

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
(X) PRÉ-PROJETO	() PROJETO	ANO/SEMESTRE: 2020/1

JSQUANTUM: PROTÓTIPO DE UM SIMULADOR DE CIRCUITOS QUÂNTICOS

Anderson Rodrigo Pozzi
Prof. Mauro Marcelo Mattos

1 INTRODUÇÃO

A Computação Quântica (CQ) é uma área que tem experimentado uma evolução muito rápida nos últimos anos, tanto do ponto de vista da pesquisa científica como do ponto de vista de empreendimentos comerciais. A CQ vale-se de certas propriedades da mecânica quântica para sugerir consideráveis ganhos em capacidade computacional caracterizando-se atualmente como um novo paradigma da área da computação (CONCEIÇÃO, 2013). Conforme Mintz et al. (2020), a CQ possui o potencial para resolver desafios computacionais que ultrapassam a capacidade dos chamados High Performance Computing (Computadores de Alta Performance, tradução nossa) os quais são concebidos justamente com a finalidade de resolução de problemas complexos e cuja configuração facilmente ultrapassa milhares de processadores com custos de milhões de dólares. Perdomo-Ortiz et al. (2018) citam, por exemplo, que na área de aprendizado de máquina (Machine Learning) as pesquisas em CQ tem se concentrado em tarefas como classificação, regressão, modelos gaussianos, quantização vetorial, análise de componentes principais.

A migração da computação tradicional para a computação quântica não é trivial o que representa uma barreira significativa para a adoção da CQ pela comunidade de pesquisa. Por extensão, a ampla adoção é ainda mais complicada pelo desafio de desenvolver aplicativos científicos que incorporam expressões quânticas para retornar resultados precisos quando executados em dispositivos quânticos de escala intermediária ruidosa (ou seja, compostos por centenas de qubits imperfeitos) (PERDOMO-ORTIZ et al., 2018). Conforme Moll et al. (2017), há uma expectativa muito grande na utilização de algoritmos de otimização que utilizam um enfoque híbrido, em outras palavras, o objetivo é encontrar um algoritmo que obtenha “um estado altamente emaranhado em um sistema quântico para um estado alvo que minimiza uma função de custo por meio da variação de alguns parâmetros de porta”. Este enfoque pode ser utilizado tanto em problemas clássicos de otimização como para problemas de química quântica (MOLL et al., 2017).

O desenvolvimento tecnológico em CQ tem por objetivo a construção de um computador quântico universal e tolerante a falhas que possa executar qualquer algoritmo muito mais rápido que um computador tradicional. Para alcançar este objetivo são necessários “milhões de qubits físicos e operações de porta de alta fidelidade para viabilizar um computador quântico tolerante a falhas universal” (MOLL et al., 2017) contudo reforçam os autores que “um sistema com estas características ainda não pode ser construído na atualidade”. Conforme Lavado (2020), os computadores comerciais ainda não superaram a marca de 100 qubits o que caracteriza a complexidade que tais projetos demandam.

A computação quântica experimental ainda é uma disciplina relativamente nova e comparável aos primeiros dias dos computadores clássicos na década de 1950. Semelhante à programação manual de um computador clássico com cartões perfurados ou um montador, os computadores quânticos de hoje exigem que o usuário especifique um algoritmo quântico como uma sequência de portas lógicas quânticas fundamentais. Assim, a implementação de um algoritmo quântico em hardware quântico real requer várias etapas em diferentes camadas de abstração. (FINGERHUTH; BABEJ; WITTEK, 2018).

Neste contexto, o presente projeto tem por objetivo o desenvolvimento de um simulador de circuitos quânticos denominado JSQUANTUM que possibilite a concepção e estudo dos conceitos fundamentais da computação quântica.

1.1 OBJETIVOS

Este projeto tem por objetivo disponibilizar um protótipo de uma ferramenta web para viabilizar a criação e simulação de circuitos quânticos.

Os objetivos específicos são:

- disponibilizar um modelo de construção de circuitos quânticos;
- disponibilizar um ambiente de simulação do funcionamento dos circuitos;
- validar simulador e circuitos lógicos quânticos através da simulação algoritmo de grover utilizando as portas lógicas quânticas implementadas, testando para cada entrada e saída respectiva.

2 TRABALHOS CORRELATOS

Foram encontrados os seguintes trabalhos correlatos: o primeiro é o trabalho de Watanabe, Suzuki e Yamazaki (2011) e descreve o programa QCAD. O segundo trabalho descreve o projeto Q.js (SMITH, 2010). O

Comentado [MCL1]: Em itálico. Corrigir em todo o texto.

Comentado [MCL2]: .

Comentado [MCL3]: .

Comentado [MCL4]: e

Comentado [MCL5]: referência

Comentado [MCL6]: citação direta exige o número da página

Comentado [MCL7]: repetir a referência

Comentado [MCL8]: sua introdução não explora a importância de construir um simulador.

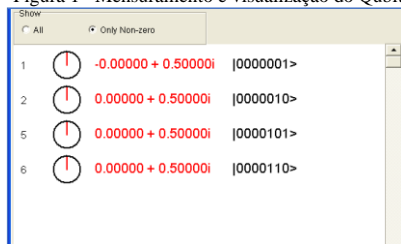
Comentado [MCL9]: Falta algo

terceiro trabalho **apresentado** por Figueiredo (2013) e **apresenta** um protótipo de simulador de circuitos quânticos.

2.1 QCAD

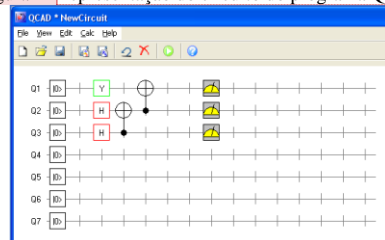
O projeto QCAD foi criado por Watanabe, Suzuki e Yamazaki (2011). QCAD é um simulador de circuitos lógicos quânticos utilizável apenas no sistema operacional Windows, no entanto tem uma boa interface gráfica e várias portas quânticas. O QCAD implementa diversos modelos de visualização como HSV View (3D ou 2D) e tem uma funcionalidade de mensuramento e visualização de qubits (Figura 1) individuais sendo possível fazer **visualização** do qubits através do modelo de **visualização** de esfera de bloch, porém ainda assim pode ser considerado um simulador simplório não tendo como opção **circuitos** como **circuito** de QFT.

Figura 1 –Mensuramento e visualização do Qubit



Fonte: Watanabe, Suzuki e Yamazaki (2011).

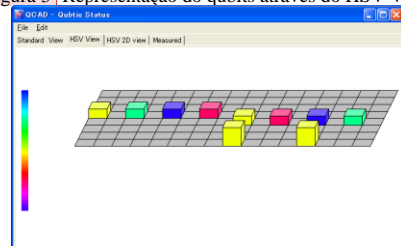
Figura 2 -Representação do circuito do programa QCAD



Fonte: Watanabe, Suzuki e Yamazaki (2011).

Dentre as possibilidades de configuração de portas pré-definidas para criação de circuitos quânticos estão as portas de Hadamart, Pauli-X, Pauli-Y, Pauli-Z, CNOT, CCNOT, Phase Shift e por fim tem pré-definido também uma porta relacionada à medição do qubit todas elas podendo ser adicionadas no diagrama de circuitos quânticos (Figura 2). Além disso, o ambiente permite salvar modelos e carregá-los através do disco possibilitando manter projetos de circuitos quânticos salvos no sistema de arquivos (WATANABE; SUZUKI; YAMAZAKI, 2010).

Figura 3 -Representação do qubits através do HSV View



Fonte: Watanabe, Suzuki e Yamazaki (2011).

A figura 3 apresenta o modelo de visualização HSV View que é um modelo de tabela em que cada coluna representa um qubit, nesse modelo por meio do sistema de cores HSV é **possível** visualizar o valor de

Comentado [MCL10]: maiúsculo

Comentado [MCL11]: A referência à figura no texto deve preceder sua inserção.

Comentado [MCL12]: Idem. Rever em todo o texto.

Comentado [MCL13]: .

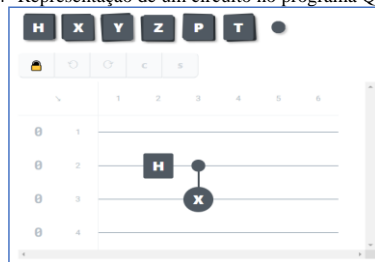
cada qubit através da cor da coluna. O modelo de visualização HSV View **auxilia análise** de dados após a simulação facilitando assim a compreensão do resultado.

O QCAD é um simulador simples que possui poucas funcionalidades, e tem apenas portas lógicas quânticas essenciais seu principal diferencial é a possibilidade de visualização de medições como HSV View, entre as vantagens do programa QCAD está utilização de diversos modelos de visualização de qubit e o modelo de diagramação de circuitos quânticos tradicional que é utilizada pelo programa, porém, não é possível visualizar os resultados na forma matricial e não tem opção de criar portas personalizadas pelo usuário o que acaba sendo uma das grandes desvantagens do **programa** (WATANABE; SUZUKI; YAMAZAKI, 2011).

2.2 Q.JS

Q.js (SMITH, 2020) é um editor desenvolvido em javascript que tem propósito de simulação de circuitos quânticos **através de codificação ou visualmente** funcionando assim como uma biblioteca de funções ou apenas simulador. Este programa possui uma interface gráfica moderna sendo considerado um **simulador circuitos** quânticos completo, visto que possui uma grande variedade de portas quânticas predefinidas para utilizar e possui um modelo eficiente de simulação. **A figura 4 apresenta uma barra superior fornece recursos** como portas pré-definidas como Hadamart, Pauli-X, Toffoli mas é importante notar que diferente das demais ferramentas é possível definir qualquer porta como tendo qubit de controle através do botão circular, ou seja, criar portas controladas de forma customizada sendo assim um grande diferencial. As simulações executam em tempo real enquanto o circuito é montado e pode ser acessada através do comando `report$()`.

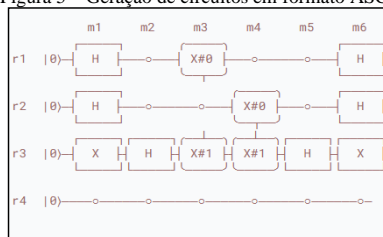
Figura 4 -Representação de um circuito no programa Quantum.js



Fonte: Smith (2020)

O simulador Q.js também possibilita gerar um modelo ASCII do circuito para melhor visualização e além disso possibilita geração dos circuitos em LaTeX para adicioná-las em documentos como artigos e relatórios (Figuras 5 e 6). A exportação de circuitos em formato LaTeX auxilia na **construção** de textos acadêmicos facilitando assim o estudo da CQ.

Figura 5 – Geração de circuitos em formato ASCII



Fonte: Smith (2020)

Comentado [MCL14]: Rever pontuação dessa frase. Está difícil ler.

Comentado [MCL15]: Confuso.

Comentado [MCL16]: Redação! Preste atenção no que escreve.

Figura 6 – Geração de circuitos em formato LaTeX

```
\Qcircuit @C=1.0em @R=0.7em {
    & \h & \qw & \targ & \qw & \qw & \h & \\
    & \h & \qw & \qw & \targ & \qw & \h & \\
    & \x & \h & \ctrl{?} & \ctrl{?} & \h & \x & \\
    & \qw & \qw & \qw & \qw & \qw & \qw & }
}
```

Fonte: Smith (2020) |

Através do comando `report$()` é possível ser acessado o mensuramento dos qubits que são atualizados em tempo real enquanto o circuito é montado. A figura 7 apresenta os resultados obtidos pelo circuito para cada combinação de resultados possíveis entre os qubits e suas respectivas porcentagens para melhor comparação entre as porcentagens é desenhada uma linha horizontal.

Figura 7 -Representação de um circuito no programa Quantum.js

1	0000>	0% chance
2	0001>	0% chance
3	0010>	0% chance
4	0011>	0% chance
5	0100>	0% chance
6	0101>	0% chance
7	0110>	0% chance
8	0111>	0% chance
9	1000>	0% chance
10	1001>	0% chance
11	1010>	0% chance
12	1011>	0% chance
13	1100>	100% chance
14	1101>	0% chance
15	1110>	0% chance
16	1111>	0% chance

Fonte: Smith (2020)

Entre suas vantagens o Q.js (SMITH, 2020) faz exportação em múltiplos formatos sendo eles LaTeX, ASCII e possibilidade de ser utilizado na codificação como biblioteca. No entanto, não possui a opção de adicionar portas personalizadas através de uma matriz de transformação podendo apenas transformar portas lógicas quânticas em portas controladas, porém não possibilitando customização completa, e é difícil visualizar os resultados devido à grande quantidade de informação em texto especialmente quando estes têm diferentes fases.

2.3 SIMULADOR DE CIRCUITOS QUÂNTICOS

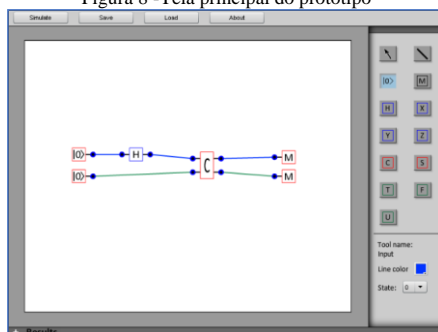
O projeto de Figueiredo (2013) foi concebido de forma a poder ser acessado em qualquer sistema operacional e ser de fácil acesso. Para isto, basta acessar uma página web onde se encontra a aplicação, iniciando o programa automaticamente ou realizando o download para o computador local, utilizando o simulador sem a necessidade de acesso à internet. A interface gráfica permite ao usuário visualizar e editar facilmente os circuitos criados para posteriormente simulá-los. A barra lateral (Figura 8) oferece várias funcionalidades para a criação de circuitos e permite a adição de portas quânticas desejadas através de uma das ferramentas de criação de portas pré-definidas, como a Hadamard ou Pauli-X. O sistema também permite a utilização da ferramenta personalizada Custom Tool que serve para criar portas quânticas customizadas através de matrizes criadas pelo usuário, desde que estas sejam unitárias. Se o usuário precisar de uma medição no final do circuito poderá fazê-lo com a ferramenta de medição Measurement tool. Em circuitos de grandes dimensões, a ferramenta permite ver as linhas de várias cores diferentes para facilitar a distinção. Assim, quando uma entrada de qubit se encontra selecionada, é possível escolher uma nova cor nas opções adicionais, alterando automaticamente a cor de todas as linhas representadas por esse qubit (FIGUEIREDO, 2013).

Comentado [MCL17]: Tirar linhas em branco

Comentado [MCL18]: Leia.

Comentado [MCL19]: Não entendi.

Figura 8 -Tela principal do protótipo



Fonte: Figueiredo (2013, p. 35)

Figura 9 -Exemplo de matriz personalizada

Fonte: Figueiredo (2013, p. 44)

A ferramenta permite adicionar novas portas personalizadas. Esta funcionalidade ativa uma opção que permite a introdução de valores da matriz de pesos e o registro de um nome de referência (Figura 9). É importante notar que no Quantum.js e QCAD tem modelo de visualização dos circuitos quânticos diferente, já no projeto de figueiredo assemelha-se mais a forma de visualização baseada em circuitos clássicos o que é de certa forma uma boa abordagem.

Finalmente, é possível realizar a simulação do circuito pressionando-se o botão `simulate` na barra superior. A figura 10 apresenta uma nova janela com os resultados obtidos do circuito. Se o circuito incluir medições no final, os resultados destas poderão ainda serem visualizados com a opção `Measurement`. A probabilidade de obter zero ou um em cada uma das medições feitas será apresentada, bem como o resultado obtido na respetiva medição. Este resultado obtido será gerado aleatoriamente pelo programa em cada simulação, respeitando as probabilidades corretas e o entrelaçamento quântico caso exista (Figura 11).

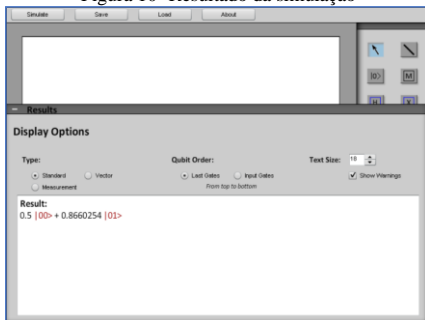
O projeto de Figueiredo (2013) tem como vantagens a possibilidade de criação de circuitos customizados e modelo de diagramação de circuitos quânticos diferenciado com uma imensa possibilidade de customização possibilitando explorar novas portas lógicas quânticas. Porém como desvantagem tem pouquíssimas possibilidades para visualização de qubits dificultando o estudo dos dados de resultado.

Comentado [MCL20]: Citada depois

Comentado [MCL21]: Mal redigido e fora de contexto. Aqui é para descrever o correlato.

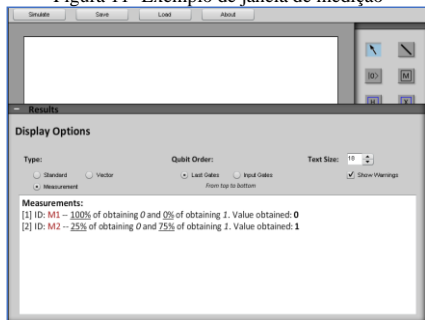
Comentado [MCL22]: Evite superlativos.

Figura 10 -Resultado da simulação



Fonte: Figueiredo (2013, p. 46)

Figura 11 -Exemplo de janela de medição



Fonte: Figueiredo (2013, p. 46)

3 PROPOSTA DA SOLUÇÃO

Nesta seção são apresentadas a justificativa para o desenvolvimento desse trabalho, os requisitos funcionais e não funcionais e a metodologia do projeto contendo o cronograma com as etapas para o desenvolvimento.

3.1 JUSTIFICATIVA

Conforme apresentado anteriormente, a área de estudos em computação quântica é uma área relativamente nova e remonta os esforços iniciais realizados na década de 1950 que redundaram na construção dos primeiros computadores digitais. Neste sentido, do ponto de vista de relevância científica, a construção de um simulador de circuitos quânticos permite aprofundar conhecimentos nesta área de estudos e a expectativa é de que o simulador produzido possa ser utilizado por outros acadêmicos interessados na área o que deve contribuir para a popularização do conteúdo.

As características dos trabalhos correlatos identificados estão apresentadas no Quadro 1 onde foram identificadas funcionalidades que se pretende venham a ser incorporadas no simulador a ser construído.

Comentado [MCL23]: É para lugar.

Quadro 1 - Comparativo de trabalhos correlatos

Características/Trabalhos correlatos	Watanabe, Suzuki e Yamazaki (2011)	SMITH (2010)	Figueiredo (2013)
Linguagem	C++	JAVASCRIPT	JAVASCRIPT
Salva em disco	Sim	Sim	Sim
Possibilita circuitos customizados	Não	Possibilita customizar se circuito é controlado	Sim
Visualização clássica de circuitos quânticos	Sim	Sim	Não
Modelos visuais de representação do qubit	HSV View, Esfera de boch	Não Possui	Não Possui
Utilização programática sem necessidade de simulador visual.	Não	Sim	Não
Possibilita transformação de todos qubits através de uma função Ex.:QFT	Não	Não	Não
Exportar circuito em diversos formatos	Não Possui	LaTeX, ASCII	Não Possui

Fonte: elaborado pelo autor

A partir do quadro 1, **percebe-se** que o trabalho de Watanabe, Suzuki e Yamazaki (2011) é um simulador com menos possibilidades de customização e mais focado nos modos de **visualização** do qubit possibilitando **visualização** através do HSV View e da esfera de Bloch. Já o Q.js (SMITH, 2020) é um simulador mais versátil pois possibilita criação de circuitos controlados através de uma customização simples além disso possui como diferencial a utilização programática sem utilização de um simulador visual e também possibilita exportação em diversos formatos como por exemplo LaTeX e ASCII. O projeto de Figueiredo (2013) **possibilita** criação de circuitos customizados através de uma matriz linear **possibilitando** a construção de um modelo mais personalizado.

A partir desta análise, conclui-se que nenhum dos projetos analisados permite **uma completa de circuitos** quânticos nem contemplam diversidade de modos de visualização dos qubits ou permitem a exportação dos modelos construídos em formatos variados. Neste contexto **o presente projeto se torna relevante na medida em que busca produzir um artefato mais completo para suportar a construção e simulação de circuitos quânticos. Destaca-se ainda que a validação do modelo será realizada através da simulação de um algoritmo amplamente conhecido na área de CQ denominado algoritmo de Grover.**

3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O JSQUANTUM deverá:

- permitir a criação de circuitos usando as portas quânticas de Hadamart, Pauli-X, Pauli-Y, Pauli-Z, CNOT, CCNOT, Toffoli, Phase Shift e Medições (Requisito Funcional (RF));
- permitir a seleção de objetos da rede: Portas, Entradas e Medições previamente inseridas no modelo; (RF);
- permitir a seleção de objetos da rede: Portas, Entradas e Medições previamente inseridas no modelo; (RF);
- permitir o reposicionamento dos componentes no layout do circuito. (RF);
- permitir a remoção de componentes do circuito. (RF);
- permitir possibilidade de visualizar qubits através do HSV View. (RF);
- permitir criação **de** portas quânticas customizadas através de uma matriz de transformação linear. (RF);
- permitir exportação dos circuitos através do formato **Latex**; (RF);
- permitir a adição de portas personalizadas, de tamanho arbitrário, em qualquer ponto do circuito de simulação (RF);
- permitir seleção de um modo educacional de modo que seja apresentado **um conjunto de desafios de construção** de circuitos. (RF);
- permitir **usuário** **visualizar** uma descrição detalhada do funcionamento de cada porta lógica quântica no modo educacional. (RF);
- permitir salvar e carregar o modelo a partir de arquivos em disco (RF);
- permitir visualizar o resultado da simulação (RF);
- ser desenvolvido em Javascript com Framework React.js (RNF);

Comentado [MCL24]: o

Comentado [MCL25]: Completa o que?

Comentado [MCL26]: .

Comentado [MCL27]: Sua justificativa ainda está bem incipiente considerando o potencial do trabalho. Deveria demarcar de forma mais específica seu diferencial e sua contribuição científica já que está propondo algo que já existe. O texto apresentado aqui é extremamente genérico.

Comentado [MCL28]: Derruba sua justificativa pois o de Smith exporta para dois formatos.

Comentado [MCL29]: São pré-definidos?

Comentado [MCL30]: As outras ações não são do usuário?

- o) ser disponibilizado através de Servidor Web do Firebase (RNF);
- p) ser compatível com navegadores como Google Chrome, Microsoft Edge, Safari, Firefox (RNF).

METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- q) levantamento bibliográfico: realizar o levantamento bibliográfico sobre a computação quântica levando em consideração a definição de qubit e transformações aplicadas pelas portas lógicas quânticas e modelos de visualização do qubit, e também atualizar o levantamento dos trabalhos correlatos que serão utilizados como referência para desenvolver o trabalho proposto;
- r) levantamento de requisitos: revisar o conjunto de requisitos funcionais e não funcionais previamente identificados nesta proposta;
- s) especificação e análise: formalizar as funcionalidades do projeto através de diagramas de caso de uso, de classe e de sequência da UML, utilizando para o isso o **software o Star UML**;
- t) implementação: implementar o software proposto de acordo com as especificações utilizando a tecnologia React.js e Firebase;
- u) testes: criar cenários de testes para validar a entrada e saída de cada porta lógica quântica a ser implementada, efetuar teste do simulador construindo modelo de circuitos simulando algoritmo de Grover **verificando** todas entradas e saídas possíveis.

As etapas serão realizadas nos períodos relacionados no Quadro 22.

Quadro 2 - Cronograma

etapas / quinzenas	2021									
	fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
levantamento dos requisitos										
especificação e análise										
implementação										
testes										

Fonte: elaborado pelo autor.

4 REVISÃO BIBLIOGRÁFICA

Esta seção tem como objetivo **introduzir principais** modelos teóricos que serão utilizados para construção do simulador abordando conceitos como: Qubit, Mensuração, Representação matemática e Transformações aplicadas pelas portas lógicas quânticas nos qubits, algoritmos.

O trabalho faz uma revisão teórica acerca do conceito de postulados da mecânica quântica, circuitos quânticos, matriz de densidade e canais de erro. Esta revisão está baseada em: Nielsen e Chang (2010), Conceição (2013), Fingerhuth, Babej e Wittek (2018), Blackman (2017) e Figueiredo (2007). Posteriormente será realizada a especificação do modelo levando em conta o trabalho de Rosa (2019).

4.1 QUBIT

Qubit é menor unidade de cálculo na programação quântica e diferentemente do bit clássico que pode apenas ter os estados 0 e 1 o valor de um qubit pode ser nenhum ou até uma sobreposição quântica de 0 e 1 **sendo representando** por duas variáveis como espaço vetorial bidimensional cada um **representando** porcentagem de estado em específico. Os qubits são **representados com dois estados distintos que é definida** através do spin de um elétron o qual pode variar entre 1/2 e -1/2. Estes valores podem ser traduzidos em sistema binário com 1 e 0 (FIGUEIREDO, 2013).

Um dos modelos mais comuns de representação de qubit único é através da esfera de Bloch (figura 12) que é uma esfera imaginária que simula um elétron. Enquanto valores clássicos binários 0 e 1 ficariam em apenas um dos pontos dos polos das esferas o qubit pode estar em qualquer ponto da esfera simulando o estado de sobreposição representado em duas coordenadas de números reais sendo assim ela possibilita a visualização de qubit como um vetor valor tridimensional real (BLACKMAN, 2017).

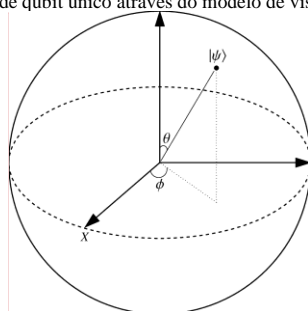
Comentado [MCL31]: Não tem métricas de desempenho para a simulação por exemplo? Quais serão os parâmetros para fazer a validação prometida nos objetivos?

Comentado [MCL32]: Isso será possível para um simulador de circuito quântico? Acho que vc deve definir um conjunto de casos de testes e identificar a confiança da simulação.

Comentado [MCL33]: Não é exatamente o que está escrito no item a da metodologia. Lá diz que iniciaria com computação quântica.

Comentado [MCL34]: Desnecessário aqui

Figura 12 - Representação de qubit unico através do modelo de visualização da esfera de bloch



Fonte: Blackman (2007)

Comentado [MCL35]: Maiúscula

Comentado [MCL36]: Na versão 1 já havia sido dito que as figuras devem ter moldura.

4.2 PORTAS QUÂNTICAS

De forma semelhante aos circuitos booleanos tradicionais, um circuito quântico utiliza portas lógicas quânticas em sequência, contudo ao contrário das portas lógicas convencionais, a portas lógicas quânticas possibilitam operações reversíveis, ou seja, deve existir operação inversa permitindo obter a entrada (o que por si só já não é possível em portas lógicas como AND, por exemplo). Também é importante notar que portas lógicas quânticas funcionam de forma que podem ser representadas por meio de matrizes lineares sendo assim o número de qubits de entrada acabam sendo semelhantes ao número de qubits de saída pois os qubits acabam sofrendo uma transformação linear (FIGUEIREDO, 2013).

Comentado [MCL37]: Está extremamente simplificada essa explicação.

Comentado [MCL38]: .

Para aplicar uma transformação de uma porta lógica quântica em um qubit basta aplicar a matriz linear correspondente a operação nas duas variáveis de porcentagem que representam qubit específico, ou seja, dado um qubit com valores [1,1] a multiplicação de matrizes resultante é saída da porta lógica quântica. É possível verificar nas figuras 13 e 14 a matriz linear correspondente de cada porta lógica quântica (NIELSEN; CHANG, 2010).

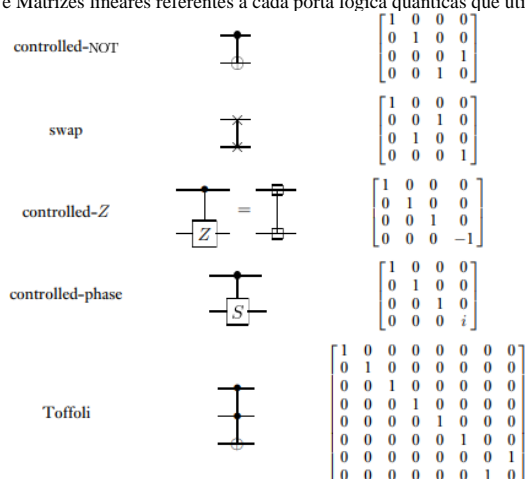
Figura 13 – Diagrama e Matrizes lineares referentes a cada porta lógica quânticas de qubit único.

Hadamard	$\text{---} \boxed{H} \text{---}$	$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X	$\text{---} \boxed{X} \text{---}$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y	$\text{---} \boxed{Y} \text{---}$	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z	$\text{---} \boxed{Z} \text{---}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Phase	$\text{---} \boxed{S} \text{---}$	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$	$\text{---} \boxed{T} \text{---}$	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

Fonte: Nielsen e Chang (2010)

Comentado [MCL39]: .

Figura 14 – Diagrama e Matrizes lineares referentes a cada porta lógica quânticas que utilizam qubit de controle.



Fonte: Nielsen e Chang (2010)

4.3 ALGORITMO DE GROVER

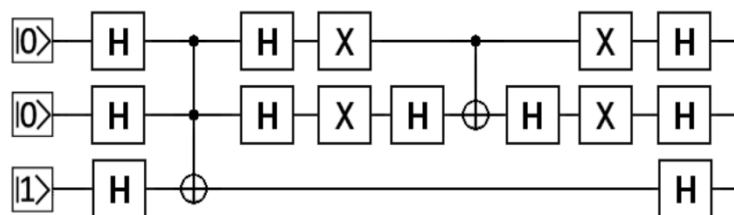
O algoritmo de Grover permite localizar com alta probabilidade um valor de entrada em uma função caixa preta ou seja uma função desconhecida que produz uma saída específica sendo que em computador com arquitetura clássica é necessário verificar cada entrada e saída respectivamente custando recursos computacionais elevados. Assim, através do algoritmo de Grover existe um ganho quadrático em performance com a verificação quântica de múltiplas possíveis entradas. Uma das possíveis aplicações deste algoritmo pode se dar em pesquisas em banco de dados evitando a busca por chaves (e índices) como é realizado atualmente. Usando CQ, a busca poderia ser realizada por múltiplas chaves simultaneamente (FIGUEIREDO, 2013).

O algoritmo funciona de forma que para cada entrada específica processada pela função caixa preta exista um estado de sobreposição quântica chamada de oráculo que possibilita dizer exatamente qual entrada é necessária para saída desejada (FIGUEIREDO, 2013). É possível simular algoritmo de Grover através de um simulador de circuitos quânticos utilizando portas de Hadamart e de Phase X (Figura 15).

Comentado [MCL40]: Melhorar redação dessa frase.

Comentado [MCL41]: Como não explicou o funcionamento das portas isso se torna abstrato.

Figura 15 - Representação do algoritmo de Grover através do diagrama de circuitos quânticos



Fonte: Figueiredo (2013, p. 50)

REFERÊNCIAS

BLACKMAN, Igor V. **Simulando Algoritmos Quânticos em um Computador Clássico**. 2017. 49 f. TCC (Graduação) - Curso de Bacharel em Ciência da Computação, Universidade Federal Fluminense, Niterói.

CONCEIÇÃO, Calebe Micael de Oliveira. Uma Arquitetura de Co-Processador para Simulação de Algoritmos Quânticos em FPGA. 2013. 82 f. Dissertação (Mestrado) - Curso de Pós-Graduação em Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013. Disponível em: <http://hdl.handle.net/10183/81297>. Acesso em: 25 nov. 2020.

Comentado [MCL42]: ABNT

FIGUEIREDO, Filipe Daniel Seabra. **Simulação de circuitos quânticos**. 2013. 42 f. Tese (Doutorado) - Curso de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, 2013.

FINGERHUTH, Mark; BABEJ, Tomáš; WITTEK, Peter. Open source software in quantum computing. **Plos One**, [S.L.], v. 13, n. 12, p. 1-28, 20 dez. 2018. Public Library of Science (PLOS). Disponível em: <https://doi.org/10.1371/journal.pone.0208561>. Acesso em: 25 nov. 2020.

LAVADO, Thiago. **IBM alcança novo patamar na computação quântica e quer dobrar capacidade**. 2020. Disponível em: <https://exame.com/tecnologia/ibm-alcanca-novo-patamar-na-computacao-quantica-e-quer-dobrar-capacidade/>. Acesso em: 25 nov. 2020.

MARQUEZINO, Franklin de Lima. **A Transformada de Fourier Quântica Aproximada e sua Simulação**. 2006. 118 f. Tese (Doutorado) - Curso de Modelagem Computacional, Laboratório Nacional de Computação Científica, Petrópolis, Rio de Janeiro, 2006.

MINTZ, Tiffany M.; MCCASKEY, Alexandre J.; F. DUMITRESCU, Eugene; V. MOORE, Shirley; POWERS, Sarah; LOUGOVSKI, Pavel. QCOR: a language extension specification for the heterogeneous quantum-classical model of computation. **Acm Journal On Emerging Technologies In Computing Systems**. [S.L.], p. 1-17. nov 2020. Disponível em: <https://doi.org/10.1145/3380964>. Acesso em: 25 nov. 2020.

MOLL, Nikolaj; BAKOUTSOS, Panagiotis; BISHOP, Lev; CHOW, Jerry. Quantum optimization using variational algorithms on near-term quantum devices. **Quantum Science And Technology**, [S.L.], v. 10, n. 3, p. 1-30, jul 2017. Disponível em: <https://doi.org/10.1088/2058-9565/aab822>. Acesso em: 25 nov. 2020.

NIELSEN, Michael A.; CHUANG, Isaac L.. **Quantum Computation and Quantum Information**. 10. ed. Cambridge, UK: Cambridge University Press, 2010. 676 p. ISBN 978-1-107-00217-3. Disponível em: . Acesso em: 25 nov. 2020.

PERDOMO-ORTIZ, Alejandro; BENEDETTI, Marcello; REALPE-GÓMEZ, John; BISWAS, Rupak. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. **Quantum Science And Technology**, [S.L.], v. 3, n. 3, p. 1-13, 19 jun. 2018. IOP Publishing,. Disponível em: <http://dx.doi.org/10.1088/2058-9565/aab859>. Acesso em: 25 nov. 2020.

ROSA, Evandro Chagas Ribeiro da. **QSystem: simulador quântico para Python**. 2019. 209 f. TCC (Graduação) - Curso de Bacharel em Ciência da Computação, Depto. de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2019.

SMITH, Stewart. **Quantum JavaScript (Q.js)**: How I'm making quantum computing accessible to anyone with a web browser. 2020. Disponível em: <https://medium.com/javascript-in-plain-english/quantum-javascript-d1effb84a619>. Acesso em: 25 nov. 2020.

WATANABE, Hiroshi; SUZUKI, Masaru; YAMAZAKI, Junnosuke. **QCAD**: gui environment for quantum computer simulator. 2011. Disponível em: <http://qcad.osdn.jp>. Acesso em: 25 nov. 2020.

ASSINATURAS

(Atenção: todas as folhas devem estar rubricadas)

Assinatura do(a) Aluno(a): _____

Assinatura do(a) Orientador(a): _____

Assinatura do(a) Coorientador(a) (se houver): _____

Observações do orientador em relação a itens não atendidos do pré-projeto (se houver):

FORMULÁRIO DE AVALIAÇÃO – PROFESSOR AVALIADOR

Acadêmico(a): Anderson Pozzi _____

Avaliador(a): Mauricio Capobianco Lopes _____

ASPECTOS AVALIADOS ¹		atende	atende parcialmente	não atende
ASPECTOS TÉCNICOS	1. INTRODUÇÃO O tema de pesquisa está devidamente contextualizado/delimitado?	x		
	O problema está claramente formulado?		x	
	1. OBJETIVOS O objetivo principal está claramente definido e é passível de ser alcançado?	x		
	Os objetivos específicos são coerentes com o objetivo principal?	x		
	2. TRABALHOS CORRELATOS São apresentados trabalhos correlatos, bem como descritas as principais funcionalidades e os pontos fortes e fracos?	x		
	3. JUSTIFICATIVA Foi apresentado e discutido um quadro relacionando os trabalhos correlatos e suas principais funcionalidades com a proposta apresentada?	x		
	São apresentados argumentos científicos, técnicos ou metodológicos que justificam a proposta?		x	
	São apresentadas as contribuições teóricas, práticas ou sociais que justificam a proposta?	x		
	4. REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO Os requisitos funcionais e não funcionais foram claramente descritos?	x		
	5. METODOLOGIA Foram relacionadas todas as etapas necessárias para o desenvolvimento do TCC?	x		
	Os métodos, recursos e o cronograma estão devidamente apresentados e são compatíveis com a metodologia proposta?	x		
	6. REVISÃO BIBLIOGRÁFICA (atenção para a diferença de conteúdo entre projeto e pré-projeto) Os assuntos apresentados são suficientes e têm relação com o tema do TCC?		x	
ASPECTOS METODOLÓGICOS	As referências contemplam adequadamente os assuntos abordados (são indicadas obras atualizadas e as mais importantes da área)?	x		
	7. LINGUAGEM USADA (redação) O texto completo é coerente e redigido corretamente em língua portuguesa, usando linguagem formal/científica?		x	
	A exposição do assunto é ordenada (as ideias estão bem encadeadas e a linguagem utilizada é clara)?	x		

PARECER – PROFESSOR AVALIADOR: (PREENCHER APENAS NO PROJETO)

O projeto de TCC ser deverá ser revisado, isto é, necessita de complementação, se:

- qualquer um dos itens tiver resposta NÃO ATENDE;
- pelo menos 5 (cinco) tiverem resposta ATENDE PARCIALMENTE.

PARECER: (x) APROVADO () REPROVADO

Comentado [MCL43]: Apesar de aprovado o trabalho deixa a desejar em muitos pontos principalmente relacionados à introdução, justificativa e revisão bibliográfica. Peço atenção para uma redação que respeite as normas da língua e que não seja tão confusa em alguns pontos.

Assinatura: _____ Data: _____

¹ Quando o avaliador marcar algum item como atende parcialmente ou não atende, deve obrigatoriamente indicar os motivos no texto, para que o aluno saiba o porquê da avaliação.