Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas

Introdução à Computação Quântica: Conceitos e Comparações com a Computação Clássica

Breno Arthur Rotte Fernandes Oliveira

TRABALHO DE CONCLUSÃO DE CURSO

ORIENTAÇÃO:
Luiz Carlos Bambirra Torres

Fevereiro, 2026
João Monlevade–MG

Breno Arthur Rotte Fernandes Oliveira

Introdução à Computação Quântica: Conceitos e Comparações com a Computação Clássica

Orientador: Luiz Carlos Bambirra Torres

Monografia apresentada ao curso de Engenharia de Computação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina "Trabalho de Conclusão de Curso II".

Universidade Federal de Ouro Preto

João Monlevade

Fevereiro de 2026

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

O482i Oliveira, Breno Arthur Rotte Fernandes.
Introdução à Computação Quântica [manuscrito]: conceitos e comparações com a computação clássica. / Breno Arthur Rotte Fernandes Oliveira. - 2026.
55 f.: il.: color.. + Histograma.

Orientador: Prof. Dr. Luiz Carlos Bambirra Torres.
Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia de Computação .

1. Algoritmos computacionais. 2. Computação. 3. Computadores quânticos - Desenvolvimento. 4. Computação quântica. 5. Sistemas de computação. I. Torres, Luiz Carlos Bambirra. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004:530.145

Bibliotecário(a) Responsável: Flavia Reis - CRB6/2431

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E APLICADAS
DEPARTAMENTO DE COMPUTACAO E SISTEMAS

FOLHA DE APROVAÇÃO

Breno Arthur Rotte Fernandes Oliveira

Introdução à Computação Quântica: Conceitos e Comparações com a Computação Clássica

Monografia apresentada ao Curso de Engenharia de Computação da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação

Aprovada em 25 de fevereiro de 2026

Membros da banca

Doutor Luiz Carlos Bambirra Torres - Orientador (Universidade Federal de Ouro Preto)
Doutora Gilda Aparecida de Assis - (Universidade Federal de Ouro Preto)
Doutor Eduardo da Silva Ribeiro - (Universidade Federal de Ouro Preto)

Luiz Carlos Bambirra Torres, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 11/03/2026

Documento assinado eletronicamente por **Luiz Carlos Bambirra Torres**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 11/03/2026, às 12:21, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1073331** e o código CRC **7E32E973**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3808-0819 - www.ufop.br

Dedico este trabalho a Deus, que me sustentou em cada etapa, à minha família, pelo amor, apoio e incentivo constantes, e aos meus amigos, que acreditaram em mim ao longo dessa jornada.

Agradecimentos

Em primeiro lugar, toda honra e toda glória a Deus. Se cheguei até aqui, foi porque Ele me sustentou em cada detalhe, em cada desafio, em cada momento em que pensei que não ia conseguir. Esta conquista é fruto de fé, disciplina e da certeza de que nenhum esforço é em vão quando existe propósito.

A trajetória até a formatura foi marcada por noites longas, renúncias, cansaço, dúvidas e também por inúmeras pequenas vitórias que me fizeram continuar. Aprendi que o processo transforma muito mais do que o resultado final, e que crescer dói, mas vale a pena.

Carrego comigo a frase que guiou meus passos: "A obsessão vence o talento natural quando o talento natural não é obcecado."Tudo o que conquistei veio da constância, do foco e da determinação de melhorar um pouco todos os dias, mesmo quando ninguém estava vendo.

Sou grato a Deus, à força que Ele me deu, aos meus pais, César e Eliane, ao meu irmão Augusto e aos amigos que estiveram ao meu lado acreditando em mim, torcendo, apoiando e acompanhando essa caminhada de perto ou de longe. Cada um deles tornou essa jornada mais leve, me ajudando nos momentos difíceis e fazendo parte de cada etapa desta conquista.

Hoje encerro um capítulo da minha história com orgulho do caminho percorrido e com a certeza de que ainda há muito mais pela frente. Esta vitória é apenas o começo de tudo o que ainda vou construir.

"O sucesso consiste em ir de fracasso em fracasso sem perder o entusiasmo."

— Winston Churchill (1874 – 1965).

Resumo

Este trabalho apresenta uma análise teórica e comparativa entre os paradigmas da computação clássica e da computação quântica, buscando destacar as diferenças fundamentais em suas unidades de informação, lógicas de processamento e arquiteturas de hardware, além de explorar o ecossistema de software quântico. O estudo mostra a transição do bit clássico para o qubit e detalha tecnologias como circuitos supercondutores e íons aprisionados em contraste com os tradicionais transistores de silício. São analisados algoritmos estratégicos, especificamente os de Shor e Grover, demonstrando como a lógica quântica oferece ganhos de eficiência em comparação com as abordagens convencionais. Além disso, o trabalho discute os desafios da atual era NISQ (*Noisy Intermediate-Scale Quantum*), abordando como o ruído ambiental causa interferências que levam à decoerência dos qubits, o que exige avanços em técnicas de correção de erros quânticos para a construção de sistemas estáveis. Para complementar a fundamentação teórica, foi desenvolvido um estudo de caso prático utilizando o *framework* Qiskit, no qual a implementação do Algoritmo de Grover em um simulador permitiu validar o uso do oráculo e do amplificador de amplitude na resolução de problemas de busca. Os resultados demonstram que, embora a computação quântica ainda enfrente limitações físicas, futuramente ela poderá superar a computação clássica em tarefas específicas, resolvendo problemas considerados intratáveis no modelo tradicional.

Palavras-chaves: Computação Quântica. Computação Clássica. Algoritmo de Grover. Era NISQ. Ecossistema de Software Quântico. Qiskit.

Abstract

This work presents a theoretical and comparative analysis between classical and quantum computing paradigms, seeking to highlight the fundamental differences in their information units, processing logic, and hardware architectures, in addition to exploring the quantum software ecosystem. The study shows the transition from classical bits to qubits and details technologies such as superconducting circuits and trapped ions in contrast to traditional silicon transistors. Strategic algorithms are analyzed, specifically Shor's and Grover's, demonstrating how quantum logic offers efficiency gains compared to conventional approaches. Furthermore, the work discusses the challenges of the current NISQ (Noisy Intermediate-Scale Quantum) era, addressing how environmental noise causes interference leading to qubit decoherence, which requires advances in quantum error correction techniques for building stable systems. To complement the theoretical foundation, a practical case study was developed using the Qiskit framework, in which the implementation of Grover's Algorithm in a simulator allowed the validation of the use of the oracle and the amplitude amplifier in solving search problems. The results demonstrate that, although quantum computing still faces physical limitations, in the future it may outperform classical computing in specific tasks, solving problems considered intractable in the traditional model.

Key-words: Quantum Computing. Classical Computing. Grover's Algorithm. NISQ Era. Quantum Software Ecosystem. Qiskit.

Lista de ilustrações

Lista de abreviaturas e siglas

ADC	Conversor Analógico–Digital
DAC	Conversor Digital–Analógico
GPU	Unidade de Processamento Gráfico
IBM	International Business Machines
NISQ	*Noisy Intermediate-Scale Quantum*
PQC	Criptografia Pós-Quântica
QEC	Correção de Erros Quânticos
QFT	Transformada de Fourier Quântica
QKD	Distribuição de Chaves Quânticas
QVM	Máquina Virtual Quântica
RAM	Memória de Acesso Aleatório
RSA	Rivest-Shamir-Adleman
SDK	Kit de Desenvolvimento de Software

Lista de símbolos

$\|0\rangle$	Estado base zero do qubit
$\|1\rangle$	Estado base um do qubit
$\|\psi\rangle$	Estado geral de um qubit
α, β	Amplitudes complexas de probabilidade
$\|\alpha\|^2$	Probabilidade de medir o estado $\|0\rangle$
$\|\beta\|^2$	Probabilidade de medir o estado $\|1\rangle$
H	Porta Hadamard
X	Porta Pauli-X
Z	Porta Pauli-Z
CNOT	Porta Controlled-NOT
CZ	Porta Controlled-Z
M	Operação de Medição
n	Número de qubits
2^n	Número de estados possíveis
$\|q_n \dots q_1 q_0\rangle$	Ordem do sistema com n qubits
q_0, q_1	Índices dos qubits
j	Unidade imaginária
c_0, c_1	Bits clássicos de medição
N	Número total de elementos no espaço de busca
$\sqrt{N}$	Custo da busca no algoritmo de Grover

Sumário

1 Introdução

A computação sempre foi marcada por avanços constantes, mas agora está passando por um momento decisivo, pois após décadas sustentada pelo paradigma clássico, ela começa a dar sinais de que sua evolução tradicional está chegando ao limite. Com o passar do tempo, esse modelo impulsionou transformações que vão desde a automação industrial até a criação de grandes redes de comunicação que conectam o mundo, moldando praticamente todos os setores da sociedade moderna. Porém, à medida que a miniaturização dos transistores alcança escalas atômicas, é evidente que esse modelo já não oferece o mesmo espaço para crescimento e isso abre caminho para um novo salto tecnológico que rompe com as limitações atuais (NOFER et al., 2023). Esse salto é representado pela computação quântica, que introduz uma forma completamente diferente de representar e processar informação, baseada em princípios que não existem no mundo clássico (RIETSCHE et al., 2022).

Ao contrário da evolução tradicional dos processadores clássicos, que ocorreu por meio de melhorias graduais, como o aumento das frequências de operação ou a adição de múltiplos núcleos, a computação quântica segue um caminho totalmente diferente. Ela utiliza princípios da mecânica quântica para resolver problemas que são, por natureza, intratáveis para a computação clássica (GILL; BUYYA, 2024). Seu potencial vai muito além de tornar cálculos mais rápidos, pois ela pode ajudar a modelar fenômenos naturais complexos, otimizar sistemas logísticos em larga escala e transformar a forma como protegemos dados na internet, impactando diretamente a segurança digital global (NANDHINI; SINGH; AKASH, 2022; SODIYA et al., 2024).

Essa mudança de paradigma vem acompanhada de desafios consideráveis, porque construir um computador quântico funcional e escalável é uma das tarefas mais complexas da ciência moderna. Para que isso seja possível, é necessário controlar partículas extremamente sensíveis e reduzir ao máximo a interferência do ambiente, que pode destruir a informação processada (BARRAL et al., 2025). Nesse cenário, a ciência da computação desempenha um papel fundamental, desenvolvendo os algoritmos, as arquiteturas de software e as ferramentas de simulação necessárias para tornar essa tecnologia acessível e útil (BASAK UPAMA et al., 2022).

Este trabalho busca apresentar de forma clara essa tecnologia em crescimento, explicando os fundamentos que diferenciam a computação quântica da computação clássica, discutindo as arquiteturas de hardware que estão sendo desenvolvidas para tornar esse paradigma viável e analisando os algoritmos que demonstram vantagens específicas sobre abordagens tradicionais. Será tratada a realidade prática da área, marcada pela necessidade

de recorrer à simulação em sistemas clássicos para validar hipóteses e desenvolver soluções, dado que o hardware quântico disponível ainda opera sob condições ruidosas e limitadas (CICERO et al., 2025). O objetivo é oferecer uma visão consolidada do estado da arte e das perspectivas futuras para a área.

1.1 Contextualização

A história da computação quântica não surgiu de forma repentina, mas como o resultado de uma construção teórica que se desenvolveu ao longo de muitos anos. O conceito começou a tomar forma no início da década de 1980, quando Paul Benioff propôs o primeiro modelo teórico de uma Máquina de Turing que operasse de acordo com os princípios da mecânica quântica (BASAK UPAMA et al., 2022). Pouco tempo depois, em 1982, o físico Richard Feynman observou que a simulação eficiente de sistemas quânticos por computadores clássicos era fundamentalmente impossível devido ao crescimento exponencial da complexidade desses sistemas. Para ele, a única maneira de simular a natureza com precisão seria construindo um computador que também fosse quântico, capaz de acompanhar essa complexidade de forma nativa (NOFER et al., 2023; RIETSCHE et al., 2022).

Durante a década de 1990, o campo deixou de ser uma curiosidade puramente teórica para se tornar uma área de relevância científica e estratégica, impulsionada pela descoberta de algoritmos que provaram a superioridade quântica para tarefas específicas. O marco definitivo ocorreu em 1994, quando Peter Shor desenvolveu um algoritmo capaz de fatorar grandes números inteiros e calcular logaritmos discretos de forma exponencialmente mais rápida do que qualquer algoritmo clássico conhecido (SHOR, 1998). Essa descoberta evidenciou que a computação quântica não oferecia apenas um ganho de velocidade incremental, mas representava uma ameaça real aos sistemas de criptografia de chave pública, como o RSA, que sustentam a segurança digital global (SODIYA et al., 2024).

Atualmente, vivemos um momento que pode ser descrito como uma nova corrida tecnológica. Diferentemente de décadas passadas, o desenvolvimento da computação quântica saiu dos laboratórios de física universitários e entrou na agenda prioritária de governos e grandes corporações de tecnologia (LAGHARI et al., 2022). Empresas como IBM, Google, Microsoft, Intel e Alibaba estão investindo bilhões de dólares no desenvolvimento de processadores quânticos mais estáveis e escaláveis, competindo para atingir a supremacia quântica, o ponto em que um computador quântico executa uma tarefa inviável para qualquer supercomputador clássico atual (RIETSCHE et al., 2022; GILL; BUYYA, 2024).

Essa corrida, no entanto, não se restringe ao hardware. Há um esforço intenso no desenvolvimento de software e algoritmos que possam operar na era atual, conhecida

como NISQ (*Noisy Intermediate-Scale Quantum*). Ao mesmo tempo, o ecossistema está se expandindo para incluir serviços de computação quântica em nuvem (*Quantum Cloud Computing*), onde o acesso a processadores reais é disponibilizado remotamente para pesquisadores e desenvolvedores, democratizando o acesso à tecnologia e acelerando a descoberta de novas aplicações em áreas como finanças, logística e inteligência artificial (GILL; BUYYA, 2024; BASAK UPAMA et al., 2022). Compreender este contexto é essencial para situar os desafios técnicos de simulação e programação que serão abordados neste trabalho.

1.2 Definição do Problema

Embora os fundamentos matemáticos e teóricos da computação quântica estejam estabelecidos há várias décadas, transformar esses conceitos em uma tecnologia prática e escalável ainda esbarra em desafios físicos e de engenharia que definem o cenário atual da área. O principal obstáculo está na imaturidade do hardware disponível, um período que os pesquisadores chamam de era NISQ (*Noisy Intermediate-Scale Quantum*) (NOFER et al., 2023). Os dispositivos quânticos existentes hoje já conseguem demonstrar fenômenos essenciais, como superposição e emaranhamento, porém operam com um número limitado de qubits e apresentam taxas de erro elevadas. Esses erros surgem porque os qubits são extremamente sensíveis a qualquer interferência do ambiente, como ruído térmico, vibrações ou campos eletromagnéticos, o que leva à perda irreversível de informação, fenômeno conhecido como decoerência (RIETSCHE et al., 2022).

Para profissionais e pesquisadores da computação, acostumados à confiabilidade e previsibilidade dos sistemas clássicos, essa instabilidade inerente ao hardware quântico representa um desafio significativo. Validar a lógica de novos algoritmos ou interpretar o comportamento real de um qubit torna-se difícil quando o próprio dispositivo introduz ruídos capazes de corromper os resultados, produzindo saídas probabilísticas que podem misturar respostas corretas com erros do sistema (NOFER et al., 2023; GILL; BUYYA, 2024). Além disso, o acesso a processadores quânticos reais ainda é restrito, caro e muitas vezes depende de serviços em nuvem que funcionam por meio de filas de espera, o que limita a experimentação contínua e a iteração rápida que são essenciais para o avanço da área (CICERO et al., 2025).

Surge uma necessidade clara de ferramentas que façam a ponte entre a teoria e a prática. A simulação de circuitos quânticos em computadores clássicos é uma das estratégias mais importantes para permitir o desenvolvimento, o teste e a depuração de algoritmos antes de executá-los em máquinas reais (CICERO et al., 2025). No entanto, a simulação também traz suas próprias limitações, visto que simular um sistema quântico é uma tarefa intensiva em recursos, pois a quantidade de memória necessária para representar

o estado quântico completo cresce de forma exponencial à medida que novos qubits são adicionados ao circuito. Esse crescimento rápido impõe um limite rígido ao tamanho e à complexidade dos algoritmos que podem ser simulados de modo eficiente sem o uso de dispositivos quânticos físicos (BASAK UPAMA et al., 2022).

O problema central que este trabalho busca investigar é a dificuldade de compreender, comparar e validar as diferenças operacionais entre a computação clássica e a computação quântica em um cenário onde o hardware quântico real ainda é ruidoso e a simulação é custosa. Existe uma lacuna na literatura voltada à computação que vá além da descrição isolada dos fenômenos físicos e estabeleça uma conexão direta com a prática de programação, comparando de forma clara a estrutura lógica dos dois paradigmas e mostrando como algoritmos quânticos podem ser implementados, visualizados e analisados por meio de simuladores, contornando as limitações da era NISQ.

1.3 Objetivos

Para facilitar o entendimento do trabalho, os objetivos foram organizados em um objetivo geral e em objetivos específicos, que são descritos a seguir.

1.3.1 Objetivo Geral

O objetivo principal deste trabalho é aprofundar a análise teórica e comparativa entre os paradigmas da computação clássica e da computação quântica, investigando as diferenças fundamentais em suas unidades de informação, lógica de processamento e arquitetura de hardware. Busca-se compreender de que forma a computação quântica, mesmo enfrentando as limitações físicas e operacionais da atual era NISQ (*Noisy Intermediate-Scale Quantum*), ainda se apresenta como um caminho promissor para a resolução de problemas que permanecem intratáveis no modelo clássico. Para consolidar essa investigação, o estudo será finalizado com um caso prático desenvolvido em um simulador, no qual será possível demonstrar a aplicação da lógica quântica na resolução de um problema de busca e comparar sua estrutura e eficiência com as abordagens convencionais.

1.3.2 Objetivos Específicos

Para atingir o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Comparar as arquiteturas físicas e lógicas de processadores clássicos, baseados em transistores de silício e lógica booleana, com as principais tecnologias utilizadas na implementação de qubits, como circuitos supercondutores e íons aprisionados, identificando os desafios de engenharia e controle presentes em cada abordagem.

- Analisar o funcionamento de algoritmos que apresentam vantagens computacionais claras, como o Algoritmo de Shor, que aborda problemas de fatoração e criptografia, e o Algoritmo de Grover, utilizado para buscas em bases não estruturadas, comparando seu desempenho com as melhores alternativas clássicas disponíveis.

- Avaliar as limitações impostas pelo ruído ambiental e pela decoerência nos dispositivos quânticos atuais, além de destacar a importância da Correção de Erros Quânticos (QEC) como elemento essencial para alcançar a computação tolerante a falhas, diferenciando a redundância quântica da clássica.

- Implementar o Algoritmo de Grover por meio do *framework* Qiskit, realizando sua execução em um simulador clássico para demonstrar as etapas de construção de um circuito quântico, a definição do oráculo, a aplicação do amplificador de amplitude e a validação dos resultados probabilísticos esperados.

1.4 Metodologia

Para alcançar os objetivos deste trabalho, foi adotada uma metodologia que combina teoria e prática, organizada em duas fases que se complementam. A primeira é dedicada à revisão bibliográfica de artigos recentes, que oferecem a base conceitual necessária para compreender o funcionamento e as diferenças entre os modelos clássicos e quânticos, enquanto a segunda fase corresponde ao estudo de caso desenvolvido por meio de simulação, que permite observar na prática o funcionamento das técnicas analisadas.

A fase teórica é desenvolvida a partir da análise de trabalhos científicos específicos que ajudam a estabelecer o paralelo entre os dois paradigmas. O foco está em entender como o hardware clássico, baseado em transistores e portas lógicas booleanas, se diferencia das tecnologias empregadas nos qubits supercondutores e nos íons aprisionados, permitindo identificar as limitações físicas que cada abordagem apresenta. A revisão contempla o estudo dos algoritmos de Shor e de Grover, analisando a estrutura matemática que possibilita a ambos alcançarem vantagens computacionais relevantes em tarefas de fatoração, criptografia e busca. São abordados também a era NISQ e o papel da correção de erros quânticos (QEC), evidenciando a necessidade de minimizar a decoerência para que a computação quântica se torne confiável e escalável.

A segunda fase corresponde ao estudo de caso prático, que tem início com a implementação de uma busca linear clássica em Python para definir a linha de base de complexidade a ser comparada posteriormente. Dadas as limitações do hardware quântico disponível hoje, utiliza-se a simulação clássica de circuitos quânticos como forma de testar e validar a lógica quântica, assim observando o comportamento dos qubits em um ambiente controlado e sem interferências externas. Será utilizado o *framework* de desenvolvimento

Qiskit, uma ferramenta de código aberto amplamente utilizada em pesquisas acadêmicas e no setor tecnológico, que permite construir, compilar e simular circuitos quânticos.

O estudo de caso quântico será baseado na implementação do Algoritmo de Grover em um espaço de busca definido, o que possibilita acompanhar de forma detalhada a evolução dos estados quânticos ao longo da execução. A implementação segue as etapas fundamentais do algoritmo, começando pela preparação dos estados quânticos, passando pela aplicação dos operadores responsáveis por marcar a solução e amplificar sua probabilidade e finalizando com o processo de medição. Os resultados obtidos na simulação serão comparados com os da busca clássica, permitindo discutir a eficiência do modelo quântico em relação às abordagens tradicionais e reforçando a importância da simulação como ferramenta essencial no desenvolvimento atual de software quântico.

1.5 Organização do trabalho

Para a realização do restante deste trabalho, os capítulos foram organizados da seguinte forma:

O Capítulo 2, Fundamentos da Informação Clássica e Quântica, apresenta a base teórica do trabalho ao explicar como as unidades básicas de informação se diferenciam. São analisadas as diferenças entre o bit e o qubit, explorando de que maneira fenômenos como superposição e emaranhamento ampliam a forma como a informação pode ser representada e manipulada, além de apresentar as diferenças entre portas clássicas e portas quânticas.

No Capítulo 3, Arquiteturas Físicas dos Sistemas, o foco passa para a camada de hardware. O texto compara a construção dos processadores de silício com as principais tecnologias usadas na criação de qubits, explicando o funcionamento de modelos como qubits supercondutores e íons aprisionados e apresentando também os desafios de engenharia presentes em cada tecnologia.

O Capítulo 4, Algoritmos e Aplicações Estratégicas, apresenta os algoritmos que mostram vantagens reais quando comparados aos métodos clássicos. São analisados, por exemplo, o Algoritmo de Shor, voltado para problemas de fatoração, e o Algoritmo de Grover, utilizado em buscas, além de serem apresentadas aplicações em áreas como criptografia, simulação de materiais e inteligência artificial.

Em seguida, o Capítulo 5, Desafios Técnicos, discute as limitações atuais da computação quântica. A seção aborda a era NISQ, a influência do ruído ambiental, a decoerência e a necessidade de técnicas específicas de correção de erros para garantir cálculos mais confiáveis.

O Capítulo 6, Ecossistema de Software Quântico, aborda os recursos para a implementação e teste de algoritmos quânticos. São apresentadas as fases de construção, as

linguagens usadas para criar circuitos e os *frameworks* de desenvolvimento, com destaque para a importância da simulação clássica como etapa essencial para validar projetos.

O Capítulo 7, Implementação do Algoritmo de Grover, descreve a parte prática do trabalho. O foco está na construção de um circuito quântico simulado para resolver um problema de busca e na análise dos resultados obtidos, permitindo comparar o comportamento observado com o desempenho esperado em teoria.

Por fim, o Capítulo 8, Conclusão, aborda os resultados gerais do estudo, retomando a comparação entre os paradigmas, apontando caminhos para pesquisas futuras e para o amadurecimento da computação quântica.

2 Fundamentos da Informação Clássica e Quântica

A compreensão da computação quântica exige primeiro uma revisão dos conceitos que servem de base para o processamento da informação. Na computação clássica, os dados são representados por estados bem definidos e seguem uma lógica determinística, o que garante previsibilidade no processamento. Já na computação quântica, a informação segue princípios que permitem a existência de múltiplos estados ao mesmo tempo e correlações não locais (SHOR, 1998). Essa mudança representa uma forma diferente de organizar e transformar a informação, já que o processamento deixa de seguir uma sequência rígida e passa a ser descrito como uma manipulação de amplitudes de probabilidade (NOFER et al., 2023).

Aqui são apresentadas as características da unidade básica de informação em cada sistema, explicando como o bit clássico se diferencia do qubit e descrevendo fenômenos da mecânica quântica, como superposição e emaranhamento. Também são analisadas as operações lógicas de cada paradigma, oferecendo uma visão geral das diferenças entre as portas utilizadas na computação clássica e as transformações aplicadas nos circuitos quânticos.

2.1 Unidades de Informação

A diferença mais básica entre a computação clássica e a computação quântica está na forma como cada uma representa a unidade fundamental de informação. Na computação clássica, toda a lógica do processamento digital é construída a partir do bit, que só pode assumir os estados 0 ou 1. Esse comportamento está ligado à física clássica e normalmente é implementado por meio de níveis de tensão em transistores de silício, seguindo uma lógica determinística na qual não há dúvidas sobre o estado em que o sistema se encontra (NANDHINI; SINGH; AKASH, 2022).

Na computação quântica, a unidade básica é o qubit, que possui dois estados de referência, chamados de estados de base e representados na notação de Dirac como $|0\rangle$ e $|1\rangle$. A diferença é que, devido aos princípios da mecânica quântica, o qubit não precisa estar em apenas um desses estados, visto que a superposição permite que ele assuma uma combinação de $|0\rangle$ e $|1\rangle$ ao mesmo tempo, o que amplia significativamente a forma de representar a informação (RIETSCHE et al., 2022).

Do ponto de vista matemático, o estado de um qubit $|\psi\rangle$ é descrito por um vetor

unitário em um espaço vetorial complexo de duas dimensões, chamado espaço de Hilbert, e pode ser escrito como:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.1}$$

Nessa expressão, α e β são números complexos chamados de amplitudes de probabilidade. A diferença crucial em relação ao modelo clássico é que essas amplitudes carregam mais informações do que apenas um valor binário, pois descrevem a contribuição de cada estado base para o estado total do sistema. No entanto, a natureza quântica determina que o resultado de uma medição física seja sempre 0 ou 1. A probabilidade de se medir o estado $|0\rangle$ é dada por $|\alpha|^2$ e a de se medir o estado $|1\rangle$ é dada por $|\beta|^2$, satisfazendo a condição de normalização $|\alpha|^2 + |\beta|^2 = 1$ (NOFER et al., 2023).

Uma forma simples de visualizar essa diferença é usar a Esfera de Bloch. Enquanto um bit clássico pode ser representado apenas por dois pontos fixos, análogos ao polo norte para o estado $|0\rangle$ e ao polo sul para o estado $|1\rangle$, o estado de um qubit pode ser representado por qualquer ponto na superfície da esfera. Isso mostra que ele pode assumir uma infinidade de estados diferentes antes da medição, o que ajuda a entender por que algoritmos quânticos conseguem explorar o espaço de estados de maneira mais rica e obter vantagens computacionais em relação aos modelos clássicos (NOFER et al., 2023; SHOR, 1998).

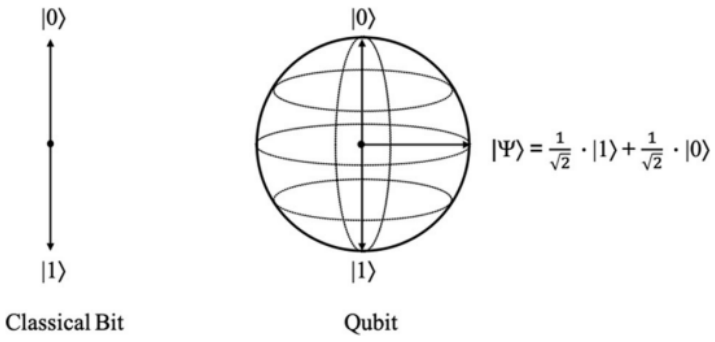

Figura 1 – Comparação entre o bit clássico e a representação do qubit na Esfera de Bloch. Fonte: (NOFER et al., 2023).

2.2 Fenômenos Mecânico-Quânticos

Para entender o real potencial da computação quântica, é preciso ir além da definição do qubit e observar os fenômenos que determinam como esses sistemas se comportam, como interagem e como evoluem ao longo do tempo. Diferente da computação clássica, que segue as leis previsíveis da física tradicional, a computação quântica utiliza propriedades fundamentais da mecânica quântica, sendo as mais notáveis a superposição,

o emaranhamento e a interferência (NOFER et al., 2023; SHOR, 1998; UGWUISHIWU et al., 2022).

A superposição é a característica que permite a um sistema quântico existir em vários estados ao mesmo tempo. Enquanto um registro clássico de n bits pode representar apenas um dos 2^n estados possíveis de cada vez, um registro quântico de n qubits em superposição pode representar uma combinação linear de todos os 2^n estados simultaneamente (NANDHINI; SINGH; AKASH, 2022). Isso dá origem ao chamado paralelismo quântico, que permite ao hardware analisar uma quantidade exponencial de informações com um número reduzido de componentes físicos, tornando possível avaliar uma função matemática para diversas entradas em um único passo computacional (NOFER et al., 2023).

Outro fenômeno central é o emaranhamento, que estabelece uma conexão profunda entre qubits. Assim, quando dois ou mais qubits se tornam emaranhados, passam a se comportar como um único sistema (SHOR, 1998; NOFER et al., 2023). O estado de cada qubit não pode ser descrito isoladamente e qualquer alteração em um deles afeta instantaneamente os demais, mesmo que estejam separados por grandes distâncias físicas (NOFER et al., 2023). O emaranhamento é um recurso essencial que possibilita a criação de protocolos de comunicação e computação que não podem ser reproduzidos no modelo clássico, permitindo que partes distintas do computador quântico funcionem de maneira integrada, como se fossem uma única unidade de processamento (LAGHARI et al., 2022).

A interferência é o mecanismo que permite transformar esse processamento paralelo em um resultado útil. Como as amplitudes de probabilidade α e β são números complexos, elas podem se somar de modo a reforçar um resultado, fenômeno conhecido como interferência construtiva, ou se anular, produzindo a interferência destrutiva, o que ocorre de maneira semelhante ao comportamento de ondas (NOFER et al., 2023). Os algoritmos quânticos são construídos para controlar essas amplitudes de probabilidade, fazendo com que os caminhos que levariam a resultados incorretos se anulem pela interferência destrutiva, enquanto os caminhos que levam à resposta correta são reforçados pela interferência construtiva, aumentando a chance de que o resultado desejado apareça no momento da medição (SHOR, 1998).

Por fim, é importante destacar uma limitação fundamental da computação quântica imposta pelas próprias leis da mecânica quântica, conhecida como Teorema da Não-Clonagem. Diferente da informação clássica, que pode ser copiada livremente, um estado quântico desconhecido não pode ser duplicado de forma perfeita (LAGHARI et al., 2022; NANDHINI; SINGH; AKASH, 2022). Essa característica cria desafios importantes para o desenvolvimento de algoritmos e para a correção de erros quânticos, pois impede o uso direto das estratégias tradicionais de cópia e exige novas formas de proteger a informação (NANDHINI; SINGH; AKASH, 2022).

2.3 Lógica Computacional

Enquanto na computação clássica o processamento da informação é baseado na Álgebra Booleana, usando portas lógicas que realizam operações bem definidas sobre bits, como AND, OR e NOT, a computação quântica segue um caminho diferente e se baseia na Álgebra Linear. No modelo quântico, o processamento não se resume a operações binárias simples, pois envolve transformações matemáticas que atuam diretamente sobre as amplitudes de probabilidade dos qubits, permitindo uma forma mais flexível de manipular a informação (NOFER et al., 2023).

As portas lógicas quânticas formam a base dos circuitos quânticos e podem atuar sobre um único qubit ou sobre vários qubits ao mesmo tempo. Entre as portas que trabalham com apenas um qubit, destacam-se a Pauli-X, que funciona de maneira semelhante à porta NOT na computação clássica, ao trocar as amplitudes de probabilidade, transformando $|0\rangle$ em $|1\rangle$ e vice-versa, e a porta Pauli-Z, que altera apenas a fase do estado $|1\rangle$ ao inverter o sinal de sua amplitude, sem mudar a probabilidade de medição, sendo crucial para criar interferência (NANDHINI; SINGH; AKASH, 2022; BASAK UPAMA et al., 2022). Outra porta fundamental é a Hadamard, que é responsável por colocar um qubit em estado de superposição, transformando um estado bem definido, como $|0\rangle$, em uma combinação de ambos os estados $|0\rangle$ e $|1\rangle$. Isso faz com que o qubit, em vez de estar exclusivamente em um estado ou outro, passe a ter uma probabilidade de ser medido em qualquer um dos dois estados, permitindo que ambos existam ao mesmo tempo até a medição ser feita (NANDHINI; SINGH; AKASH, 2022; NOFER et al., 2023).

Para operações envolvendo múltiplos qubits, o que é essencial para a criação do emaranhamento, utilizam-se as portas controladas. A mais fundamental é a CNOT (Controlled-NOT), que atua sobre dois qubits, sendo um deles o qubit de controle e o outro o qubit alvo. O funcionamento é condicional: quando o qubit de controle estiver no estado $|0\rangle$, o alvo permanece inalterado, e quando o qubit de controle estiver no estado $|1\rangle$, o qubit alvo tem seu estado invertido, trocando de $|0\rangle$ para $|1\rangle$ ou vice-versa (SHOR, 1998; NANDHINI; SINGH; AKASH, 2022). De forma análoga, a porta CZ (Controlled-Z) aplica uma mudança de fase ao estado $|11\rangle$, invertendo o sinal da amplitude apenas quando ambos os qubits estão nesse estado, sendo muito utilizada na construção de oráculos e em algoritmos de busca quântica (CICERO et al., 2025; BASAK UPAMA et al., 2022).

Uma característica importante das portas quânticas é que elas são reversíveis, o que significa que não há perda de informação durante o processamento quântico e que, em teoria, é possível desfazer operações realizadas e retornar ao estado inicial. Essa característica se diferencia do comportamento de muitas portas clássicas e está diretamente relacionada à maneira como a informação é tratada no contexto quântico (SHOR, 1998; BASAK UPAMA et al., 2022).

Após a aplicação das portas lógicas, o processamento é encerrado pela operação de medição (M), que é irreversível e tem como função extrair a informação do sistema, fazendo com que o estado em superposição seja colapsado em um resultado clássico, 0 ou 1. Como esse resultado é probabilístico, os algoritmos quânticos costumam ser executados várias vezes para que a distribuição de resultados possa ser analisada e a resposta correta seja identificada com maior confiança (NOFER et al., 2023; CICERO et al., 2025).

3 Arquiteturas Físicas dos Sistemas

A passagem dos conceitos teóricos para a construção de computadores quânticos que funcionem de forma confiável representa um dos maiores desafios da engenharia atual. Enquanto a computação clássica se desenvolveu a partir da tecnologia de semicondutores e arquiteturas bem estabelecidas, a computação quântica ainda não possui uma arquitetura física padronizada. Assim, diferentes tecnologias estão sendo desenvolvidas para a implementação de qubits, cada uma apresentando vantagens e desvantagens específicas, exigindo soluções de engenharia únicas para superar as limitações impostas pela própria natureza quântica.

3.1 Construção de Processadores

A construção física de um processador quântico exige uma infraestrutura de suporte muito mais complexa do que apenas o chip onde os qubits são implementados. Um dos principais desafios da engenharia nesta etapa é garantir que o sistema permaneça praticamente isolado do ambiente, já que os estados quânticos são altamente sensíveis a interferências externas, como ruído térmico, vibrações e radiação eletromagnética. Para reduzir esses efeitos, a maior parte dos processadores quânticos atuais opera dentro de refrigeradores de diluição, que mantêm o chip em vácuo e em temperaturas extremamente baixas, próximas ao zero absoluto (CICERO et al., 2025).

Além do controle da temperatura, o funcionamento do sistema depende de uma camada de controle complexa, formada por eletrônica de alta precisão e por cabos específicos responsáveis por enviar pulsos de micro-ondas ou lasers aos qubits. A integração entre o hardware quântico e os sistemas clássicos de controle é um dos principais obstáculos para a escalabilidade, visto que à medida que o número de qubits aumenta, torna-se mais difícil organizar os cabos e controlar a dissipação de calor dentro do refrigerador sem comprometer o estado de coerência do sistema (NOFER et al., 2023).

A fabricação do chip quântico exige materiais com um nível de pureza muito maior do que o normalmente utilizado na indústria de semicondutores. Mesmo pequenos defeitos na estrutura dos materiais ou impurezas nas superfícies do processador podem levar à perda rápida da informação quântica. Por esse motivo, a engenharia de materiais tem um papel central nesse processo, buscando técnicas de fabricação e deposição que reduzam essas perdas e permitam que os qubits permaneçam estáveis pelo tempo necessário para a realização de cálculos úteis (CICERO et al., 2025).

3.2 Principais Tecnologias de Implementação

A computação clássica alcançou um padrão bem definido ao longo do tempo com o uso dos transistores de silício, enquanto a computação quântica ainda passa por uma fase de experimentação, na qual diferentes tecnologias físicas estão sendo estudadas para entender qual delas pode se tornar o padrão no futuro. O desenvolvimento do hardware quântico busca equilibrar o tempo de coerência dos qubits, a velocidade de processamento e a possibilidade de produção em larga escala. Atualmente, a busca pelo hardware ideal foca nos circuitos supercondutores e nos sistemas de íons aprisionados, que são as duas abordagens principais (NOFER et al., 2023; NANDHINI; SINGH; AKASH, 2022).

A arquitetura de circuitos supercondutores tem origem na eletrônica de estado sólido e é a mais adotada por empresas como Google, IBM e Intel. Essa tecnologia utiliza o fenômeno da supercondutividade para criar circuitos que funcionam em temperaturas próximas ao zero absoluto dentro de câmaras criogênicas, com o objetivo de reduzir os efeitos do ruído térmico do ambiente. O componente central desse hardware é a junção de Josephson que, em conjunto com um capacitor, permite a implementação física do qubit no chip. Cada qubit pode ser controlado de forma individual por sinais de radiofrequência ajustados às suas frequências de ressonância, o que possibilita a realização das operações lógicas por meio de pulsos bem definidos e facilita a integração com os sistemas clássicos de controle (CICERO et al., 2025; NANDHINI; SINGH; AKASH, 2022).

Apesar dessas vantagens, a escalabilidade dos circuitos supercondutores ainda enfrenta limitações importantes relacionadas à infraestrutura física. Um dos principais problemas está na quantidade de cabos de entrada e saída que se conectam ao chip por meio da câmara criogênica, já que cada cabo transfere calor ao sistema e aumenta o nível de ruído térmico. Além disso, a forma como os qubits são conectados no chip geralmente permite a interação apenas entre os vizinhos próximos, o que exige operações extras para realizar cálculos entre qubits que não possuem conexão direta. A necessidade de recalibrações frequentes para compensar interferências do ambiente e a presença de impurezas na superfície do chip dificultam a manutenção da estabilidade do hardware conforme o sistema cresce (NOFER et al., 2023; CICERO et al., 2025).

Os sistemas de íons aprisionados seguem um caminho diferente e utilizam partículas atômicas carregadas positivamente, mantidas suspensas no vácuo por campos eletromagnéticos. A informação quântica é armazenada nos estados eletrônicos de cada íon. Como essas partículas possuem a mesma carga elétrica positiva, elas se repelem de forma natural, o que cria uma ligação devido à força de repulsão entre os íons aprisionados. Para a realização dos cálculos, são utilizados lasers de alta precisão que controlam cada qubit de forma individual e aproveitam essa força de repulsão para permitir que a informação de um íon seja transmitida aos demais por meio do movimento coletivo e quantizado das próprias partículas dentro de uma armadilha compartilhada. Essa característica torna

a arquitetura uma alternativa promissora para a construção de computadores quânticos universais e versáteis (NANDHINI; SINGH; AKASH, 2022).

A comparação entre essas duas arquiteturas mostra um equilíbrio entre a velocidade das operações e a estabilidade da informação quântica. Os circuitos supercondutores se destacam por permitir a execução de portas lógicas em altas velocidades, porém eles apresentam tempos de coerência limitados, que variam de microssegundos até cerca de um milissegundo. Dessa forma, o sistema de controle deve ser rápido o suficiente para executar milhares de operações durante esse tempo. Já os íons aprisionados oferecem qubits naturalmente mais estáveis por serem baseados em átomos idênticos, mas enfrentam dificuldades para escalar o conjunto de lasers necessário para controlar grandes conjuntos de partículas simultaneamente. Em ambos os casos, o funcionamento do hardware depende de uma camada de controle complexa, na qual conversores digital-analógicos (DAC) e analógico-digitais (ADC) são responsáveis por converter os comandos da computação clássica em pulsos físicos que manipulam os qubits (NANDHINI; SINGH; AKASH, 2022; CICERO et al., 2025).

O avanço dessas arquiteturas é sustentado pelo uso de simulações em diferentes níveis do sistema. No nível mais próximo do hardware, essas simulações ajudam a estudar os materiais e o comportamento físico dos qubits, levando em conta aspectos mecânicos, térmicos e eletromagnéticos que influenciam na construção do hardware. À medida que os sistemas passam a operar com centenas de qubits, a necessidade de memória e de capacidade de processamento para simular o comportamento do sistema cresce exponencialmente, o que reforça a importância de otimizações de software e do suporte oferecido pelo hardware clássico para sustentar a evolução da computação quântica (CICERO et al., 2025).

4 Algoritmos e Aplicações Estratégicas

Este capítulo apresenta os algoritmos quânticos e explica como eles usam os princípios da mecânica quântica para superar as limitações dos sistemas tradicionais. O foco é mostrar como esses algoritmos podem ser aplicados em áreas estratégicas, como em problemas de otimização e segurança da informação.

4.1 Criptografia e Segurança

4.1.1 O Algoritmo de Shor e o impacto na segurança RSA

Grande parte da segurança digital utilizada atualmente depende de sistemas de criptografia assimétrica para proteger comunicações na internet. O protocolo RSA é um dos principais exemplos, pois sua proteção está ligada à dificuldade de fatorar números inteiros muito grandes, algo que a computação clássica não consegue realizar em tempo viável. Na prática, multiplicar dois números primos para gerar uma chave é simples, mas realizar o processo inverso e descobrir os fatores originais exige um tempo de processamento inviável. O Algoritmo de Shor muda essa lógica ao explorar a superposição e a interferência quântica, permitindo analisar vários estados ao mesmo tempo e encontrar os fatores de um número de forma eficiente (NOFER et al., 2023; SHOR, 1998).

Com o uso da Transformada de Fourier Quântica (QFT), esse algoritmo consegue identificar padrões matemáticos que tornam possível quebrar a proteção do RSA de maneira exponencialmente mais rápida do que qualquer supercomputador atual. Estimativas teóricas indicam que, enquanto as abordagens clássicas levariam cerca de 300 trilhões de anos para fatorar uma chave RSA de 2048 bits, um computador quântico escalável operando com aproximadamente 20 milhões de qubits poderia realizar essa mesma tarefa em um intervalo de apenas oito horas. Essa vantagem computacional mostra como os sistemas baseados em fatoração tendem a se tornar inadequados à medida que o hardware quântico evolui (SHOR, 1998; BASAK UPAMA et al., 2022; CICERO et al., 2025).

O potencial de execução do Algoritmo de Shor coloca em risco a confiança dos sistemas de segurança usados atualmente. A fragilização da criptografia convencional expõe serviços essenciais, como operações bancárias realizadas pela internet, a novas vulnerabilidades. Essa exposição afeta diretamente a proteção de informações sensíveis, tanto no setor público quanto no privado, que precisam permanecer seguras por longos períodos. Surge, então, a necessidade de desenvolver e adotar novos modelos de segurança capazes de resistir ao poder de processamento dos computadores quânticos (NOFER et al., 2023; CICERO et al., 2025).

4.1.2 Criptografia Pós-Quântica (PQC) e Distribuição de Chaves (QKD)

A Criptografia Pós-Quântica (PQC) se baseia no desenvolvimento de algoritmos construídos a partir de problemas matemáticos que os computadores quânticos não conseguem resolver de forma eficiente. A principal vantagem desta abordagem é que ela pode ser implementada via software nas arquiteturas atuais. Isso permite que os sistemas sejam atualizados sem a necessidade imediata de trocar o hardware, facilitando a transição para modelos de segurança mais robustos (SODIYA et al., 2024; NOFER et al., 2023; NANDHINI; SINGH; AKASH, 2022).

Diferente da abordagem baseada em algoritmos, a Distribuição de Chaves Quânticas (QKD) fundamenta sua segurança nas leis da física para proteger a transmissão de dados. A chave criptográfica é transmitida por meio de partículas quânticas, geralmente fótons, que são enviados um a um por um canal de fibra óptica. Cada fóton é preparado em um estado quântico específico, como diferentes ângulos em que a luz vibra, que representam valores binários. Por exemplo, uma polarização vertical pode representar o valor 1, enquanto uma polarização horizontal pode representar o valor 0. Esses estados funcionam como portadores de informação, de modo que a sequência de fótons transmitidos corresponde à sequência de bits que compõem a chave criptográfica. Quando ocorre uma tentativa de interceptação, a medição altera o estado quântico do fóton, permitindo que a interferência seja detectada pelo receptor, já que o sinal recebido apresentará diferenças em relação ao original. Dessa forma, a QKD assegura que a troca de chaves criptográficas não dependa apenas da complexidade de cálculos, mas de propriedades físicas do próprio sistema. Isso impede o acesso não autorizado às informações sem que o sistema identifique a intrusão (SODIYA et al., 2024; LAGHARI et al., 2022; NANDHINI; SINGH; AKASH, 2022).

A adoção dessas tecnologias representa um passo importante para reduzir os riscos associados à evolução do poder computacional quântico. Atualmente, o foco da área está voltado para a validação de protocolos que permitam a substituição gradual dos algoritmos criptográficos tradicionais por soluções resistentes a ataques quânticos. A incorporação dessas novas defesas é essencial para recuperar a confiança na infraestrutura digital e garantir a integridade das comunicações em um cenário pós-quântico (LAGHARI et al., 2022; SODIYA et al., 2024).

4.2 Otimização e Busca

A computação quântica traz avanços importantes em otimização e busca em bases de dados desestruturadas. O Algoritmo de Grover se destaca nessas áreas por permitir a localização de itens em listas desorganizadas de forma mais eficiente. Enquanto a busca clássica depende de uma análise sequencial dos dados, a abordagem quântica utiliza a superposição para representar todas as possibilidades ao mesmo tempo. Com base nessa

superposição, o algoritmo aplica a interferência quântica para ampliar a probabilidade da resposta correta, reduzindo drasticamente o esforço computacional necessário (SODIYA et al., 2024; UGWUISHIWU et al., 2022).

Essa diferença fica clara quando se compara o número de tentativas necessárias em cada abordagem. Em uma base com N elementos, um computador clássico precisa, em média, de $N/2$ tentativas para encontrar o item correto, já um sistema quântico é capaz de realizar essa busca em aproximadamente $\sqrt{N}$ operações. Esse ganho de desempenho reduz de maneira significativa o tempo de processamento necessário para lidar com grandes volumes de dados (RIETSCHE et al., 2022; CICERO et al., 2025).

No futuro, a aplicação completa desse algoritmo exigirá mudanças importantes nas defesas digitais, principalmente na criptografia de chave simétrica. Como o Algoritmo de Grover acelera a busca por essas chaves, a segurança efetiva dos sistemas atuais acaba sendo reduzida pela metade. Na prática, isso significa que uma chave de 256 bits em um contexto clássico passa a oferecer um nível de proteção semelhante ao de uma chave de 128 bits diante de ataques quânticos, tornando necessário o uso de chaves maiores para manter o sigilo das informações. Por outro lado, essa capacidade de processamento otimizado abre novas possibilidades em áreas como a inteligência artificial aplicada à cibersegurança. O algoritmo possibilita uma análise mais rápida de grandes volumes de dados para identificar anomalias e padrões de ameaças com uma agilidade superior à dos métodos clássicos. Assim, o Algoritmo de Grover se consolida como um recurso estratégico tanto para o aprimoramento da proteção quanto para a eficiência de sistemas críticos (SODIYA et al., 2024; OLORUNSOGO; JACKS; AJALA, 2024).

4.3 Simulação Quântica

A possibilidade de simular o comportamento da matéria em nível molecular é considerada uma das aplicações mais promissoras da computação quântica (NANDHINI; SINGH; AKASH, 2022). Diferente do processo tradicional de desenvolvimento de medicamentos e ingredientes ativos, que é frequentemente longo, custoso e baseado em métodos de tentativa e erro no mundo real, a computação quântica permitirá a replicação virtual desses comportamentos com base nos próprios princípios da física quântica. Isso abre caminho para um futuro em que a descoberta de novos compostos pode acontecer de forma muito mais rápida e precisa do que os limites da computação clássica conseguem oferecer atualmente. Essa abordagem possui o potencial de substituir processos de alto custo por pesquisas baseadas em simulação de alta fidelidade (RIETSCHE et al., 2022; SODIYA et al., 2024).

Nesse contexto, algoritmos quânticos vão ser empregados para modelar estruturas químicas complexas e prever interações medicamentosas. Empresas de biotecnologia, como

a ProteinQure, já exploram o uso desses sistemas para a modelagem de proteínas, o que poderá revolucionar o tratamento de doenças cardiovasculares e do câncer (GILL; BUYYA, 2024).

Além do setor farmacêutico, a simulação quântica também vem sendo aplicada em outros campos, como a análise de sequências genéticas e a produção de fertilizantes (NOFER et al., 2023). Na ciência dos materiais, essa tecnologia permite o desenvolvimento de compostos mais resistentes à corrosão e unidades de armazenamento de energia mais eficientes. Um exemplo prático é o uso de computadores quânticos por montadoras como Volkswagen e Daimler para modelar a composição química de baterias de veículos elétricos, com o objetivo de melhorar o desempenho e reduzir o impacto ambiental (NANDHINI; SINGH; AKASH, 2022; GILL; BUYYA, 2024). Por fim, a aplicação desses modelos estende-se à infraestrutura urbana, onde algoritmos quânticos preliminares estão sendo testados para, no futuro, gerenciar e simular o fluxo de tráfego em grandes centros urbanos (GILL; BUYYA, 2024).

5 Desafios Técnicos

Este capítulo trata dos principais desafios técnicos que ainda limitam o avanço e o uso prático dos sistemas quânticos em larga escala. Serão abordados os aspectos físicos e operacionais que marcam o estágio atual dessa tecnologia, oferecendo uma visão geral das dificuldades envolvidas para manter a estabilidade do hardware e preservar a informação quântica em ambientes ruidosos.

5.1 A Era NISQ

Hoje, a computação quântica encontra-se na chamada era NISQ (*Noisy Intermediate-Scale Quantum*), conceito apresentado pelo físico John Preskill em 2018, que pode ser traduzido como "Computação Quântica de Escala Intermediária Ruidosa". Esta geração de dispositivos quânticos, embora represente um avanço importante em relação aos sistemas clássicos, ainda opera sob grandes limitações de hardware (NOFER et al., 2023).

O termo Escala Intermediária está relacionado à quantidade limitada de qubits presentes nos processadores quânticos atuais, que costuma variar entre dezenas e algumas centenas. Embora esse número já permita superar o desempenho de supercomputadores clássicos em tarefas específicas, ele ainda é insuficiente para a computação universal tolerante a falhas, que exigiria milhares ou até milhões de qubits para aplicações realmente robustas (RIETSCHE et al., 2022; BASAK UPAMA et al., 2022).

Já o termo Ruidosa refere-se à alta taxa de erro e à imperfeição dos qubits físicos. Diferente da computação clássica, os dispositivos NISQ não possuem correção de erros nativa, o que significa que os qubits são extremamente sensíveis a interferências ambientais, resultando em cálculos que podem ser corrompidos antes da conclusão. Por esse motivo, muitos algoritmos quânticos atuais são projetados sob uma arquitetura híbrida, na qual um computador clássico atua em conjunto com o processador quântico para ajudar na otimização dos cálculos e reduzir os impactos da instabilidade do hardware (CICERO et al., 2025; RIETSCHE et al., 2022).

5.2 Decoerência

A decoerência quântica é um dos principais desafios para tornar a computação quântica viável em larga escala. Variações em campos elétricos, ruído térmico, vibrações e outras interferências externas podem provocar a perda da informação armazenada nos qubits, afetando diretamente a superposição e o emaranhamento. Esse contato com o

ambiente acaba forçando os estados quânticos a colapsar para valores binários, 0 ou 1, antes que o processamento seja finalizado (GILL; BUYYA, 2024; NOFER et al., 2023).

A fragilidade desses sistemas é tão elevada que pequenas variações de temperatura, na escala de milikelvins, ou interferências de campos magnéticos são suficientes para comprometer o funcionamento dos processadores. Esse nível de sensibilidade exige que os dispositivos quânticos operem em condições extremas de isolamento e vácuo para prolongar a vida útil da informação armazenada nos estados quânticos (CICERO et al., 2025).

A estabilidade desses sistemas é medida pelo chamado tempo de coerência, que define o intervalo de tempo disponível para a realização das operações lógicas. Segundo (CICERO et al., 2025), em circuitos supercondutores, esse tempo, que em modelos de ponta chega a um milissegundo, impõe um limite direto à profundidade do circuito, pois determina quantas portas quânticas podem ser aplicadas em sequência.

Caso a execução de um algoritmo ultrapasse o tempo de coerência disponível no hardware, os erros acumulados passam a comprometer o cálculo. Nessa situação, o processador deixa de produzir resultados confiáveis e começa a gerar saídas puramente aleatórias, invalidando qualquer tentativa de computação útil (RIETSCHE et al., 2022).

5.3 Correção de Erros Quânticos (QEC)

A Correção de Erros Quânticos (QEC) é o conjunto de métodos desenvolvidos para proteger a informação quântica contra a decoerência e contra falhas que surgem naturalmente durante o processamento. Ao contrário da computação convencional, que utiliza a simples replicação de bits para criar redundância e identificar erros, a computação quântica é limitada pelo Teorema da Não-Clonagem. Esse princípio fundamental da mecânica quântica estabelece que é impossível criar uma cópia exata de um estado quântico desconhecido, o que impede a aplicação direta dos códigos de repetição tradicionais para a detecção e correção de erros (NANDHINI; SINGH; AKASH, 2022).

Para lidar com essa limitação, a QEC distribui a informação de um qubit lógico em um sistema emaranhado composto por múltiplos qubits físicos ruidosos. Essa abordagem permite identificar e corrigir padrões de erro a partir das correlações quânticas, sem a necessidade de realizar uma medição direta que causaria o colapso da superposição original. Os qubits lógicos são unidades de informação protegidas que visam alcançar a estabilidade e a tolerância a falhas. A implementação de um qubit lógico funcional pode exigir o controle coordenado de diversas unidades físicas, gerando uma sobrecarga de hardware que os processadores atuais ainda não conseguem suportar. No estágio atual da tecnologia, as taxas de erro das portas lógicas permanecem acima do limite necessário para que a aplicação de códigos de correção produza ganhos computacionais reais. Como ainda não é possível aplicar a correção de erros quânticos em larga escala com os recursos disponíveis,

os sistemas acabam recorrendo principalmente a técnicas de mitigação de erros, que buscam reduzir o impacto do ruído nos resultados finais, sem depender de uma estrutura completa de correção (NOFER et al., 2023; RIETSCHE et al., 2022).

6 Ecossistema de Software Quântico

O uso de processadores quânticos depende de um ecossistema de software responsável por organizar a execução dos algoritmos e fazer a comunicação com o hardware. Os principais componentes desse ecossistema envolvem as linguagens de programação, os ambientes de desenvolvimento utilizados para a construção de circuitos quânticos e o uso de simuladores clássicos para a validação de códigos, o que possibilita identificar falhas antes da execução dos algoritmos em processadores reais.

6.1 A Pilha de Software

O desenvolvimento de software quântico tem um papel essencial ao tornar mais simples a criação e a execução de algoritmos, permitindo que pesquisadores explorem o potencial dessa tecnologia na resolução de problemas complexos. O ecossistema atual é formado por diferentes plataformas que surgiram ao longo do tempo para apoiar a construção de programas quânticos, com destaque para o lançamento do Qiskit pela IBM em 2017 e do Q# pela Microsoft no mesmo ano. Nos anos seguintes, outras ferramentas passaram a integrar esse cenário, como o Cirq do Google, o PyQuil da Rigetti, o Silq da ETH Zurich e o TKET da Quantinuum. Essas ferramentas ampliam o uso da computação quântica em áreas como química, ciência dos materiais, finanças e problemas de otimização (GILL; BUYYA, 2024).

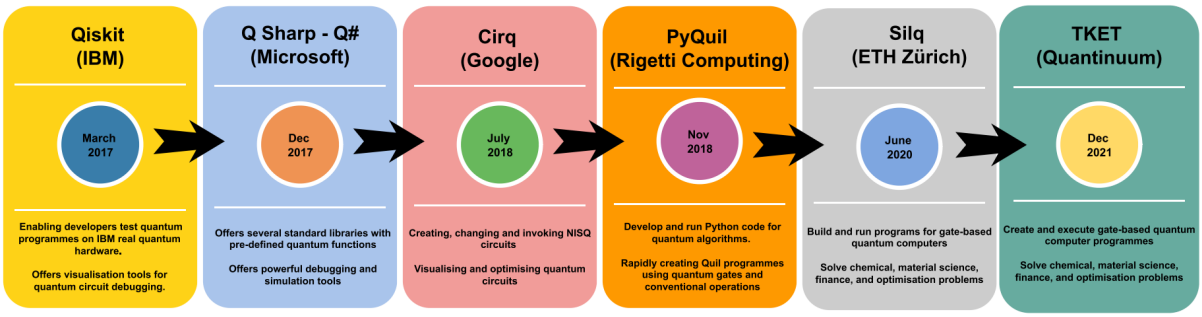

Figura 2 – Ferramentas de Software e Plataformas para Programação Quântica.
Fonte: (GILL; BUYYA, 2024).

Entre essas tecnologias, o Cirq se destaca por ser uma biblioteca de código aberto voltada para a criação e otimização de circuitos quânticos de escala intermediária ruidosa, sendo utilizada tanto em simuladores quanto em dispositivos reais acessados pela nuvem. A estruturação desse ambiente segue uma hierarquia funcional que começa nos simuladores de circuitos e nos serviços em nuvem, passa pelo *framework* principal de programação e alcança bibliotecas de pesquisa especializadas. Essa organização facilita a integração

com ferramentas como o TensorFlow Quantum para modelos híbridos de aprendizado de máquina, o OpenFermion para aplicações em química quântica e soluções como ReCirq e PennyLane voltadas à otimização variacional (BASAK UPAMA et al., 2022).

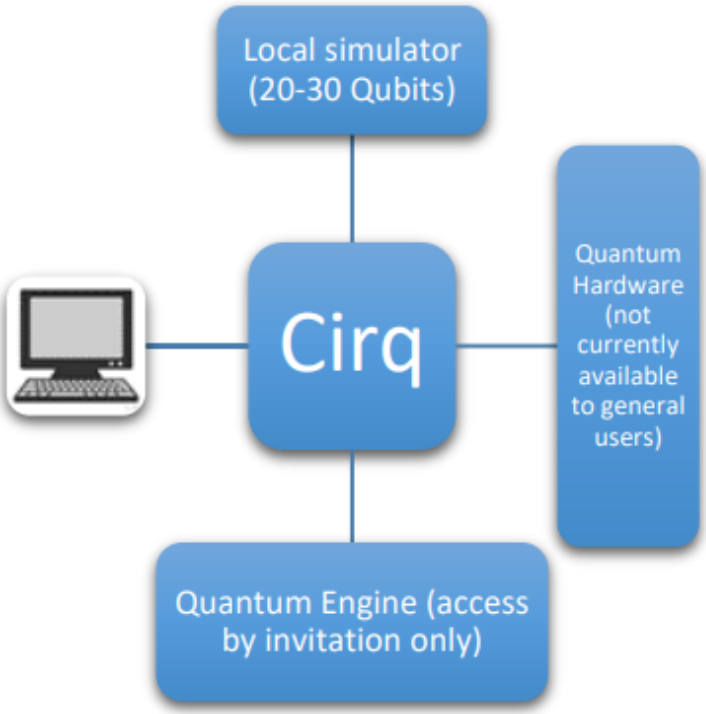

Figure 6: A general overview of Cirq

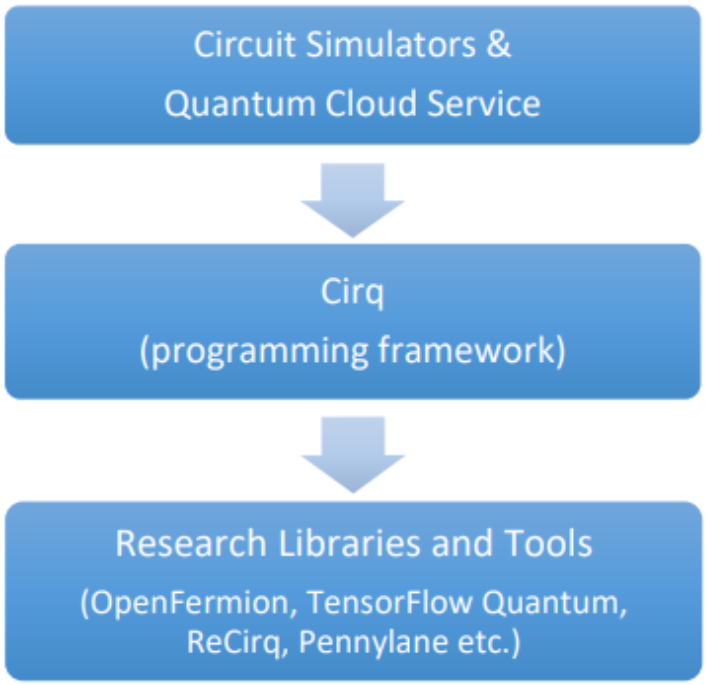

Figura 3 – Como o TensorFlow Quantum funciona com o Cirq.
Fonte: (BASAK UPAMA et al., 2022).

No ecossistema criado pela Microsoft, o Quantum Development Kit surge como uma solução que integra a linguagem de programação Q# ao ambiente Visual Studio. O funcionamento desse kit estabelece uma conexão cíclica entre as ferramentas de desenvolvimento, o computador quântico físico e o conceito de qubit topológico, disponibilizando bibliotecas padrão com funções já definidas. A plataforma também oferece recursos de simulação e depuração, o que facilita a passagem da lógica do algoritmo para sua execução final no hardware quântico (BASAK UPAMA et al., 2022).

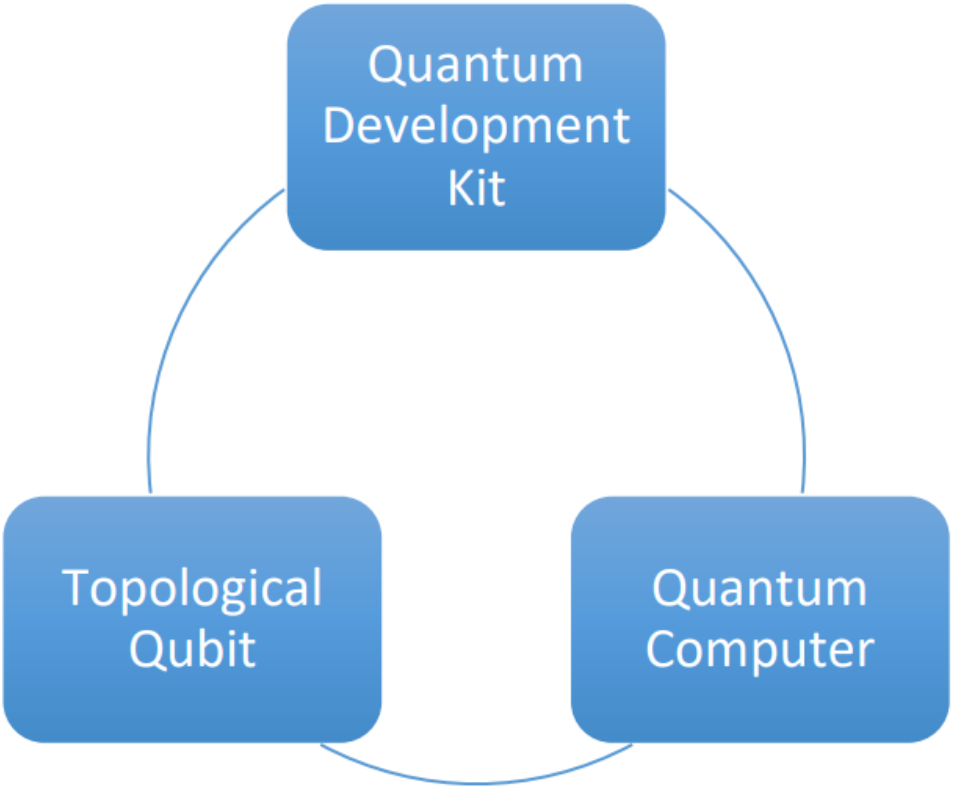

Figura 4 – Funcionamento do Microsoft Quantum Development Kit.
Fonte: (BASAK UPAMA et al., 2022).

O Quantum Information Science Kit, mais conhecido como Qiskit, consolidou-se como uma das ferramentas mais utilizadas nesse ecossistema. Baseado em Python, ele permite a criação de circuitos quânticos que podem ser executados em processadores quânticos reais por meio da nuvem. A plataforma IBM Quantum Experience organiza esse processo em diferentes módulos, que incluem tutoriais introdutórios, ferramentas para a configuração de portas quânticas, simuladores e o acesso direto às máquinas disponíveis. Um dos elementos centrais desse *framework* é o Qiskit Aer, responsável por oferecer simulações tanto em ambientes sem ruído quanto em ambientes que consideram modelos realistas de ruído, servindo de base para diversos tipos de experimentos quânticos (GILL; BUYYA, 2024; BASAK UPAMA et al., 2022; CICERO et al., 2025).

A pilha de software quântico também é ampliada por outros conjuntos de ferramen-

tas, como a Rigetti Forest, que reúne a linguagem Quil, a biblioteca pyQuil e o ambiente de simulação QVM, integrando o hardware quântico a recursos virtuais. *frameworks* como o ProjectQ permitem desenvolver programas em Python com uma sintaxe simples e traduzi-los para diferentes *backends*, enquanto o Strawberry Fields oferece plataformas de código aberto voltadas para aprendizado de máquina e simulação de diversos algoritmos. Dentro dessa estrutura, o software gerencia a comunicação entre o computador clássico e os aceleradores quânticos, coordenando o envio das instruções lógicas e o recebimento dos dados processados pelo hardware (BASAK UPAMA et al., 2022; GILL; BUYYA, 2024).

6.2 Simulação Clássica de Circuitos Quânticos

Esta seção apresenta a simulação clássica de circuitos quânticos, explicando os diferentes níveis de execução e destacando as limitações técnicas que dificultam a escalabilidade do processo.

6.2.1 O papel dos simuladores na validação de lógica quântica

A validação da lógica de um algoritmo em simuladores clássicos é uma etapa essencial no desenvolvimento de software quântico, principalmente por causa que o hardware quântico atual ainda é muito sensível a ruídos. Ao contrário dos computadores quânticos físicos, que sofrem com erros frequentes causados por interferências do ambiente, o simulador oferece um ambiente de execução estável, no qual é possível verificar se o algoritmo foi implementado corretamente, sem que falhas do equipamento interfiram nos resultados. Essa separação ajuda o desenvolvedor a entender se um resultado incorreto vem de uma falha na estrutura do algoritmo ou das instabilidades térmicas e eletromagnéticas dos processadores quânticos reais, garantindo que o código esteja matematicamente correto antes de ser executado em máquinas físicas de alto custo (CICERO et al., 2025).

As formas de simulação variam de acordo com o nível de detalhe exigido em cada teste. Na simulação em nível de porta, o simulador acompanha a evolução física das operações, imitando como os sinais de controle e os pulsos afetam o estado do sistema ao longo do tempo, o que é importante para compreender o comportamento físico de circuitos específicos. Já a simulação em nível algorítmico torna esse processo mais simples ao considerar apenas a aplicação das operações lógicas, calculando diretamente o resultado final de cada instrução sem entrar nos detalhes da física do chip. Ao ignorar essas características do hardware, essa abordagem permite uma execução mais rápida e possibilita testar circuitos com um número maior de qubits quando comparada à simulação focada na física do dispositivo (CICERO et al., 2025).

6.2.2 O custo computacional de simular sistemas quânticos em máquinas clássicas

A simulação de sistemas quânticos em computadores clássicos encontra uma limitação significativa associada ao custo computacional, que cresce de forma exponencial conforme a complexidade do sistema aumenta. O problema principal é a memória RAM, já que a quantidade de dados necessários para representar um estado quântico dobra a cada novo qubit adicionado ao circuito, seguindo a progressão de 2^n, em que n representa o número de qubits. Dentro dessa lógica, um sistema com 30 qubits já exige memória na escala de gigabytes, enquanto simulações de 50 qubits demandam capacidades na escala de petabytes, sendo que um único petabyte corresponde a 1024 terabytes, valor que supera a capacidade de armazenamento dos maiores supercomputadores atuais. O tempo de processamento necessário para realizar as operações matemáticas de cada porta passa a ser inviável, o que acaba limitando a simulação clássica a sistemas de menor escala (CICERO et al., 2025).

Para lidar com essas limitações de infraestrutura, são utilizadas ferramentas de alto desempenho voltadas à otimização do uso de hardware clássico. O QuEST, por exemplo, possibilita distribuir o processamento entre múltiplos núcleos e GPUs, explorando o paralelismo para acelerar os cálculos. Já o Intel Quantum SDK foca na eficiência de execução ao otimizar o código quântico para rodar com alta performance diretamente em processadores convencionais. Outra abordagem importante é a simulação incremental, adotada pelo *framework* qTask, que evita a necessidade de reprocessar todo o experimento ao reaproveitar estados intermediários que já foram calculados. Ao permitir que ajustes pontuais na lógica do algoritmo sejam testados de forma mais ágil, essa técnica reduz de maneira significativa o tempo total de computação (CICERO et al., 2025).

7 Implementação do Algoritmo de Grover

A aplicação prática do Algoritmo de Grover permite reforçar os conceitos teóricos apresentados ao longo do trabalho por meio da construção e da execução de um circuito quântico em ambiente simulado. Neste capítulo, o desenvolvimento é apresentado em quatro etapas principais que incluem a definição do problema de busca em bases não estruturadas, a organização lógica das operações do algoritmo, a análise dos resultados obtidos e a comparação do desempenho com abordagens clássicas.

7.1 Definição do Cenário

Para este estudo de caso, o cenário experimental foi definido para comparar as lógicas de busca clássica e quântica. No modelo clássico, utiliza-se uma lista desordenada formada por 11 elementos numéricos: [1, 3, 5, 2, 4, 9, 5, 8, 0, 7, 6]. O objetivo é localizar o número 7, que se encontra no índice 9 da lista. Esse exemplo representa o comportamento de computadores comuns, que precisam analisar os elementos um a um até encontrar o valor desejado, demonstrando uma perda de eficiência conforme a quantidade de dados aumenta.

Implementação do Oráculo Clássico

Código Python: Oráculo e Busca Linear

```python
# Lista de dados desordenados
minha_lista = [1, 3, 5, 2, 4, 9, 5, 8, 0, 7, 6]

# O Oráculo clássico que apenas verifica se a entrada é o número 7
def oraculo(entrada):
    vencedor = 7
    if entrada == vencedor:
        return True
    else:
        return False

# Busca linear que testa cada elemento no oráculo até encontrar o alvo
for indice, numero_testado in enumerate(minha_lista):
    if oraculo(numero_testado) is True:
        print(f'Vencedor encontrado no índice {indice}')
        print(f'{indice + 1} chamadas ao Oráculo foram usadas')
        break
```

```
Saída do Console
Vencedor encontrado no índice 9
10 chamadas ao Oráculo foram usadas
```

No cenário quântico, a proposta é mostrar como essa mesma tarefa de busca pode ser resolvida de forma mais eficiente, evidenciando o potencial da computação quântica. Com o uso de $n = 2$ qubits, o sistema é capaz de processar de maneira simultânea $2^n = 2^2 = 4$ estados possíveis $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. No início da execução, todos esses estados estão em superposição e apresentam exatamente 25% de probabilidade de serem medidos. Ao definir o estado $|11\rangle$ como alvo de busca, o objetivo é validar a superioridade do Algoritmo de Grover em tarefas específicas, mostrando como ele consegue amplificar a amplitude de probabilidade na resposta correta, reduzindo de forma significativa o esforço computacional quando comparado às abordagens clássicas.

7.2 Desenvolvimento e Construção do Circuito

A implementação foi realizada utilizando a plataforma Anaconda para organizar e gerenciar as bibliotecas necessárias, garantindo um ambiente isolado e estável. O Jupyter Notebook foi adotado como ambiente de trabalho, permitindo combinar o código em Python com os recursos do *framework* Qiskit de forma clara e interativa. Essa escolha facilitou o desenvolvimento do código e permitiu a visualização direta do circuito quântico. Para acompanhar o comportamento interno do sistema, foi utilizado o recurso `Statevector`, que possibilita observar as amplitudes dos estados quânticos de forma direta, sem a necessidade de executar medições que eliminariam a superposição.

A primeira etapa do desenvolvimento focou na criação de um componente isolado chamado oráculo, estruturado em um circuito de dois qubits. Na computação clássica, este bloco funciona como uma caixa-preta, que conhece a resposta de um problema, mas não a revela, apenas confirma se a tentativa está correta. Na computação quântica, o oráculo não retorna um valor, ele inverte o sinal do estado correto, realizando a marcação do estado alvo. No código, esse comando segue a forma `oraculo.cz(qubit de controle, qubit alvo)`, onde o primeiro valor indica o qubit de controle e o segundo valor indica o qubit alvo. No Qiskit, o índice do qubit (0, 1, 2, ...) corresponde ao índice do bit na string, contado da direita para a esquerda, seguindo a ordem $|q_n \dots q_1 q_0\rangle$. Dessa forma, no comando `oraculo.cz(0, 1)`, o índice 0, que no caso é o qubit de controle, aponta para o bit na posição q_0 e o índice 1, que é o qubit alvo, aponta para o bit na posição q_1. Como a porta CZ (Controlled-Z) atua de forma seletiva, invertendo o sinal (multiplicando por -1) apenas quando ambos os qubits envolvidos são iguais a 1, a aplicação nos quatro estados disponíveis ($2^n = 2^2 = 4$) resulta no seguinte comportamento:

- No estado $|00\rangle$, onde $q_0 = 0$ e $q_1 = 0$, a porta permanece inativa;
- Em $|01\rangle$, com $q_0 = 1$ e $q_1 = 0$, nada muda;
- Em $|10\rangle$, com $q_0 = 0$ e $q_1 = 1$, o sistema permanece inalterado;
- No estado $|11\rangle$, onde $q_0 = 1$ e $q_1 = 1$, o sinal é invertido, transformando o estado em $-|11\rangle$.

Assim, o oráculo marca o estado $|11\rangle$ como o vencedor para que o algoritmo identifique que sua amplitude de probabilidade deve ser amplificada.

Essa forma de endereçar os qubits é essencial para compreender a estrutura do sistema: se temos 2 qubits, operamos com 2 bits; se temos 3 qubits, operamos com 3 bits. No caso de 3 qubits ($2^n = 2^3 = 8$), resultam os 8 estados: $|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle$. No Qiskit, a ordem dos bits é $|q_2 q_1 q_0\rangle$, contando da direita para a esquerda, onde q_0, q_1 e q_2 são os índices dos bits, ou seja, em qual posição ele está, se está na primeira posição é o índice q_0, se está na segunda posição é o índice q_1 e se está na terceira posição é o índice q_2. Ao executar o comando `oraculo.cz(0, 1)`, o primeiro qubit aponta para o índice do bit q_0 e o segundo qubit aponta para o índice do bit q_1. Como exemplo, no estado $|011\rangle$, onde $q_0 = 1$ e $q_1 = 1$, o sinal é invertido. No estado $|111\rangle$, onde $q_0 = 1$ e $q_1 = 1$, o sinal também sofre inversão; já no $|110\rangle$, onde $q_0 = 0$ e $q_1 = 1$, não se inverte o sinal. Portanto, os únicos estados afetados seriam $|011\rangle$ e o $|111\rangle$, pois tanto o qubit de controle (q_0) quanto o qubit alvo (q_1) são 1. Em um cenário com 5 qubits ($2^n = 2^5 = 32$), resultando em 32 estados: $|00000\rangle, |00001\rangle, |00010\rangle$ até $|11111\rangle$, onde a ordem dos bits vai de 0 a 4, $|q_4 q_3 q_2 q_1 q_0\rangle$. Assim, é possível usar a porta CZ em qualquer par deles: no comando `oraculo.cz(0, 4)`, se a primeira posição (q_0) e a quinta posição (q_4) forem iguais a 1, o sinal do estado do qubit será invertido. Da mesma forma, em `oraculo.cz(2, 3)`, a inversão ocorre se a terceira posição (q_2) e a quarta posição (q_3) forem iguais a 1.

Configuração do Ambiente e Definição do Oráculo

Código Python: Importações e Definição do Oráculo

```python
# Ferramentas para criar e compilar circuitos quânticos
from qiskit import QuantumCircuit, transpile

# Simulador que emula a execução de um hardware quântico
from qiskit_aer import AerSimulator

# Função para visualizar os resultados das medições em forma de histograma
from qiskit.visualization import plot_histogram

# Permite ver o estado quântico do circuito sem medir
from qiskit.quantum_info import Statevector

# Biblioteca usada para cálculos numéricos e arredondamento de valores
import numpy as np

# --- PARTE 1 - Definindo o Oráculo ---

# Cria o circuito quântico do oráculo, com 2 qubits e rótulo "oraculo"
oraculo = QuantumCircuit(2, name='oraculo')

# Aplica a porta CZ para inverter o sinal do estado |11>, marca como vencedor
oraculo.cz(0, 1)

# Converte o circuito do oráculo em uma porta quântica única
porta_do_oraculo = oraculo.to_gate()

# Desenha o circuito do oráculo para visualização
oraculo.draw()
```

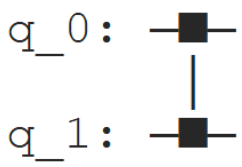

Figura 5 – Representação visual do subcircuito do Oráculo Quântico.
Fonte: Elaborado pelo autor (2026).

No circuito principal, a preparação dos dados começa com a criação da superposição. Por convenção no *framework* Qiskit, sempre que se instancia um objeto `QuantumCircuit(n)`, o estado inicial de todos os n qubits é definido automaticamente como $|0\rangle$. Para transformar esse estado base em um espaço de busca, aplica-se a porta

Hadamard (H) nos qubits 0 e 1 simultaneamente. Com isso, o estado passa de $|00\rangle$ para uma superposição uniforme: $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$. Na prática, o computador agora carrega todas as quatro possibilidades com a mesma amplitude de 0,5, permitindo que o algoritmo processe todos os estados ao mesmo tempo. Somente após essa etapa é que o oráculo é inserido via comando `append()`. Como o estado $|11\rangle$ compõe a superposição, a porta CZ do oráculo inverte o seu sinal, resultando em: $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle)$.

Preparação da Superposição e Aplicação do Oráculo

Código Python: Superposição e Validação do Oráculo

```python
# --- PARTE 2 - Superposição e Validando o Oráculo ---

# Cria o simulador quântico configurado p/ calcular as amplitudes dos estados
simulador = AerSimulator(method='statevector')

# Cria o circuito principal com 2 qubits e 2 bits clássicos para a medição
circuito_grover = QuantumCircuit(2, 2)

# Aplica Hadamard nos qubits para colocar todos os estados em superposição
circuito_grover.h([0, 1])

# Insere o oráculo no circuito principal, invertendo o sinal do estado |11>
circuito_grover.append(porta_do_oraculo, [0, 1])

# Permite visualizar as amplitudes e seus sinais sem realizar a medição
vetor = Statevector.from_instruction(circuito_grover)

# Exibe as amplitudes do estado quântico arredondadas
print("Amplitudes do estado quântico após a aplicação do oráculo:")
print(np.around(vetor, 2))

# Desenha o circuito principal para visualização
circuito_grover.draw()
```

Saída do Console

```
Amplitudes do estado quântico após a aplicação do oráculo:
[0.5+0.j 0.5+0.j 0.5+0.j -0.5+0.j]
```

```
Amplitudes do estado quântico após a aplicação do oráculo:
[ 0.5+0.j  0.5+0.j  0.5+0.j -0.5+0.j]
```

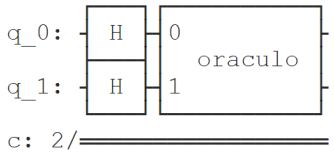

Figura 6 – Circuito principal contendo a superposição e o bloco do oráculo.
Fonte: Elaborado pelo autor (2026).

A etapa seguinte envolve a construção do Amplificador de Amplitude, também conhecido como circuito de reflexão, cujo objetivo é utilizar o sinal negativo gerado pelo oráculo para aumentar a probabilidade da resposta correta e diminuir a das incorretas. Como o hardware quântico não possui uma porta específica capaz de realizar diretamente a reflexão sobre o estado médio das amplitudes, o algoritmo utiliza uma sequência conhecida como "Sanduíche de Hadamard". Essa sequência começa com a aplicação de portas Hadamard (H) para mudar a base do sistema de forma que o estado $|00\rangle$ passe a representar a média das amplitudes de toda a superposição.

Em seguida, aplica-se a porta Z em ambos os qubits, responsável pela inversão de fase. Essa porta não altera o valor lógico do qubit, que continua sendo 0 ou 1, mas muda o sinal da amplitude quando o qubit está no estado $|1\rangle$. Em um sistema com dois qubits, isso significa que o estado $|00\rangle$ não sofre alteração, enquanto os estados $|01\rangle$, $|10\rangle$ e $|11\rangle$ têm sua fase invertida, resultando na configuração: $\frac{1}{2}(|00\rangle - |01\rangle - |10\rangle + |11\rangle)$. Como o objetivo do amplificador é inverter apenas a fase do estado médio $|00\rangle$ para realizar a reflexão desejada, e o hardware não possui uma instrução direta para isso, o algoritmo utiliza a estratégia de inverter a fase de todos os estados, exceto o $|00\rangle$. Essa inversão parcial permite a aplicação da porta CZ entre os qubits 0 e 1. Como a CZ inverte o sinal apenas quando ambos os qubits estão no estado 1, somente o estado $|11\rangle$ é afetado, retornando ao sinal negativo. O efeito combinado das portas Z e CZ mantém o estado $|00\rangle$ positivo e deixa os estados $|01\rangle$, $|10\rangle$ e $|11\rangle$ com sinal negativo, formando a configuração: $\frac{1}{2}(|00\rangle - |01\rangle - |10\rangle - |11\rangle)$.

Este padrão de sinais é intencional e faz parte do funcionamento do amplificador. As inversões de fase aplicadas dentro do chamado "Sanduíche de Hadamard" não alteram imediatamente as probabilidades de medição, mas preparam o sistema para a etapa final do processo, que é a reaplicação da porta Hadamard (H) em ambos os qubits. Nesta fase, a Hadamard mistura os estados e redistribui as amplitudes, convertendo o padrão de sinais positivos e negativos em diferenças reais de amplitude através da interferência. O resultado é que o estado marcado pelo oráculo passa a ter uma amplitude maior, enquanto os demais ficam com amplitudes reduzidas; é neste momento que a amplificação acontece de fato. O Algoritmo de Grover alcança esse efeito global de reflexão em torno da média utilizando apenas portas simples como H, Z e CZ, pois o ponto central não é qual estado teve a fase

invertida durante o processo, mas sim como a interferência final reforça o estado alvo e aumenta a sua probabilidade de ser medido.

Construção do Módulo de Reflexão

Código Python: Amplificador de Amplitude

```python
# --- PARTE 3 - Amplificador de Amplitude (Circuito de Reflexao) ---

# Cria o circuito de reflexao com 2 qubits para amplificar o estado marcado
reflexao = QuantumCircuit(2, name='reflexao')

# Aplica Hadamard para mudar a base e iniciar a reflexão do algoritmo
reflexao.h([0, 1])

# Aplica Z nos dois qubits, invertendo a fase de todos os estados exceto |00>
reflexao.z([0, 1])

# Aplica CZ para ajustar as fases e preparar a amplificação de amplitude
reflexao.cz(0, 1)

# Aplica Hadamard novamente para converter fases em amplificação de amplitude
reflexao.h([0, 1])

# Converte o circuito de reflexão em uma porta quântica única
porta_de_reflexao = reflexao.to_gate()

# Desenha o circuito de reflexão para visualização
reflexao.draw()
```

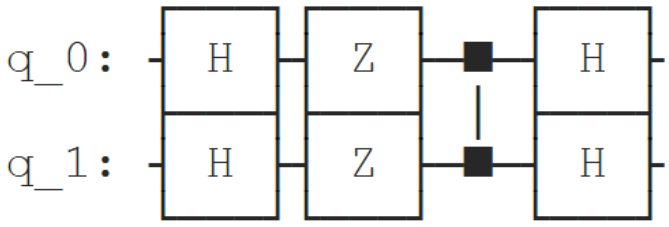

Figura 7 – Subcircuito de Reflexão responsável pela amplificação da probabilidade.
Fonte: Elaborado pelo autor (2026).

Para a execução prática, foi utilizado o `AerSimulator` configurado com o método `statevector`. Essa configuração é essencial para o desenvolvimento, pois permite ao simulador calcular e observar os valores exatos das amplitudes de probabilidade, como $0,5$ e $-0,5$, algo que não é possível visualizar em um hardware quântico real. O circuito

principal foi estruturado com o `QuantumCircuit(2, 2)`, que é composto por dois qubits e dois bits clássicos, destinados a armazenar o resultado final da busca após a medição.

A montagem final do circuito seguiu uma sequência lógica de execução. Primeiro, após a instanciação do objeto `QuantumCircuit(n)`, o *framework* define automaticamente o estado inicial de todos os n qubits como $|0\rangle$. Dessa forma, aplicou-se a porta Hadamard (H) para converter o estado inicial $|00\rangle$ em uma superposição uniforme, na qual todos os estados possuem a mesma amplitude. Em seguida, utilizou-se o comando `append()` para inserir os módulos do oráculo, através da `porta_do_oraculo`, e do circuito de reflexão, através da `porta_de_reflexao`. Enquanto o oráculo marca o estado alvo $|11\rangle$ por meio da inversão do seu sinal, o bloco de reflexão utiliza essa diferença para amplificar sua amplitude de probabilidade e reduzir a dos demais estados. Por fim, o comando `measure()` foi utilizado para mapear e armazenar a informação dos qubits q_0 e q_1 nos bits clássicos c_0 e c_1. Essa operação colapsa a superposição, transformando o estado quântico em uma saída clássica binária, como $00, 01, 10$ ou 11.

Montagem Final e Medição

Código Python: Integração do Algoritmo de Grover

```python
# --- PARTE 4 - Construção do Circuito Final ---

# Cria o simulador quântico configurado p/ calcular as amplitudes dos estados
simulador = AerSimulator(method='statevector')

# Cria o circuito principal com 2 qubits e 2 bits clássicos para a medição
circuito_grover = QuantumCircuit(2, 2)

# Aplica Hadamard nos qubits para colocar todos os estados em superposição
circuito_grover.h([0, 1])

# Insere o oráculo no circuito, invertendo o sinal do estado |11>
circuito_grover.append(porta_do_oraculo, [0, 1])

# Insere o amplificador de amplitude, aumentando a chance do estado marcado
circuito_grover.append(porta_de_reflexao, [0, 1])

# Mede os qubits e salva o resultado nos bits clássicos
circuito_grover.measure([0, 1], [0, 1])

# Desenha o circuito completo do Algoritmo de Grover
circuito_grover.draw()
```

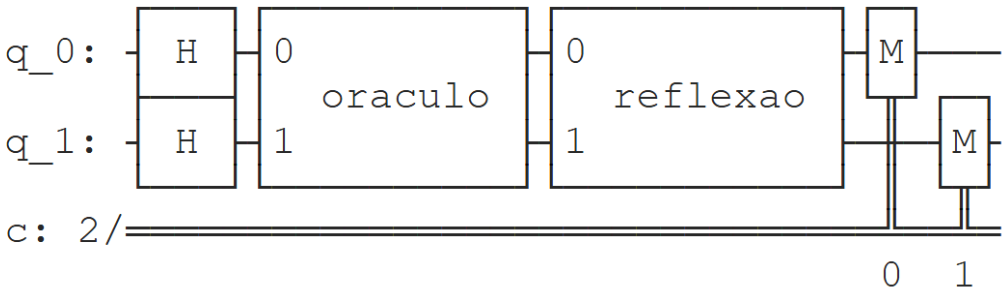

Figura 8 – Circuito completo do Algoritmo de Grover com medição final.
Fonte: Elaborado pelo autor (2026).

7.3 Execução e Análise dos Resultados

Após a conclusão da montagem lógica do circuito, inicia-se a etapa de execução prática com o uso do comando `transpile`. Esse comando funciona como um tradutor e otimizador que adapta o circuito abstrato para as instruções que o simulador quântico consegue interpretar corretamente. O Qiskit é capaz de identificar e eliminar redundâncias para economizar recursos computacionais. Por exemplo, caso existissem duas portas X seguidas, o *transpiler* reconheceria que elas se anulam e as eliminaria automaticamente. Esse processo garante que o circuito seja simplificado e que cada instrução seja perfeitamente compatível com o conjunto de portas suportado pelo *backend* escolhido, que neste estudo de caso foi o simulador `AerSimulator`.

A execução do circuito foi realizada por meio do método `run()`, configurado para realizar 1024 execuções, conhecidas como *shots*. Essa escolha de 2^{10} repetições é uma prática comum na área por ser um número grande o suficiente para gerar um histograma estável e pequeno o bastante para manter a simulação rápida. Como a computação quântica possui um comportamento probabilístico, rodar o circuito múltiplas vezes é essencial para estimar as probabilidades de medição com segurança. Caso a quantidade de *shots* fosse muito pequena, os resultados estariam sujeitos a variações estatísticas, o que poderia gerar histogramas imprecisos ou até mesmo indicar falsas probabilidades em estados incorretos, comprometendo a precisão e a confiabilidade da análise final.

O processo de execução ocorre em quatro etapas principais: primeiro, o circuito otimizado é enviado ao simulador; segundo, o simulador executa o circuito 1024 vezes, registrando um resultado clássico em cada medição; terceiro, o sistema contabiliza a frequência de cada saída possível, como $00, 01, 10$ e 11; e quarto, todos esses dados são organizados na variável `resultado`, que pode ser acessada pelo usuário. Para a análise final, utiliza-se o método `get_counts()`, que retorna um dicionário onde a chave representa o estado medido, como o alvo 11, e o valor indica quantas vezes esse estado foi detectado.

Ao analisar o histograma obtido, percebe-se que o estado $|11\rangle$ obteve todas as

frequências de medição. Esse resultado mostra de forma clara o efeito da manipulação estratégica dos sinais aplicada nas etapas anteriores, em que o oráculo e o amplificador de amplitude atuaram em conjunto para reduzir as chances de medição dos estados $|00\rangle$, $|01\rangle$ e $|10\rangle$, caracterizando uma interferência destrutiva, enquanto reforçavam o estado alvo, configurando uma interferência construtiva. Esse comportamento valida a construção lógica do algoritmo, mostrando que ele foi capaz de isolar a resposta correta com precisão absoluta dentro do ambiente simulado.

Execução do Circuito e Análise dos Resultados

Código Python: Execução e Coleta de Dados

```python
# --- PARTE 5 - Execução e Análise ---

# Compila/traduz o circuito para o formato que o simulador consegue executar
circuito_compilado = transpile(circuito_grover, simulador)

# Executa o circuito 1024 vezes para gerar uma estatística estável de medição
trabalho = simulador.run(circuito_compilado, shots=1024)

# Coleta o resultado final do trabalho executado no simulador
resultado = trabalho.result()

# Extrai as contagens de cada saída medida (00, 01, 10, 11)
contagens = resultado.get_counts()

# Mostra as contagens no terminal
print(f"Resultados: {contagens}")

# Mostra um gráfico com as contagens para visualizar o estado mais provável
plot_histogram(contagens)
```

Saída do Console

```
Resultados: {'11': 1024}
```

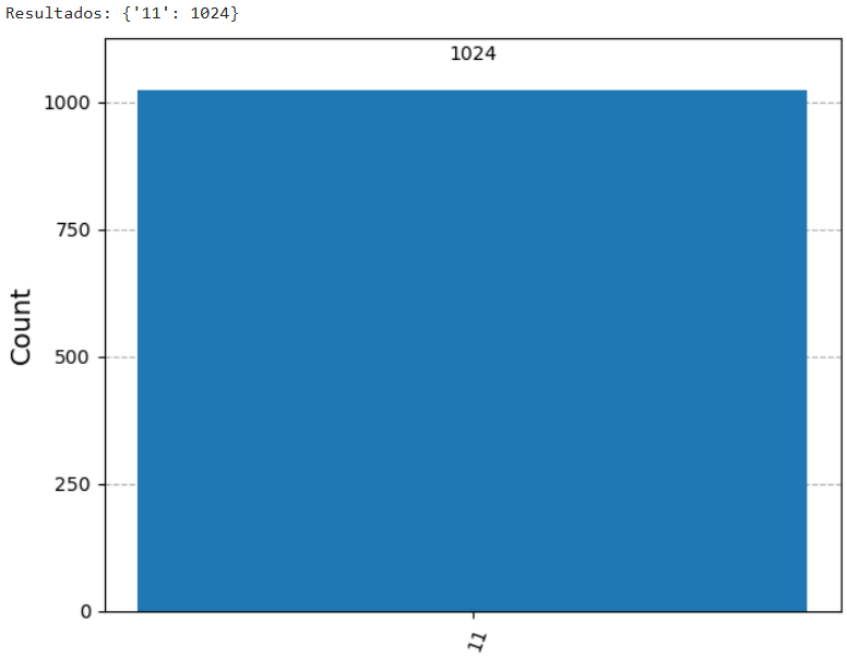

Figura 9 – Histograma de probabilidades resultante da execução no simulador.
Fonte: Elaborado pelo autor (2026).

7.4 Discussão Comparativa

Ao comparar os resultados obtidos, fica evidente a diferença de desempenho entre a busca clássica e a abordagem quântica. Na computação tradicional, o esforço de busca é proporcional ao número total de elementos (N). Assim, em um banco de dados com 1 milhão de registros, seriam necessárias, no pior caso, 1 milhão de comparações individuais. Já o Algoritmo de Grover opera em uma escala de raiz quadrada ($\sqrt{N}$), o que reduziria esse mesmo volume de processamento para apenas mil consultas. Na prática realizada neste trabalho, enquanto o processamento clássico precisou de 10 iterações para encontrar o alvo, o circuito quântico isolou o estado $|11\rangle$ após uma única iteração do algoritmo.

Essa diferença de desempenho acontece porque, na execução clássica, o sistema realiza uma busca exaustiva fazendo uma verificação item a item para encontrar o elemento desejado, dependendo da sorte ou da ordem da lista. Já na execução quântica, o oráculo marca o estado alvo ao inverter o seu sinal dentro do conjunto de estados possíveis, permitindo que o amplificador de amplitude reduza a amplitude das respostas incorretas por meio da interferência e concentre a probabilidade de medição na solução correta.

O Algoritmo de Grover obteve estabilidade de 100% no histograma dentro do ambiente simulado, com 1024 acertos em 1024 *shots*. Com isso, fica evidente o potencial da computação quântica para problemas de busca em bases de dados não estruturadas, ao demonstrar que a manipulação controlada dos estados quânticos pode superar a estratégia de força bruta usada na computação clássica.

8 Conclusão

Este trabalho permitiu uma compreensão clara das diferenças fundamentais entre a computação clássica e a computação quântica. Observou-se que a base do processamento quântico não se limita a um ganho de velocidade, mas propõe uma nova forma de manipular a informação. Enquanto a computação tradicional utiliza o bit e transistores de silício em uma lógica determinística, o modelo quântico utiliza qubits implementados por tecnologias como circuitos supercondutores e íons aprisionados. Essa mudança na estrutura do processamento possibilita explorar fenômenos como a superposição e o emaranhamento, permitindo que os dados sejam tratados juntos e de forma simultânea.

A análise teórica dos algoritmos de Shor e Grover mostrou que a lógica quântica apresenta um potencial superior para resolver problemas de fatoração e de busca em bases de dados não estruturadas em cenários específicos. No entanto, essa vantagem não representa uma substituição completa da computação clássica, que continua sendo a principal ferramenta para o processamento de grandes volumes de dados nas aplicações atuais. O trabalho evidenciou que estamos na era NISQ (*Noisy Intermediate-Scale Quantum*), na qual o ruído ambiental provoca a decoerência dos qubits, o que representa um obstáculo significativo que exige o desenvolvimento de técnicas de correção de erros para que o sistema se torne plenamente tolerante a falhas.

A aplicação prática realizada por meio do *framework* Qiskit validou os conceitos teóricos abordados, evidenciando que o atual ecossistema de software quântico amadureceu a ponto de tornar o desenvolvimento acessível a programadores e pesquisadores para a experimentação de códigos quânticos. Com a construção do circuito do Algoritmo de Grover, foi possível observar como o uso do oráculo e do amplificador de amplitude permite manipular as amplitudes de probabilidade para isolar a resposta correta de forma eficiente dentro de um ambiente controlado. A implementação mostrou na prática como a lógica quântica otimiza a busca em bases de dados não estruturadas, demonstrando o potencial da programação quântica. A simulação atua como uma ferramenta essencial dentro desse ecossistema, pois permite testar o funcionamento dos circuitos e confirmar os resultados probabilísticos esperados com precisão, sem precisar de sistemas físicos para a execução.

A computação quântica abre um caminho promissor para resolver problemas considerados intratáveis para os computadores convencionais. Esse avanço, no entanto, convive com desafios práticos, já que a transição da força bruta clássica para a manipulação de estados quânticos ainda depende de avanços científicos graduais capazes de superar as limitações atuais da área. Para trabalhos futuros, destaca-se a importância de investigar novas soluções de hardware e acompanhar os progressos na correção de erros quânticos,

fatores que serão decisivos para que essa tecnologia alcance sua maturidade e aplicação efetiva no mundo real.

Referências

BARRAL, D. et al. Review of distributed quantum computing: From single QPU to high performance quantum computing. *Computer Science Review*, 2025. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013725000231>. Citado na página 14.

BASAK UPAMA, P. et al. Evolution of quantum computing: A systematic survey on the use of quantum computing tools. In: *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. [S.l.]: IEEE, 2022. p. 1–8. Citado 10 vezes nas páginas 14, 15, 16, 17, 24, 29, 33, 37, 38 e 39.

CICERO, A. et al. Simulation of quantum computers: review and acceleration opportunities. *arXiv preprint arXiv:2410.12660v2*, 2025. Disponível em: <https://arxiv.org/pdf/2410.12660>. Citado 14 vezes nas páginas 15, 16, 24, 25, 26, 27, 28, 29, 31, 33, 34, 38, 39 e 40.

GILL, S. S.; BUYYA, R. Transforming research with quantum computing. *Computer*, IEEE Computer Society, v. 57, n. 6, p. 99–104, 2024. Citado 8 vezes nas páginas 14, 15, 16, 32, 34, 36, 38 e 39.

LAGHARI, A. A. et al. A review on quantum computing trends & future perspectives. *EAI Endorsed Transactions on Cloud Systems*, v. 8, n. 30, p. 1–10, 2022. Citado 3 vezes nas páginas 15, 23 e 30.

NANDHINI, S. S.; SINGH, H.; AKASH, U. N. An extensive review on quantum computers. *Advances in Engineering Software*, Elsevier, v. 174, p. 103337, 2022. Citado 10 vezes nas páginas 14, 21, 23, 24, 27, 28, 30, 31, 32 e 34.

NOFER, M. et al. Quantum computing. *Business & Information Systems Engineering*, Springer, v. 65, n. 4, p. 361–367, 2023. Citado 16 vezes nas páginas 14, 15, 16, 21, 22, 23, 24, 25, 26, 27, 29, 30, 32, 33, 34 e 35.

OLORUNSOGO, T.; JACKS, B. S.; AJALA, O. A. Leveraging quantum computing for inclusive and responsible ai development: A conceptual and review framework. *Computer Science & IT Research Journal*, v. 5, n. 3, p. 671–680, 2024. Citado na página 31.

RIETSCHE, R. et al. Quantum computing. *Electronic Markets*, Springer, v. 32, p. 2525–2536, 2022. Citado 8 vezes nas páginas 14, 15, 16, 21, 31, 33, 34 e 35.

SHOR, P. W. Quantum computing. *Documenta Mathematica*, Extra Volume ICM 1998, p. 467–486, 1998. Disponível em: <https://www2.math.upenn.edu/~ted/210S14/References/Shor.MAN.pdf>. Citado 6 vezes nas páginas 15, 21, 22, 23, 24 e 29.

SODIYA, E. O. et al. Quantum computing and its potential impact on U.S. cybersecurity. *World Journal of Advanced Research and Reviews*, v. 21, n. 2, p. 1396–1403, 2024. Citado 4 vezes nas páginas 14, 15, 30 e 31.

UGWUISHIWU, C. H. et al. A review on the fundamental concepts of quantum elements, efficient quantum algorithms and quantum error correcting codes. *International Journal of Scientific & Technology Research*, v. 11, p. 1–7, 2022. Citado 2 vezes nas páginas 23 e 31.