. T. *Mecânica Quântica para Matemáticos em Formação*. Rio de Janeiro, Brasil: IMPA, 2011. Citado na página 29.

BIOLCHINI, J. et al. Systematic review in software engineering. Rio de Janeiro, Brasil, 2005. Citado 3 vezes nas páginas 17, 29 e 53.

HOWARD, A. *Elementary Linear Algebra*. 11th. ed. USA: Wiley, 2013. ISBN 978-1-118-43441-3. Citado 4 vezes nas páginas 17, 29, 35 e 41.

IEZZI, G. *Fundamentos de Matemática Elementar*. 8. ed. [S.l.]: Saraiva, 2013. v. 6. Citado na página 29.

IMRE, S.; BALáZS, F. *Quantum Computing and Communications*: An engineering approach. 1th. ed. USA: Wiley, 2005. ISBN 978-0470869024. Citado na página 30.

KITCHENHAM, B. A. *Evidence-Based Software Engineering and Systematic Reviews*. Keele University: [s.n.], 2015. Citado 2 vezes nas páginas 29 e 53.

VITOR, C. F. e O. *Física Moderna - Origens Clássicas e Fundamentos Quânticos*. 2th. ed. [S.l.]: LTC, 2016. Citado 2 vezes nas páginas 17 e 47.

YANOFSKY, N. S.; MANNUCCI, M. A. *Quantum Computing for Computer Scientists*. 1th. ed. USA: Cambridge University Press, 2008. ISBN 978-0-521-879965. Citado na página 30.