Uma introdução à Computação Quântica com o Microsoft Quantum Development Kit *

Mateus Karvat Camara[1], **Adriana Postal**[1]

[1]Centro de Ciências Exatas e da Terra – Universidade Estadual do Oeste do Paraná
Cascavel – PR – Brasil

mateus.camara@unioeste.br, adriana.postal@unioeste.br

Abstract. *Quantum Computing is a multidisciplinary area that encompasses Math, Physics and Computer Science, using highly intricate concepts. In light of this, this work aims to shed light on the core concepts of this new body of knowledge, in order to provide a solid foundation to further studies in the area. The concepts to be studied are qubits, superposition, quantum gates and quantum circuits. Based on these concepts, the algorithms of quantum teleportation, quantum paralelism and the Grover's algorithm will be discussed. For that matter, bibliographic research shall be used, as well as the Quantum Development Kit environment, where qubits, quantum entanglement, quantum gates and the Grover's algorithm will be implemented. The analysis of Grover's algorithm's implementation will be favoured due to the meaningful gain this algorithm presents in comparison to traditional algorithms that solve the same problem.*

Keywords: *Qubits. Quantum Algorithms. Quantum Development Kit.*

Resumo. *A Computação Quântica é uma área multidisciplinar que faz uso da Matemática, Física e Ciência da Computação, se valendo de conceitos com alto nível de complexidade em seus estudos. Tendo isso em vista, o presente trabalho busca elucidar os conceitos centrais dessa nova área do conhecimento, a fim de prover uma sólida base teórica a estudos posteriores acerca do tema. Os conceitos a serem estudados são: qubits, superposição, portas lógicas quânticas e circuitos quânticos. A partir desses conceitos, serão discutidos o algoritmo de teletransporte quântico, o paralelismo quântico e o algoritmo de Grover. Para tanto, será utilizada pesquisa bibliográfica da literatura existente e o ambiente Quantum Development Kit, onde serão implementados qubits, a propriedade de entrelaçamento, portas lógicas quânticas e o algoritmo de Grover. A análise da implementação do algoritmo de Grover receberá destaque em virtude do ganho significativo que tal algoritmo apresenta em relação a algoritmos não quânticos que solucionam o mesmo problema.*

Palavras-chave: *Qubits. Algoritmos quânticos. Quantum Development Kit.*

1. Introdução

Computação Quântica é um modelo computacional que tem como base a Mecânica Quântica em detrimento da elétrica utilizada em computadores tradicionais. Enquanto

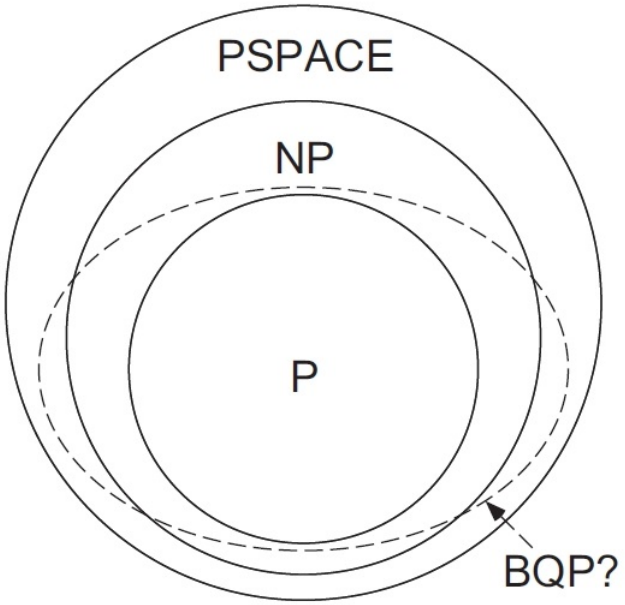
*An introduction to Quantum Computing with Microsoft Quantum Development Kit

um computador convencional realiza seus cálculos utilizando bits (0 ou 1), um computador quântico realiza seus cálculos utilizando qubits (bits quânticos), os quais têm uma infinidade de estados possíveis, além dos clássicos estados de 0 e 1 [Gershon 2017].

Valendo-se de propriedades da Mecânica Quântica como superposição e entrelaçamento, os computadores quânticos permitem a realização de cálculos em tempo muito menor que o levado por computadores tradicionais (até mesmo supercomputadores) [Gershon 2017].

A análise dos recursos necessários à realização dos cálculos computacionais (entre eles, o tempo) é feita pela Teoria da Computação. Essa área do conhecimento separa os problemas computáveis em classes de problemas: os problemas do tipo P são problemas cuja resolução se dá em tempo polinomial (ou seja, o tempo necessário à resolução do problema é uma função polinomial de n, sendo n o tamanho de entrada do problema), já problemas do tipo NP são problemas cuja resolução se dá em tempo não polinomial, mas que podem ser verificados em tempo polinomial. Problemas do tipo NP têm sua resolução dificultada em computadores tradicionais, podendo levar milhares de anos para serem resolvidos num supercomputador. Ainda não se sabe qual classe de problemas seriam resolvidos em tempo polinomial por computadores quânticos (classe que poderia ser denominada BQP, do inglês *bounded-error quantum polynomial time*), mas acredita-se que eles possam resolver problemas do tipo NP melhor que computadores tradicionais, graças a seu princípio de funcionamento [Nielsen and Chuang 2010]. Logo, a classe BQP conteria a classe P e parte da classe NP, conforme Figura 1.

Figura 1. Classe BQP



Fonte: [Nielsen and Chuang 2010].

Tendo isso em vista, a aplicação prática da Computação Quântica promete proporcionar grandes avanços em determinadas áreas. A criptografia, por exemplo, pode ser revolucionada, visto que atualmente ela é baseada na fatoração de grandes números. Enquanto um computador tradicional levaria muitos anos efetuando os cálculos necessários

para descobrir a chave de um sistema de criptografia, um computador quântico realizaria essas operações em minutos [Bennett 2016]. Uma criptografia pós-quântica utilizaria dados quânticos para o compartilhamento de chaves, sendo um sistema ainda mais seguro que o utilizado atualmente [Nielsen and Chuang 2010].

Além disso, a simulação de sistemas quânticos (como moléculas e proteínas) promete ser enormemente aprimorada com o uso de computadores quânticos. Um exemplo utilizado por [Nielsen and Chuang 2010] é o de um sistema quântico composto por 50 qubits (os quais, a critério de exemplificação, podem ser considerados 50 átomos). Esse sistema é composto por 2^{50}, ou 10^{15}, amplitudes complexas. Atribuindo precisão de 128 bits para cada amplitude, seriam necessários 32 petabytes para simular esse sistema num computador tradicional. Em contrapartida, bastariam 50 qubits para simular esse sistema num computador quântico.

Visto que qubits são entidades matemáticas, computadores quânticos podem ser construídos a partir de quaisquer sistemas físicos que tenham as mesmas propriedades que qubits. Em outras palavras, quaisquer sistemas físicos que apresentem propriedades quânticas [Nielsen and Chuang 2010].

Atualmente, há três principais sistemas físicos utilizados na construção de computadores quânticos, cada um com suas vantagens e desvantagens. Técnicas ópticas fazem uso de radiação eletromagnética. Nelas, a informação é armazenada em fótons, os quais são altamente estáveis para essa função. Uma grande desvantagem desse modelo é que fótons não interagem entre si, necessitando de um mediador, como um átomo, para as interações. Outro sistema, bastante utilizado atualmente, é o de aprisionamento de átomos. Nesse sistema, tanto íons quanto átomos neutros podem ser aprisionados e a informação é armazenada nos próprios átomos. Por fim, sistemas de ressonância nuclear magnética (NMR) armazenam a informação no spin dos elétrons. Esses últimos dois sistemas têm como vantagem a interação facilitada entre os qubits, mas têm como desvantagem sua instabilidade, necessitando de temperaturas inferiores a um kelvin para operarem ($1K = -272,15°C$) [Nielsen and Chuang 2010].

O presente trabalho foi desenvolvido por meio de pesquisa bibliográfica à literatura existente acerca do tema e do uso do ambiente Quantum Development Kit. Na Seção 2.1, serão estudados os conceitos básicos da Computação Quântica: qubits e circuitos quânticos. Em seguida, na Seção 2.2, serão estudados algoritmos quânticos simples como o teletransporte quântico e o paralelismo quântico, bem como o importante algoritmo de Grover, o qual apresenta ganho significativo em relação a algoritmos clássicos na resolução de problemas de busca não estruturada. Na Seção 2.3, o ambiente Quantum Development Kit será descrito, sendo seu funcionamento demonstrado na Seção 3.1 pela implementação de circuitos quânticos básicos. Por fim, a Seção 3.2 analisa a implementação do algoritmo de Grover no Quantum Development Kit, demonstrando o ganho deste algoritmo em relação a algoritmos clássicos. A Seção 4 conclui o trabalho analisando a situação atual da Computação Quântica e a relevância deste trabalho para o tema.

2. Revisão bibliográfica

2.1. Conceitos básicos

Esta seção apresenta os conceitos fundamentais da Computação Quântica necessários ao estudo dos algoritmos quânticos, sendo eles Qubits e Circuitos Quânticos.

2.1.1. Qubits

Qubits são entidades matemáticas, o que significa que eles podem ser estudados isoladamente, não importando o meio físico utilizado para sua realização [Nielsen and Chuang 2010].

Um qubit pode existir numa superposição dos estados clássicos 0 e 1. Essa superposição de estados colapsa ao ser realizada uma medição, quando o estado se torna 0 ou 1 [Bennett 2016].

A representação mais simples para um qubit é pela notação de Dirac (Equação 1).

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

Nessa notação, α é a amplitude de $|0\rangle$ e β é a amplitude de $|1\rangle$. $|\alpha|^2$ representa a probabilidade de que se obtenha 0 ao medir o qubit e $|\beta|^2$ a probabilidade de se obter 1. É importante ressaltar que α e β são variáveis complexas e que $|\alpha|^2 + |\beta|^2 = 1$ [Nielsen and Chuang 2010].

Um qubit pode também ser representado como um vetor num espaço vetorial bidimensional complexo - noção importante para operações realizadas por portas lógicas quânticas. A notação de um qubit de forma vetorial se dá na forma $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, sendo α e β as mesmas variáveis da notação de Dirac [Nielsen and Chuang 2010].

A superposição não ocorre apenas para um qubit isolado, mas também para um grupo de qubits [Nielsen and Chuang 2010]. Um par de qubits, por exemplo, pode existir em superposição, de modo que cada estado computacional possível tem uma amplitude associada (Equação 2).

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{2}$$

Assim, um par de qubits apresenta novos estados possíveis em relação a cada qubit tomado individualmente [Chuang and Shor 2018].

Um estado importante para a Computação Quântica é o chamado estado de Bell (também conhecido como par EPR, em homenagem a Einstein, Podolsky e Rosen, que o descobriram) [Nielsen and Chuang 2010]. Dois qubits constituintes de um par EPR têm uma forte relação entre si, que é utilizada em vários algoritmos quânticos. Tomando, por exemplo, o estado β_{00} na Equação 3.

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{3}$$

A probabilidade de se obter o estado 00 na medição é de 50% e a de se obter o estado 11 também é de 50%. Essa correlação permite o uso de propriedades únicas, inexistentes fora da Mecânica Quântica [Nielsen and Chuang 2010]. Além do estado β_{00}, há também os estados representados nas Equações 4, 5 e 6.

$$|\beta_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \tag{4}$$

$$|\beta_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \tag{5}$$

$$|\beta_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \tag{6}$$

2.1.2. Circuitos quânticos

Tal qual na computação tradicional, a Computação Quântica faz uso de portas lógicas para realizar operações em seus dados [Chuang and Shor 2018]. Tendo em vista a representação vetorial de qubits, as portas lógicas quânticas podem ser descritas como matrizes. Algumas portas quânticas bastante importantes são as portas NOT (Equação 7), Z (Equação 8) e Hadamard (Equação 9).

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} (Transforma\ \alpha|0\rangle + \beta|1\rangle\ em\ \beta|0\rangle + \alpha|1\rangle) \tag{7}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} (Transforma\ \alpha|0\rangle + \beta|1\rangle\ em\ \alpha|0\rangle - \beta|1\rangle) \tag{8}$$

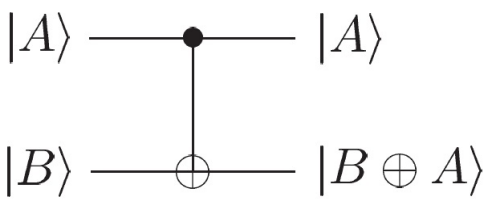
$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} (Transforma\ \alpha|0\rangle + \beta|1\rangle\ em\ \alpha\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle - |1\rangle}{\sqrt{2}}) \tag{9}$$

Para portas que agem sobre múltiplos qubits, é mais conveniente representá-las através de um circuito. Na representação de um circuito quântico, uma linha representa a passagem do tempo para cada qubit, enquanto quadrados representam portas lógicas. À esquerda da linha tem-se o estado do qubit de entrada, enquanto à direita tem-se o estado de qubit de saída após a realização das operações [Nielsen and Chuang 2010].

No circuito da Figura 2, o círculo fechado representa que aquele é um qubit de controle. Ele realizará a operação apenas se tiver valor igual a 1. Já o símbolo $\oplus$ representa adição módulo 2 (ou seja, o resto da divisão da soma de A e B por 2). Assim, a porta CNOT inverte B caso o valor de A seja 1, e se o valor de A for 0, B mantém-se intacto [Chuang and Shor 2018].

Para exemplificar a passagem de dois qubits por uma porta CNOT, o estado da Equação 10 se tornaria o estado da Equação 11. Percebe-se que, para as amplitudes α_{10}

Figura 2. Porta CNOT (NOT Controlado)



Fonte: [Nielsen and Chuang 2010].

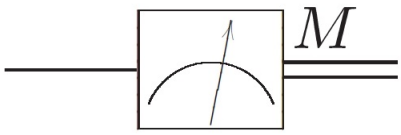
e α_{11}, em que o primeiro qubit é 1, houve inversão do segundo qubit, de acordo com o princípio de funcionamento da porta CNOT.

$$|\psi_0\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{10}$$

$$|\psi_1\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|11\rangle + \alpha_{11}|10\rangle \tag{11}$$

Na representação de circuitos quânticos, indica-se a medição de qubits pelo símbolo da Figura 3. Nele, as linhas duplas após o símbolo de medição indicam que o qubit colapsou para seu estado clássico, isto é, para um bit [Nielsen and Chuang 2010].

Figura 3. Medição de qubit



Fonte: [Nielsen and Chuang 2010].

2.2. Algoritmos Quânticos

Utilizando os conceitos citados anteriormente, discutiremos o algoritmo de teletransporte quântico, o funcionamento do paralelismo quântico e o algortimo de Grover.

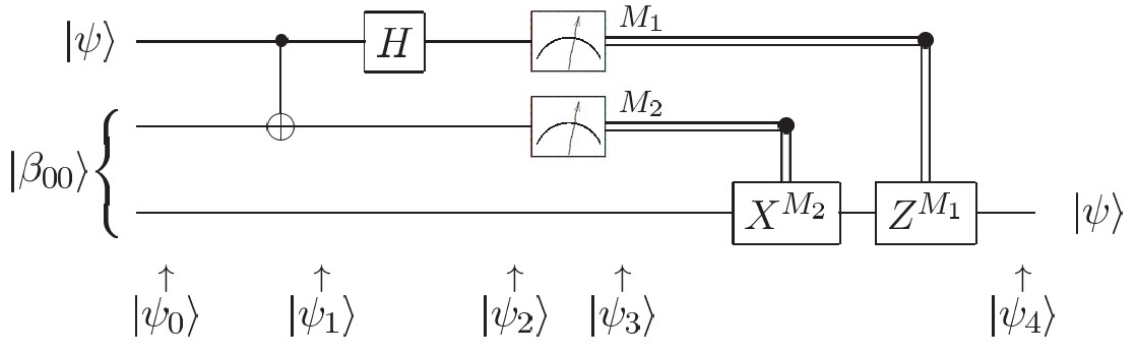
2.2.1. Teletransporte Quântico

Para ilustrar o algoritmo de teletransporte quântico, apresentado por [Nielsen and Chuang 2010], suponhamos duas pessoas: Alice e Bob. Alice tem um qubit desconhecido ψ que deseja enviar para Bob. Eles compartilham um par EPR e ela tem que enviar seu qubit desconhecido a Bob tal qual está.

Para que esse procedimento seja realizado, pode-se realizar o algoritmo de Teletransporte quântico, representado na Figura 4. Suponhamos que o par EPR compartilhado por Alice e Bob seja o par $|\beta_{00}\rangle$ (apresentado anteriormente na Equação 3).

O estado inicial é descrito pela Equação 12.

Figura 4. Representação do algoritmo de teletransporte quântico



Fonte: [Nielsen and Chuang 2010].

$$|\psi_0\rangle = |\psi\rangle|\beta_{00}\rangle = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)] \tag{12}$$

Passando o primeiro e segundo qubit por uma porta CNOT, tem-se o estado da Equação 13.

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)] \tag{13}$$

Passando então o primeiro qubit por uma porta Hadamard, tem-se o estado da Equação 14.

$$|\psi_2\rangle = \frac{1}{2}[\alpha(|0\rangle + |1\rangle)(|00\rangle + |11\rangle) + \beta(|0\rangle - |1\rangle)(|10\rangle + |01\rangle)] \tag{14}$$

Esse termo pode ser reescrito na forma representada pela Equação 15.

$$|\psi_2\rangle = \frac{1}{2}[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)] \tag{15}$$

Assim, conforme Equação 15, se Alice realizar uma medição e obtiver como resultado 00, então o qubit de Bob estará no estado $\alpha|0\rangle + \beta|1\rangle$. Similarmente para outros resultados da medição de Alice, podemos determinar, a partir da Equação 15, o estado do qubit de Bob. Sabendo o resultado da medição de Alice, Bob pode "arrumar" seu qubit para obter o qubit inicial ψ de Alice.

Na Figura 4, a medição de Alice é representada pelos bits tradicionais M_1 e M_2. As operações que Bob deve realizar sobre seu qubit são representadas pelas portas lógicas X^{M_2} e Z^{M_1}. Ou seja, se $M_2 = 1$, Bob deve passar seu qubit pela porta lógica X, não devendo realizar tal operação caso $M_2 = 0$, seguindo a mesma lógica para a porta lógica Z e o bit M_1.

A Tabela 1 resume os possíveis estados encontrados após a medição, apresentando o resultado da medição de Alice, os bits M_1 e M_2, os respectivos estados do qubit de Bob, as portas lógicas pelas quais Bob deve passar seu qubit e o resultado final, sempre igual ao qubit ψ de Alice.

Tabela 1. Possíveis estados ψ_3 (M_1, M_2 e Qubit de Bob) e posterior estado ψ_4

Medição de Alice	M_1	M_2	**Qubit de Bob**	**Portas lógicas**	**Qubit final ψ_4**
00	0	0	$\alpha\|0\rangle + \beta\|1\rangle$	-	$\alpha\|0\rangle + \beta\|1\rangle$
01	0	1	$\alpha\|1\rangle + \beta\|0\rangle$	X	$\alpha\|0\rangle + \beta\|1\rangle$
10	1	0	$\alpha\|0\rangle - \beta\|1\rangle$	Z	$\alpha\|0\rangle + \beta\|1\rangle$
11	1	1	$\alpha\|1\rangle - \beta\|0\rangle$	X e Z	$\alpha\|0\rangle + \beta\|1\rangle$

Fonte: [Nielsen and Chuang 2010].

Percebe-se que Bob precisa da informação acerca da medição de Alice para obter ψ. Dessa forma, o teletransporte quântico não se trata de um teletransporte no sentido usual do termo, visto que Bob precisa receber essa informação, a qual tem sua velocidade de transmissão limitada pela velocidade da luz. Mesmo assim, o teletransporte quântico se mostra uma forma simples e eficiente de compartilhar qubits.

2.2.2. Paralelismo quântico

O paralelismo quântico é a propriedade que permite computadores quânticos calcularem uma função $f(x)$ para diferentes valores de x simultaneamente [Nielsen and Chuang 2010].

Para ilustrar o paralelismo quântico em uma função, consideremos um computador quântico de dois qubits que inicia no estado $|x, y\rangle$. Com uma sequência apropriada de portas lógicas quânticas, é possível transformar esse estado em $|x, y \oplus f(x)\rangle$. Logo, se $y = 0$, o estado final do segundo qubit é apenas o valor de $f(x)$ [Nielsen and Chuang 2010]. A partir disso, se $x = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$, o estado final resultante é representado na Equação 16.

$$\frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}} \tag{16}$$

Esse estado é importante, pois nele cada termo corresponde a um valor distinto de $f(x)$, mostrando que o valor da função foi calculado simultaneamente para valores distintos de x.

Esse procedimento pode ser generalizado para funções atuando sobre um número arbitrário de bits, bastando passar cada qubit por uma porta Hadamard simultaneamente. A passagem de n qubits por n portas Hadamard simultaneamente é descrita pela notação $H^{\otimes n}$. Desse modo, será criado um estado no qual há a superposição de todos os estados computacionais possíveis para aquele número de bits [Nielsen and Chuang 2010].

Resultados notáveis surgem quando pensamos num grande número de bits, visto que a sequência de operações realizadas sobre eles seria executada uma única vez, tendo como resultado o valor de $f(x)$ para todos os diferentes valores de x utilizados. Em

contrapartida, um computador tradicional realizaria o cálculo de $f(x)$ para cada x sequencialmente.

2.2.3. Algoritmo de Grover

O algoritmo de Grover é também chamado de algoritmo quântico de busca por realizar busca não estruturada, isto é, a busca em um conjunto de entradas sem ordenação prévia. Em um problema de busca não estruturada, tem-se uma função booleana $f : X \to \{0, 1\}$ atuando sobre um conjunto $X = \{x_1, x_2, ..., x_N\}$ de N elementos tal que o objetivo é encontrar os M elementos x^* pertencentes a X tal que $f(x^*) = 1$ [Wright 2015]. Em outras palavras, tem-se M elementos que satisfazem uma propriedade desejada (indicados por x^*) em um conjunto de N elementos, tal que a função f retorna 1 para estes M elementos. Visto que a busca a ser realizada pelo algoritmo é não estruturada, algoritmos clássicos requerem um tempo de ordem $\Theta(N/M)$ para encontrar os elementos x^*. O algortimo de Grover, por sua vez, utiliza de propriedades quânticas e requer apenas um tempo de ordem $\Theta(\sqrt{N/M})$ para encontrar o elemento [Nielsen and Chuang 2010].

Para realizar a busca, analisaremos os elementos com base em seus índices, de modo que o conteúdo destes elementos não é relevante para o algoritmo em si, mas sim para a função $f(x)$, que operará com base nos índices destes elementos e identificará aqueles que atendem à propriedade desejada. Assumindo que $N = 2^n$ (caso N não seja uma potência de 2, podemos acrescentar lixo ao conjunto até que se tenha um n inteiro que satisfaça essa igualdade [Wright 2015]), precisaremos de n qubits para armazenar os índices. Assim, a função descrita anteriormente é simplificada para $f : \{0, 1\}^n \to \{0, 1\}$.

O algoritmo inicia com n qubits inicializados em $|0\rangle$. O primeiro passo é criar uma superposição de todos os estados possíveis, a qual é efetuadada aplicando a porta Hadamard sobre esse conjunto de qubits, de modo bastante similiar ao paralelismo quântico. Tal superposição é descrita na Equação 17 [Strubell 2011].

$$|\psi\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \sum_{x=0}^{2^n-1} \frac{1}{\sqrt{N}}|x\rangle \tag{17}$$

A seguir, realiza-se uma série de transformações conhecida como *Iteração de Grover* ou *Operador de Grover* [Nielsen and Chuang 2010]. Nela, cada iteração se dá pela aplicação de um oráculo que realiza a função f sobre os qubits superpostos, seguida da realização da *transformada difusa*, a qual realiza inversão em torno da média. A Iteração de Grover é descrita por diferentes autores utilizando as diferentes representações de qubits e operações sobre qubits: [Watrous 2006a] e [Watrous 2006b] utilizam a notação angular, [Strubell 2011] utiliza a notação algébrica, [Wright 2015] utiliza a notação artimética e [Nielsen and Chuang 2010] aborda todas as três notações. Tomaremos como base a notação aritmética.

O oráculo a ser utilizado é uma "caixa-preta", isto é, um conjunto de operadores que transforma uma determinada entrada na saída esperada, sem que se faça necessário o conhecimento de seu funcionamento interno. A utilização do oráculo pelo algoritmo de Grover é um ponto positivo [Watrous 2006b], pois ele pode ser aplicado a qualquer

problema de busca, independente dos tipos de dados a serem analisados, visto que cada problema utilizaria um oráculo distinto. Utilizando as notações quânticas, pode-se descrever o oráculo como um operador O, conforme Equação 18, em que um qubit $|q\rangle$ é invertido se $f(x) = 1$ e mantém-se intacto caso contrário. Nota-se que o objetivo do oráculo é implementar a função $f(x)$ sobre o conjunto inicial de qubits, destacando os elementos desejáveis do conjunto de dados.

$$|\psi\rangle|q\rangle \xrightarrow{\text{O}} |\psi\rangle|q \oplus f(x)\rangle \tag{18}$$

É, então, conveniente utilizar $|q\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$, pois a inversão resultará no estado $-|q\rangle$ (a inversão de $|q\rangle$ resulta em $\frac{-|0\rangle + |1\rangle}{\sqrt{2}} = -|q\rangle$). Tendo isso em vista, pode-se escrever a ação do oráculo na forma da Equação 19.

$$|\psi\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \xrightarrow{\text{O}} (-1)^{f(x)} |\psi\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{19}$$

Pode-se observar que o qubit $|q\rangle$ não é alterado pela atuação do oráculo, tal que ele pode ser omitido da Equação 19, resultando na Equação 20. A partir dela, dizemos que o oráculo *marca* as soluções para o problema ao inverter a fase da solução [Nielsen and Chuang 2010].

$$|\psi\rangle \xrightarrow{\text{O}} (-1)^{f(x)} |\psi\rangle \tag{20}$$

Após a aplicação do oráculo sobre o conjunto de qubits $|\psi\rangle$, realiza-se a *transformada difusa*, que é descrita aritmeticamente pela Equação 21 [Wright 2015].

$$\sum_{x=0}^{2^n-1} \alpha_x |x\rangle \mapsto \sum_{x=0}^{2^n-1} (2\mu - \alpha_x) |x\rangle \tag{21}$$

Na Equação 21, α_x é a amplitude de um qubit x e μ é a média das amplitudes de todos os qubits, conforme Equação 22 [Wright 2015].

$$\mu = \frac{1}{N} \sum_{x=0}^{2^n-1} \alpha_x \tag{22}$$

A partir disso, a *Iteração de Grover*, que consiste na aplicação sucessiva dos passos descritos nas Equações 20 e 21, deve ser realizada por um número R de vezes, sendo R descrito pela Equação 23. Por fim, realiza-se a medição do estado resultante.

$$R \leq \left\lfloor \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rfloor \tag{23}$$

Para melhor compreender o algoritmo, tomemos o exemplo dado por [Strubell 2011] de um sistema contendo $N = 8$ estados em que se busca um estado x^*

de índice 011. Vale ressaltar que o conteúdo de x^* é irrelevante ao algoritmo, apenas seu índice 011 será considerado. Visto que utilizaremos o Algoritmo de Grover para encontrar tal elemento, devemos verificar se N é potência de 2, ou seja, se existe n tal que $N = 2^n$. Como $2^3 = 8$, então $n = 3$ e não é necessário acrescentar lixo ao conjunto. Temos, então, uma função booleana $f : \{0,1\}^3 \rightarrow \{0,1\}$, em que $f(011) = 1$ e $f(i) = 0$ para todos os outros índices i diferentes de 011.

O algortimo de Grover inicia com n qubits inicializados em $|0\rangle$, tal que, sendo $n = 3$, teremos o estado $|\psi_0\rangle$ descrito pela Equação 24.

$$|\psi_0\rangle = 1|000\rangle \tag{24}$$

O próximo passo é aplicar a porta Hadamard ao estado $|\psi_0\rangle$, o que criará uma superposição de todos os estados possíveis para 3 qubits, tendo cada estado a mesma amplitude. Conforme Equação 17, o estado resultante será igual ao somatório dos estados x com amplitude $\frac{1}{\sqrt{N}}$. Como $N = 8$, $\sqrt{N} = 2\sqrt{2}$ e como x varia de 0 a $2^n - 1$, temos x variando de 0 a 7 (que são representados, no sistema numérico binário, variando de 000 a 111). Com base nessas considerações, a superposição descrita é representada na Equação 25.

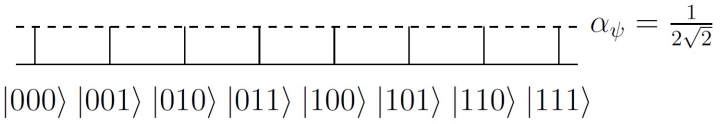
$$H^3|000\rangle = \frac{1}{2\sqrt{2}}|000\rangle + \frac{1}{2\sqrt{2}}|001\rangle + ... + \frac{1}{2\sqrt{2}}|111\rangle = \sum_{x=0}^{7} \frac{1}{2\sqrt{2}}|x\rangle = |\psi_1\rangle \tag{25}$$

No estado $|\psi_1\rangle$, a média das amplitudes μ_1 pode ser calculada somando-se todas as amplitudes e dividindo pelo número de estados (conforme Equação 22), o que é feito na Equação 26.

$$\mu_1 = \frac{1}{8}\sum_{x=0}^{7} \alpha_x = \frac{1}{8}\frac{8}{2\sqrt{2}} = \frac{1}{2\sqrt{2}} \tag{26}$$

Tendo isso em vista, o estado $|\psi_1\rangle$ é graficamente representado na Figura 5, em que a amplitude de cada estado é representada por uma barra vertical proporcional à amplitude daquele estado. Como todas as amplitudes são iguais, todas as barras verticais têm mesmo comprimento, o qual é equivalente a α_ψ.

Figura 5. Representação do estado $|\psi_1\rangle$



Fonte: [Strubell 2011].

Antes de iniciar as iterações de Grover, é importante verificar quantas iterações devem ser realizadas. Utilizando a Equação 23, tem-se que 2 iterações de Grover devem ser realizadas, o que é explicitado pela Equação 27.

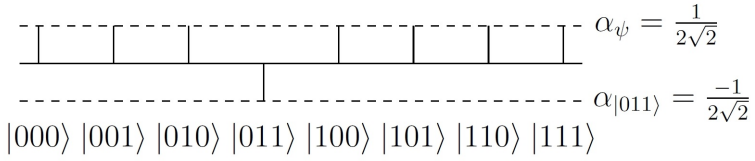
$$R \leq \left\lfloor \frac{\pi}{4}\sqrt{\frac{8}{1}} \right\rfloor = \left\lfloor \frac{\pi 2\sqrt{2}}{4} \right\rfloor = 2 \tag{27}$$

Iniciando a primeira iteração, aplica-se o oráculo sobre o estado $|\psi_1\rangle$, o qual, conforme Equação 20, inverterá a amplitude de todos os elementos em que $f(x) = 1$ e manterá a mesma amplitude para todos os elementos em que $f(x) = 0$. Como descrito anteriormente, apenas o estado $|011\rangle$ satisfaz a função booleana do oráculo, originando o estado $|\psi_2\rangle$, representado na Equação 28.

$$|\psi_2\rangle = \frac{1}{2\sqrt{2}}|000\rangle + \frac{1}{2\sqrt{2}}|001\rangle + \frac{1}{2\sqrt{2}}|010\rangle - \frac{1}{2\sqrt{2}}|011\rangle + ... + \frac{1}{2\sqrt{2}}|111\rangle \tag{28}$$

Nota-se, no estado $|\psi_2\rangle$, que a amplitude de todos os estados é $\frac{1}{2\sqrt{2}}$, exceto do estado $|011\rangle$, em que a amplitude é $-\frac{1}{2\sqrt{2}}$. Por esse motivo, a representação gráfica deste qubit posicionará a barra vertical do estado $|011\rangle$ abaixo do eixo, o que pode ser observado na Figura 6. Percebe-se que os demais estados não foram alterados.

Figura 6. Representação do estado $|\psi_2\rangle$



Fonte: [Strubell 2011].

A mudança no estado $|011\rangle$ afeta, porém, o valor da média das amplitudes. Assim, a média das amplitudes μ_2 para o estado $|\psi_2\rangle$ é calculada na Equação 29.

$$\mu_2 = \frac{1}{8}\sum_{x=0}^{7}\alpha_x = \frac{1}{8}\frac{6}{2\sqrt{2}} = \frac{3}{8\sqrt{2}} \tag{29}$$

A aplicação da transformada difusa (descrita na Equação 21) a $|\psi_2\rangle$ resulta na Equação 30.

$$\sum_{x=0}^{7}\alpha_x|x\rangle \mapsto \sum_{x=0}^{7}(2\frac{3}{8\sqrt{2}} - \alpha_x)|x\rangle \tag{30}$$

Sabendo-se que α_x representa a amplitude do estado x e que todos os estados, exceto $|011\rangle$, têm amplitude $\frac{1}{2\sqrt{2}}$, a transformada difusa alterará todos esses estados igualmente, reduzindo sua amplitude, como se observa na Equação 31.

$$2\frac{3}{8\sqrt{2}} - \frac{1}{2\sqrt{2}} = \frac{3}{4\sqrt{2}} - \frac{1}{2\sqrt{2}} = \frac{1}{4\sqrt{2}} \tag{31}$$

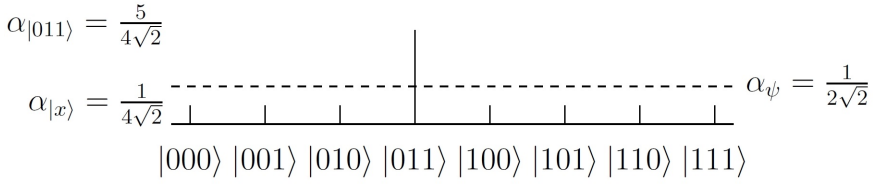
Todavia, para o estado $|011\rangle$, a amplitude α_{011} é igual a $-\frac{1}{2\sqrt{2}}$, tal que sua transformada difusa será distinta da transformada dos demais estados, o que pode ser observado na Equação 32.

$$2\frac{3}{8\sqrt{2}} - \left(-\frac{1}{2\sqrt{2}}\right) = \frac{3}{4\sqrt{2}} + \frac{1}{2\sqrt{2}} = \frac{5}{4\sqrt{2}} \tag{32}$$

Devido a isso, a aplicação da transformada difusa sobre $|\psi_2\rangle$ resulta no estado $|\psi_3\rangle$, o qual é representado na Equação 33 e na Figura 7.

$$|\psi_3\rangle = \frac{1}{4\sqrt{2}}|000\rangle + \frac{1}{4\sqrt{2}}|001\rangle + \frac{1}{4\sqrt{2}}|010\rangle + \frac{5}{4\sqrt{2}}|011\rangle + ... + \frac{1}{4\sqrt{2}}|111\rangle \tag{33}$$

Figura 7. Representação do estado $|\psi_3\rangle$

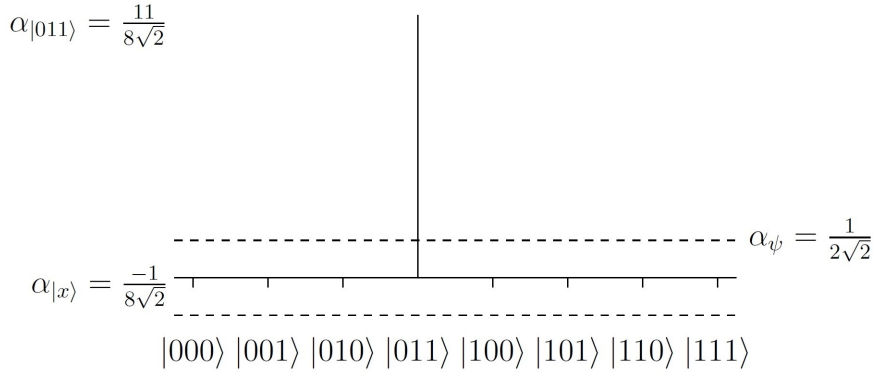


Fonte: [Strubell 2011].

Encerrada a primeira iteração de Grover, realiza-se a segunda iteração. Assim como na primeira, efetua-se a aplicação do oráculo sobre o estado e então a transformada difusa. A segunda iteração tem como resultado o estado $|\psi_4\rangle$, representado na Equação 34 e na Figura 8. Destaca-se a amplitude significativamente maior do estado $|011\rangle$, em relação aos demais estados.

$$|\psi_4\rangle = -\frac{1}{8\sqrt{2}}|000\rangle - \frac{1}{8\sqrt{2}}|001\rangle - \frac{1}{8\sqrt{2}}|010\rangle + \frac{11}{8\sqrt{2}}|011\rangle - ... - \frac{1}{8\sqrt{2}}|111\rangle \tag{34}$$

Figura 8. Representação do estado $|\psi_4\rangle$



Fonte: [Strubell 2011].

Após as duas iterações, efetua-se a medição, na qual o estado $|011\rangle$ tem probabilidade $|\frac{11}{8\sqrt{2}}|^2 = \frac{121}{128} \approx 94.5\%$ de ser medido, enquanto a probabilidade de encontrar um estado incorreto é de $\frac{7}{128} \approx 5.5\%$. Assim, a probabilidade de se ter uma medição correta é 17 vezes maior de que a probabilidade de se obter uma medição incorreta. Ainda que o algoritmo de Grover seja probabilístico, para $N \geq 3$, o maior erro é de $\approx 5.5\%$ e ele tende a 0 conforme N tende ao infinito [Watrous 2006a].

2.3. Quantum Development Kit

O Quantum Development Kit é um ambiente de programação e simulação de algoritmos quânticos desenvolvido pela Microsoft e lançado em Dezembro de 2017 [Team 2017]. Ele foi desenvolvido não apenas para que iniciantes à programação quântica possam implementar algoritmos quânticos, mas também para que especialistas possam desenvolver novos algoritmos. O ambiente conta com [Microsoft 2017c]:

- Linguagem Q# e compilador: A linguagem Q# foi desenvolvida exclusivamente para expressar algoritmos quânticos, sendo utilizada para escrever subprogramas que executam num simulador quântico operando em um computador tradicional;
- Biblioteca Q#: Contém operações e funções que suportam os algoritmos quânticos;
- Simulador quântico local: Um simulador completo de estados quânticos otimizado para simulação rápida e precisa;
- Simulador quântico *trace*: o simulador *trace* não simula o ambiente como o simulador quântico local. Ele é usado para estimar os recursos necessários para execução de um programa quântico e permitir *debugging* mais rápido;
- Estimador de recursos: Estima os recursos necessários para executar uma determinada instância de uma operação em Q# num computador quântico;
- Extensão do Visual Studio e Visual Studio Code: Contém os padrões para os arquivos e projetos Q# bem como suporte para realce sintático.

O passo a passo da instalação do Quantum Development Kit é descrito em detalhes em sua página oficial [Microsoft 2017b].

O ambiente foi escolhido para a implementação de algoritmos quânticos por se tratar de um software livre e open source, o que não apenas permitiu o acesso gratuito a suas funcionalidades, mas propiciou a análise e a modificação de seu código fonte. Além disso, conta com uma comunidade ativa de desenvolvedores e recebe atualizações constantes, fatores que asseguram sua confiabilidade e estabilidade de uso.

3. Resultados

A partir da revisão bibliográfica a respeito do tema, analisaremos a implementação de alguns algoritmos quânticos simples no Quantum Development Kit.

3.1. Portas lógicas no Quantum Development Kit

O site oficial do Quantum Development Kit contém um tutorial para a implementação de portas lógicas simples, como a porta NOT, Hadamard e CNOT, além da realização da medição dos qubits [Microsoft 2017d]. Analisaremos o funcionamento das portas citadas no ambiente tendo tal tutorial como base.

O projeto que compõe o tutorial contém dois arquivos: o arquivo *Operations.qs*, que contém o código em Q#, e o arquivo *Driver.cs*, que contém o *driver* em C# responsável por construir o simulador quântico, computar os argumentos necessários à execução do algoritmo quântico, executar o algoritmo quântico e processar os resultados da operação. Cada operação em Q# gera uma classe homônima em C#, a qual contém um método chamado *Run* que executa de forma assíncrona a operação (visto que um computador tradicional é incapaz de realizar operações sobre um grande número de qubits simultaneamente).

O tutorial inicia com a porta NOT e a operação de Medição, as quais são implementadas na `Operação Set`:

`Algoritmo 1: Operação Set`

```
1  operation Set (desired: Result, q1: Qubit) : Unit {
2          if (desired != M(q1)) {
3              X(q1);
4          }
5      }
```

Fonte: [Microsoft 2017d].

A `Operação Set` inicializa um qubit $q1$ no estado *desired*. Para tanto, ela realiza a medição do qubit (`Set` - Linha 2: $M(q1)$), o qual é invertido pela porta NOT caso o resultado seja distinto do valor do parâmetro *desired* (`Set` - Linha 3: $X(q1)$).

`Algoritmo 2: Operação BellTest`

```
1  operation BellTest (count : Int, initial: Result) :
2          (Int, Int) {
3
4          mutable numOnes = 0;
5          using (qubit = Qubit()) {
6
7              for (test in 1..count) {
8                  Set (initial, qubit);
9                  let res = M (qubit);
10
11                 if (res == One) {
12                     set numOnes += 1;
13                 }
14             }
15             Set(Zero, qubit);
16         }
17
18         return (count-numOnes, numOnes);
19     }
```

Fonte: [Microsoft 2017d].

A `Operação BellTest` é utilizada para facilitar a visualização das operações realizadas. A partir dos parâmetros *count* e *Init* inseridos no *driver*, ela inicializa 1000 qubits no estado 0 e 1000 qubits no estado 1. Para cada estado 1 medido (`BellTest`

- Linha 9), a variável *numOnes* é incrementada, tal que, ao final da operação, o número de estados 0 e 1 medidos é exibido. A Saída obtida demonstra que as operações foram realizadas com sucesso, visto que o parâmetro *Init* determina o estado de todos os qubits a serem medidos:

```
1 Init:Zero 0s=1000 1s=0
2 Init:One  0s=0    1s=1000
3 Press any key to continue...
```

Para demonstrar o funcionamento da porta NOT, basta fazer uma pequena alteração na Operação *BellTest*, inserindo, imediatamente antes da medição (`BellTest` - Linha 9), o comando $X(qubit)$. A saída obtida demonstra a inversão dos estados:

```
1 Init:Zero 0s=0    1s=1000
2 Init:One  0s=1000 1s=0
3 Press any key to continue...
```

Tendo isso em vista, ao implementar a porta Hadamard antes da medição (substituindo a implementação da porta NOT por $H(qubit)$), cria-se superposição e o resultado esperado é o qubit $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, onde há probabilidade de 50% de que os estados 0 e 1 sejam medidos. Novamente, a saída obtida comprova o sucesso da operação (vale ressaltar que a probabilidade ocorre para cada medição, tal que o valor obtido não é igual a 500, mas muito próximo a ele):

```
1 Init:Zero 0s=502  1s=498
2 Init:One  0s=496  1s=504
3 Press any key to continue...
```

Por fim, para demonstrar a propriedade do entrelaçamento, o tutorial acrescenta um segundo qubit $q1$, o qual é inicializado no estado zero (`Entrelaçamento` - Linha 8). O primeiro qubit $q0$ é colocado em superposição (`Entrelaçamento` - Linha 11) e ambos passam por uma porta CNOT (`Entrelaçamento` - Linha 12). Após essas operações, os qubits $q0$ e $q1$ estão entrelaçados, tal que as medições realizadas sobre ambos devem trazer os mesmos resultados. A fim de verificar esse resultado, atribui-se à variável *res* o resultado da medição do qubit $q0$ (`Entrelaçamento` - Linha 13), a qual é comparada com o resultado da medição do qubit $q1$ (`Entrelaçamento` - Linha 15), incrementando uma variável *agree* caso as medições sejam coincidentes.

Algoritmo 3: Entrelaçamento

```
1  operation BellTest(count:Int, initial:Result):
2          (Int, Int, Int){
3          mutable numOnes = 0;
4          mutable agree = 0;
5          using ((q0, q1) = (Qubit(), Qubit())) {
6              for (test in 1..count)
7              {
8                  Set (initial, q0);
9                  Set (Zero, q1);
10
11                 H(q0);
12                 CNOT(q0,q1);
13                 let res = M (q0);
```

```
14
15                  if (M(q1) == res) {
16                      set agree += 1;
17                  }
18
19                  if (res == One) {
20                      set numOnes += 1;
21                  }
22
23              }
24
25              Set(Zero, q0);
26              Set(Zero, q1);
27          }
28
29          return (count-numOnes, numOnes, agree);
30      }
```

Fonte: [Microsoft 2017d].

A saída obtida apresenta o resultado esperado, com a varíavel *agree* apresentando valor 1000:

```
1 Init:Zero  0s=526   1s=474   agree=1000
2 Init:One   0s=512   1s=488   agree=1000
3 Press any key to continue...
```

Comprova-se, assim, o funcionamento do ambiente Quantum Development Kit na simulação de portas quânticas e circuitos simples.

3.2. Algoritmo de Grover no Quantum Development Kit

Muito além de simples portas lógicas quânticas, o Quantum Development Kit pode executar algoritmos quânticos complexos, como o Algoritmo de Grover. Uma implementação deste algoritmo é disponibilizado pela própria Microsoft [Microsoft 2017a] e será utilizada neste trabalho. A implementação disponibilizada foi ligeiramente modificada para efetuar 1000 tentativas para todos os algoritmos, bem como calcular o número de iterações e acessos ao oráculo necessários com base na literatura existente. Tal alteração só foi possível em virtude da natureza *open source* da plataforma. Considerando tal implementação, focaremos na saída apresentada pelo programa, a qual descreve satisfatoriamente o funcionamento do algoritmo de Grover no ambiente, comparando-o com um algoritmo clássico de busca não estruturada.

Primeiramente, o programa executa uma busca não estruturada utilizando um algoritmo clássico que seleciona aleatoriamente um dos elementos pertencentes ao conjunto em que se realiza a busca. Esse algoritmo não realizará ordenação dos elementos do conjunto nem eliminará os elementos indesejados de futuras buscas, de modo que o espaço de busca é sempre o mesmo, possibilitando uma melhor comparação entre o poder computacional dos computadores tradicionais com computadores quânticos. O conjunto considerado tem 8 elementos, tal que a probabilidade de sucesso para cada tentativa é de 0.125, ou 12.5%. A partir disso, 1000 tentativas são realizadas, havendo a exibição do resultado a cada 100 tentativas, em que é exibido o status de sucesso da tentativa (One para

sucesso e Zero para falha) e qual foi o elemento encontrado, com base em seu índice (a tentativa 99, por exemplo, encontrou o elemento de índice 111). A probabilidade exibida a cada 100 tentativas é a razão entre o número de tentativas bem sucedidas e o número total de tentativas realizadas. Nota-se que o valor da probabilidade se mantém próximo ao valor esperado, mas, se tratando de um cenário probabilístico, o valor nunca é exatamente o mesmo.

Busca clássica

```
Classical random search for marked element in
 database.
  Database size: 8.
  Success probability:  0,125

Attempt 99. Success: One,  Probability: 0,14
Found database index One, One, One
Attempt 199. Success: Zero,  Probability: 0,16
Found database index One, One, Zero
Attempt 299. Success: Zero,  Probability: 0,15
Found database index Zero, One, One
Attempt 399. Success: Zero,  Probability: 0,162
Found database index One, Zero, One
Attempt 499. Success: Zero,  Probability: 0,162
Found database index One, One, Zero
Attempt 599. Success: Zero,  Probability: 0,15
Found database index Zero, Zero, One
Attempt 699. Success: One,  Probability: 0,146
Found database index One, One, One
Attempt 799. Success: Zero,  Probability: 0,142
Found database index Zero, One, One
Attempt 899. Success: Zero,  Probability: 0,142
Found database index One, Zero, One
Attempt 999. Success: Zero,  Probability: 0,139
Found database index Zero, One, One


Press any key to continue...
```

Tendo em vista a natureza probabilística da busca realizada, convém executar o programa novamente a fim de verificar se os valores de probabilidade encontrados em cada tentativa são distintos, o que se comprova na `Busca clássica - Tentativa 02`:

Busca clássica - Tentativa 02

```
Classical random search for marked element in
 database.
  Database size: 8.
  Success probability:  0,125

Attempt 99. Success: Zero,  Probability: 0,19
Found database index Zero, One, Zero
Attempt 199. Success: Zero,  Probability: 0,16
Found database index One, Zero, Zero
Attempt 299. Success: Zero,  Probability: 0,167
```

```
Found database index Zero, Zero, One
Attempt 399. Success: Zero,  Probability: 0,15
Found database index Zero, Zero, Zero
Attempt 499. Success: Zero,  Probability: 0,146
Found database index One, Zero, One
Attempt 599. Success: Zero,  Probability: 0,137
Found database index One, One, Zero
Attempt 699. Success: One,  Probability: 0,139
Found database index One, One, One
Attempt 799. Success: Zero,  Probability: 0,135
Found database index One, Zero, One
Attempt 899. Success: Zero,  Probability: 0,14
Found database index One, Zero, One
Attempt 999. Success: Zero,  Probability: 0,137
Found database index Zero, One, Zero


Press any key to continue...
```

Tendo efetuado a `Busca clássica` em um espaço de busca pequeno, o programa parte para a realização de uma busca utilizando o algoritmo de Grover. O espaço considerado tem 64 elementos, necessitando de 6 iterações de Grover (conforme Equação 23). A probabilidade de sucesso de um algoritmo de busca clássico efetuando a busca de modo aleatório é de 1.5625%, enquanto a probabilidade de sucesso do algoritmo quântico é de $\approx 99.6586\%$. A probabilidade de sucesso do algoritmo quântico é dada pela Equação 35, em que N representa o tamanho do espaço de busca e M representa o número de elementos que satisfazem a busca (ou seja, o número de elementos desejáveis) [Watrous 2006b].

$$P = sin^2\left(\left(2\left\lfloor\frac{\pi}{4}\sqrt{\frac{N}{M}}\right\rfloor + 1\right) sin^{-1}\left(\sqrt{\frac{M}{N}}\right)\right) \tag{35}$$

De modo similar à implementação da `Busca clássica`, o programa exibe os elementos encontrados a cada 100 iterações, em que são exibidas a condição de Sucesso e a Probabilidade descritas anteriormente, bem como o *Speedup*, que indica quantas vezes, em média, o algoritmo quântico opera mais rápido que o algoritmo clássico (levando em conta as probabilidades apresentadas e o número de iterações de Grover realizadas). O valor encontrado indica que o algoritmo quântico é aproximadamente 10 vezes mais rápido.

`Busca quântica por um elemento`

```
Quantum search for marked element in database.
  Database size: 64.
  Classical success probability: 0,015625
  Queries per search: 6
  Quantum success probability: 0,996585680786799

Attempt 99. Success: One,  Probability: 1 Speedup: 10,667
Found database index One, One, One, One, One, One
Attempt 199. Success: One,  Probability: 1 Speedup: 10,667
Found database index One, One, One, One, One, One
```

```
Attempt 299. Success: One,  Probability: 1 Speedup: 10,667
Found database index One, One, One, One, One, One
Attempt 399. Success: One,  Probability: 1 Speedup: 10,667
Found database index One, One, One, One, One, One
Attempt 499. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One
Attempt 599. Success: One,  Probability: 0,997 Speedup: 10,635
Found database index One, One, One, One, One, One
Attempt 699. Success: One,  Probability: 0,997 Speedup: 10,635
Found database index One, One, One, One, One, One
Attempt 799. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One
Attempt 899. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One
Attempt 999. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One


Press any key to continue...
```

Tal como na `Busca Clássica`, obtém-se resultados ligeiramente diferentes ao se efetuar a `Busca quântica por um elemento - Tentativa 02`:

`Busca quântica por um elemento - Tentativa 02`

```
Quantum search for marked element in database.
  Database size: 64.
  Classical success probability: 0,015625
  Queries per search: 6
  Quantum success probability: 0,996585680786799

Attempt 99. Success: One,  Probability: 1 Speedup: 10,667
Found database index One, One, One, One, One, One
Attempt 199. Success: One,  Probability: 1 Speedup: 10,667
Found database index One, One, One, One, One, One
Attempt 299. Success: One,  Probability: 1 Speedup: 10,667
Found database index One, One, One, One, One, One
Attempt 399. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One
Attempt 499. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One
Attempt 599. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One
Attempt 699. Success: One,  Probability: 0,999 Speedup: 10,656
Found database index One, One, One, One, One, One
Attempt 799. Success: One,  Probability: 0,999 Speedup: 10,656
Found database index One, One, One, One, One, One
Attempt 899. Success: One,  Probability: 0,999 Speedup: 10,656
Found database index One, One, One, One, One, One
Attempt 999. Success: One,  Probability: 0,998 Speedup: 10,645
Found database index One, One, One, One, One, One


Press any key to continue...
```

Por fim, é realizada uma busca com o algoritmo de Grover num conjunto de 256

elementos em que há 4 elementos (de índices 0, 39, 101 e 234) que satisfazem a operação. Aqui, a probabilidade de sucesso do algoritmo quântico é a mesma que para a `Busca quântica por um elemento` tendo em vista que a Equação 35 apresenta o mesmo valor. Além disso, o algoritmo clássico para este conjunto tem mesma probabilidade de sucesso que no exemplo anterior, visto que $\frac{4}{256} = \frac{1}{64} = 0,015625$.

`Busca quântica por quatro elementos`

```
1  Quantum search for marked element in database.
2    Database size: 256.
3    Marked elements: 0,39,101,234
4    Classical success probability: 0,015625
5    Queries per search: 6
6    Quantum success probability: 0,996585680786799
7
8  Attempt 99. Success: One,  Probability: 0,98 Speedup: 10,453
9  Found database index 234
10 Attempt 199. Success: One,  Probability: 0,98 Speedup: 10,453
11 Found database index 101
12 Attempt 299. Success: One,  Probability: 0,987 Speedup: 10,528
13 Found database index 39
14 Attempt 399. Success: One,  Probability: 0,99 Speedup: 10,56
15 Found database index 39
16 Attempt 499. Success: One,  Probability: 0,992 Speedup: 10,581
17 Found database index 101
18 Attempt 599. Success: One,  Probability: 0,993 Speedup: 10,592
19 Found database index 234
20 Attempt 699. Success: One,  Probability: 0,993 Speedup: 10,592
21 Found database index 234
22 Attempt 799. Success: One,  Probability: 0,994 Speedup: 10,603
23 Found database index 234
24 Attempt 899. Success: One,  Probability: 0,992 Speedup: 10,581
25 Found database index 234
26 Attempt 999. Success: One,  Probability: 0,993 Speedup: 10,592
27 Found database index 234
28
29 Press any key to continue...
```

Novamente, a execução de uma segunda tentativa apresenta resultados ligeiramente diferentes em virtude da natureza probabilística do problema.

`Busca quântica por quatro elementos - Tentativa 02`

```
1  Quantum search for marked element in database.
2    Database size: 256.
3    Marked elements: 0,39,101,234
4    Classical success probability: 0,015625
5    Queries per search: 6
6    Quantum success probability: 0,996585680786799
7
8  Attempt 99. Success: One,  Probability: 0,99 Speedup: 10,56
9  Found database index 0
10 Attempt 199. Success: One,  Probability: 0,99 Speedup: 10,56
11 Found database index 39
12 Attempt 299. Success: One,  Probability: 0,99 Speedup: 10,56
```

```
Found database index 0
Attempt 399. Success: One,  Probability: 0,99 Speedup: 10,56
Found database index 234
Attempt 499. Success: One,  Probability: 0,99 Speedup: 10,56
Found database index 101
Attempt 599. Success: One,  Probability: 0,992 Speedup: 10,581
Found database index 39
Attempt 699. Success: One,  Probability: 0,993 Speedup: 10,592
Found database index 101
Attempt 799. Success: One,  Probability: 0,994 Speedup: 10,603
Found database index 101
Attempt 899. Success: One,  Probability: 0,994 Speedup: 10,603
Found database index 0
Attempt 999. Success: One,  Probability: 0,995 Speedup: 10,613
Found database index 101

Press any key to continue...
```

4. Conclusão

A partir da implementação do algoritmo de Grover no ambiente Quantum Development Kit, evidencia-se o poder da Computação Quântica sobre a computação tradicional. O algoritmo de Grover pode ser utilizado para acelerar a resolução de problemas da classe NP como fatoração e problemas com soluções fatoriais [Nielsen and Chuang 2010], sendo este apenas um dos vários algoritmos quânticos que superam seus equivalentes clássicos.

Durante a realização deste trabalho, a Computação Quântica ainda se encontrava em fase de pesquisa, sem possuir aplicações de impacto na sociedade. Acredita-se que tais aplicações surgirão após a Computação Quântica alcançar a chamada "supremacia quântica", em que computadores quânticos serão capazes de realizar operações que computadores clássicos são incapazes de realizar. Para se alcançar tal marco, muita pesquisa e investimento devem ser direcionados a essa área.

Nota-se, a partir dos conceitos expostos e dos algoritmos analisados, que computadores quânticos diferenciam-se de computadores tradicionais não apenas por seus princípios de funcionamento, mas pelo modo como lidam com dados e realizam operações sobre estes. Nesse sentido, o avanço desta área emergente requer não apenas o desenvolvimento da tecnologia por trás dos computadores quânticos, mas da disseminação do conhecimento referente à execução e ao desenvolvimento de algoritmos quânticos.

Tendo isso em vista, o presente trabalho surge como ponto de partida para estudos futuros na área. Há, na literatura existente acerca do tema, algoritmos de maior complexidade e maior poder computacional, como algoritmos baseados na transformada de Fourier e algoritmos de simulação quântica. Além disso, há um grande número de simuladores quânticos os quais podem auxiliar no estudo da área.

Referências

Bennett, C. H. (2016). Information is quantum. Disponivel em https://www.youtube.com/watch?v=EqXv40kCahM. Consultado em 08/08/2018.

Chuang, I. L. and Shor, P. (2018). Quantum information science i, part 1. Disponivel em https://courses.edx.org/courses/course-v1:MITx+8.370.1x+1T2018/course. Consultado em 03/08/2018.

Gershon, T. (2017). A beginner's guide to quantum computing. Disponivel em https://www.youtube.com/watch?v=S52rxZG-zi0. Consultado em 03/08/2018.

Microsoft (2017a). Microsoft quantum development kit samples. Disponivel em https://github.com/Microsoft/Quantum. Consultado em 29/04/2019.

Microsoft (2017b). Quantum development kit. Disponivel em https://www.microsoft.com/en-us/quantum/development-kit. Consultado em 20/05/2019.

Microsoft (2017c). Welcome to the microsoft quantum development kit preview. Disponivel em https://docs.microsoft.com/en-us/quantum/?view=qsharp-preview. Consultado em 20/05/2019.

Microsoft (2017d). Writing a quantum program. Disponivel em https://docs.microsoft.com/en-us/quantum/quickstart?view=qsharp-preview&tabs=tabid-vscode. Consultado em 29/04/2019.

Nielsen, M. A. and Chuang, I. L. (2010). *Quantum Computation and Quantum Information*. Cambridge University Press.

Strubell, E. (2011). An introduction to quantum algorithms. *COS498 - Chawathe*. Consultado em 29/04/2019.

Team, M. Q. (2017). Announcing the microsoft quantum development kit. Disponivel em https://cloudblogs.microsoft.com/quantum/2017/12/11/announcing-microsoft-quantum-development-kit/. Consultado em 20/05/2019.

Watrous, J. (2006a). Lecture 12: Grover's algorithm. Disponivel em https://cs.uwaterloo.ca/~watrous/LectureNotes/CPSC519.Winter2006/13.pdf. Consultado em 29/04/2019.

Watrous, J. (2006b). Lecture 13: Grover's algorithm (continued). Disponivel em https://cs.uwaterloo.ca/~watrous/LectureNotes/CPSC519.Winter2006/12.pdf. Consultado em 29/04/2019.

Wright, J. (2015). Lecture 4: Grover's algorithm. Disponivel em https://www.cs.cmu.edu/~odonnell/quantum15/lecture04.pdf. Consultado em 29/04/2019.