UNIVERSIDADE FEDERAL DO OESTE DA BAHIA
CENTRO DE CIÊNCIAS EXATAS E DAS TECNOLOGIAS
CURSO DE BACHARELADO EM FÍSICA

Anna Beatriz Macedo de Souza

A Transformada Quântica de Fourier como bloco fundamental da computação quântica

Barreiras, Bahia
13 de Janeiro de 2025

Anna Beatriz Macedo de Souza

A Transformada Quântica de Fourier como bloco fundamental da computação quântica

Monografia apresentado ao curso de bacharelado em Física, como requisito para a aprovação na componente curricular CET0318-Trabalho de Conclusão de Curso III e obtenção do título de bacharel(a) em Física, pelo Centro de Ciências Exatas e das Tecnologias da Universidade Federal do Oeste da Bahia

Universidade Federal do Oeste da Bahia

Centro de Ciências Exatas e das Tecnologias

Curso de Bacharelado em Física

Orientador: Clebson dos Santos Cruz

Barreiras, Bahia

13 de Janeiro de 2025

FICHA CATALOGRÁFICA

F866 Anna Beatriz Macedo de Souza.

A transformada quântica de fourier como bloco fundamental da computação quântica. / Anna Beatriz Macedo de Souza. – 2025

41f.
Orientador: Prof. Dr. Clebson dos Santos Cruz.
Monografia (Graduação) – Bacharelado em Física. Universidade Federal do Oeste da Bahia. Centro das Ciências Exatas e das Tecnologias. Barreiras, BA, 2025.

1. Computação Quântica. 2.Mecânica Quântica. 3.Transformada De Fourier. I. Cruz, Clebson dos Santos. II. Universidade Federal do Oeste da Bahia - Centro das Ciências Exatas e das Tecnologias. III. Título.

CDD 530

Biblioteca Universitária de Barreiras - UFOB

Anna Beatriz Macedo de Souza

A Transformada Quântica de Fourier como bloco fundamental da computação quântica

Monografia apresentado ao curso de bacharelado em Física, como requisito para a aprovação na componente curricular CET0318- Trabalho de Conclusão de Curso III e obtenção do título de bacharel(a) em Física, pelo Centro de Ciências Exatas e das Tecnologias da Universidade Federal do Oeste da Bahia

Trabalho aprovado. Barreiras, Bahia, 13 de Janeiro de 2025:

Documento assinado digitalmente
gov.br
CLEBSON DOS SANTOS CRUZ
Data: 17/01/2025 18:18:18-0300
Verifique em https://validar.iti.gov.br

Clebson dos Santos Cruz
Orientador

Documento assinado digitalmente
gov.br
ELIAS BRITO ALVES JUNIOR
Data: 18/01/2025 05:29:22-0300
Verifique em https://validar.iti.gov.br

Elias Brito Alves Júnior
Convidado 1

Documento assinado digitalmente
gov.br
MURILO SODRE MARQUES
Data: 17/01/2025 17:50:54-0300
Verifique em https://validar.iti.gov.br

Murilo Sodré Marques
Convidado 2

Barreiras, Bahia
13 de Janeiro de 2025

Agradecimentos

Gostaria de agradecer pelo apoio de meus pais, Maria Rosa e Valdivino, que sempre me apoiaram em meus estudos e interesses. Sou grata pelo meu companheiro, Felipe Corado, que me acompanha nessa jornada desde o segundo semestre, me apoiando dentro e fora da universidade. Quero também mostrar minha gratidão por todos os familiares e amigos que me apoiam, tendo destaque minha madrinha Sebastiana e meus amigos da Comunidade, Vitória, Hannah, Adriana e Frankle, além dos demais amigos feitos durante o curso, Ana Clara, Lucas Axel e Fabrício. Quero agradecer também à minha amiga Simone Pereira, que mesmo de longe pôde me ajudar com correções gramaticais no trabalho. Gostaria de agradecer a todos os professores que fizeram parte da minha jornada, principalmente aos que formaram a banca avaliativa, professores Elias Brito e Murilo Sodré, que ajudaram em correções e ajustes para melhora deste trabalho. Por último, mas não menos importante, quero agradecer ao meu orientador Clebson Cruz, que faz parte da minha carreira acadêmica desde minha primeira pesquisa com a transformada de Fourier e seguiu prestando apoio nos demais projetos aos quais me interessei.

Resumo

A Transformada Quântica de Fourier é uma transformação linear aplicada entre qubits, permitindo uma aceleração exponencial de processos computacionais em comparação a algoritmos clássicos. O presente trabalho contém uma análise da computação quântica, em especial das portas que formam o circuito da transformada. A transformada de Fourier, em sua versão clássica, é uma transformação responsável pela mudança no espaço de uma função. Derivadas da TF clássica estão a transformada discreta e a transformada quântica que são utilizadas, respectivamente, em análise de dados eletrônicos e descrevendo sistemas em espaços quânticos. Os resultados mostram o circuito da transformada e a distribuição de estados quando implementada em dois *frameworks* distintos de computação quântica, myQLM e Qiskit. Por fim, são estudados alguns algoritmos que contém a QFT como sub-rotina, conseguindo assim um desempenho melhor que o visto em seus análogos clássicos.

Palavras-chave: Computação Quântica; Mecânica Quântica; Transformada De Fourier;

Abstract

The Quantum Fourier Transform is a linear transformation applied to qubits, allowing for an exponential acceleration of computational processes when compared to classical algorithms. The present work has an analysis of quantum computing, especially the quantum gates present on the circuit of the transform. The Fourier transform, in its classical version, is a transformation responsible for the change of a function's space. Deriving from the classical FT are the discrete and quantum Fourier transforms which are used, respectively, for analyzing electronic data and describing systems in quantum space. Results show the circuit of the transform and the distribution of states when implemented in two distinct quantum computing frameworks, myQLM and Qiskit. Finally, some algorithms that contain the QFT as a subroutine are studied, as they present a better performance than their classical counterparts.

Keywords: Quantum Computing; Quantum Mechanics; Fourier Transform.

Lista de ilustrações

Lista de abreviaturas e siglas

CCET	Centro de Ciências Exatas e das Tecnologias
DFT	*Discrete Fourier Transform*
FFT	*Fast Fourier Transform*
FT	*Fourier Transform*
QFT	*Quantum Fourier Transform*
QPE	*Quantum Phase Estimation*

Sumário

1 Introdução

A evolução computacional das últimas décadas levou a miniaturização de componentes eletrônicos, gerando novos confortos com as tecnologias desenvolvidas. Com o processo de diminuição do tamanho das peças, os computadores passaram a ter maior potência. Porém, tal processo de miniaturização tem suas limitações visto que a influência de efeitos quânticos passa a ser importante para os componentes de escala nanométrica. Neste contexto surge a computação quântica, como uma forma de buscar uma melhora de desempenho se utilizando dos efeitos quânticos que surgem. Com a computação quântica, espera-se alcançar avanços significativos na velocidade de processamento e na capacidade de armazenamento de dados. Essa nova abordagem promete revolucionar a forma como lidamos com problemas complexos na ciência e na tecnologia, já que tem o potencial de resolver problemas os quais seriam impossíveis de serem resolvidos pelos computadores clássicos atuais, como a fatoração de números inteiros grandes em tempo hábil. Com isso, a computação quântica representa uma nova fronteira na área da computação e promete transformar diversos setores da sociedade.

Atualmente, a computação quântica tem sido amplamente investigada por empresas e instituições de pesquisa ao redor do mundo; a expectativa é que, com o avanço nessa tecnologia, sejam alcançadas soluções inovadoras para desafios complexos em diversas áreas do conhecimento. A transformada de Fourier é uma função de transformação linear que modifica o espaço de uma função. Em sua versão quântica é capaz de mudar a base do estado quântico. Neste cenário, a transformada quântica se apresenta como uma ferramenta fundamental na ascensão da computação quântica, por apresentar um desempenho com menor gasto computacional que seu análogo clássico. Neste trabalho, será explorado o potencial da transformada quântica de Fourier em aplicações práticas, visando otimizar processos e solucionar desafios atuais com maior eficiência. A expectativa é que a utilização dessa técnica possa abrir portas para novas descobertas e inovações no campo da computação quântica (NIELSEN; CHUANG, 2001).

Para melhor entendimento de seu funcionamento, este trabalho é organizado com a apresentação das ferramentas utilizadas para a implementação dos algoritmos. Na seção 4.1 é feita uma apresentação das ferramentas Qiskit e myQLM, utilizadas para escrita dos algoritmos, assim como a plataforma IBMQe que permite execução dos algoritmos em um computador quântico real (IBM, 2023a). (ATOS, 2024) (IBM, 2023b) Após isso, em 2 é feita uma introdução à computação quântica em seus princípios mais básicos. Os conceitos de qubit e as portas quânticas são apresentadas. As equações que definem a transformada de Fourier em suas versões clássica, discreta rápida e quântica são mostradas na seção 3. (ARFKEN; WEBER; HARRIS, 2011) Nos resultados parciais (seção 4.2), serão

apresentados os valores obtidos ao implementar o algoritmo da transformada quântica de Fourier, além de uma breve introdução e resultados dos algoritmos de Estimativa de Fase e Fatoração de Shor, utilizando a transformada como sub-rotina. Sendo assim tem-se a consolidação da transformada quântica e sua ampla aplicabilidade em outros algoritmos quânticos importantes.(SHOR, 1994)(COPPERSMITH, 1994)

2 Um novo paradigma da computação

A computação teve seu início com o desenvolvimento dos modelos teóricos de Alan Turing e John von Neumann. O primeiro modelo trata-se de uma formulação onde uma máquina, posteriormente conhecida como máquina de Turing, é capaz de reproduzir processos através de algoritmos de forma universal. [1] Isto é, qualquer máquina de Turing pode obter os mesmos resultados que qualquer outra máquina de Turing por seguir o mesmo procedimento. O segundo modelo, desenvolvido por von Neumann, apresenta a arquitetura com componentes físicos capaz de implementar uma máquina de Turing. A arquitetura proposta é formada por uma central de processamento que se liga a um componente de memória e outro responsável pela entrada e saída de dados. No cerne do componente de processamento, encontra-se a chave do desenvolvimento computacional das últimas décadas.

Os transistores são componentes eletrônicos presentes nos processadores. Feitos com materiais semicondutores, são capazes de mudar e amplificar sinais elétricos. O número de transistores por processador aumentou continuamente como previsto por Gordon Moore em 1965. Tal aumento leva a uma maior capacidade de processamento e melhora nos tempos necessários para cada processo. O avanço da computação se deu então por termos de *hardware*, ou seja, dos componentes físicos que formam as máquinas.

A previsão feita originalmente em 1965 pode ser vista no gráfico da Figura 1, cobrindo apenas a primeira década. Já na Figura 2 está uma análise da quantidade de

[1] O termo algoritmo pode ser inicialmente entendido como um passo-a-passo que não necessariamente se limita a computação. Porém, mais à frente neste trabalho, o termo algoritmo se refere único e exclusivamente a rotinas computacionais executadas em linguagem Python.

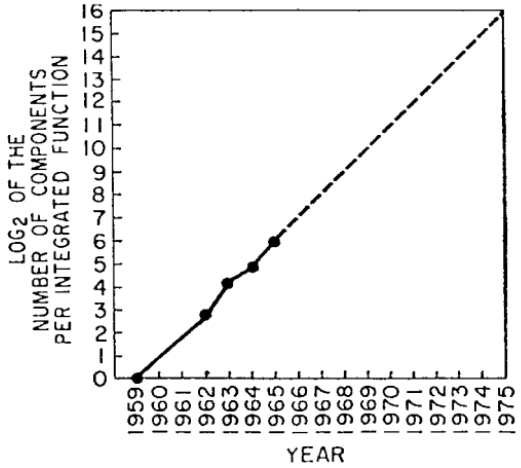


Figura 1 – Número de componentes por função integrada
Fonte: (MOORE, 1965)

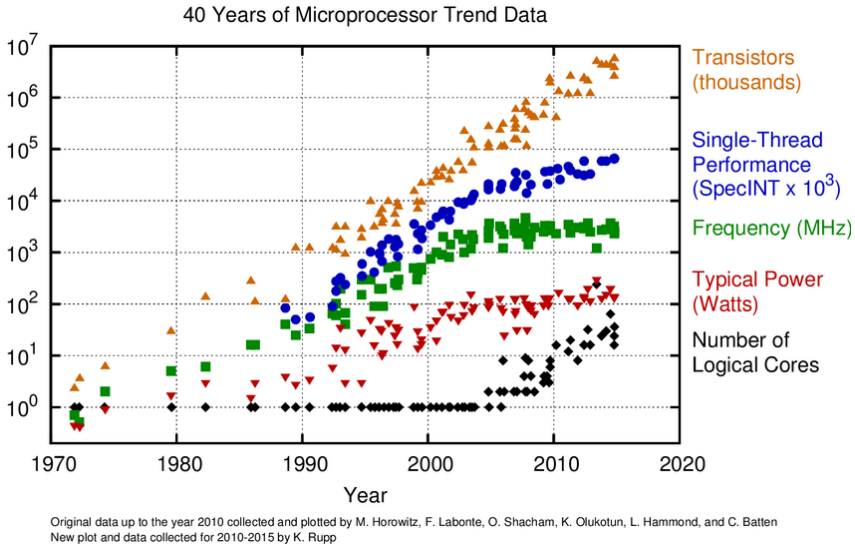


Figura 2 – Análise de componentes e desempenho computacional nos últimos 40 anos. Fonte: (MöLLER; VUIK, 2017)

transistores, performance em *Thread* única, frequência, potência e núcleos lógicos. Para uma melhor comparação com os dados previstos por Moore, pode-se focar apenas dados em amarelo, que mostram a quantidade de transistores.

Há, porém, uma ameaça a tal evolução computacional: com peças cada vez menores existe um limite de transistores que podem ser alocados por processador e, dado a ordem de seu comprimento efeitos quânticos passam a surgir, causando interferência nos sinais eletrônicos. É necessário então buscar meios de conseguir avanço computacional sem uma dependência exclusiva de componentes de *hardware*; uma das soluções propostas é a computação quântica.

2.1 Introdução à Mecânica Quântica

O estudo da matéria e sua estrutura mais fundamental levou a descoberta de partículas e interações observadas apenas em pequena escala. A natureza mais ínfima da matéria é objeto de interesse filosófico desde o século VI a.C., porém apenas descobertas feitas nos últimos cem anos foram capazes de melhor descrever fenômenos quânticos. A partir do desenvolvimento das formulações de dualidade onda-partícula da luz e da matéria, a quantização de diversas propriedades foi teorizada, entre elas o momento angular do elétron, o que possibilitou o estudo ainda mais aprofundado de efeitos de interferência.

Os experimentos feitos por Otto Stern e Walther Gerlach evidenciaram a existência do spin. O experimento consistia em um aparato que emitia um feixe de partículas e aplicava um campo magnético, como representado pela Figura 3. A partir da aplicação do campo, acontece uma divisão entre os feixes, o que só é possível já que cada elétron tem uma característica intrínseca. O conceito de spin muitas vezes gera confusão pois seu nome,

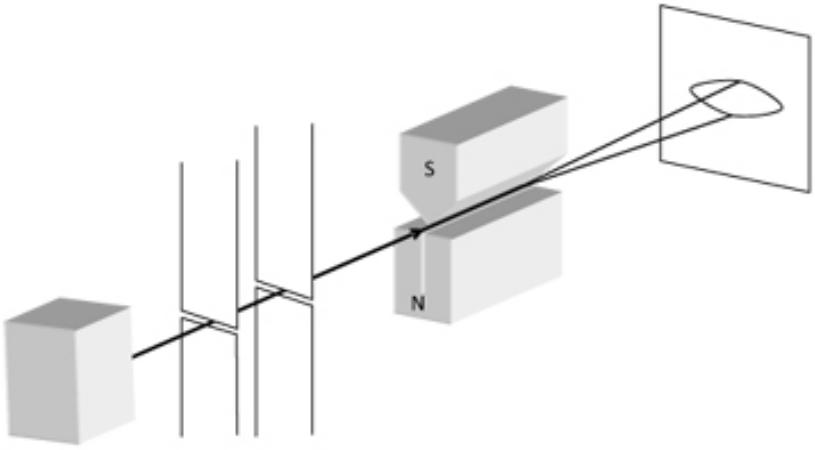


Figura 3 – Esquema representativo do experimento de Stern-Gerlach
Fonte: (GOMES; PIETROCOLA, 2011)

em inglês, dá a ideia de rotação, o que não acontece em uma partícula unidimensional. Pode-se pensar no spin pelo que este é: uma propriedade da matéria, assim como a massa e a carga elétrica. Como resultados do experimento de Stern-Gerlach, foram observados uma relação de dependência com a direção em que é aplicada o campo magnético. Uma das sequências utilizadas no experimento consistia na aplicação de um campo em direção Z, outro em direção X e novamente uma aplicação em direção Z. Como resultado, o feixe sempre retornava a sua configuração inicial. O comportamento dos spins pôde ser descrito a partir das matrizes desenvolvidas por Wolfgang Pauli:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.1}$$

Cada uma das matrizes σ representa uma direção de aplicação do campo magnético e todas são hermitianas, isso é, são iguais a seu transposto conjugado.

2.2 Computação Quântica

A computação quântica surge como um novo paradigma, que busca conseguir avanços ao aplicar efeitos quânticos na computação. Nesta seção a computação quântica será discutida como solução de *software*, fazendo analogias a circuitos e componentes de computação clássica, porém isso só é possível devido ao desenvolvimento de máquinas capazes de representar os sistemas quânticos aqui discutidos. (FEYNMAN, 2018)

Uma das primeiras diferenças entre as formas de computação, clássica e quântica, dá-se a partir da unidade fundamental de processamento que cada uma utiliza. A computação clássica tem o *bit*, termo que vem do inglês e significa parte ou pedaço. O bit é o elemento mais fundamental de contagem na computação e só pode ter valores de 0 e 1, de forma que

uma sequência de pulsos elétricos com um valor do tipo ”Verdadeiro” ou ”Falso” pode ser utilizado para representar processos muito mais sofisticados. Todo dado que percorre um computador tem sua representação física nos impulsos elétricos, que podem ser convertidos a partir da aplicação de portas lógicas. Já a computação quântica conta com os *qubits*, ou bits quânticos, que apesar de apresentarem similaridades com os bits clássicos, têm propriedades puramente quânticas como a superposição.

O comportamento do qubit se diferencia de seu análogo clássico devido ao fato de ser uma representação de estados quânticos. Os qubits $|0\rangle$ e $|1\rangle$ são vetores coluna, que na notação de Dirac são denominados *kets*. Por representar um estado quântico, o qubit pode ser escrito:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.2}$$

Outro análogo que se pode traçar entre as computações clássica e quântica é o uso de portas sucessivas na manipulação da informação formando circuitos. Na computação clássica, os circuitos são representados por fios, que carregam os valores de cada bit.

Na Figura 4 os pontos A, B e C são entradas de dados e o ponto S é a saída após aplicação das portas lógicas.

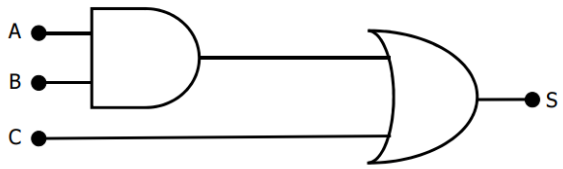


Figura 4 – Circuito de portas clássicas. Fonte: (BARANAUSKAS, 2012)

De modo geral, os circuitos quânticos têm estrutura idêntica, com linhas que representam diferentes qubits e símbolos para cada porta aplicada.

As portas quânticas são representadas por operadores quânticos unitários. Tal propriedade dos operadores faz com que estes sejam reversíveis, isto é, não há necessidade de gasto energético na mudança dos estados. Vale ressaltar que nem todo operador unitário é uma porta quântica, mas toda porta quântica deve ser um operador unitário.

As três portas principais de rotação podem ser representadas pelas matrizes de Pauli.As matrizes de Pauli são três operadores base da mecânica quântica utilizados para explicar os resultados do experimento de Stern-Gerlach. [2]

A seguir estão as descrições das principais portas quânticas, junto às matrizes e suas representações nos circuitos. A porta X é responsável por uma rotação π na direção X.

[2] Mais informações sobre os experimentos de Stern-Gerlach e as matrizes de Pauli podem ser encontradas no apêndice 2.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \text{q — X —} \tag{2.3}$$

A porta Y de forma similar faz uma rotação de π na direção Y.

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \text{q — Y —} \tag{2.4}$$

E analogamente, a porta Z aplica uma rotação de π na direção Z.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \qquad \text{q — Z —} \tag{2.5}$$

A porta Hadamard é a primeira porta exclusivamente quântica utilizada, isto é, não há formas de reproduzir os efeitos desta porta com os mecanismos clássicos já existentes. A porta Hadamard faz a normalização da distribuição de probabilidade entre estados.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \text{q — H —} \tag{2.6}$$

O resultado da aplicação da porta Hadamard (2.7) mostra o efeito de **superposição**. Depois da aplicação da porta, os dois estados se sobrepõem, tendo igual probabilidade de serem encontrados. Esse efeito é um dos responsáveis pelo ganho computacional da computação quântica: um único qubit é capaz de carregar informações sobre dois estados simultaneamente. A aplicação da porta nos estados $|0\rangle$ e $|1\rangle$ gera outros dois estados muito comuns, $|+\rangle$ e $|-\rangle$.

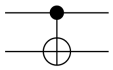
$$\begin{aligned} H|0\rangle &= |+\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ H|1\rangle &= |-\rangle = \tfrac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \end{aligned} \tag{2.7}$$

A porta de fase é representada, tanto em sua matriz quanto em circuitos, pela letra P do inglês *Phase*. Normalmente sua representação é feita com a letra P seguida do valor θ que é aplicado na rotação.

$$P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix} \tag{2.8}$$

Por último, é importante conhecer a porta CNOT, a única porta aplicada obrigatoriamente dois qubits utilizada no circuito. Com sua nomenclatura também vindo do inglês,

Controled NOT, a CNOT é uma porta que executa um "não"controlado, isto é, inverte o valor de um qubit alvo (q_1) com base no valor de um qubit de controle (q_0). Por ser aplicada em dois qubits, é representada por uma matriz 4×4. O primeiro quadrante da matriz representa o qubit de controle; Caso este tenha valor nulo, a porta não tem efeito. Caso q_0 tenha valor $|1\rangle$, então é aplicada uma porta X que está representada pelo quarto quadrante da matriz.

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \text{  } \tag{2.9}$$

Sua representação em circuitos é um pouco diferente das portas já estudadas. O símbolo $\bullet$ é utilizado para sinalizar o qubit de controle e $\oplus$ indica o qubit no qual será feita a inversão; os dois símbolos são conectados por uma linha.

$$\begin{aligned} CNOT\,|00\rangle &= |00\rangle \\ CNOT\,|01\rangle &= |01\rangle \\ CNOT\,|10\rangle &= |11\rangle \\ CNOT\,|11\rangle &= |10\rangle \end{aligned} \tag{2.10}$$

Ao ser aplicada em um estado com superposição, a porta CNOT gera o efeito quântico de **emaranhamento**. No caso de estados emaranhados, qualquer mudança em um dos qubits leva a mudanças no outro, de modo que os estados não possam evoluir de forma independente. Nestas situações, o sistema não pode ser descrito com o uso de produtos tensoriais. O conjunto carrega informação, não as partes que o formam. Vale ressaltar que o efeito de emaranhamento só ocorre para estados em superposição, isto é, na aplicação de uma porta Hadamard seguida por uma porta CNOT.

O emaranhamento pode ser visto no caso de aplicação da CNOT em dois qubits, em um cenário onde foram combinados os qubits $|+\rangle$ e $|0\rangle$. Inicialmente, forma-se o novo estado analisado a partir do produto tensorial:

$$\begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \bigotimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} \tag{2.11}$$

E a partir da aplicação da porta CNOT, tem-se:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle \tag{2.12}$$

O resultado da equação (2.12) evidencia o emaranhamento visto que, na medição do primeiro qubit, o segundo sempre assumirá valor igual, sendo assim impossível obter $|0\rangle$ no segundo qubit caso o primeiro tenha valor $|1\rangle$, e vice-versa. Também não é possível representar o estado final como produto tensorial. Este resultado é também o primeiro estado de Bell. Os estados de Bell são quatro estados representando as configurações de emaranhamento máximo que pode ser obtido entre dois qubits. Os demais estados de Bell se encontram na equação (2.13).

$$\begin{aligned} |\Phi^+\rangle &= \tfrac{|00\rangle+|11\rangle}{\sqrt{2}} \\ |\Phi^-\rangle &= \tfrac{|00\rangle-|11\rangle}{\sqrt{2}} \\ |\Psi^+\rangle &= \tfrac{|01\rangle+|10\rangle}{\sqrt{2}} \\ |\Psi^-\rangle &= \tfrac{|01\rangle-|10\rangle}{\sqrt{2}} \end{aligned} \tag{2.13}$$

O comportamento observado com o emaranhamento é puramente quântico, já que permite a quebra da localidade clássica. Os efeitos de medição entre partículas emaranhadas pode ser observados em objetos mesmo em longas distâncias.

3 As Transformadas de Fourier

Em física matemática há um grupo de transformações lineares que se dão através da integral de uma função núcleo K que, ao gerar a função transformada faz sua mudança de espaço através de um processo de mapeamento. (ARFKEN; WEBER; HARRIS, 2011) De modo geral temos:

$$g(\alpha) = \int_a^b f(t)K(\alpha, t)dt, \tag{3.1}$$

onde a nova função g no espaço de α tem mesma proporcionalidade que a função f no espaço t. Podem ser definidas também as transformadas inversas que desfazem a mudança de espaço. A existência das transformadas diretas e inversas permitem a utilização de tais ferramentas para levar uma função que tem difícil solução em um espaço para outro onde a solução é mais simples, desfazendo a mudança no resultado, obtendo assim a solução no espaço original.

As séries de Fourier foram desenvolvidas em 1822 por Jean-Baptiste Fourier em seus estudos sobre o fluxo de calor na termodinâmica. Elas são utilizadas para representar funções periódicas e contínuas em um certo intervalo a partir de uma soma de senoides. De modo geral, uma função só pode ser descrita em termos de uma série de Fourier se seguir as condições de Dirichlet: função contínua e periódica, onde por períodos exista um número finito de extremos e uma integral convergente. (RILEY; HOBSON; BENCE, 2006)

O efeito das somas ser visto na Figura 5 cujo comportamento é análogo ao da

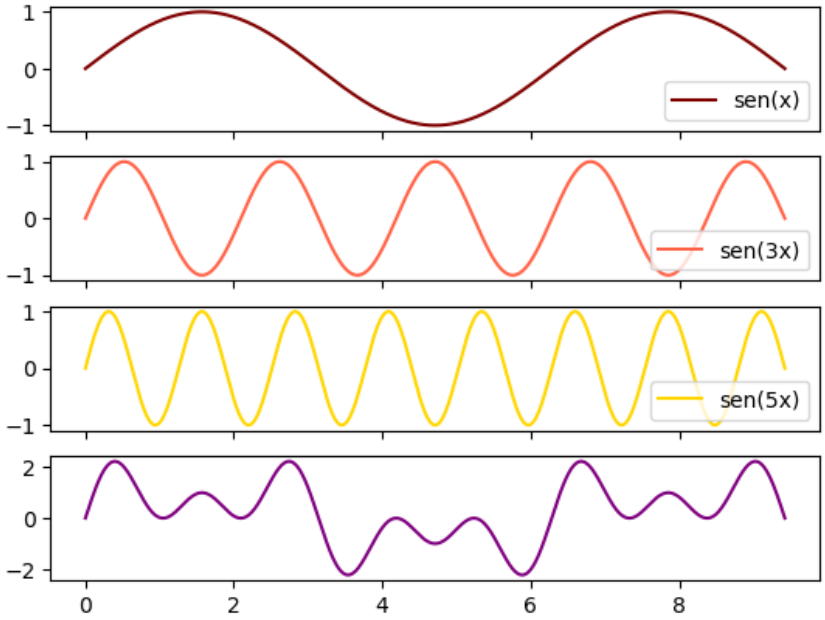


Figura 5 – Função desenvolvida a partir da soma de senoides. Fonte: Autoria própria

interferência de ondas, os picos e vales interagem entre si dando origem a um novo formato. Isso ocorre já que funções do tipo seno e cosseno são capazes de descrever um espaço completo. O efeito das somas também pode ser interpretado num espaço de Euler, com rotação dos vetores, gerando imagens mais complexas. [1]

3.1 A Transformada de Fourier Clássica

A Transformada de Fourier Clássica (FT, do inglês *Fourier Transform*) é uma função integral utilizada na mudança de espaço de um sistema, e surgiu como uma generalização da série de Fourier, decompondo funções como uma soma contínua de ondas harmônicas. Dada pela expressão:

$$F(s) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-i2\pi xs} dx \tag{3.2}$$

Onde o termo $1/\sqrt{2\pi}$ é um termo de normalização das transformadas. (BRACEWELL; KAHN, 1966) (ARFKEN; WEBER; HARRIS, 2011)

A Transformada Inversa de Fourier tem o mesmo termo de normalização e núcleo similar. (WANG, 1984) Devido às propriedades das transformadas direta e inversa, é possível levar a função para um novo espaço e recuperar seu valor, de modo que a transformada inversa de uma transformada dá a função inicial.

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(s) e^{i2\pi xs} ds \tag{3.3}$$

Por ser uma generalização da série, a transformada pode ser aplicada em funções não periódicas, desde que siga as seguintes condições:

- A função tem um número finito de descontinuidades
- A função é absolutamente integrável em todo o espaço

Por ter núcleo exponencial, a FT pode também ser estudada como uma combinação de seno e cosseno. Dada a relação de Euler:

$$e^{\pm X} = cos(X) \pm isen(X)$$

A transformada de Fourier é baseada no núcleo $e^{i\omega t}$ e suas partes real e imaginária tomadas separadamente, cos ωt e sen ωt. Como esses núcleos são as funções usadas para descrever ondas, as transformadas de Fourier aparecem com frequência em estudos de ondas e na extração de informações de ondas, em particular quando estão envolvidas informações de fase. (ARFKEN; WEBER; HARRIS, 2011)

[1] Mais sobre o efeito da série de Fourier para geração de imagens pode ser visto pelo Youtube no canal 3Blue1Brown.

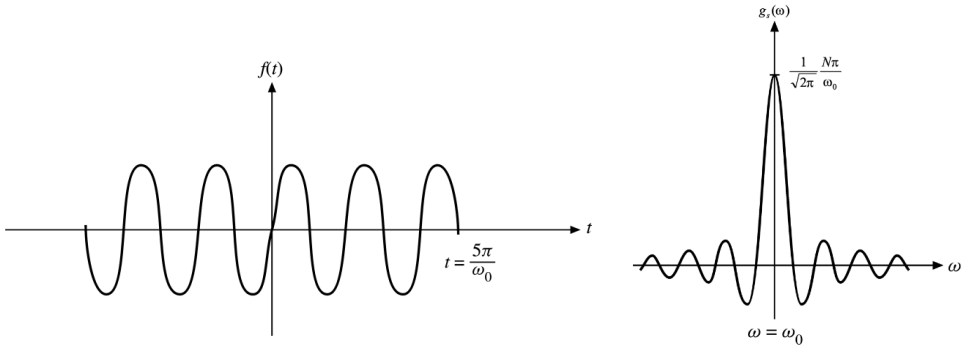


Figura 6 – Trem de onda finito à esquerda e sua respectiva frequência obtida a partir da FT à direita. Fonte: (ARFKEN; WEBER; HARRIS, 2011)

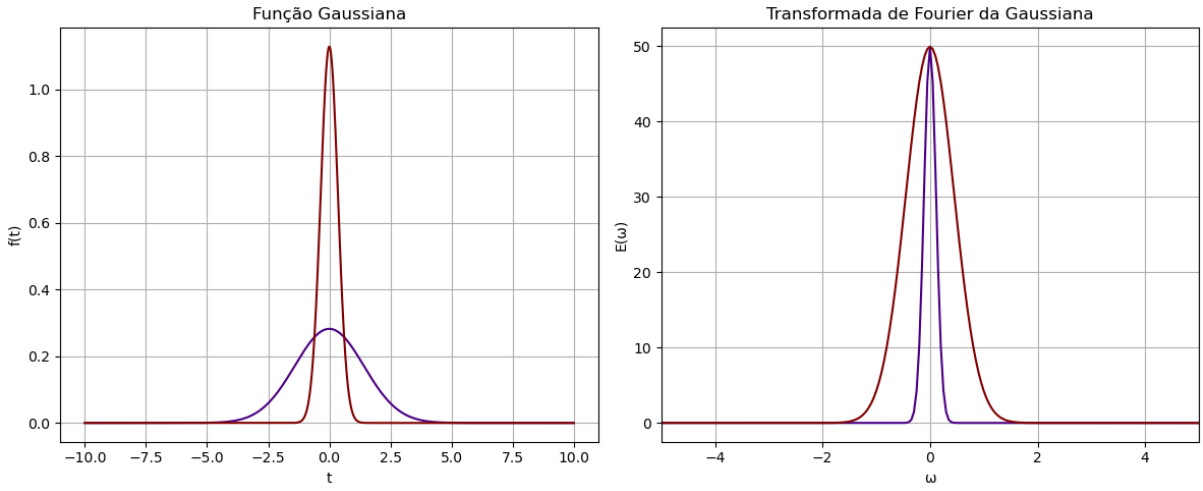


Figura 7 – Gráfico da transformada de Fourier de duas funções Gaussianas distintas. Fonte: Autoria Própria

Na Figura 6 é possível perceber o efeito da aplicação da FT. No gráfico à esquerda está a representação de um trem de ondas finito em termos do tempo. Já à direita há um gráfico de sua respectiva transformada em termos da frequência ω.

Os espaços s e x na FT usualmente correspondem a tempo e frequência. A FT pode ser utilizada como uma ferramenta matemática para estudo do comportamento de diversas incertezas entre grandezas físicas, sendo a incerteza de Heisenberg o exemplo mais comum. Além disso, a FT pode ser entendida em casos de incerteza clássicos, como na observação de pulsos com uma certa frequência depende do tempo em que estes ocorrem. Diz-se então que tais duas grandezas estão numa relação de incerteza. O efeito da incerteza entre as grandezas pode ser visto a partir do achatamento da gaussiana na Figura 7 onde uma função que é originalmente mais achatada tem, na transformada, um pico mais fino.

3.2 A Transformada Discreta

A Transformada de Fourier discreta, ou *Discrete Fourier Transform*(DFT) foi introduzida para estudo de áreas da ciência que dependem de valores discretizados, como o processamento de sinais. Seu comportamento é análogo ao da transformada clássica, porém os dados de entrada e saída são valores pontuais e não uma função, como estudado anteriormente.

A DFT tem uma ordem de $\mathcal{O}(N^2)$ operações, sendo dada pela expressão:

$$Y(m) = \frac{1}{\sqrt{N}} \sum_{v=0}^{N-1} X(v) e^{2\pi i/N \sum v_j m_k 2^{j+k}} \tag{3.4}$$

Por ter uma ordem de crescimento exponencial, é utilizada apenas em casos em que o conjunto de dados analisados seja pequeno, ou onde o grande tempo computacional não seja um problema.

Há uma versão adaptada chamada Transformada Rápida de Fourier, ou FFT do inglês *Fast Fourier Transform*, que utiliza sinais eletrônicos e os transforma entre si, sendo amplamente utilizada em análise de sinais. A FFT é construída de modo que sua complexidade computacional seja $\mathcal{O}(NlogN)$. Por ter desempenho mais rápido que a DFT, a FFT é utilizada em aparelhos como o osciloscópio, onde o tempo de processamento é crítico para a visualização em tempo real do comportamento de uma função. Outras aplicações do FFT e DFT incluem a codificações de áudio e estimativa de espectro de potência, fazendo parte assim de diversas operações de telecomunicação. (SELESNICK; SCHULLER et al., 2001)

3.3 A Transformada Quântica

A transformada quântica de Fourier, ou QFT do inglês *Quantum Fourier Transform*, é um análogo direto da transformada discreta, porém aplicado a estados quânticos. Desenvolvida em 1994 por Don Coppersmith e Peter Shor, ambos pesquisadores da IBM no campo de criptografia e proteção de dados. Enquanto Coppersmith investigava o comportamento da QFT, Shor já estudava seu algoritmo de fatoração de números primos, do qual a transformada faz parte.

A QFT é aplicada em qubits como segue:

$$|a\rangle = \frac{1}{\sqrt{N}} \sum^{N} e^{2\pi i ac/q} |c\rangle \tag{3.5}$$

E assim como no caso clássico, é possível utilizar uma transformada quântica inversa, que desfaz a mudança de estado executada pela primeira. Tal mudança está explicitada na

	DFT	FFT	QFT
2	4	0.602	0.091
5	25	3.49	0.488
10	100	10	1
N	$\mathcal{O}(N^2)$	$\mathcal{O}(NlogN)$	$\mathcal{O}(log^2N)$

Tabela 1 – Ordem computacional das transformadas discreta, rápida e quântica para N elementos.

equação abaixo:

$$|c\rangle = \frac{1}{\sqrt{N}} \sum^{N} e^{-2\pi iac/q} |a\rangle \tag{3.6}$$

A versão quântica da transformada de Fourier tem funcionamento análogo à FFT, porém se aplica a mudança de base dos estados quânticos, e não em sinais digitais. Além de estar adaptada para sistemas quânticos, o que melhor combina com os computadores que estão em desenvolvimento, a QFT tem vantagem por complexidade computacional, o que pode ser visto na Tabela 1. "Enquanto o algoritmo clássico de FFT possui complexidade $\mathcal{O}(NlogN)$, com N correspondendo a quantidade de elementos do vetor de entrada, sua versão quântica é exponencialmente mais rápida, $\mathcal{O}(log^2N)$". (MARQUEZINO, 2006)

É preciso notar também que, apesar da QFT apresentar uma melhora computacional em comparação com a FFT, nem sempre há um ganho com sua execução: alguns problemas têm melhor resultado com o uso de ferramentas quânticas alternativas ou até algoritmos clássicos. O presente estado da computação quântica dificulta maiores comparações entre as transformadas clássica e quântica, já que os dispositivos atuais apresentam taxas de erros que crescem exponencialmente com a quantidade de qubits e portas quânticas utilizados.

Além da aplicação abordada neste trabalho em computação quântica, a QFT é utilizada em outras áreas de pesquisa e desenvolvimento. Na indústria, a QFT foi testada na examinação de materiais de forma que não houvesse perda de informações durante medição. Também foi aplicada na filtragem de imagens, sendo combinada com o algoritmo de codificação Representação Flexível de imagens quânticas, FRQI, do inglês *Flexible Representation of Quantum Images*. Este uso da QFT apresentou um ganho computacional quando combinado ao FRQI. Ao diminuir a profundidade dos circuitos construídos, levou a sucesso no estudo de imagens até 4 vezes maiores que processos quânticos que não utilizam da QFT. (BARTSCH; SCHöBEL, 2021) (KHAN, 2019)

Em estudos mais matemáticos, a aceleração computacional da QFT em relação a FFT é discutida, pensando-se em decomposição matricial e produtos de Kronecker (CAMPS; BEEUMEN; YANG, 2021). Já em aprendizado de máquinas, o uso da QFT na solução do Problema de Subgrupo Oculto inspirou o estudo de inferências para comparação de dados. Tal abordagem leva a avanços em aprendizado de máquina clássico e quântico. (WAKEHAM; SCHULD, 2024)

4 QFT na prática

4.1 Ferramentas utilizadas

4.1.1 myQLM

O myQLM, do inglês *my Quantum Leaning Machine*, é uma ferramenta desenvolvida pela empresa de tecnologia Atos, que permite desenvolvimento de algoritmos quânticos. Seu grande diferencial é oferecer um ambiente de simulação algébrica, focada nos autoestados, sem interferências. Para o desenvolvimento deste trabalho foi utilizada sua versão gratuita que permite a escrita de códigos quânticos com portas customizadas. Há também uma versão paga dessa ferramenta que permite uso de simuladores com ruídos, porém essa versão não foi utilizada, então todos os valores obtidos a partir do myQLM são os valores esperados em teoria, sem variações. (ATOS, 2024)

4.1.1.1 Qiskit

A ferramenta Qiskit, do inglês *Quantum Information Science Kit*, é um kit de desenvolvimento de informação quântica disponibilizado pela empresa IBM. Há um conjunto de softwares que permitem implementação de algoritmos quânticos, criação de circuitos, pós-processamento e aprendizado de máquina. De fácil instalação, o Qiskit conta com diversos tutoriais em sua documentação, além de permitir acesso a plataforma IBMQe que será introduzida mais à frente. O Qiskit funciona tendo como base a linguagem de programação Python e pode ser instalada em diversas IDE's. (IBM, 2023b)

4.1.1.2 IBMQe e máquinas utilizadas

O IBMQe, ou IBM *Quantum Experience* é um programa de acesso de máquinas quânticas reais disponibilizadas pela IBM. A plataforma tem sua versão de acesso gratuito com limitações de tempo de utilização mensal, e máquinas disponíveis. Durante a implementação dos algoritmos presentes neste trabalho, foram utilizadas diversas máquinas da IBMQe. É importante conhecer as estruturas dos computadores utilizados já que isso traz impacto direto aos valores encontrados. As taxas de erro presentes nas Figuras 8 e 9 são menores para os tons mais escuros de azul e roxo, e mais altas para os tons mais claros.

Na Figura 8, tem-se o mapa de erro da máquina quântica de 5 qubits Quito. Esta máquina foi utilizada para a implementação do algoritmo da QFT. Já a Figura 9 contém o mapa de erro para a máquina *Brisbane*, empregada na implementação dos demais algoritmos devido à sua maior quantidade de qubits.

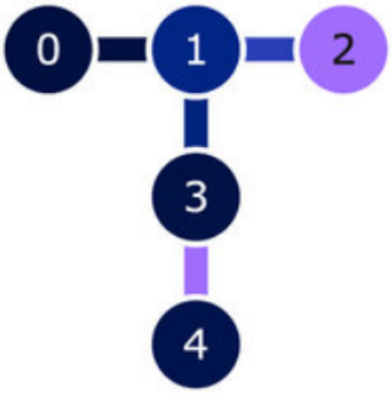


Figura 8 – Arquitetura com taxas de erro da máquina quântica Quito.
Fonte: (IBM, 2023b)

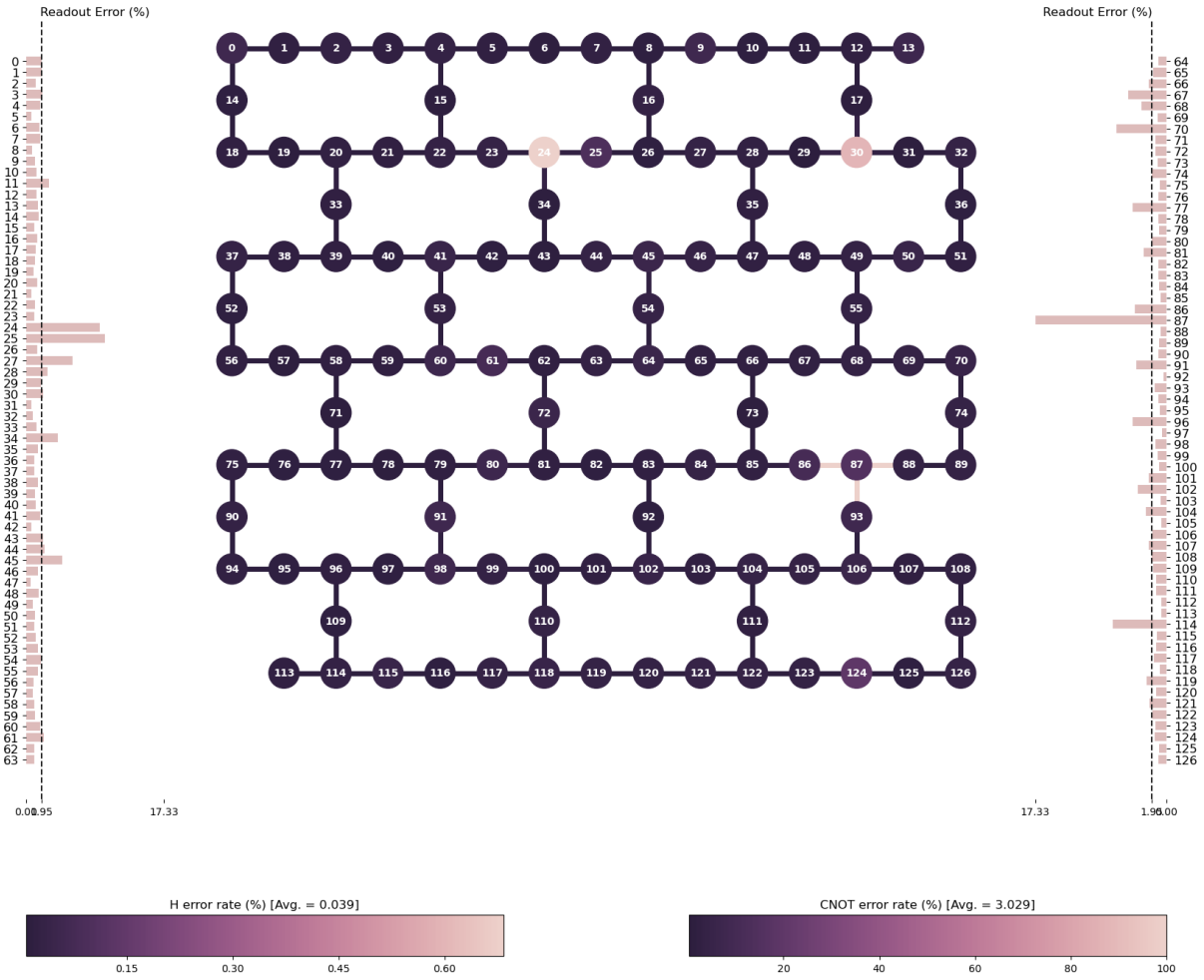


Figura 9 – Arquitetura com taxas de erro da máquina quântica Brisbane.
Fonte: (IBM, 2023b)

4.2 O algoritmo e resultados

Nesta seção estão os resultados relativos a implementação do algoritmo da QFT. O circuito, obtido através de implementação com o Qiskit, assim como as esferas de Bloch, independem do processo de medidas, sendo então iguais em todos os sistemas. Já os histogramas apresentados variam de acordo com a ferramenta utilizada.

A Figura 10 contém o esquema de aplicação das portas que formam o circuito da QFT. Inicialmente é aplicada a porta Hadamard no primeiro qubit, seguida das portas de fase controlada. A fase aplicada é reduzida pela metade a cada qubit de controle utilizado. O mesmo processo se repete para todos os n qubits, até que no último apenas uma porta Hadamard seja executada, sem rotações. Os valores são então invertidos entre si, partindo dos extremos para os mais internos, com o uso da porta Swap. A última etapa é a medida dos valores de cada qubit.

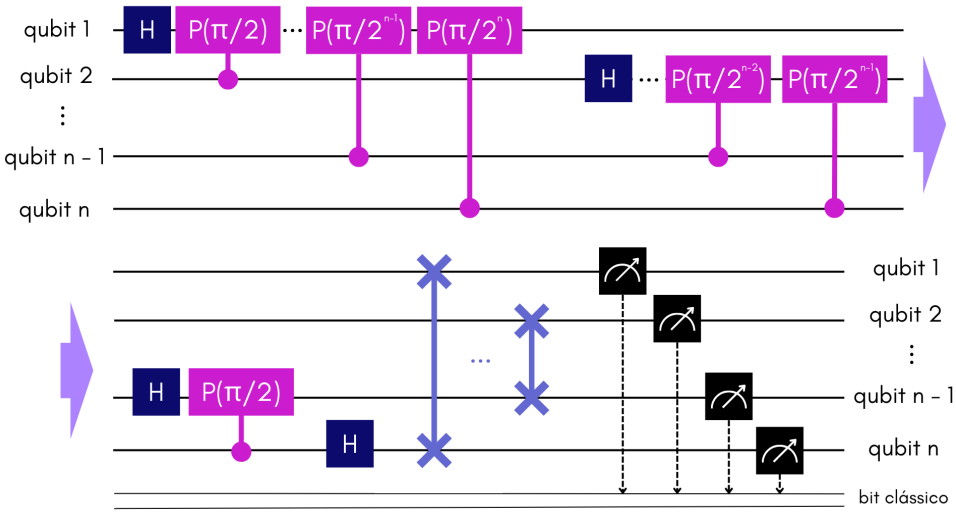


Figura 10 – Esquema do circuito da QFT para n qubits. Fonte: Autoria própria

Na Figura 11 está o circuito encontrado ao fazer a implementação da QFT em um sistema de três qubits utilizando o Qiskit. É importante perceber que por escolha de notação da própria ferramenta os qubits aparecem em ordem invertida, sendo o primeiro qubit o que está mais abaixo no circuito. Com um número ímpar de qubits há a aplicação de uma quantidade ímpar de portas swap, isso porque o qubit central não precisa de ter sua posição modificada.

O resultado da distribuição dos estados em um qubit de valor inicial $|100\rangle$ após a aplicação da QFT utilizando o Qiskit pode ser visto na Figura 12. É possível perceber que os valores assumem valores próximos, o que é esperado já que o algoritmo deve fazer a distribuição das probabilidades entre todos os estados possíveis. Por outro lado, em uma simulação perfeita obtida com a ferramenta myQLM, presente na Figura 13 é possível ver tal distribuição perfeita entre as probabilidades. A diferença entre esses resultados

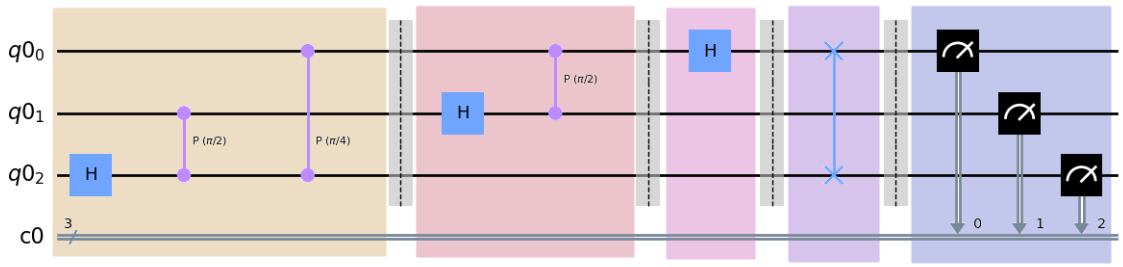


Figura 11 – Circuito do algoritmo da QFT implementado com o Qiskit aplicado em três qubits

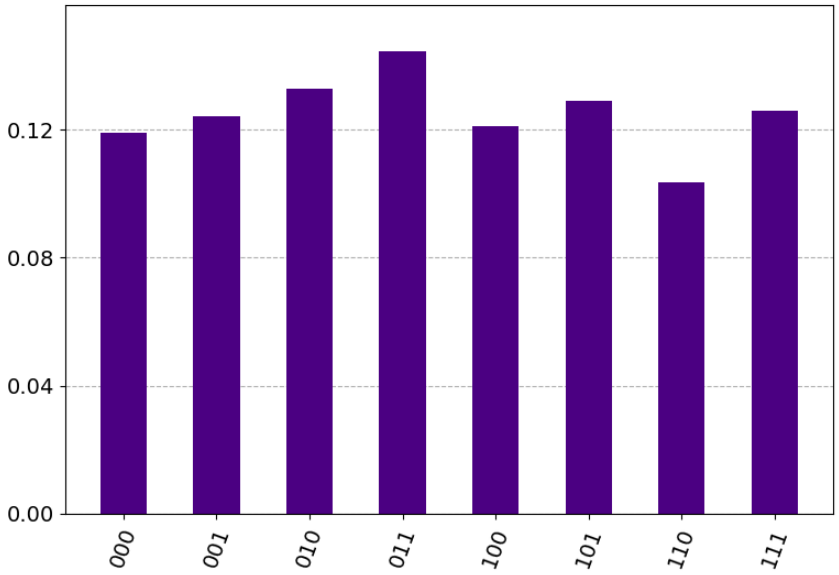


Figura 12 – Histograma das probabilidades, obtido ao aplicar a QFT em uma simulação no estado $|100\rangle$ com uso da ferramenta Qiskit

Figura 13 – Histograma das probabilidades obtido ao aplicar a QFT em uma simulação no estado $|100\rangle$ com uso da ferramenta myQLM

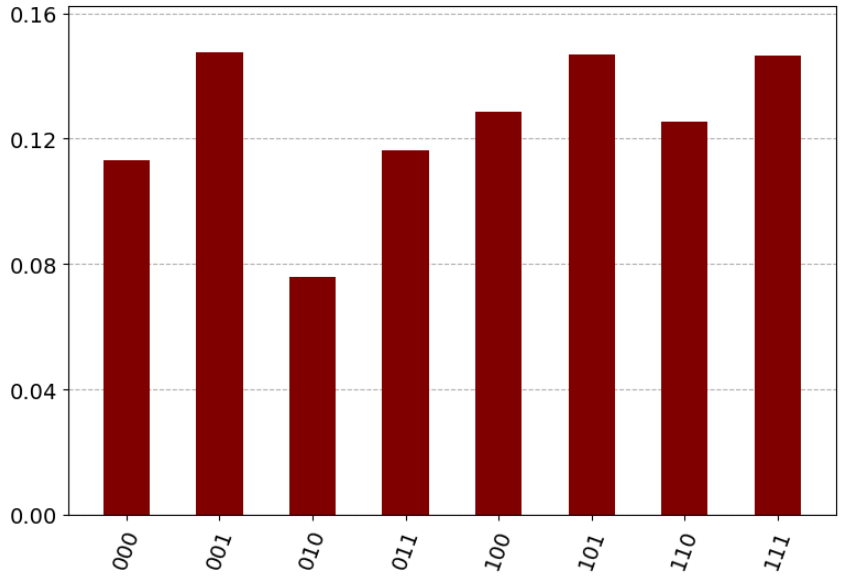


Figura 14 – Histograma contendo a probabilidade dos vetores de estado obtidos com a aplicação da QFT no estado $|100\rangle$ em um computador quântico real de acesso remoto da IBM

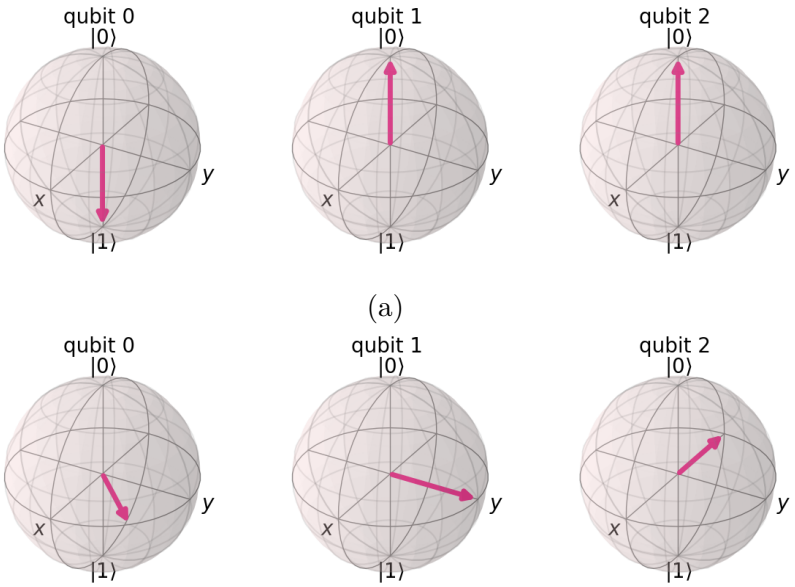
ocorre porque o myQLM tem uma simulação baseada no cálculo do autoestado, que é um processo puramente algébrico, enquanto o Qiskit se baseia na evolução do sistema repetidas vezes, processo que está sujeito a variações estatísticas.

Um resultado similar é encontrado na Figura 14, onde a QFT, também aplicada a um vetor de estado $|100\rangle$, foi executada em um computador quântico real. Neste caso, são encontradas maiores variações entre os vetores de estado medidos, devido ao erro inerente à máquina. É possível perceber assim os efeitos de ruído em uma computador real.

Devido ao comportamento do algoritmo, que como explicado anteriormente, aplica rotações aos qubits, é importante analisar seus efeitos através da esfera de Bloch presente nas Figuras 15 e 16. A esfera de Bloch é gerada em um momento anterior à medição do sistema, então seu valor é o mesmo em simulações quânticas e computadores reais.

Percebe-se que cada qubit de valor inicial $|1\rangle$ gera uma rotação $\pi/2^n$. Na Figura 15 gera rotações de $\pi/4$, $\pi/2$ e π respectivamente. Já a soma das rotações pode ser vista nas esferas da Figura 16 (b). A primeira esfera apresenta a combinação de uma rotação π e $\pi/2$ devido aos dois qubits de valor um no estado inicial $|011\rangle$. Já a segunda esfera contém apenas uma rotação π, enquanto a última está na origem da direção x.

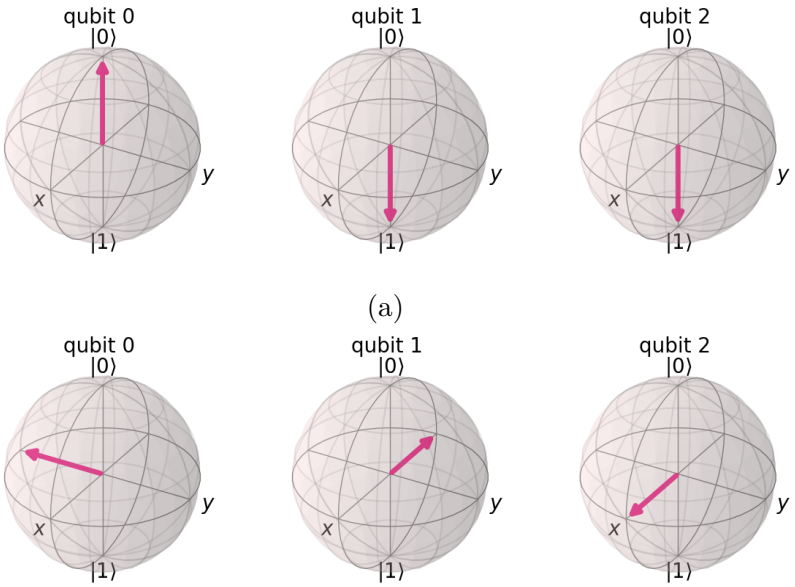
A principal importância da QFT reside não apenas no seu funcionamento, mas também na sua aplicação como sub-rotina em outros algoritmos. A QFT, por si só, é uma ferramenta e não o produto final; por isso, é fundamental conhecer alguns algoritmos que a utilizam e entender como eles operam.



(a)

(b)

Figura 15 – Esferas de Bloch que representam (a) o estado inicial $|100\rangle$ e (b) o vetor de estado após a aplicação da QFT



(a)

(b)

Figura 16 – Esferas de Bloch que representam (a) o estado inicial $|011\rangle$ e (b) o vetor de estado após a aplicação da QFT

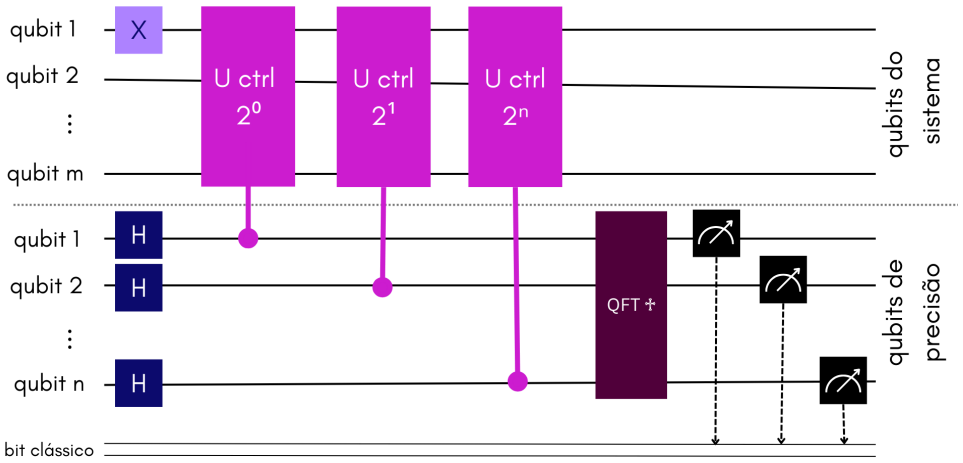


Figura 17 – Circuito do algoritmo QPE

4.3 Aplicações da QFT

Nesta seção serão apresentados alguns dos algoritmos construídos a partir da QFT. É importante salientar que esses algoritmos podem ser sub-rotina de algoritmos maiores, e as aplicações da transformada não se limitam aos exemplos apresentados.

4.3.1 Estimativa de Fase Quântico

O algoritmo de Estimativa de Fase Quântica, ou QPE (do inglês *Quantum Phase Estimation*), calcula a fase θ do autoestado de um operador unitário qualquer $\mathbb{U}$.

$$\mathbb{U}\left|\psi\right\rangle = e^{2i\pi\theta}\left|\psi\right\rangle \tag{4.1}$$

A implementação do algoritmo requer a alocação de dois conjuntos de qubits. O primeiro conjunto tem a função de guardar informações sobre o operador unitário através de seu autovetor ψ. Já o segundo conjunto consiste nos qubits de precisão que, como o nome sugere, fazem com que a fase estimada se aproxime cada vez mais do valor da fase de um operador $\mathbb{U}$.

Nos qubits de precisão é feita a aplicação de uma porta Hadamard, seguida por um operador $\mathbb{U}$ controlado. Tal operador da posição do qubit de controle e age como um oráculo, isto é, seu funcionamento não é conhecido em detalhes, mas sabe-se que tem o comportamento esperado que condiz com a transformação desejada. (COMBARRO; GONZÁLEZ-CASTILLO; MEGLIO, 2023)

O próximo passo na implementação do algoritmo é a aplicação da transformada quântica de Fourier inversa. O circuito com este algoritmo pode ser visto na Figura 17. Como em primeiro momento são aplicadas operações com rotação inerente θ na esfera

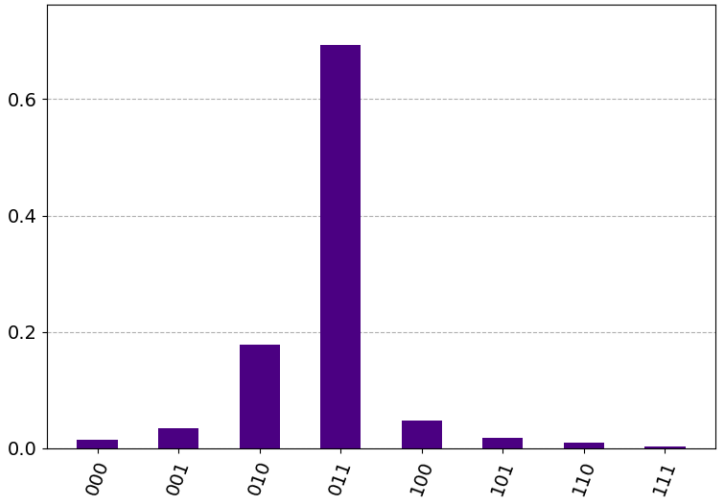


Figura 18 – Histograma das probabilidades obtido com o algoritmo QPE construído com 3 qubits de precisão

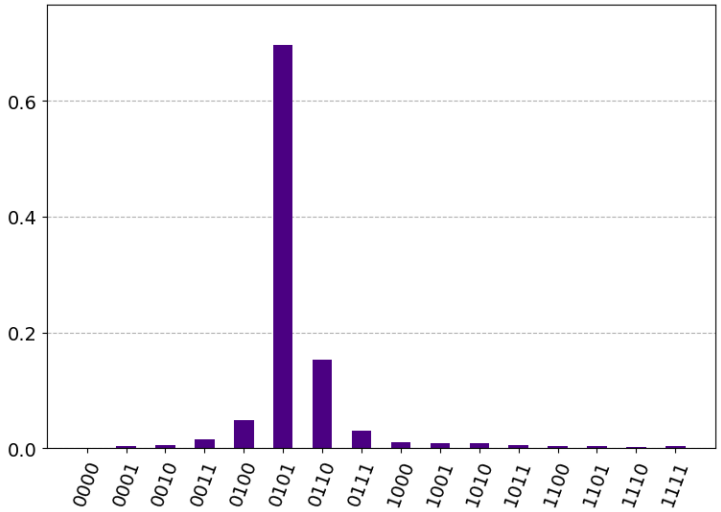


Figura 19 – Histograma das probabilidades obtido com o algoritmo QPE construído com 4 qubits de precisão

de Bloch, é adicionada a IQFT para conversão desses ângulos em qubits. (NIELSEN; CHUANG, 2001)

Os efeitos gerados pela escolha no número de qubits de controle podem ser vistos ao comparar os histogramas em 18 e 19. A Figura 18 contém o histograma obtido ao aplicar a QPE com uma porta de rotação $2\pi/3$ sobre um qubit representando o autoestado $|1\rangle$ em uma simulação utilizando um computador clássico com 3 qubits de precisão. Seu estado de maior destaque foi o qubit $|011\rangle$. Já a Figura 19, obtida com mesmas propriedades e nas mesmas circunstâncias, modificando apenas o número de qubits de precisão, apresentou um resultado diferente, onde o qubit de maior probabilidade foi $|0101\rangle$. O resultado analítico obtido neste caso foi de 0.334. Para comparação pode-se converter os resultados obtidos para 3 e 4 qubits de precisão, sendo eles respectivamente 0.375 e 0.312. Fica evidente assim

que uma maior quantidade de qubits de controle leva a uma maior precisão no algoritmo. A conversão entre números binários e decimais para análise do QPE está na equação (**??**), representando o exemplo da Figura 18.

$$|0.011\rangle \rightarrow \frac{0}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} = 0.375 \tag{4.2}$$

A escolha do número de qubits de controle deve ser feita com cautela, pois um número muito baixo indica uma menor taxa de precisão no algoritmo. No entanto, escolher um número excessivamente alto também gera consequências negativas. Com uma quantidade muito elevada de qubits, os efeitos de ruído e erro inerentes ao processo podem causar uma propagação de erro que afeta as medidas, além de resultar em um gasto computacional acima do necessário.

4.3.2 O algoritmo de Shor

O algoritmo de fatoração de Shor é um dos mais conhecidos na computação quântica. O algoritmo, desenvolvido em 1994 por Peter Shor, é capaz de resolver o problema de fatoração com um ganho operacional. A técnica de Shor consiste em transformar o problema de fatoração em um problema de cálculo de fase, onde é aplicada a sub-rotina QPE. (SHOR, 1994)

Devido ao uso da QFT, o processo de fatoração se torna mais rápido que os métodos clássicos existentes como o GNFS, do inglês *General Number Field Sieve* e o Método *rho* de Pollard. (BUHLER; LENSTRA; POMERANCE, 1993) (BACH, 1991)

A aceleração gerada no processo de fatoração, por sua vez, é importante por impactar o funcionamento da criptografia moderna. Atualmente, a segurança de dados em diversas áreas e ferramentas é feita com a distribuição de chaves públicas e privadas, na qual a chave privada é formada por números primos grandes que não são facilmente fatorados. Se torna então inegável a influência de um processo acelerado de fatoração na segurança digital, com o algoritmo de Shor implementado em uma máquina quântica robusta pode quebrar a segurança de dados na internet.(RIVEST; SHAMIR; ADLEMAN, 1978)

O algoritmo de Shor é divido em três partes principais. Na primeira é feita a preparação do sistema quântico, alocando a quantidade de qubits necessários para execução do QPE e carregando a informação do número que deve ser fatorado. A segunda parte é o próprio circuito do QPE que filtra os estados com a maior probabilidade de terem relação com N. A terceira e última parte é o processamento dos dados encontrados e o cálculo dos fatores a partir deles. Essa etapa de interpretação do resultado é feita a partir de um período, r, obtido como a razão entre o resultado encontrado no processamento e uma potência de dois elevado ao número de qubits de precisão m, ou seja:

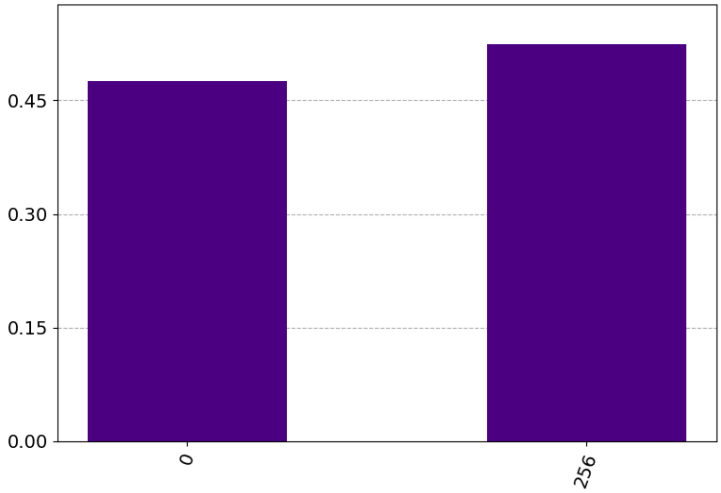


Figura 20 – Resultados das probabilidades para o algoritmo de Shor fatorando N = 21 com chute a = 13 em uma simulação quântica

$$r = \frac{resultado}{2^m} \tag{4.3}$$

O valor do período pode ser posteriormente aplicado na forma de uma exponencial do número chute. De modo que a expressão $a^{r/2} \pm 1$ carrega duas equações, onde cada uma tem um fator em comum com o número desejado.

Todos os testes para o algoritmo de Shor foram feitos com um circuito de 9 qubits de precisão. A influência do número de chute escolhido pode ser visto nas Figuras 20 e 21. A Figura 20 contém o histograma dos estados obtidos para fatoração do número 21 tendo como chute o número 13. Seu resultado é direto, tendo valores apenas para o estado 0 e 256[1]. Calculando r, tem-se então:

$$r = \frac{256}{2^9} = \frac{1}{2} \tag{4.4}$$

e os fatores são:

$$\begin{cases} MDC(a^{r/2} + 1, N) = MDC(14, 21) = 7 \\ MDC(a^{r/2} - 1, N) = MDC(12, 21) = 3 \end{cases} \tag{4.5}$$

Já na Figura 21 está o histograma representando os estados obtidos para a fatoração do número 21 tendo como chute 10. Neste caso, cinco estados surgem com alta probabilidade, todos resultando em pelo menos uma das raízes de 21. Alguns estados, como o 256 encontrado anteriormente, fornecem informações sobre ambas as raízes. O estado 0 que aparece em todos os histogramas é a solução trivial, pois dá a informação que o número

[1] Os valores nos histogramas do algoritmo de Shor vêm da conversão dos qubits em base binária para base decimal, onde cada qubit de valor um corresponde a uma soma de 2^n onde n é a posição de tal qubit.

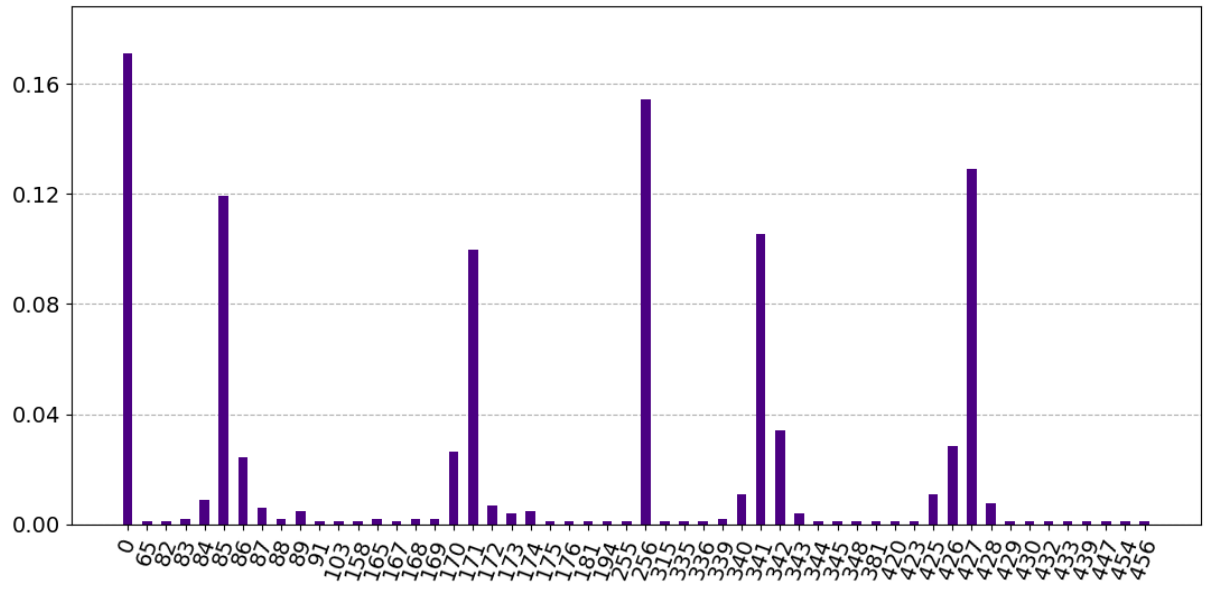


Figura 21 – Histograma das probabilidades do algoritmo de Shor fatorando N = 21 com chute a = 10 em uma simulação quântica

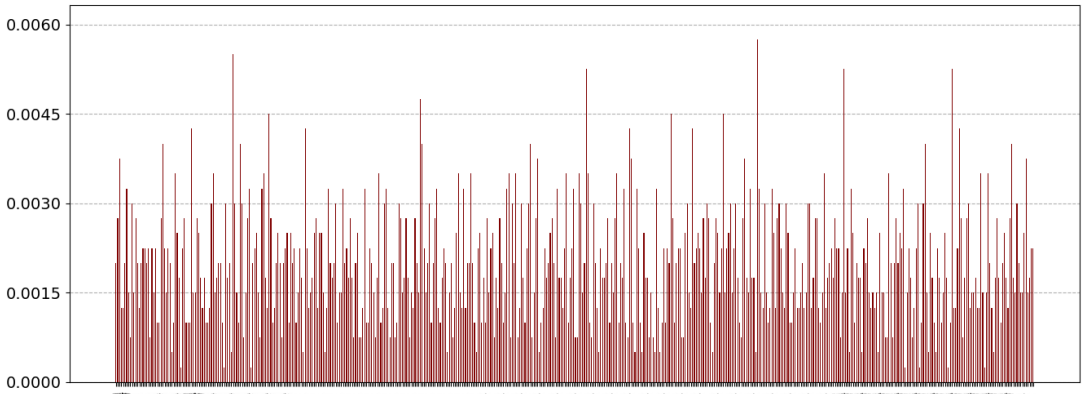


Figura 22 – Histograma das probabilidades após aplicação do algoritmo de Shor para fatoração de N = 21, a partir de um chute a = 13 na máquina quântica Brisbane da IBM

estudado tem 1 como fator. É importante observar que não existe uma fórmula mágica para definir o melhor número de chute nem o melhor resultado para o cálculo das raízes. Por fim, a Figura 22 contém o histograma obtido com a aplicação do algoritmo de Shor no número 21 tendo como chute o número 13 em um computador quântico real. Devido influência de ruído, o algoritmo não performa bem, apresentando uma vasta distribuição entre todos os estados. Com probabilidades abaixo de 1% cada, conclui-se que os computadores atualmente disponíveis não são capazes de executar o algoritmo de Shor com resultados satisfatórios sem uso de ferramentas de redução de ruído.

Como visto nos resultados já discutidos, o algoritmo de Shor, apesar de ser promissor, ainda não pode ser amplamente implementado para fatoração de números grandes por limitações de hardware.

4.3.3 Harrow Hassidim Lloyd

Normalmente referenciado como HHL, o algoritmo desenvolvido pelos pesquisadores Aram Harrow, Avinatan Hassidim e Seth Lloyd é um algoritmo quântico utilizado no cálculo de sistemas de equações lineares. Sendo um dos algoritmos desenvolvidos em tempo mais recente, no ano de 2009, o HHL pode ser aplicado em diversas áreas de física e engenharia. (HARROW; HASSIDIM; LLOYD, 2009)

$$\mathcal{A}\vec{x} = \vec{b} \tag{4.6}$$

A equação (4.6) contém a matriz hermitiana $\mathcal{A}$ e os vetores $\vec{x}$ e $\vec{b}$, que devem ter mesma dimensão. O número de qubits necessário no sistema é proporcional a dimensão dos vetores, de modo que $Dim = 2^n$, onde n é o número de qubits. As amplitudes de $\vec{x}$ e $\vec{b}$ formam a base do sistema.

O algoritmo começa com a preparação do estado, são utilizados três conjuntos de qubit durante o processo:

- b, com o valor inicial do vetor $\vec{b}$
- *clock*, guardando informação das fases
- *ancilla*, auxiliando na verificação da solução

Após a preparação do estado é feita uma estimativa de fase quântica, que evidencia a fase autovalor da porta. Tal fase deve ser proporcional ao autovalor da matriz $\mathcal{A}$ estudada. Com isso, parte da estrutura está pronta, é utilizado um *ancilla qubit*, um qubit descartável, cuja rotação serve de controle das informações de autovalor do sistema. Os *ancilla* qubit podem ser medidos para validação do cálculo, já que este adquire valor $|1\rangle$ no colapso do sistema. Por fim é feita a aplicação de uma QPE inversa para que o resultado obtido fique na base correta do sistema, e não na base temporária criada na aplicação da QPE. (ZAMAN; MORRELL; WONG, 2023)

Para melhor compreensão do funcionamento do HHL, é possível analisar sua performance no seguinte problema:

$$\begin{pmatrix} 1 & -\frac{1}{3} \\ -\frac{1}{3} & 1 \end{pmatrix} \vec{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \tag{4.7}$$

Com esse sistema relativamente simples, encontra-se os valores de $x_0 = 9/8$ e $x_1 = 1/8$, o que evidencia uma proporcionalidade $9:1$ entre os resultados. Na Figura 23 o estado $|11\rangle$ representa x_1 enquanto o valor de x_0 está presente em $|01\rangle$. A proporcionalidade

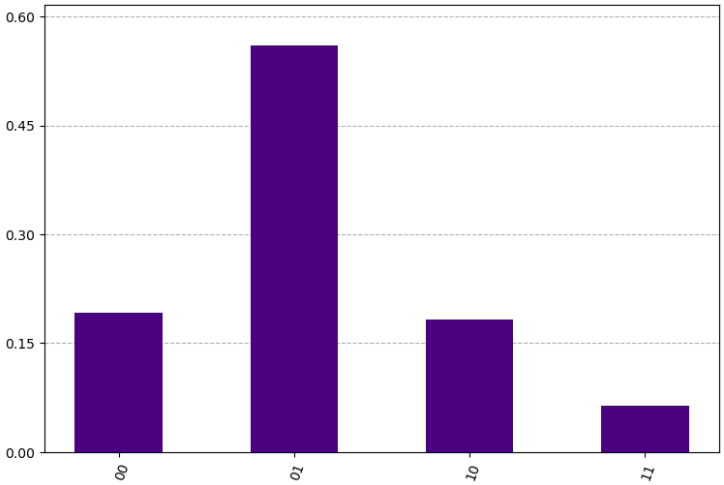


Figura 23 – Histograma das probabilidades medidas no HHL para o problema proposto.

calculada entre esses estados foi de aproximadamente 8.78, o que condiz com o resultado analítico.

Todos os algoritmos apresentados tem ganho computacional quando comparados a algoritmos clássicos que desempenham mesma função. A aceleração exponencial gerada pela redução de qubits de entrada da QFT permite que os algoritmos vistos tenham bom desempenho. Há além destes, outros algoritmos que utilizam desta aceleração como o algoritmo de contagem quântica e o de distribuição de chaves. O último, que apesar de ser formulado para a comunicação quântica sem uso da QFT pode ser adaptado, apresentando melhoras com após sua aplicação.

5 Conclusão

No presente trabalho foi apresentado o circuito formado pela Transformada Quântica de Fourier, assim como as probabilidades obtidas após aplicação de tal circuito em ferramentas como o Qiskit, myQLM e IBM-Qe. Foram abordados também algoritmos derivados da QFT, como o de estimativa de fase quântico, o de fatoração desenvolvido por Shor e o de resolução de sistemas de equações lineares. Tais algoritmos têm grande destaque na área de informação quântica, e só apresentam ganho computacional graças ao uso da QFT como sub-rotina. Os resultados apresentados evidenciam assim o funcionamento da transformada, mostrando o ganho computacional obtido com o uso desta. Perspectivas futuras incluem o estudo de demais algoritmos que utilizando a QFT como sub-rotina, como os algoritmos de contagem quântica, assim como sua aplicação em distribuição de chaves, na área de comunicação quântica. Desta forma, a construção da QFT feita neste trabalho, desde suas origens clássicas na Física Matemática até sua implementação em computadores quânticos, mostrou sua importância na computação quântica como uma área emergente da tecnologia.

Referências

ARFKEN, G. B.; WEBER, H. J.; HARRIS, F. E. *Mathematical methods for physicists: a comprehensive guide*. [S.l.]: Academic press, 2011. 7, 11, 20, 21, 22

ATOS. *myQLM - Quantum Computing Framework - Atos*. 2024. Disponível em: <https://atos.net/en/lp/myqlm>. 11, 25

BACH, E. Toward a theory of pollard's rho method. *Information and Computation*, Elsevier, v. 90, n. 2, p. 139–155, 1991. 33

BARANAUSKAS, J. A. Funções lógicas e portas lógicas. *Aula)–Departamento de Computação e Matemática–USP. Ribeirão Preto–SP*, 2012. 7, 16

BARTSCH, M. K. V.; SCHöBEL, A. *Quantum Fourier Transformation in Industrial Applications*. 2021. Disponível em: <https://ercim-news.ercim.eu/en128/special/quantum-fourier-transformation-in-industrial-applications>. 24

BRACEWELL, R.; KAHN, P. B. The fourier transform and its applications. *American Journal of Physics*, American Association of Physics Teachers, v. 34, n. 8, p. 712–712, 1966. 21

BUHLER, J. P.; LENSTRA, H. W.; POMERANCE, C. Factoring integers with the number field sieve. In: SPRINGER. *The development of the number field sieve*. [S.l.], 1993. p. 50–94. 33

CAMPS, D.; BEEUMEN, R. V.; YANG, C. Quantum fourier transform revisited. *Numerical Linear Algebra with Applications*, Wiley Online Library, v. 28, n. 1, p. e2331, 2021. 24

COMBARRO, E. F.; GONZÁLEZ-CASTILLO, S.; MEGLIO, A. D. *A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-on Approach to Modern Quantum Algorithms*. [S.l.]: Packt Publishing Ltd, 2023. 31

COPPERSMITH, D. An approximate fourier transform useful in quantum factoring. *arXiv preprint quant-ph/0201067*, 1994. 12

FEYNMAN, R. P. Simulating physics with computers. In: *Feynman and computation*. [S.l.]: cRc Press, 2018. p. 133–153. 15

GOMES, G. G.; PIETROCOLA, M. The stern-gerlach experiment and the electron spin: a quasi history example. *Revista Brasileira de Ensino de Física*, SciELO Brasil, v. 33, p. 2604, 2011. 7, 15

HARROW, A. W.; HASSIDIM, A.; LLOYD, S. Quantum algorithm for linear systems of equations. *Physical review letters*, APS, v. 103, n. 15, p. 150502, 2009. 36

IBM. *QFT*. 2023. Disponível em: <https://qiskit.org/documentation/stubs/qiskit.circuit.library.QFT.html>. 11

IBM, Q. *Qiskit 0.45 documentation*. 2023. Disponível em: <https://qiskit.org/documentation/>. 7, 11, 25, 26

KHAN, R. A. An improved flexible representation of quantum images. *Quantum Information Processing*, Springer, v. 18, p. 1–19, 2019. 24

MARQUEZINO, F. de L. *A transformada de fourier quântica aproximada e sua simulação*. Tese (Doutorado) — Master's thesis, Laboratório Nacional de Computação Científica, 2006. 24

MOORE, G. Cramming more components onto integrated circuits (1965). 1965. 7, 13

MöLLER, M.; VUIK, C. On the impact of quantum computing technology on future developments in high-performance scientific computing. *Ethics and Information Technology*, v. 19, 12 2017. 7, 14

NIELSEN, M. A.; CHUANG, I. L. Quantum computation and quantum information. *Phys. Today*, v. 54, n. 2, p. 60, 2001. 11, 32

RILEY, K. F.; HOBSON, M. P.; BENCE, S. J. *Mathematical methods for physics and engineering: a comprehensive guide*. [S.l.]: Cambridge university press, 2006. 20

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, ACM New York, NY, USA, v. 21, n. 2, p. 120–126, 1978. 33

SELESNICK, I. W.; SCHULLER, G. et al. The discrete fourier transform. *The transform and data compression handbook*, CRC Press LLC Boca Raton, p. 37–74, 2001. 23

SHOR, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In: IEEE. *Proceedings 35th annual symposium on foundations of computer science*. [S.l.], 1994. p. 124–134. 12, 33

WAKEHAM, D.; SCHULD, M. Inference, interference and invariance: How the quantum fourier transform can help to learn from data. *arXiv preprint arXiv:2409.00172*, 2024. 24

WANG, Z. Fast algorithms for the discrete w transform and for the discrete fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, IEEE, v. 32, n. 4, p. 803–816, 1984. 21

ZAMAN, A.; MORRELL, H. J.; WONG, H. Y. A step-by-step hhl algorithm walkthrough to enhance understanding of critical quantum computing concepts. *IEEE Access*, IEEE, 2023. 36